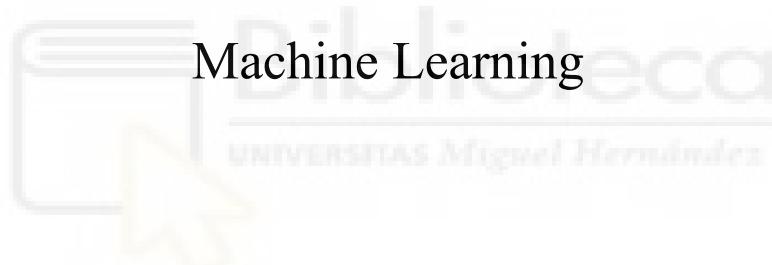




Gestión de Carteras Basada en Análisis Técnico y Machine Learning



Universidad Miguel Hernández

Facultad Ciencias de Ciencias Sociales y Jurídicas

Grado: Estadística Empresarial

4º curso

Fernando Díaz Lifante

Tutor: Jose Luis Sainz-Pardo Auñon

Curso 2024- 2025

Índice

1.	Resumen	4
2.	¿Por qué?	5
3.	Ley de la oferta y la demanda.....	6
4.	Conceptos financieros básicos	6
5.	Mercados financieros	7
6.	Representación del histórico de precios.....	9
8.	Gráfico de barras	10
9.	Análisis Fundamental y Técnico	11
9.1.	Análisis Fundamental	11
9.2.	Componentes Clave del Análisis Fundamental.....	11
9.3.	Aplicación del Análisis Fundamental	11
9.3.1.	Análisis Técnico	12
9.3.2.	Aplicación del Análisis Técnico	12
9.3.3.	Comparación y uso complementario	12
10.	Herramientas del análisis técnico	12
10.1.	Medias Móviles.....	13
10.2.	MACD (Moving Average Convergence Divergence)	14
10.3.	Bandas de Bollinger.....	15
10.4.	RSI	16
10.5.	Volumen	16
11.	Rentabilidad	17
12.	Propósitos	18
13.	Metodología	18
13.1.	Estrategia principal	19
13.1.1.	Gestión del riesgo y control de capital	20
14.	Resultados Machine Learning y Estrategias Tradicionales	20
14.1.	Bitcoin	21
14.2.	Bitcoin con posición siempre abierta.....	28
14.3.	Nasdaq100	34
14.4.	Nasdaq100 con posición siempre abierta.....	42
14.5.	Bitcoin, Machine Learning con apalancamiento y Stop-loss.....	48
14.6.	Nasdaq100, Machine Learning con apalancamiento y Stop-loss.....	61
15.	Proyecto a futuro	74
16.	Agradecimientos	74

17.	Bibliografía.....	75
18.	Anexos.....	76
18.1.	BTC Machine Learning y Estrategias Tradicionales.....	76
18.2.	BTC Machine Learning y Estrategias Tradicionales con posición siempre abierta	83
18.3.	Nasdaq100 Machine Learning y Estrategias Tradicionales	90
18.4.	Nasdaq100 Machine Learning y Estrategias Tradicionales con posición siempre abierta	96
18.5.	BTC Machine Learning con apalancamiento y Stop-loss	103
18.6.	Nasdaq 100 Machine Learning con apalancamiento y Stop-loss.....	111



1. Resumen

El objetivo de este Trabajo de Fin de Grado es aplicar técnicas de inversión basadas en Machine Learning y estrategias tradicionales sobre dos activos financieros. Para ello, se han diseñado y simulado sistemas de trading automatizados que generan señales de compra y venta, utilizando como objetivo de predicción (target) una fluctuación del ±1,5 % en el precio de cierre diario. De esta manera, se podrá saber cuáles son las estrategias claves para futuras operaciones.

Los activos financieros empleados en este proyecto son el Bitcoin (BTC) y el Nasdaq 100 (bolsa de valores formada por las empresas tecnológicas más relevantes, la cual se encuentra ubicada en New York).

Una vez se tengan los datos que queramos y se hayan desarrollado las estrategias se hará backtesting para el modelo de Machine Learning. Es decir, se van a comprobar si se obtienen buenos o malos resultados desde el 1 de Enero de 2024 hasta el 17 de febrero de 2025. Este será nuestro período de prueba para evaluar si son efectivas o no las estrategias que se han creado.

Se han implementado dos enfoques distintos de gestión de posiciones. En el primero, la estrategia cierra la posición al final del día si la señal deja de estar activa o cambia, siguiendo un enfoque tradicional de gestión diaria. En el segundo enfoque, la posición permanece abierta indefinidamente hasta que la señal cambie (por ejemplo, de compra a venta), asegurando así una operativa continua mientras persista la condición original. Esto permite comparar el impacto de cerrar posiciones diariamente frente a mantenerlas abiertas hasta un cambio de señal.

Se ha incluido también un Stop-loss y un apalancamiento a las estrategias empleadas.

La temporalidad que se ha usado ha sido diaria. Se ha optado por una temporalidad diaria ya que nuestro objetivo es obtener un 1.5% de fluctuación de un día para otro.

Finalmente, se han comentado y analizado con detalle los resultados obtenidos.

2. ¿Por qué?

Por qué realizar este proyecto, la idea de hacer este proyecto nació mucho antes de cursar la asignatura “Gestión de Carteras e Inversiones”, en el último año de mi carrera. Mucho antes, me interesó mucho la estadística viendo un partido de tenis, en donde, Rafael Nadal jugaba contra Daniil Medvedev, iba perdiendo 4-6 4-6 3-5 y con bola de partido para Daniil, en la pantalla se reflejó que Rafa tenía un 1% de ganar y desde ahí me interesé en cómo consiguieron ese 1% que después ganó el partido cuando la estadística daba por perdido el partido. Pero desde ahí ya sabía que quería hacer una carrera relacionada con la estadística. Pocos años después descubrí la carrera que estoy a punto de terminar “Estadística Empresarial” y de la que estoy súper agradecido de todo lo que me ha brindado. A lo largo de la carrera me interesé más por las inversiones y empezando a operar en este mundillo, es con todo esto por lo que acabé haciendo mi Trabajo Final de Grado de la asignatura que más me identifica que es la nombrada anteriormente; “Gestión de Carteras e Inversiones” impartida por Jose Luis Sainz-Pardo Auñon la cual me encantó por estas razones no tuve ninguna duda de que mi TFG sería sobre este tema. De esta manera podría observar si generar rentabilidad invirtiendo es posible.

Antes de todo, vamos a hablar y profundizar sobre los aspectos fundamentales del mercado de valores, para así entender cómo funcionan los mercados financieros.

La bolsa de valores es un mercado organizado donde se lleva a cabo la compraventa de acciones, bonos, y otros valores financieros, es decir, un lugar donde las personas pueden intercambiar activos financieros, como, por ejemplo, acciones de empresas.

Este intercambio se basa en la relación entre la cantidad de acciones que están disponibles para la compra (oferta) y la cantidad que las personas desean adquirir (demanda).

Además de los factores económicos, como la salud financiera de las empresas, también pueden influir otros aspectos en este proceso de compra y venta.

Por una parte, este mercado proporciona a las empresas grandes cantidades de dinero ya que hay muchas personas inversionistas que invierten en ellas. Esto hace que las empresas puedan expandirse y llevar a cabo proyectos de desarrollo e investigación.

Por otra parte, este mercado brinda a los inversores la oportunidad de participar en el crecimiento de las empresas y conseguir ganar dinero a través de la compra y venta de activos.

Es aquí donde se encuentra el foco de este trabajo. Se quiere invertir en los mercados financieros con la finalidad de obtener ganancias de ellos. Para ello se empleará el análisis técnico (análisis fundamental para encontrar tendencias, patrones, etc). Para poder lograrlo debemos tener en cuenta unos conceptos importantes, que son los siguientes y que explicaremos en el propio trabajo.

3. Ley de la oferta y la demanda

Para explicar la ley de la oferta y la demanda y cómo funciona en el trading, se va a exponer un ejemplo para que se comprenda de manera más clara. Imagina que estamos en el Mercadona y muchas personas quieren comprar peras y hay pocas, pues, Mercadona puede subir el precio porque saben que la gente está dispuesta a pagar más por conseguirlas (alta demanda). Por otra parte, si hay pocas personas interesadas en comprar manzanas y existen muchas manzanas, Mercadona tendrá que bajar el precio si quiere conseguir venderlas (baja demanda). Ahora, vamos a poner un ejemplo sencillo en nuestro caso: Por una parte, imagina que se rumorea que las acciones de una empresa van a subir mucho porque han innovado un nuevo producto, esto hará que mucha gente quiera comprar acciones de ella, pero no hay suficientes personas que quieran vender. Por lo que el precio de las acciones aumentará, (alta demanda, baja oferta). Por otra parte, imagina que la empresa tiene un escándalo o malas noticias. Esto hará que muchos inversores quieran vender sus acciones, pero pocas personas querrán comprarlas. Por lo tanto, el precio de las acciones disminuirá. En resumen, si más gente quiere comprar que vender, el precio sube. Y si más gente quiere vender que comprar, el precio baja.

4. Conceptos financieros básicos

Para comprender todavía mejor qué es la bolsa de valores, se va a indicar en qué instrumentos financieros se pueden invertir en ella.

Primero de todo hay que tener muy claro que invertir en la bolsa significa destinar dinero en algún tipo de activo con la expectativa de obtener un rendimiento en el futuro.

Se puede invertir, entre otros, en los siguientes instrumentos financieros:

-Acciones: Representan la propiedad de una fracción de una empresa. Cuando compras acciones de una empresa, te conviertes en accionista y tienes derecho a una parte de las ganancias de la empresa y a votar en las decisiones corporativas.

-Bonos: Son instrumentos de deuda emitidos por gobiernos, municipios o empresas para financiar sus operaciones. Al comprar un bono, estás prestando dinero al emisor a cambio de pagos periódicos de intereses y la devolución del capital al vencimiento.

-Fondos de inversión: Son vehículos de inversión que reúnen el dinero de varios inversores para invertir en una cartera diversificada de activos, como acciones, bonos u otros instrumentos financieros.

-Índice bursátil: Es una medida que refleja el comportamiento general del mercado de valores. Los índices bursátiles se calculan utilizando una ponderación de las acciones de las empresas incluidas en el índice y se utilizan como referencia para medir el rendimiento del mercado.

-Materias primas: Invertir en materias primas es una estrategia popular para diversificar una cartera de inversiones y protegerse contra la inflación.

-Criptomonedas: Son monedas digitales que existen únicamente en Internet y se caracterizan por su descentralización, lo que significa que no están controladas por bancos o gobiernos. Funcionan a través de una tecnología segura llamada blockchain, que es como un libro de contabilidad digital que registra todas las transacciones de forma transparente y segura.

Estos son algunos de los instrumentos financieros con los que cualquier persona puede invertir en ellos. En este trabajo vamos a invertir en acciones, en índices bursátiles (como el NASDAQ 100 en nuestro caso) y en Bitcoin que es una criptomoneda.

5. Mercados financieros

Los mercados financieros son como plataformas donde se hacen transacciones de instrumentos financieros. Se va a exponer una breve descripción de los mercados financieros más importantes:

- Mercado de valores: Es donde se negocian acciones y otros instrumentos de capital. Los inversores compran y venden acciones de empresas cotizadas en bolsas de valores como la Bolsa de Nueva York (NYSE) y el NASDAQ. Este mercado proporciona a las empresas una forma de obtener capital para financiar sus operaciones y a los inversores la oportunidad de participar en el crecimiento de estas empresas.
- Mercado de bonos: En este mercado se negocian bonos y otros instrumentos de deuda emitidos por gobiernos, empresas y entidades gubernamentales. Los bonos son títulos de deuda que pagan intereses periódicos y devuelven el capital al vencimiento. Los inversores pueden comprar bonos como una forma de generar ingresos regulares y diversificar sus carteras de inversión.
- Mercado de divisas (Forex): Es donde se negocian las monedas extranjeras. Los participantes en este mercado incluyen bancos, instituciones financieras, inversores institucionales y minoristas. El mercado de divisas es el mercado financiero más grande y líquido del mundo, con un volumen diario de transacciones que supera los trillones de dólares estadounidenses.
- Mercado de materias primas: En este mercado se negocian materias primas como petróleo, oro, plata, maíz, trigo, café, entre otros. Los inversores pueden operar con materias primas a través de contratos de futuros, opciones y ETFs (Exchange-Traded Funds). Este mercado es importante para la fijación de 11 precios y la gestión de riesgos para productores y consumidores de materias primas.
- Mercado de criptomonedas: Es un mercado emergente donde se negocian criptomonedas como Bitcoin, Ethereum, entre otras. Estas monedas digitales operan en una red descentralizada utilizando tecnología blockchain. El mercado de criptomonedas es altamente volátil y especulativo, es decir, invertir en ellas conlleva mucho riesgo asociado.

Es importante entender cómo funcionan estos mercados y cómo pueden afectar tus decisiones de inversión

6. Representación del histórico de precios

Hay diferentes formas de visualizar y ver el valor de los precios de un activo a golpe de vista. Gracias a los gráficos, podemos observar el valor de los precios del activo en cualquier periodo. Las velas japonesas, el gráfico de líneas y el gráfico de barras son tres tipos de esquemas que permitirán analizar el precio de un activo financiero.

Estos dos gráficos del análisis técnico cuentan con características diferentes, pero nos pueden ayudar para el mismo fin: ver el precio de apertura, cierre, el máximo y mínimo, por lo que ninguno es mejor que el otro. El tipo de gráfico dependerá del estilo de trading que más se adecue a las características de cada uno

7. Velas japonesas

Las velas japonesas son un tipo de gráfica que se utiliza para mostrar el comportamiento del precio de un activo financiero en un período de tiempo determinado. Cada vela está compuesta por un cuerpo y dos líneas. El cuerpo refleja el precio de apertura y de cierre, mientras que los extremos de las líneas representan el máximo y el mínimo del precio alcanzado en la temporalidad escogida. Lo podemos ver con más detalle en la imagen 1.

El libro “Las velas japonesas” de Steve Nison (x), ofrece una visión completa de las velas japonesas y las técnicas para seguir en el mercado con estas herramientas.

En general, las velas son de color verde o rojo. El verde se correlaciona cuando sube el precio, mientras que el rojo significa que el precio ha bajado

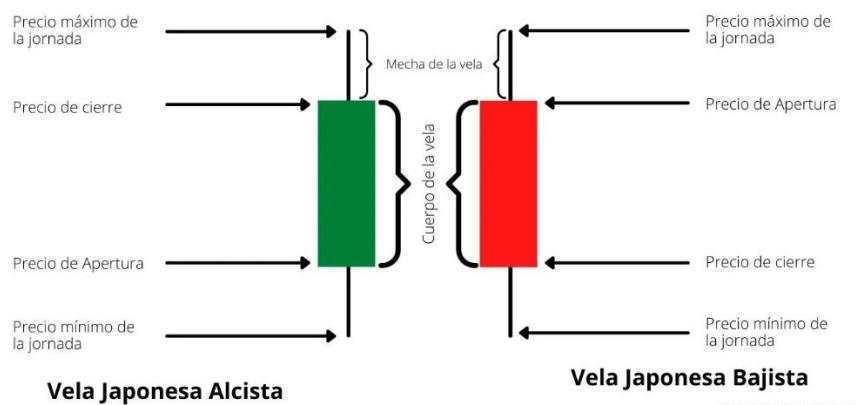


Imagen 1: Velas Japonesas. Fuente: clubdecapitales.com

En la imagen podemos observar dos velas, una de color rojo y otra verde. La primera indica que el precio del activo ha subido en la temporalidad escogida. Mientras que el precio del activo ha bajado en la temporalidad escogida. En la primera podemos ver que el precio de cierre está por arriba del de apertura y en la segunda al revés. El mínimo y máximo del valor del precio es el mismo para las dos velas, es decir, el precio más bajo se corresponde al extremo inferior y el precio máximo al extremo superior. Esta información puede resultar útil a la hora de encontrar tendencias o patrones de velas con el fin de predecir el movimiento del precio

8. Gráfico de barras

Los gráficos de barras también sirven para representar el precio de cualquier activo financiero. A diferencia de las velas japonesas, el gráfico de barras consiste en una línea vertical que representa el precio de la acción, con dos líneas horizontales a cada lado. La línea izquierda indica el precio de apertura y la línea derecha muestra el precio de cierre. Como se aprecia en la imagen 2. Durante la formación de la barra, el precio de cierre se actualiza continuamente hasta que se completa el período de tiempo, momento en el que se establece el precio de cierre de la barra y comienza una nueva.



Imagen 2: Gráfico de Barras. Fuente: Clubdecapitales.com

9. Análisis Fundamental y Técnico

Este trabajo, como se ha comentado anteriormente, se basa principalmente en el análisis técnico con Machine Learning. Pero, para realizar correctas operaciones de inversión, lo idílico sería utilizar tanto el fundamental como el técnico.

9.1. Análisis Fundamental

El análisis fundamental se centra en determinar cuál es el valor intrínseco de un activo financiero, como una acción, mediante el estudio de factores económicos, financieros y cualitativos.

El objetivo es identificar si el precio del mercado financiero de un activo refleja realmente su valor.

9.2. Componentes Clave del Análisis Fundamental

-Factores Cuantitativos: Incluyen métricas financieras obtenidas de los estados financieros de una empresa, como ingresos, beneficios netos, flujos de caja, deuda y ratios financieros (por ejemplo: PER, ROE, margen de beneficio).

-Factores Cualitativos: Incluyen los aspectos no numéricos, como la calidad del equipo de trabajo, la posición de mercado, la ventaja competitiva y las perspectivas del sector.

-Análisis Macroeconómicos: Este análisis evalúa el entorno económico general, incluyendo tasas de interés, inflación, políticas gubernamentales y tendencias económicas que puedan afectar al rendimiento de la empresa.

9.3. Aplicación del Análisis Fundamental

Los inversores utilizan este análisis para identificar activos infravalorados o sobrevalorados. Si el valor intrínseco estimado de una acción es superior a su precio de mercado, puede considerarse una oportunidad de compra y viceversa. Este análisis se centra en inversores que les gusta más invertir a largo plazo y que buscan beneficios sostenibles sin arriesgar demasiado.

9.3.1. Análisis Técnico

El análisis técnico se basa en el estudio de datos históricos de precios y volúmenes de negociación para predecir movimientos futuros del mercado. La idea de este análisis es que los precios reflejan toda la información disponible y que los patrones históricos tienden a repetirse. Este análisis no tiene en cuenta las noticias que son vitales para los inversores. Asumen que los precios se mueven de manera lógica y predecible, aunque no siempre sea así.

9.3.2. Aplicación del Análisis Técnico

Este análisis es utilizado principalmente por traders e inversores a corto plazo que buscan aprovechar movimientos de precios a corto plazo. Al identificar tendencias y patrones, los analistas técnicos intentan determinar puntos óptimos de entrada y salida en el mercado.

9.3.3. Comparación y uso complementario

Aunque pueda parecer que estos dos análisis sean opuestos, muchos inversores los utilizan de manera complementaria. Por ejemplo, un inversor puede usar el análisis fundamental para seleccionar activos sólidos y luego aplicar el análisis técnico para decidir el momento óptimo de compra o venta.

10. Herramientas del análisis técnico

El análisis técnico se enfoca en anticipar el comportamiento futuro del precio de una acción mediante el uso de herramientas y métodos matemáticos. Su finalidad es ayudar a tomar decisiones sobre cuándo comprar (si se espera que el precio suba) o vender (si se prevé que el precio bajará).

Este tipo de análisis se basa principalmente en la identificación de tendencias, patrones y movimientos en el precio. Además, recurre a indicadores técnicos para detectar posibles momentos adecuados de entrada o salida del mercado.

Los indicadores técnicos funcionan como señales que brindan información sobre la situación actual del precio. Un ejemplo es el RSI (Índice de Fuerza Relativa), que puede

señalar si una acción está sobrevalorada (lo que sugeriría vender) o infravalorada (lo que sugeriría comprar).

Otro elemento importante en el análisis técnico es el volumen de negociación, que representa la cantidad de acciones compradas o vendidas en un periodo determinado. Un volumen alto suele reflejar un fuerte interés en el activo, mientras que un volumen bajo indica desinterés. Generalmente, un mayor volumen se asocia con una mayor volatilidad en el precio.

Actualmente, el análisis técnico continúa desarrollándose, incorporando nuevos indicadores y herramientas gráficas. A día de hoy, sigue siendo una herramienta fundamental para traders e inversores en los mercados financieros.

10.1. Medias Móviles

La media móvil es una medida de tendencia central que recoge el promedio (media) de una serie de datos históricos de precios. Se pueden usar tanto medias móviles simples (SMA) como exponenciales (EMA). La diferencia entre ellas está en que en las exponenciales las cotizaciones más recientes tienen mayor peso que las más antiguas.

$$SMA = \frac{P_1 + P_2 + P_3 + \dots + P_n}{n}$$

Fórmula Cálculo de la media móvil simple

Donde: P1, P2, P3, hasta Pn, son los precios de cierre de un activo durante los n períodos seleccionados.

n es el número de períodos de tiempo

$$EMA_{hoy} = (P_{hoy} \times \alpha) + (EMA_{ayer} \times (1 - \alpha))$$

Fórmula Cálculo de la media móvil exponencial

Donde: Phoy es el precio de cierre del día de hoy.

EMA ayer es el valor de la EMA del día anterior.

El α es el factor de suavizado, calculado como $\alpha = 2/(n+1)$, donde n es el número de períodos.

La finalidad de este método es eliminar el ruido producido por movimientos bruscos en el precio.

10.2. MACD (Moving Average Convergence Divergence)

El MACD, es un indicador técnico muy común utilizado en el análisis técnico de los mercados financieros. Es un indicador de tendencia por lo que resulta más fiable en períodos de tendencia.

Este indicador se forma mediante dos medias móviles exponenciales, una de corto ($NC=12$ períodos por ejemplo) y otra de largo plazo ($NL=26$ períodos por ejemplo). El número de períodos depende de los criterios de cada analista. El MACD se obtiene a través de la siguiente fórmula:

$$MACD = M. Móvil Exponencial (valor, NC) - M. Móvil Exponencial (valor, NL)$$

Fórmula Cálculo del valor del MACD

Para saber interpretarlo, se necesita el cálculo de la señal, que se calcula mediante una media móvil exponencial más corta aún. ($NCC=8$, por ejemplo)

$$Señal = M. Móvil Exponencial (MACD, NCC)$$

Fórmula Cálculo del valor de la señal

Y después de eso hay que calcular la diferencia:

$$Diferencia = MACD - Señal$$

Fórmula Cálculo de la diferencia

Así pues, ya se puede interpretar esta estrategia. Cuando la línea MACD cruce de abajo hacia arriba la línea de señal se interpretará como una señal de compra, que esto quiere

decir que el precio del activo financiero subirá. En cambio, cuando la línea MACD cruza de arriba hacia abajo la línea de señal, se interpretará como una señal de venta, que esto quiere decir que el precio del activo financiero bajará.

Hay más maneras de interpretar esta estrategia, como por ejemplo con la diferencia. Cuando la diferencia pase de ser positiva a negativa significa que existe una señal de compra, y si la diferencia pasa de ser positiva a negativa significa que hay una señal de venta.

A parte de servir como referencia para posibles compras o ventas, las intersecciones entre la línea MACD y su línea de señal también pueden interpretarse como indicios de un cambio en la dirección del mercado. Por ejemplo, cuando la línea MACD atraviesa hacia arriba la línea de señal, suele interpretarse como una posible transición de una fase bajista a una alcista. Por el contrario, si la línea MACD cruza hacia abajo la línea de señal, podría estar anticipando un giro de tendencia hacia la baja.

En esencia, el MACD es un indicador técnico que se basa en dos medias móviles exponenciales, junto con una tercera que actúa como línea de señal. Esta combinación no solo ayuda a detectar oportunidades de entrada y salida en el mercado, sino que también es eficaz para reconocer posibles giros en la dirección de los precios. Por ello, se ha convertido en una herramienta clave para inversores y traders que buscan tomar decisiones fundamentadas en sus estrategias de inversión.

10.3. Bandas de Bollinger

Este indicador se forma por tres líneas plasmadas en un gráfico de precios: una banda superior, una línea central que por defecto es una media móvil simple, y una banda inferior.

Las bandas superior e inferior se calculan utilizando una desviación estándar sobre la media móvil. Las bandas fluctúan en función de la volatilidad del mercado. Cuanto más ruido haya, más se expanden, y cuanto menos volatilidad haya, más se contraen las

bandas. Todo esto, proporciona información muy útil sobre posibles zonas de sobrecompra o sobreventa de un activo.

Si los precios están cerca de la banda superior, esto puede indicar sobrecompra, es decir, la acción está “overvalue”, que quiere decir que la acción está cara y podríamos venderla (señal de venta). En cambio, si los precios están cercanos a la banda inferior, puede indicar sobreventa, es decir, la acción está “undervalue” que quiere decir, barata y por lo tanto compraremos (señal de compra).

En conclusión, este indicador expresa que al sobrepasar los precios en cualquiera de las bandas se trata de una “señal”, indicando un posible cambio de tendencia consistente, por tanto, habría posibilidad de realizar una operación financiera.

10.4. RSI

Se utiliza naturalmente para medir la velocidad y el cambio de los movimientos de los precios en un activo financiero, puede ser una acción, un índice, una divisa...

El RSI funciona mejor en periodos donde no existe tendencia, es decir, en tendencia lateral, a diferencia de lo que pasaba con los indicadores que utilizan medias móviles como el MACD o las Bandas de Bollinger.

10.5. Volumen

En el contexto financiero, el volumen se refiere simplemente a la cantidad de acciones que se han negociado, es decir, compradas y vendidas durante un periodo determinado. Este dato es clave para quienes analizan los mercados desde un enfoque técnico, ya que proporciona indicios sobre la solidez o debilidad de una tendencia.

Cuando un cambio de precio viene acompañado de un alto volumen, suele interpretarse como una señal de que ese movimiento tiene fuerza y es probable que continúe. En cambio, si el volumen es bajo, podría estar indicando que la tendencia pierde impulso y que un giro en la dirección del mercado podría estar cerca.

El volumen también puede emplearse como base para generar señales de entrada o salida en una estrategia de inversión. Más adelante, en la sección 7 de este trabajo, se explicará con detalle cómo se construye una estrategia centrada en este indicador.

11. Rentabilidad

La rentabilidad, en el ámbito de la inversión, indica la ganancia o pérdida obtenida por una inversión en comparación con el capital inicialmente invertido. Es un indicador fundamental que ayuda a los inversores a analizar el rendimiento de sus inversiones y realizar elecciones informadas. Al invertir dinero en un activo como acciones, bonos, propiedades o fondos, la rentabilidad indica cuán lucrativa ha sido dicha inversión, es decir, cuánto se ha ganado o perdido en relación con la cantidad que se invirtió al principio.

Hay diversas maneras de expresar la rentabilidad, pero la más simple se determina como el porcentaje de cambio entre el valor final y el valor inicial de la inversión. Por ejemplo, si una persona invierte 1.000 euros y después de un año tiene 1.100, la rentabilidad lograda sería del 10 %. No obstante, este cálculo fundamental se puede ajustar con otros métodos más concretos según el tipo de análisis que se deseé llevar a cabo.

Entre los tipos más importantes de rentabilidad se hallan la rentabilidad nominal, que representa el rendimiento bruto sin considerar factores como la inflación; y la rentabilidad real, que corrige ese resultado restando la inflación, proporcionando así una perspectiva más exacta del poder adquisitivo obtenido. Igualmente, relevante es la rentabilidad anualizada, que facilita la comparación de inversiones con distintas duraciones, y la rentabilidad acumulada, que indica el total de ganancias obtenidas en un periodo específico. En contextos más teóricos, también se menciona la rentabilidad anticipada, fundamentada en proyecciones de rendimiento futuro conforme a escenarios plausibles. Numerosos factores pueden influir en la rentabilidad final de una inversión. Entre ellos destacan el nivel de riesgo asumido (pues a mayor riesgo, suele esperarse mayor rentabilidad), el plazo durante el cual se mantiene la inversión, y los costes asociados, como comisiones o impuestos, que pueden reducir la ganancia neta. Otro aspecto clave es la reinversión de beneficios, ya que reinvertir las ganancias puede generar un efecto compuesto, multiplicando los rendimientos con el paso del tiempo.

La rentabilidad no solo facilita evaluar el éxito de una inversión, sino que también es crucial para comparar diversos activos o enfoques. Sin embargo, su análisis debe estar

siempre ligado a la evaluación del riesgo, dado que una alta rentabilidad no siempre es favorable si implica una gran volatilidad o riesgo de pérdida. Por esta razón, se utilizan instrumentos como el ratio de Sharpe, que vincula rentabilidad y riesgo para obtener una perspectiva más equilibrada del desempeño.

En resumen, la rentabilidad es un concepto fundamental en el ámbito financiero. Conocer cómo calcularla, interpretarla y vincularla con otros factores como el riesgo o el tiempo es esencial para quienes quieren invertir de manera informada en los mercados. Su adecuada evaluación posibilita optimizar beneficios y administrar eficazmente un portafolio de inversiones.

12. Propósitos

Lo que se ha querido reflejar con este trabajo es saber si sería rentable invertir en ciertos activos utilizando el análisis técnico juntamente con el Machine Learning. Se ha elaborado una estrategia de Machine Learning utilizando un modelo Random Forest con una ventana de 60, 40, 20 y 10 días para entrenar al modelo que a su vez se han utilizado 10 variables que comentaremos más adelante.

13. Metodología

Para la realización de este trabajo, se ha llevado a cabo una estrategia de inversión automatizada mediante la utilización de técnicas de Machine Learning y enfoques tradicionales. Todo el análisis se ha realizado en Python, un lenguaje de programación de alto nivel que es muy utilizado en las áreas financiera y científica.

Se llevó a cabo un backtesting de operaciones de compra y venta, simulando el rendimiento de las estrategias con datos históricos de dos activos financieros: el Bitcoin (BTC) y el Nasdaq100. Se comienza con un capital inicial de 1000 €, el cual se utiliza para hacer inversiones según las señales que se generen. El intervalo examinado se extiende desde el 1 de enero de 2024 hasta el 17 de febrero de 2025.

Se han investigado diversas configuraciones, como la utilización o no de apalancamiento,

la implementación de un Stop-loss dinámico, y estrategias que permiten mantener posiciones abiertas hasta recibir nueva señal. Asimismo, se ha analizado el efecto del tamaño de la ventana, que indica la cantidad de datos pasados empleados para entrenar el modelo previo a cada predicción.

13.1. Estrategia principal

La estrategia principal como he comentado antes es la de Machine Learning. En la cual se ha utilizado un modelo Random Forest, que se basa en conjuntos de árboles de decisión. Lo que permite que se maneje relaciones no lineales entre las variables.

Para el entrenamiento del modelo, se han utilizado 10 variables que son de las más famosas analizando el comportamiento del mercado:

RSI_14 (Relative Strength Index)

MACD (Moving Average Convergence Divergence)

EMA_10 (Media Móvil Exponencial de 10 días)

SMA_10 (Media Móvil Simple de 10 días)

Precio > SMA_10

Precio > EMA_10

Precio > Banda Superior de Bollinger

Precio < Banda Inferior de Bollinger

Precio > SMA_20 (Media Móvil Simple de 20 días)

Precio > STD_20 (Desviación típica de 20 días)

Para asegurar la veracidad del modelo, el entrenamiento se lleva a cabo a diario empleando una ventana deslizante de datos previos, que llamamos tamaño de ventana. En este estudio, se han evaluado diferentes longitudes de ventana (por ejemplo, 10, 20, 40 y 60 días) para examinar cómo influye la cantidad de datos pasados en la eficacia predictiva del modelo. Esto significa que, para cada día de inversión, el modelo se entrena solo con

los datos de los n días previos (de acuerdo con el tamaño de la ventana seleccionado), sin recurrir a información del presente o del futuro, lo que simula situaciones reales de inversión.

Asimismo, se han analizado varias configuraciones operativas. Por un lado, se han analizado estrategias que cierran automáticamente la posición al cambiar la señal con otras que mantienen la posición abierta hasta el final del periodo.

El objetivo del modelo es predecir la dirección del mercado para el día siguiente. Y, para que se lleve a cabo el objetivo, se necesitará una variable que se definirá como “target” que se basa en el rendimiento diario siguiente:

Se asigna un valor de 1 si el precio sube más de un 1.5%

Se asigna un valor de -1 si el precio baja más de un 1.5%

Se asigna 0 si el cambio está entre -1.5% y 1.5%

Las decisiones de inversión son:

Si la predicción es 1, se realiza una compra al precio de apertura del día siguiente.

Si la predicción es -1, se ejecuta una venta en corto al precio de apertura del día siguiente.

Si la predicción es 0, no se realiza ninguna operación.

Las posiciones abiertas se cierran cuando la predicción cambia o si se fuerza el cierre al final del período.

13.1.1. Gestión del riesgo y control de capital

Además, se ha implementado un apalancamiento financiero ($\times 5$ y $\times 10$), junto a un Stop-loss dinámico del 10 %, que restringe las pérdidas máximas por operación según el capital disponible al momento de abrirla. Estas variaciones posibilitan evaluar el efecto del riesgo y de la gestión activa en los resultados finales de las estrategias.

14. Resultados Machine Learning y Estrategias Tradicionales

14.1. Bitcoin

En primer lugar, se llevó a cabo un análisis de las diversas estrategias empleadas en el activo BTC sin incluir apalancamiento ni métodos de protección como el Stop-loss. En esta investigación se ha analizado el desempeño del modelo de Machine Learning y de las metodologías tradicionales en diversas configuraciones de tamaño de ventana de entrenamiento, específicamente con valores de 10, 20, 40 y 60 días.

Como se anticipaba, los resultados del modelo de ML presentan una notable sensibilidad al tamaño de ventana: al disminuir el tamaño de ventana, el modelo tiene menos historial para su entrenamiento, lo que puede perjudicar la calidad de las predicciones. Por ejemplo, al utilizar un tamaño de ventana de 60 como podemos ver en la tabla 1.1 y en el gráfico 1.1.1, el modelo acaba con un capital de 831,98 €, mientras que al emplear una ventana de solo 10 días como podemos ver en la tabla 1.4 y en el gráfico 1.4.1, el resultado se reduce a 575,05 €. Esta tendencia sugiere que un entrenamiento con escasa información previa puede resultar en decisiones de inversión más equivocadas.

Por otro lado, las estrategias tradicionales evidencian resultados estables sin importar el tamaño de la ventana. Esto se debe a que no utilizan un proceso de aprendizaje, sino que funcionan directamente a partir de señales binarias o indicadores técnicos previamente calculados en el conjunto de datos. De esta manera, tácticas como “Precio > SMA_10” o “Precio > EMA_10” como se observa en las tablas 1.1, 1.2, 1.3, 1.4 y en los gráficos 1.1.2, 1.2.2, 1.3.2 y 1.4.2, preservan el mismo capital final en todos los casos estudiados: 1.664,03 € y 1.617,06 € en su respectivo orden. Esta estabilidad se debe a que el backtest de estas estrategias revisa diariamente si se satisfacen o no las condiciones para abrir o cerrar una posición, sin depender de una ventana de entrenamiento móvil anterior.

En conjunto, estos hallazgos resaltan que, si bien el Machine Learning brinda un enfoque flexible y posiblemente más avanzado, su efectividad está muy vinculada a una adecuada elección del tamaño de la ventana. Las estrategias convencionales, en cambio, ofrecen una mayor solidez ante alteraciones en la configuración del backtest, aunque pueden no ser tan adaptables a cambios repentinos en el mercado.

Tabla 1.1: Tabla comparativa BTC con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-16.80 %	831.98
Señal_RSI	-50.06 %	499.37
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1214.53
Señal_EMA	+15.42 %	1154.21
Precio > SMA_10	+66.40 %	1664.03
Precio > EMA_10	+61.71 %	1617.06
Precio > Banda Superior	+4.84 %	1048.40
Precio < Banda Inferior	+4.70 %	1047.02

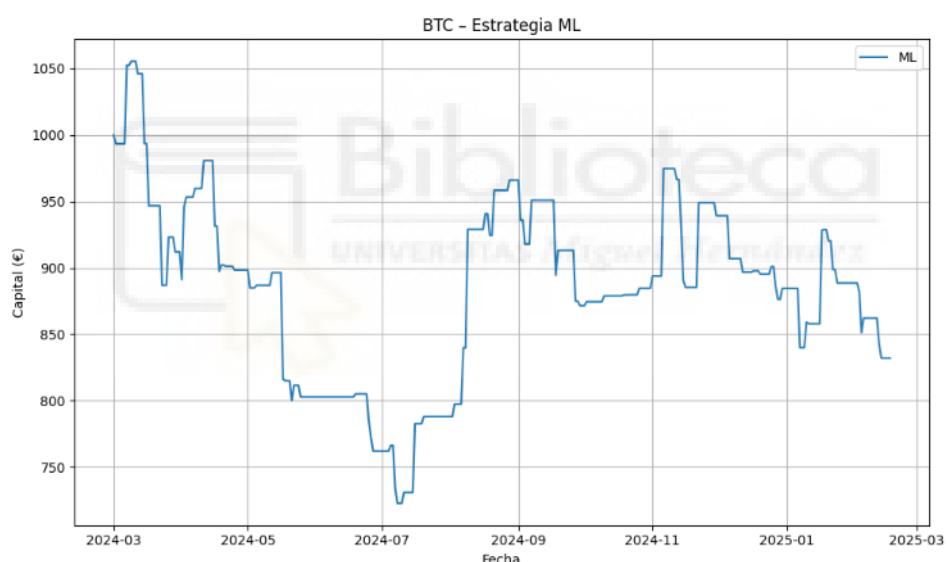


Gráfico 1.1.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con Window Size 60. Fuente: Elaboración Propia

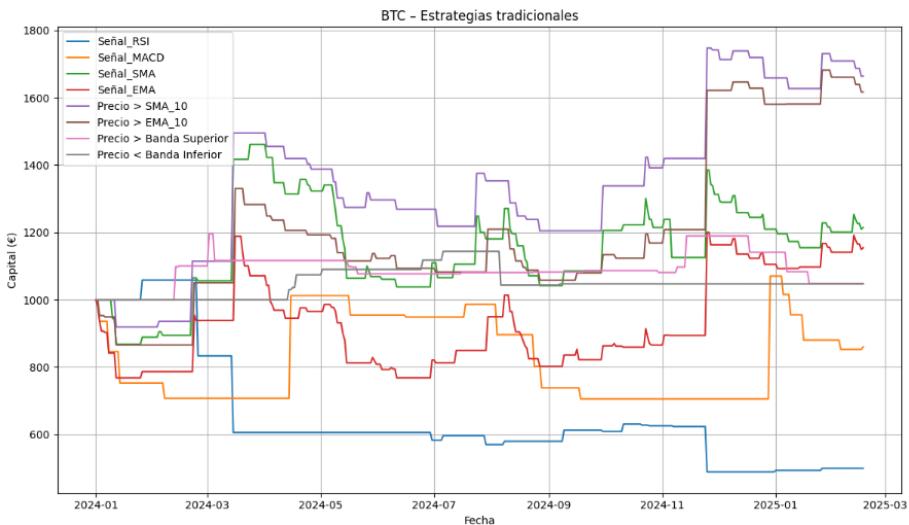


Gráfico 1.1.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con Window Size 60. Fuente: Elaboración Propia

Tabla 1.2: Tabla comparativa BTC con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-4.24 %	957.61
Señal_RSI	-50.06 %	499.37
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1214.53
Señal_EMA	+15.42 %	1154.21
Precio > SMA_10	+66.40 %	1664.03
Precio > EMA_10	+61.71 %	1617.06
Precio > Banda Superior	+4.84 %	1048.40
Precio < Banda Inferior	+4.70 %	1047.02

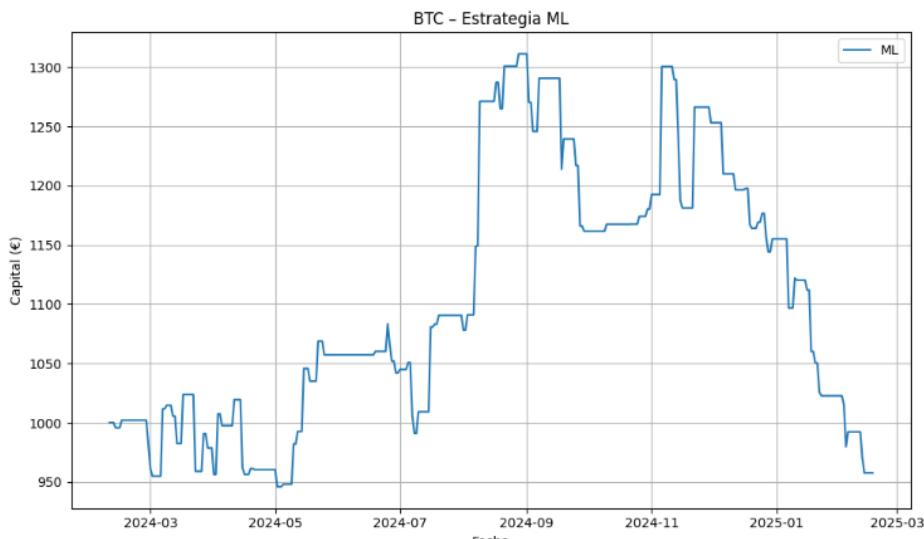


Gráfico 1.2.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con Window Size 40. Fuente: Elaboración Propia

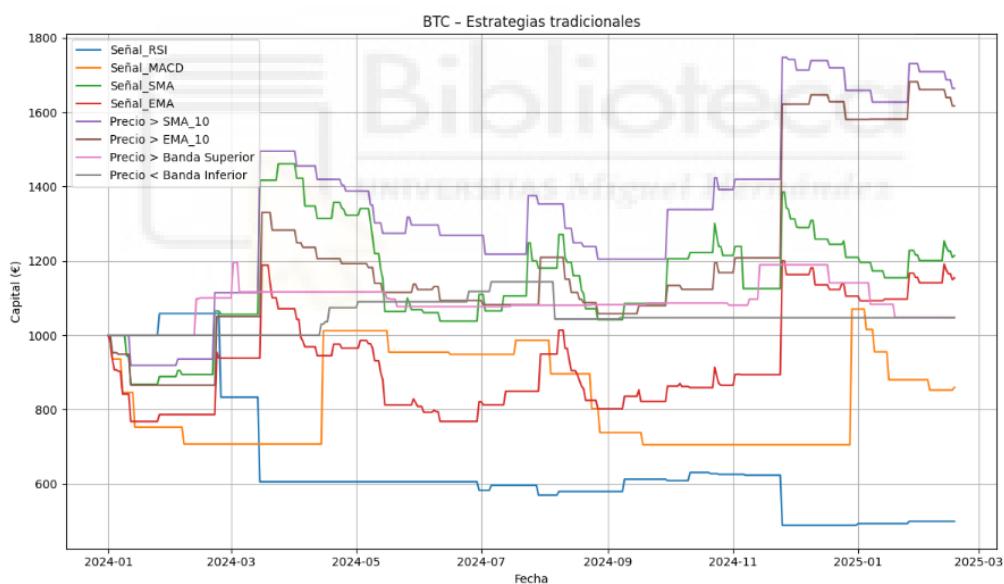


Gráfico 1.2.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con Window Size 40. Fuente: Elaboración Propia

Tabla 1.3: Tabla comparativa BTC con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
------------	-----------------	-------------------

ML	-33.72 %	662.79
Señal_RSI	-50.06 %	499.37
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1214.53
Señal_EMA	+15.42 %	1154.21
Precio > SMA_10	+66.40 %	1664.03
Precio > EMA_10	+61.71 %	1617.06
Precio > Banda Superior	+4.84 %	1048.40
Precio < Banda Inferior	+4.70 %	1047.02



Gráfico 1.3.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con Window Size 20. Fuente: Elaboración Propia

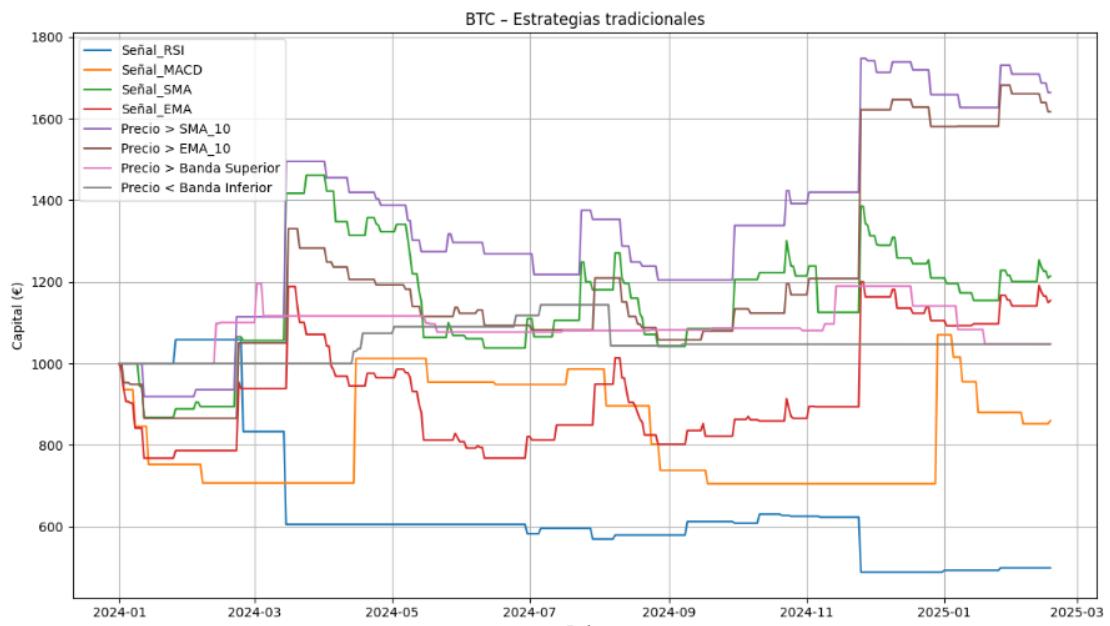


Gráfico 1.3.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con Window Size 20. Fuente: Elaboración Propia

Tabla 1.4: Tabla comparativa BTC con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-42.49 %	575.05
Señal_RSI	-50.06 %	499.37
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1214.53
Señal_EMA	+15.42 %	1154.21
Precio > SMA_10	+66.40 %	1664.03
Precio > EMA_10	+61.71 %	1617.06
Precio > Banda Superior	+4.84 %	1048.40
Precio < Banda Inferior	+4.70 %	1047.02

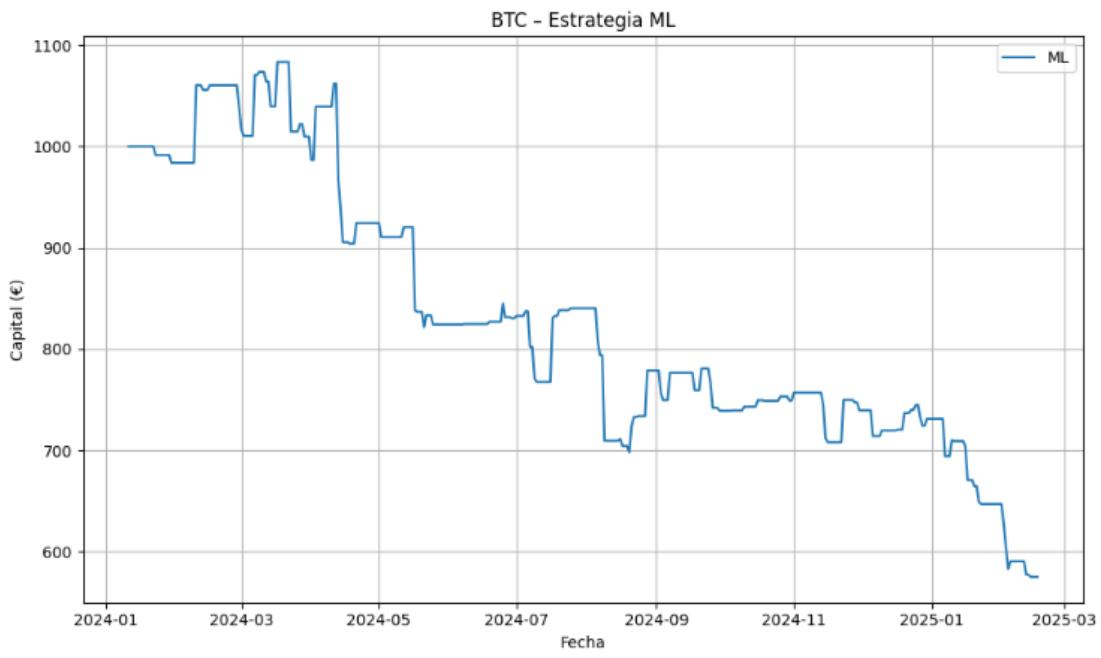


Gráfico 1.4.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con Window Size 10. Fuente: Elaboración Propia

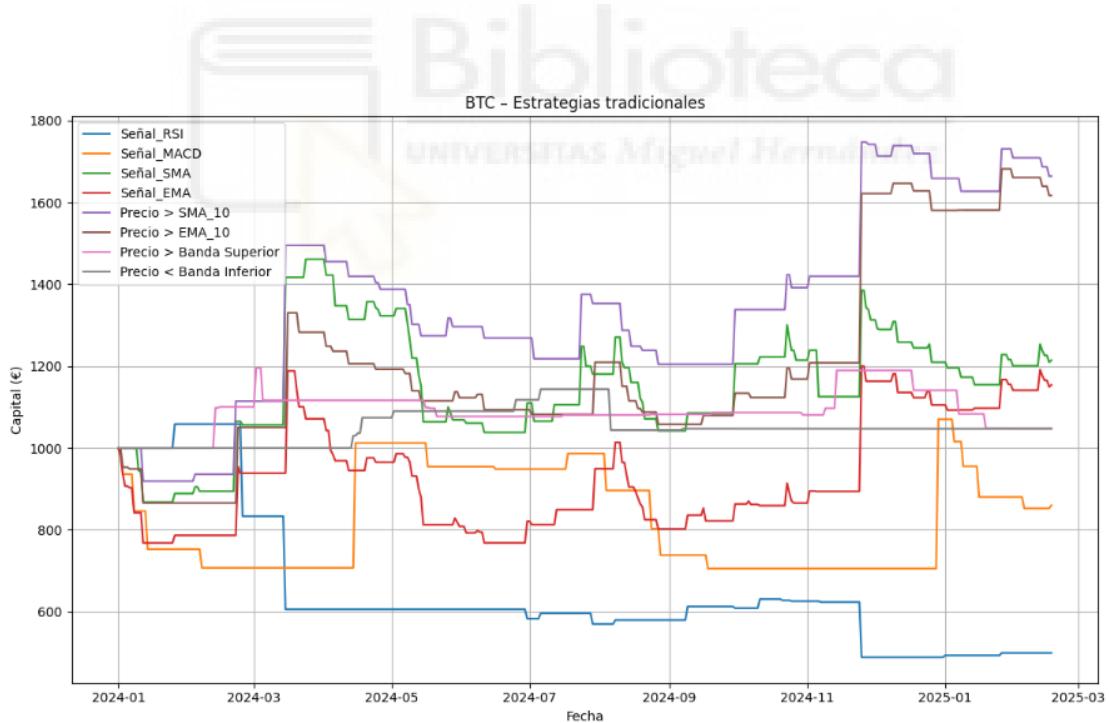


Gráfico 1.4.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con Window Size 20. Fuente: Elaboración Propia

14.2. Bitcoin con posición siempre abierta

En esta situación se ha analizado el desempeño de varias estrategias de inversión en el activo BTC, utilizando un método sin apalancamiento, sin stop-loss y manteniendo las posiciones abiertas (es decir, sin cerrar de manera explícita al recibir señal neutra). Se examinaron los resultados logrados con diferentes tamaños de ventana (window size): 10, 20, 40 y 60 días.

Se nota una discrepancia notable en el desempeño del modelo de Machine Learning (ML) en comparación con las estrategias convencionales. Conforme se extiende la ventana de entrenamiento, el rendimiento del modelo de ML se incrementa notablemente: desde una reducción del -62.88 % con una ventana de 10 días como se observa en la tabla 2.4 y en el gráfico 2.4.1, hasta una mejora del +20.37 % con una ventana de 40 días como se ve en la tabla 2.2 y en el gráfico 2.2.1, e incluso un +9.56 % con una ventana de 60 días como se aprecia en la tabla 2.1 y en el gráfico 2.1.1 .Este comportamiento indica que un historial de datos más extenso facilita el entrenamiento de modelos más sólidos y con menor sobreajuste, aunque los resultados continúan siendo afectados por la calidad de las señales producidas.

En cambio, las estrategias tradicionales presentan resultados muy consistentes en todos los tamaños de ventana. Esto se debe a que estas estrategias no dependen del tamaño de la ventana para tomar decisiones, ya que realizan operaciones basadas en indicadores técnicos establecidos. De esta manera, técnicas como SMA, EMA, Precio > SMA_10 o Precio > EMA_10 generan ganancias positivas en todas las situaciones, resaltando especialmente Precio > SMA_10 y Precio > EMA_10 con rendimientos de +28.62 % y +17.52 % respectivamente, mientras que Precio < Banda Inferior presenta un desempeño consistentemente negativo (-72.48 %). Estos resultados los podemos ver en cualquier tabla desde la 2.1 hasta la 2.4 y en los gráficos desde el 2.1.2 hasta el 2.4.2 que son los mismos.

Todo esto indica que, al optar por un enfoque sin limitaciones (sin Stop-loss ni cierres), las estrategias fundamentadas en medias móviles simples y exponenciales tienden a aventajar al modelo de ML en situaciones con escasa profundidad de entrenamiento (window size pequeños), aunque este último puede obtener resultados comparables con un historial adecuado. La consistencia de las estrategias tradicionales se opone a la

elevada fluctuación del rendimiento del modelo de ML, lo que es un aspecto fundamental a tener en cuenta para aplicaciones prácticas.

Tabla 2.1: Tabla comparativa BTC con posición siempre abierta con Window Size 60.

Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	+9.56 %	1,095.61
Señal_RSI	-54.00 %	459.98
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1,214.53
Señal_EMA	+15.42 %	1,154.21
Precio > SMA_10	+28.62 %	1,286.18
Precio > EMA_10	+17.52 %	1,175.24
Precio > Banda Superior	-53.46 %	465.44
Precio < Banda Inferior	-72.48 %	275.16



Gráfico 2.1.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning, con posición siempre abierta y con Window Size 60. Fuente: Elaboración Propia

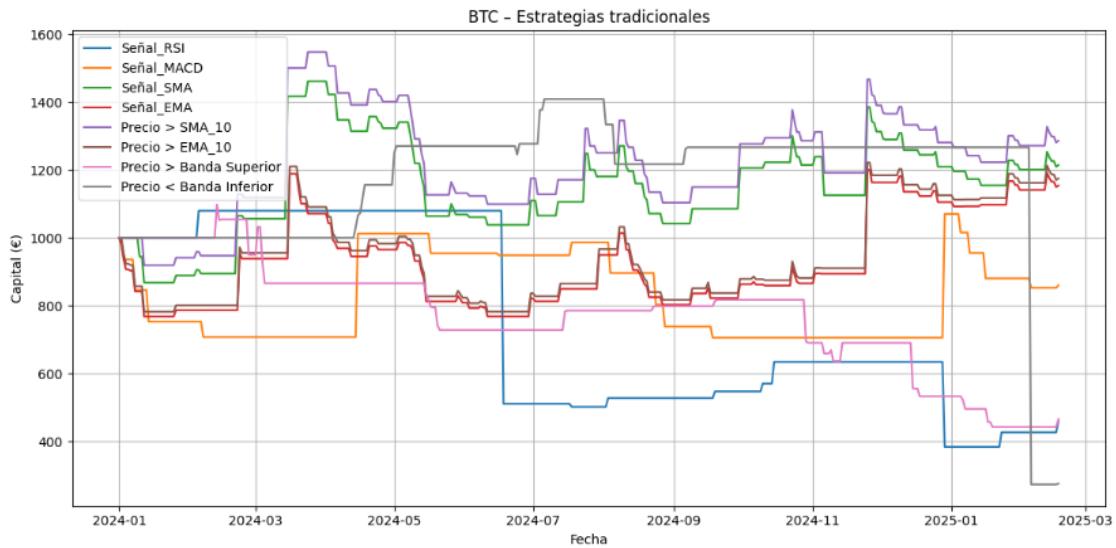


Gráfico 2.1.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales, con posición siempre abierta y con Window Size 60. Fuente: Elaboración Propia

Tabla 2.2: Tabla comparativa BTC con posición siempre abierta con Window Size 40.
Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	+20.37 %	1,203.73
Señal_RSI	-54.00 %	459.98
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1,214.53
Señal_EMA	+15.42 %	1,154.21
Precio > SMA_10	+28.62 %	1,286.18
Precio > EMA_10	+17.52 %	1,175.24
Precio > Banda Superior	-53.46 %	465.44
Precio < Banda Inferior	-72.48 %	275.16

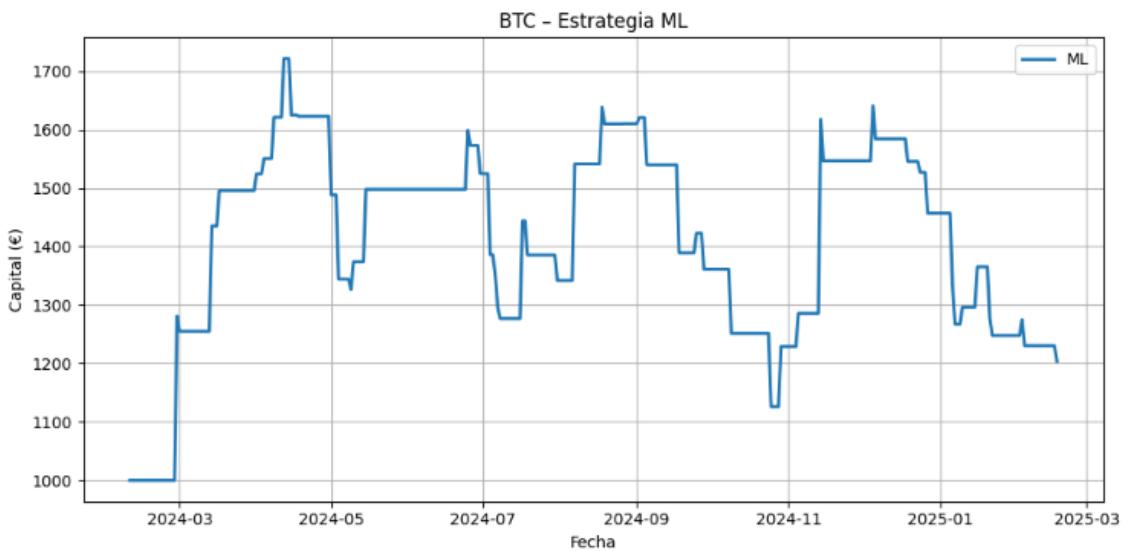


Gráfico 2.2.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning, con posición siempre abierta y con Window Size 40. Fuente: Elaboración Propia

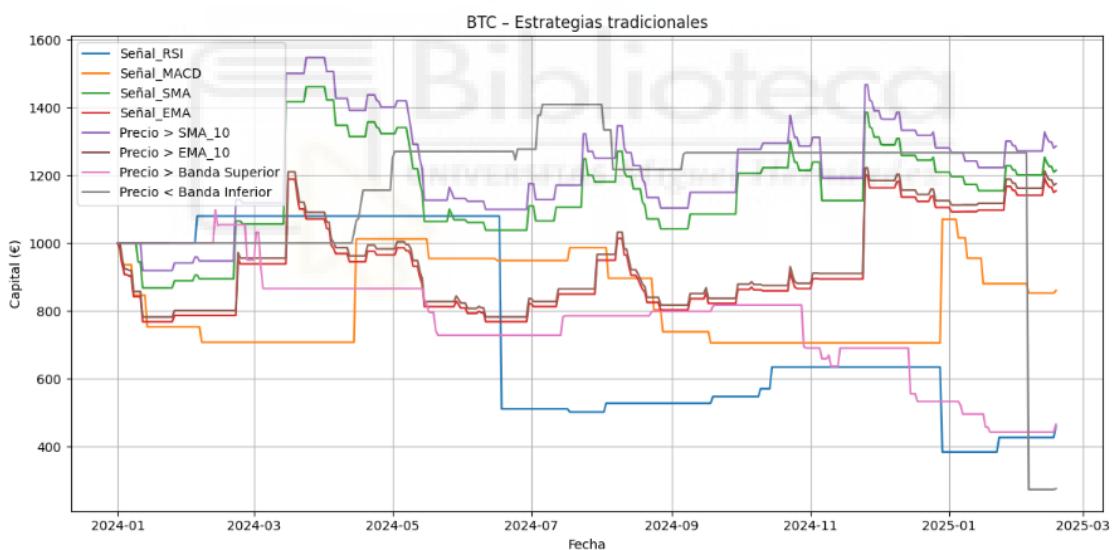


Gráfico 2.2.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales, con posición siempre abierta y con Window Size 40. Fuente: Elaboración Propia

Tabla 2.3: Tabla comparativa BTC con posición siempre abierta con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-53.53 %	464.70

Señal_RSI	-54.00 %	459.98
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1,214.53
Señal_EMA	+15.42 %	1,154.21
Precio > SMA_10	+28.62 %	1,286.18
Precio > EMA_10	+17.52 %	1,175.24
Precio > Banda Superior	-53.46 %	465.44
Precio < Banda Inferior	-72.48 %	275.16



Gráfico 2.3.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning, con posición siempre abierta y con Window Size 20. Fuente: Elaboración Propia

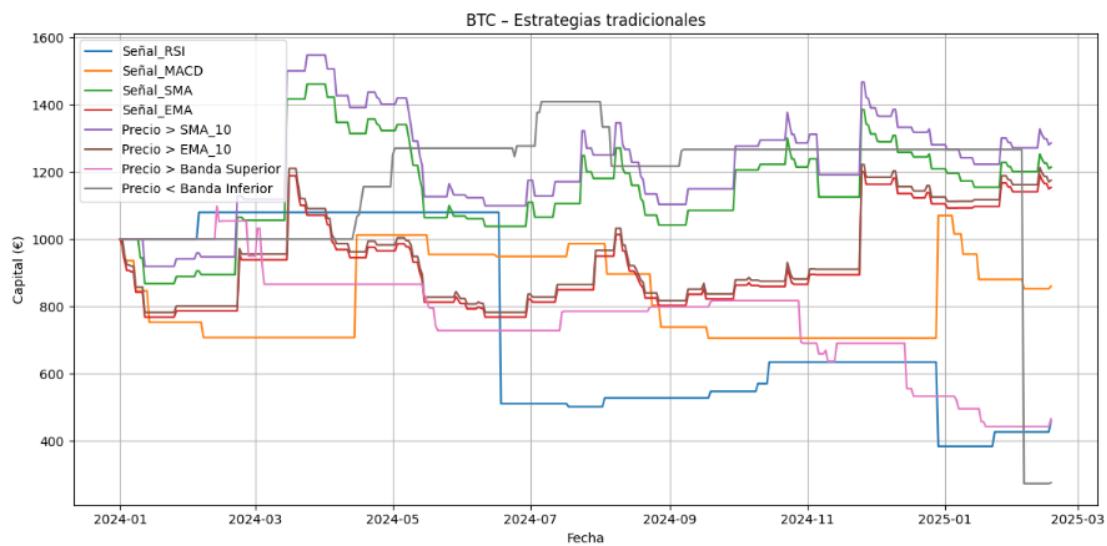


Gráfico 2.3.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales, con posición siempre abierta y con Window Size 20. Fuente: Elaboración Propia

Tabla 2.4: Tabla comparativa BTC con posición siempre abierta con Window Size 10.

Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-62.88 %	371.22
Señal_RSI	-54.00 %	459.98
Señal_MACD	-14.01 %	859.92
Señal_SMA	+21.45 %	1,214.53
Señal_EMA	+15.42 %	1,154.21
Precio > SMA_10	+28.62 %	1,286.18
Precio > EMA_10	+17.52 %	1,175.24
Precio > Banda Superior	-53.46 %	465.44
Precio < Banda Inferior	-72.48 %	275.16



Gráfico 2.4.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning, con posición siempre abierta y con Window Size 10. Fuente: Elaboración Propia

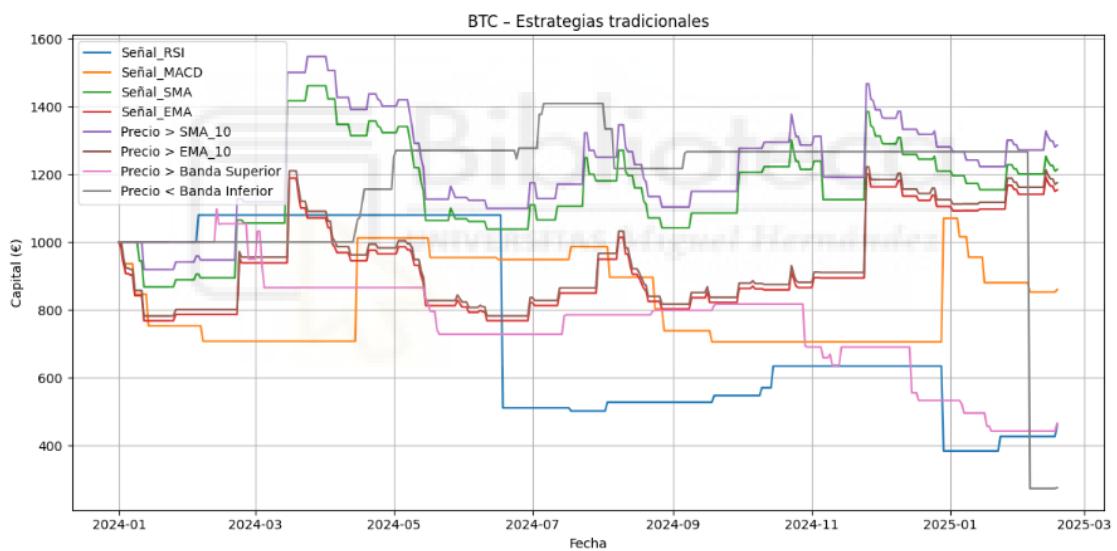


Gráfico 2.4.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales, con posición siempre abierta y con Window Size 10. Fuente: Elaboración Propia

14.3. Nasdaq100

Antes de utilizar de los efectos del apalancamiento y la gestión del riesgo mediante Stop-loss, se llevó a cabo un análisis preliminar del rendimiento de las estrategias sin apalancamiento ni límites de pérdidas, utilizando diferentes valores de window size para

la estrategia de Machine Learning (ML). La inversión inicial fue de 1.000 € en todas las situaciones, y se analizó el desarrollo hasta el cierre del periodo temporal disponible.

Los hallazgos indican que las estrategias tradicionales permanecen constantes a lo largo de las diversas ventanas temporales, con pequeñas fluctuaciones en el capital final. Por ejemplo, tácticas como Precio > Banda Superior, MACD o Precio > SMA_10 generaron ganancias moderadas en donde en la tabla 3.1, 3.2, 3.3 y 3.4 y los gráficos 3.1.2, 3.2.2, 3.3.2 y 3.4.2 nos muestran rentabilidades entre +3 % y +4,5 %, mientras que otras como Precio < Banda Inferior o EMA muestran pequeñas pérdidas, alrededor de -1,5 % y -2,9 %, como nos muestran las tablas 3.1, 3.2, 3.3 y 3.4 respectivamente.

A diferencia de esto, la estrategia de Machine Learning (ML) muestra una mayor sensibilidad al parámetro tamaño de ventana. Para un tamaño de ventana de 20, es donde presenta el mejor rendimiento, con una leve pérdida de -1,60 % que podemos ver en la tabla 3.1 y en el gráfico 3.1.1. No obstante, al disminuir o aumentar la ventana, el desempeño disminuye: por ejemplo, con un tamaño de ventana de 40 en la tabla 3.2 y en el gráfico 3.2.1 y un tamaño de 60 en la tabla 3.1 se observan pérdidas del -9,57 % y -7,15 %, respectivamente. Esto indica que, sin apalancamiento y sin mecanismos de protección como el Stop-loss, el Machine Learning necesita un ajuste de su ventana de entrenamiento para igualar el rendimiento de las estrategias tradicionales.

En conclusión, sin el uso de apalancamiento ni la administración de pérdidas, las estrategias tradicionales presentan una mayor estabilidad, mientras que el Machine Learning presenta una variabilidad más alta en su desempeño, según la configuración del modelo.

Tabla 3.1: Tabla comparativa Nasdaq100 con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-7.15 %	928.48
Señal_RSI	-2.77 %	972.31
Señal_MACD	+3.42 %	1,034.16

Señal_SMA	-0.77 %	992.32
Señal_EMA	-2.89 %	971.10
Precio > SMA_10	+3.12 %	1,031.20
Precio > EMA_10	+1.03 %	1,010.33
Precio > Banda Superior	+4.46 %	1,044.61
Precio < Banda Inferior	-1.56 %	984.45

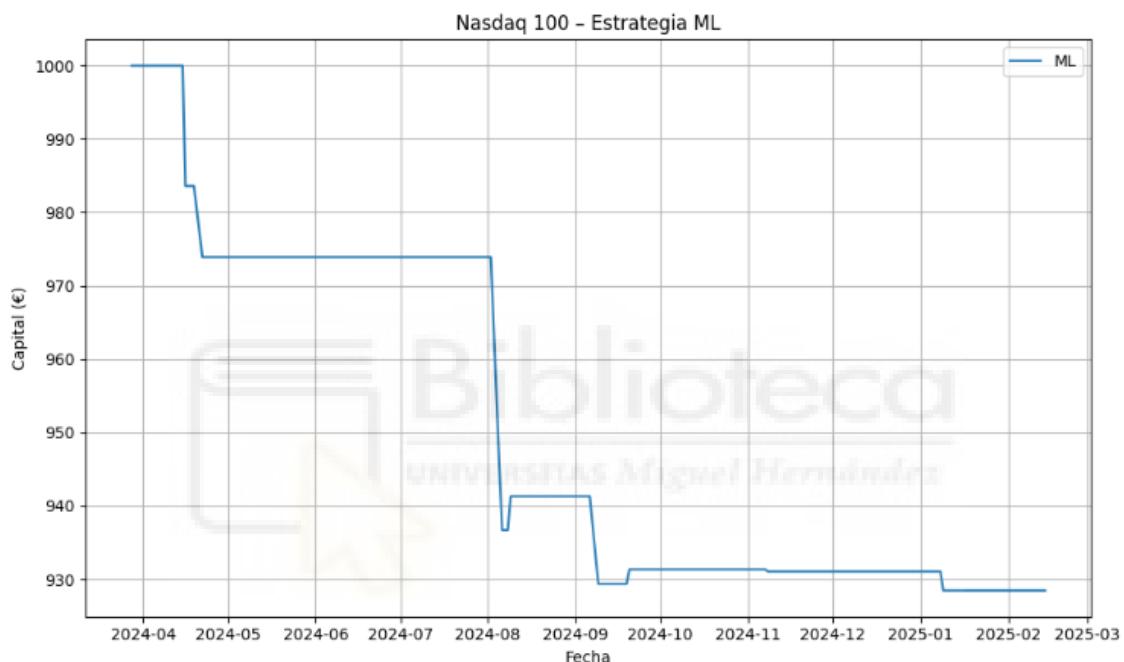


Gráfico 3.1.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con Window Size 60. Fuente: Elaboración Propia

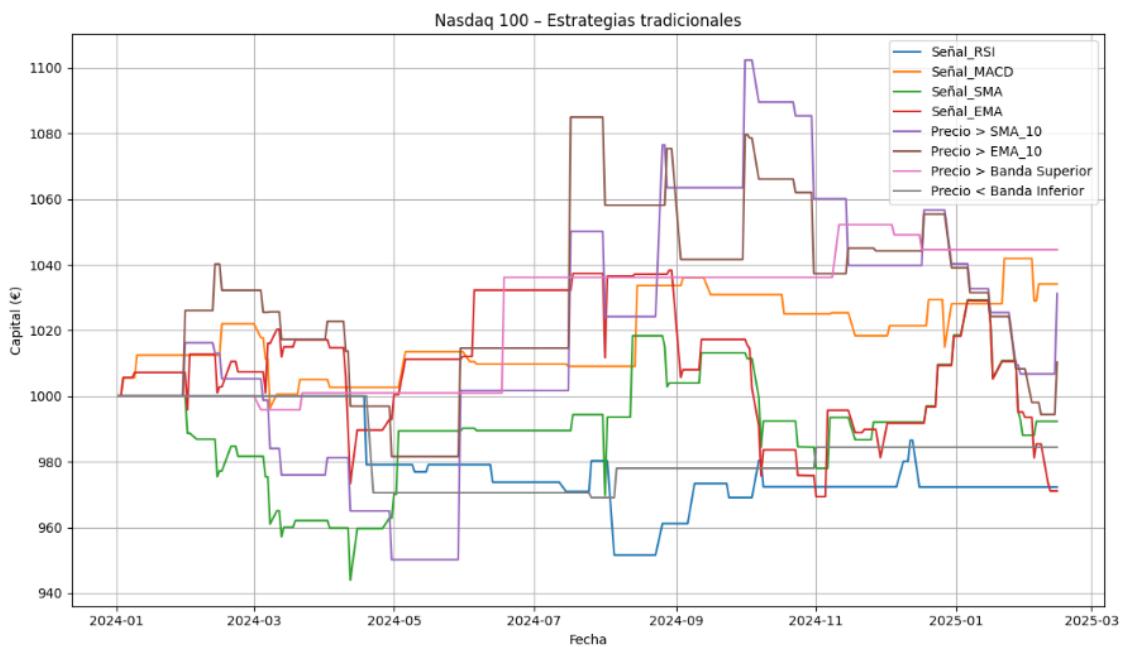


Gráfico 3.1.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con Window Size 60. Fuente: Elaboración Propia

Tabla 3.2: Tabla comparativa Nasdaq100 con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-9.57 %	904.27
Señal_RSI	-2.77 %	972.31
Señal_MACD	+3.42 %	1,034.16
Señal_SMA	-0.77 %	992.32
Señal_EMA	-2.89 %	971.10
Precio > SMA_10	+3.12 %	1,031.20
Precio > EMA_10	+1.03 %	1,010.33
Precio > Banda Superior	+4.46 %	1,044.61
Precio < Banda Inferior	-1.56 %	984.45

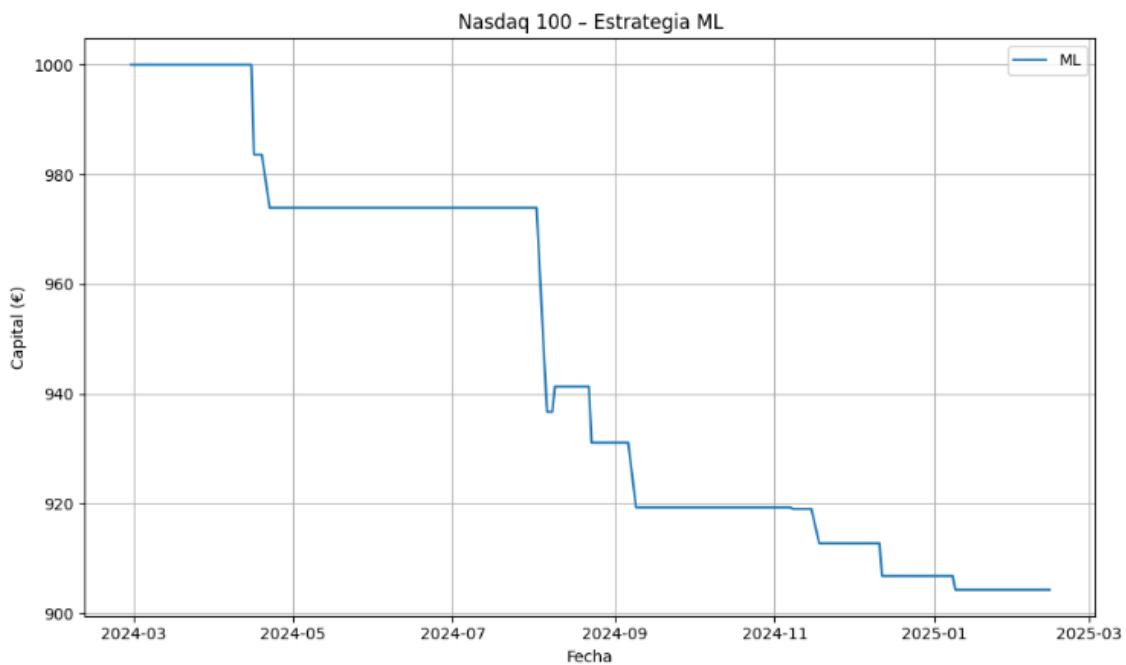


Gráfico 3.2.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con Window Size 40. Fuente: Elaboración Propia

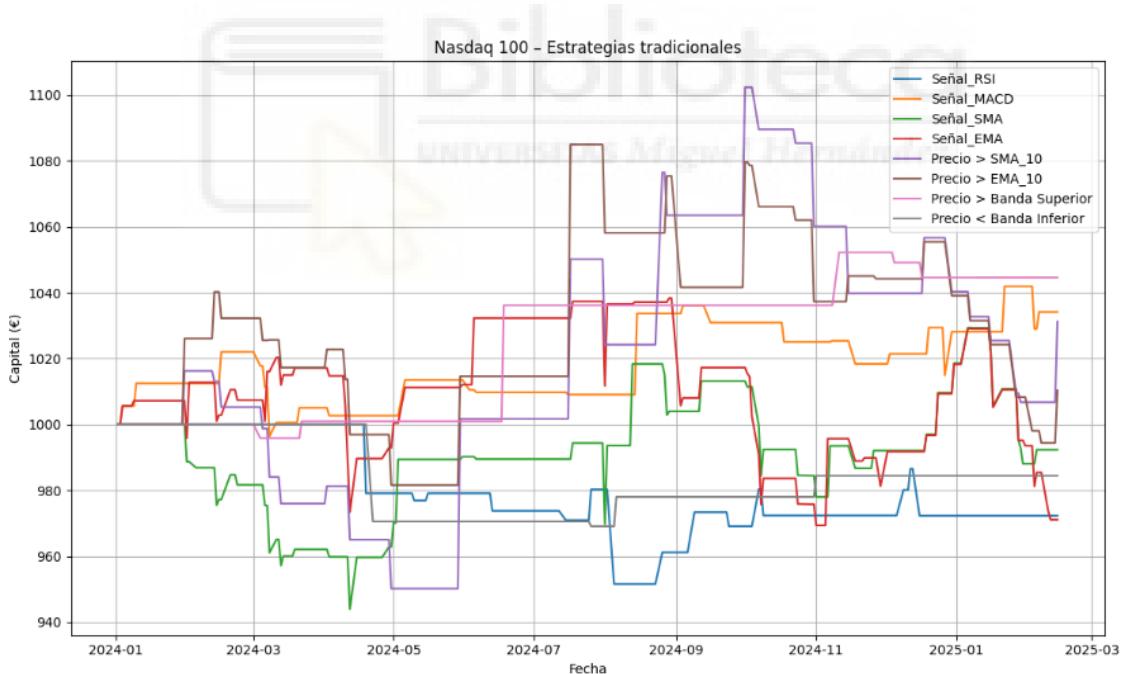


Gráfico 3.2.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con Window Size 40. Fuente: Elaboración Propia

Tabla 3.3: Tabla comparativa Nasdaq100 con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-1.60 %	983.96
Señal_RSI	-2.77 %	972.31
Señal_MACD	+3.42 %	1,034.16
Señal_SMA	-0.77 %	992.32
Señal_EMA	-2.89 %	971.10
Precio > SMA_10	+3.12 %	1,031.20
Precio > EMA_10	+1.03 %	1,010.33
Precio > Banda Superior	+4.46 %	1,044.61
Precio < Banda Inferior	-1.56 %	984.45

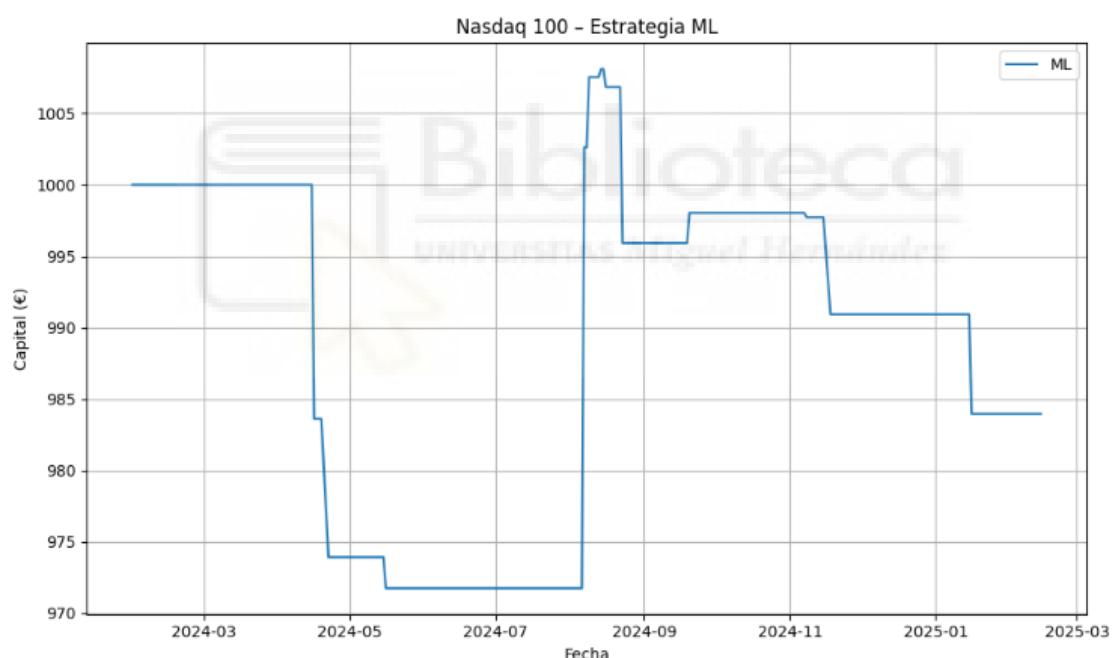


Gráfico 3.3.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con Window Size 20. Fuente: Elaboración Propia

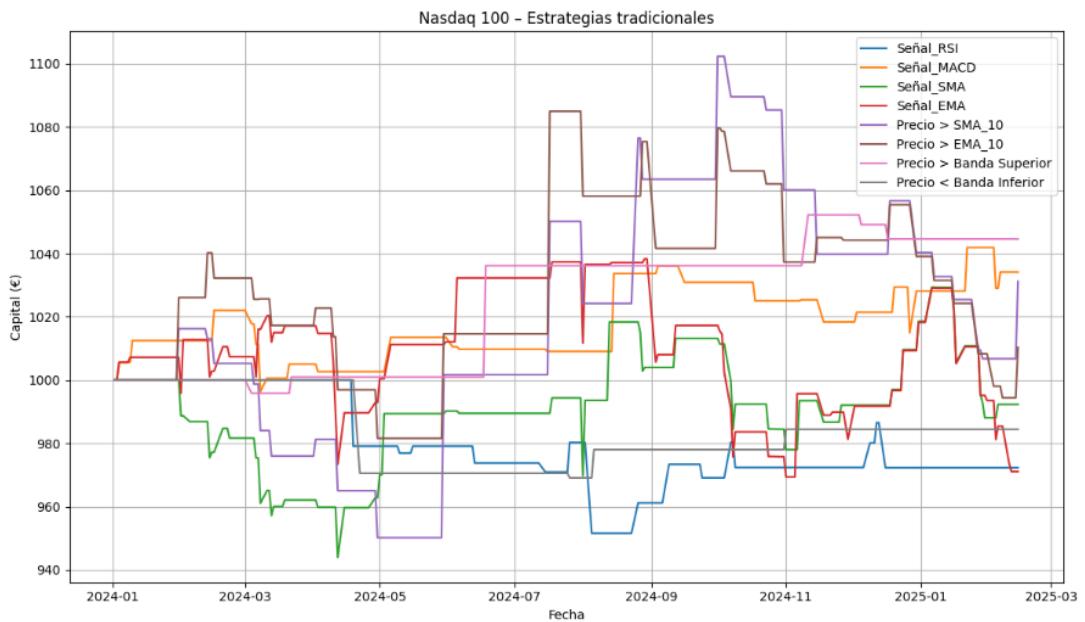


Gráfico 3.3.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con Window Size 20. Fuente: Elaboración Propia

Tabla 3.4: Tabla comparativa Nasdaq100 con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-5.11 %	948.92
Señal_RSI	-2.77 %	972.31
Señal_MACD	+3.42 %	1,034.16
Señal_SMA	-0.77 %	992.32
Señal_EMA	-2.89 %	971.10
Precio > SMA_10	+3.12 %	1,031.20
Precio > EMA_10	+1.03 %	1,010.33
Precio > Banda Superior	+4.46 %	1,044.61
Precio < Banda Inferior	-1.56 %	984.45

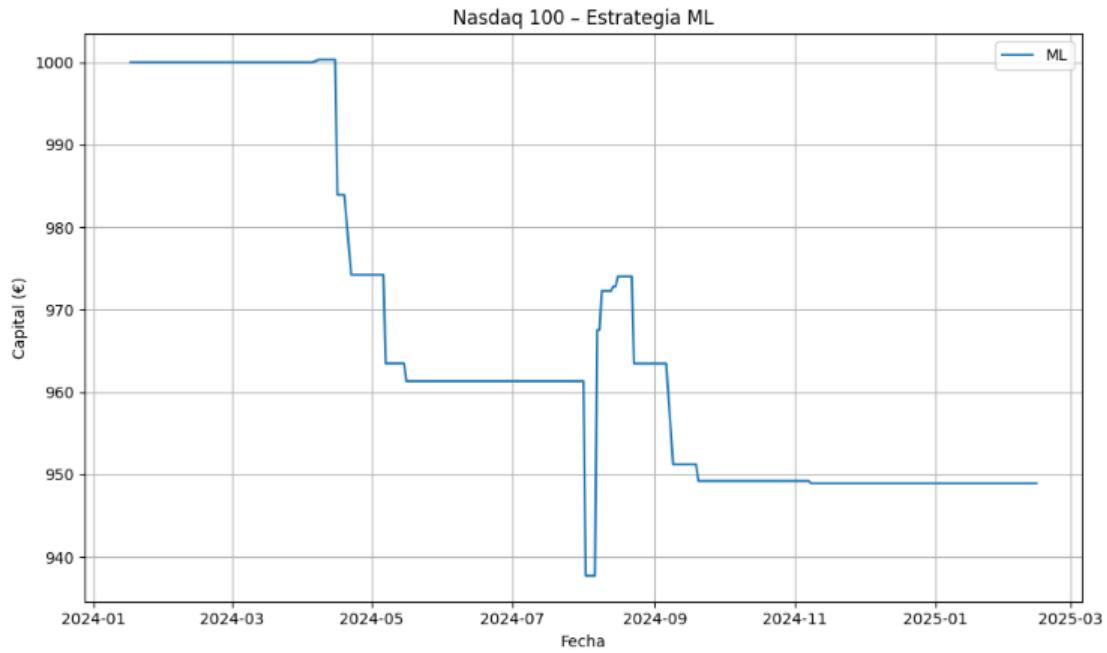


Gráfico 3.4.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con Window Size 10. Fuente: Elaboración Propia

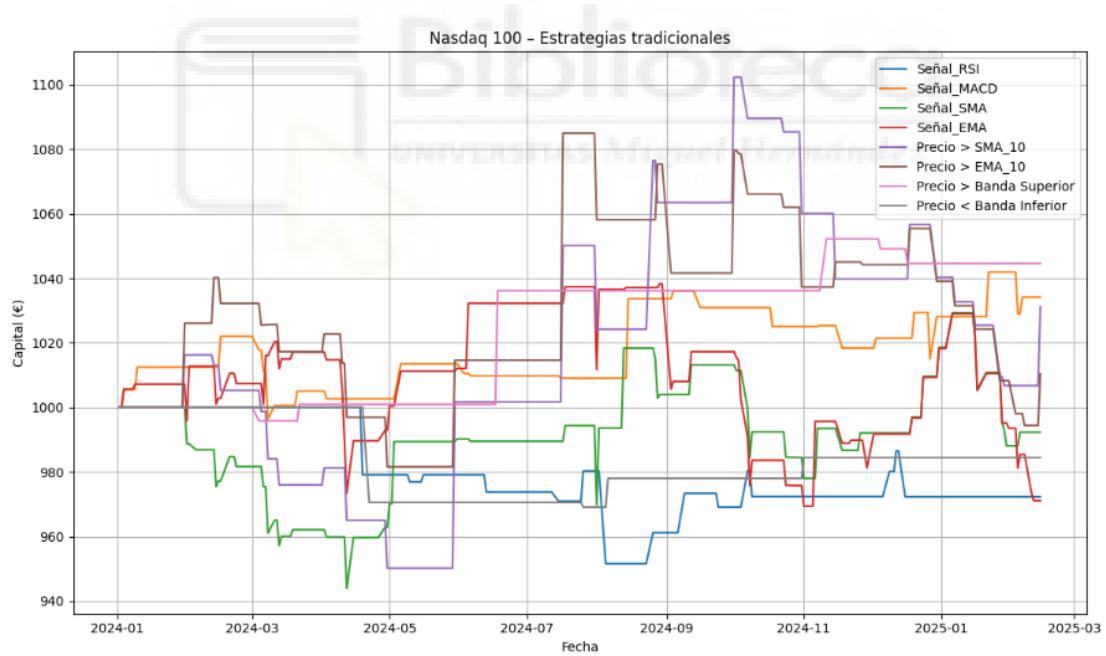


Gráfico 3.4.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con Window Size 10. Fuente: Elaboración Propia

14.4. Nasdaq100 con posición siempre abierta

En este análisis final se examinan los resultados de las estrategias implementadas en el Nasdaq100 sin utilizar apalancamiento ni Stop-loss, conservando las posiciones abiertas hasta el cierre del periodo, aun en ausencia de señales contrarias. Bajo estas circunstancias, se nota una mejora notable en el desempeño de diversas estrategias tradicionales en comparación con los escenarios previos.

En particular, las estrategias tradicionales como “Precio > SMA_10” y “Precio > EMA_10” presentan resultados notables y consistentes, con rendimientos acumulados cercanos al +30 %, sin importar el tamaño de ventana (window size) utilizado (Tabla 4.1, 4.2, 4.3 y 4.4). Esto indica que en un entorno de tendencia duradera, sostener posiciones abiertas de forma continua puede optimizar las ganancias.

En cambio, el desempeño del modelo de Machine Learning se optimiza en comparación con sus variantes que utilizan Stop-loss o cierre instantáneo. A pesar de no superar las mejores estrategias tradicionales, se aprecia una tendencia favorable al disminuir el tamaño de la ventana, logrando una pérdida de apenas -8.25 % con window size = 10 como vemos en la tabla 4.4 y en el gráfico 4.4.1. Este comportamiento sugiere que el modelo tiene un mejor aprendizaje en períodos cortos.

Finalmente, otras estrategias clásicas como RSI o MACD se mantienen, sin grandes pérdidas ni ganancias con un +7.56% y un -7.88% respectivamente viendo las tablas 4.1, 4.2, 4.3 y 4.4 que dan los mismos resultados en las estrategias tradicionales.

Tabla 4.1: Tabla comparativa Nasdaq100 con posición si con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-28.45 %	715.47
Señal_RSI	+7.56 %	1,075.59
Señal_MACD	-7.88 %	921.17
Señal_SMA	-20.66 %	793.41
Señal_EMA	-24.39 %	756.12

Precio > SMA_10	+30.82 %	1,308.24
Precio > EMA_10	+32.09 %	1,320.91
Precio > Banda Superior	+21.04 %	1,210.38
Precio < Banda Inferior	+24.17 %	1,241.66

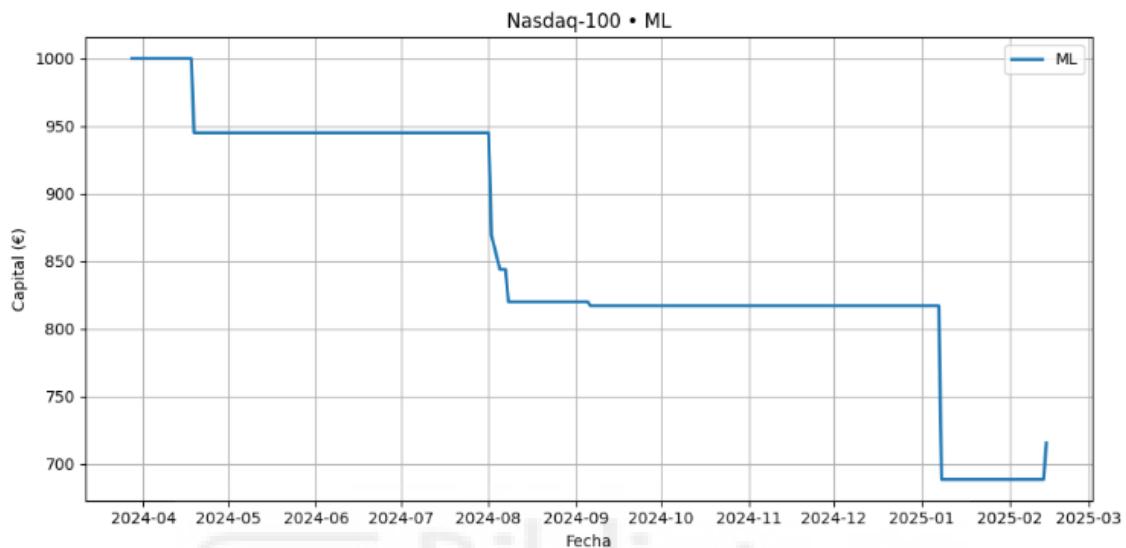


Gráfico 4.1.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con posición siempre abierta con Window Size 60. Fuente: Elaboración Propia

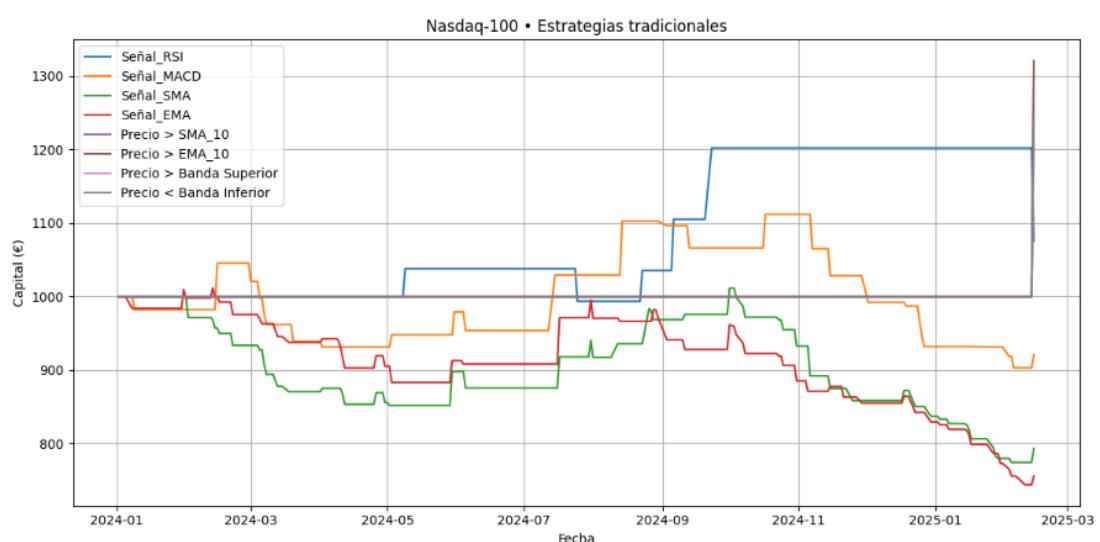


Gráfico 4.1.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con posición siempre abierta con Window Size 60. Fuente: Elaboración Propia

Tabla 4.2: Tabla comparativa Nasdaq100 con posición siempre abierta con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-21.73 %	782.74
Señal_RSI	+7.56 %	1,075.59
Señal_MACD	-7.88 %	921.17
Señal_SMA	-20.66 %	793.41
Señal_EMA	-24.39 %	756.12
Precio > SMA_10	+30.82 %	1,308.24
Precio > EMA_10	+32.09 %	1,320.91
Precio > Banda Superior	+21.04 %	1,210.38
Precio < Banda Inferior	+24.17 %	1,241.66

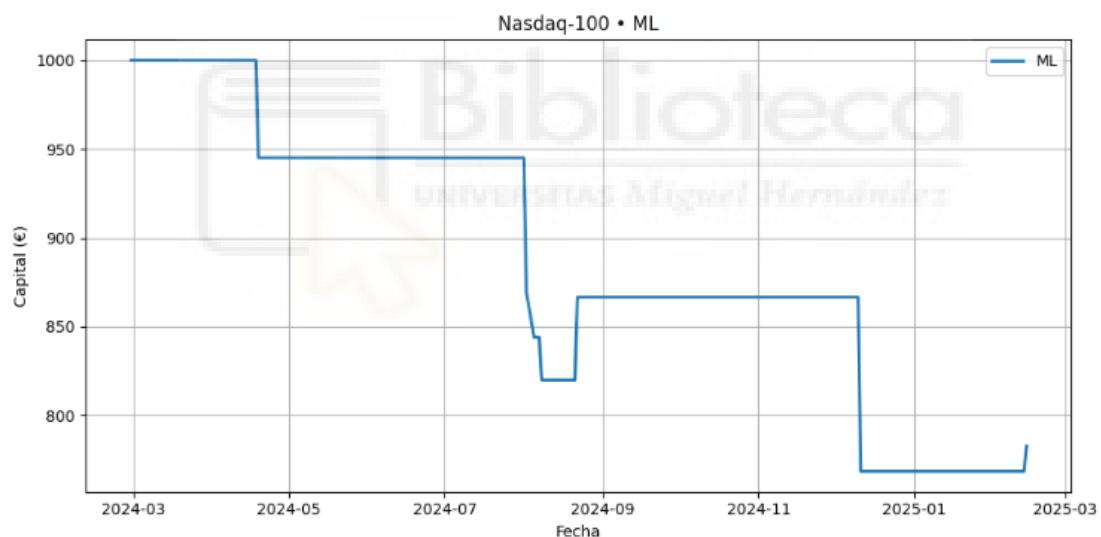


Gráfico 4.2.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con posición siempre abierta con Window Size 40. Fuente: Elaboración Propia

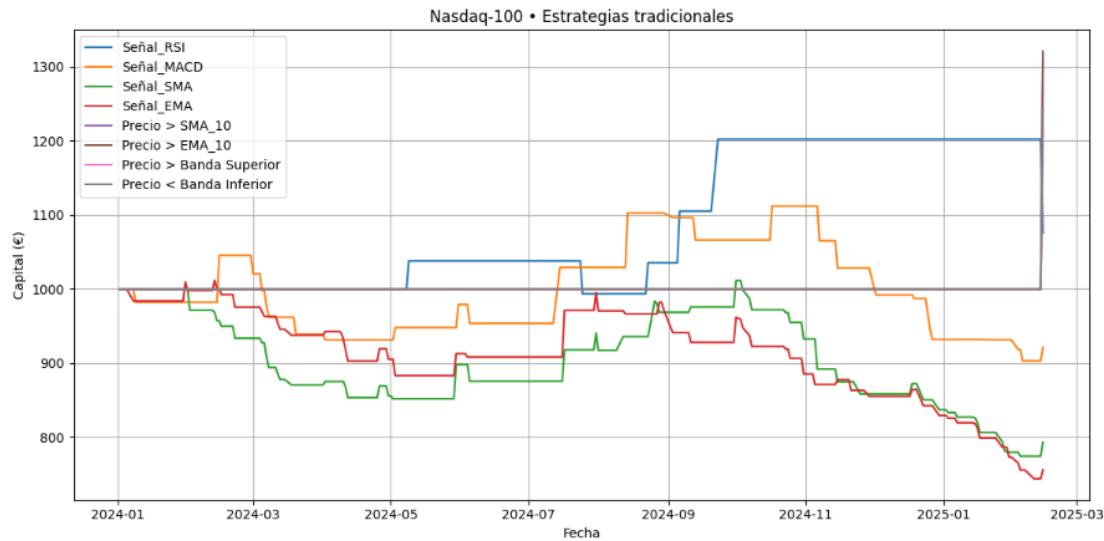


Gráfico 4.2.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con posición siempre abierta con Window Size 40. Fuente: Elaboración Propia

Tabla 4.3: Tabla comparativa Nasdaq100 con posición siempre abierta con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-14.76 %	852.40
Señal_RSI	+7.56 %	1,075.59
Señal_MACD	-7.88 %	921.17
Señal_SMA	-20.66 %	793.41
Señal_EMA	-24.39 %	756.12
Precio > SMA_10	+30.82 %	1,308.24
Precio > EMA_10	+32.09 %	1,320.91
Precio > Banda Superior	+21.04 %	1,210.38
Precio < Banda Inferior	+24.17 %	1,241.66

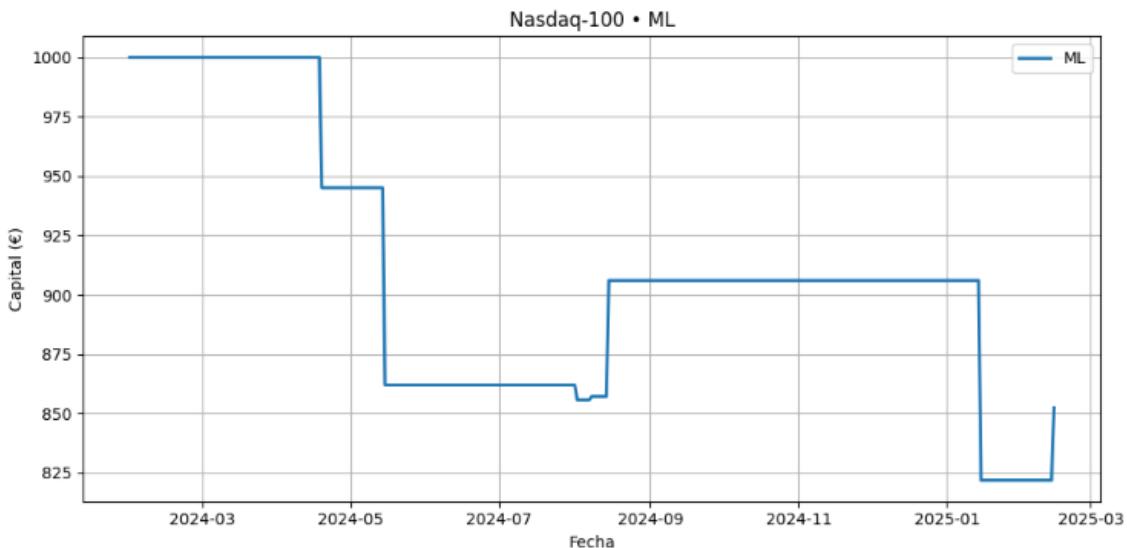


Gráfico 4.3.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con posición siempre abierta con Window Size 20. Fuente: Elaboración Propia

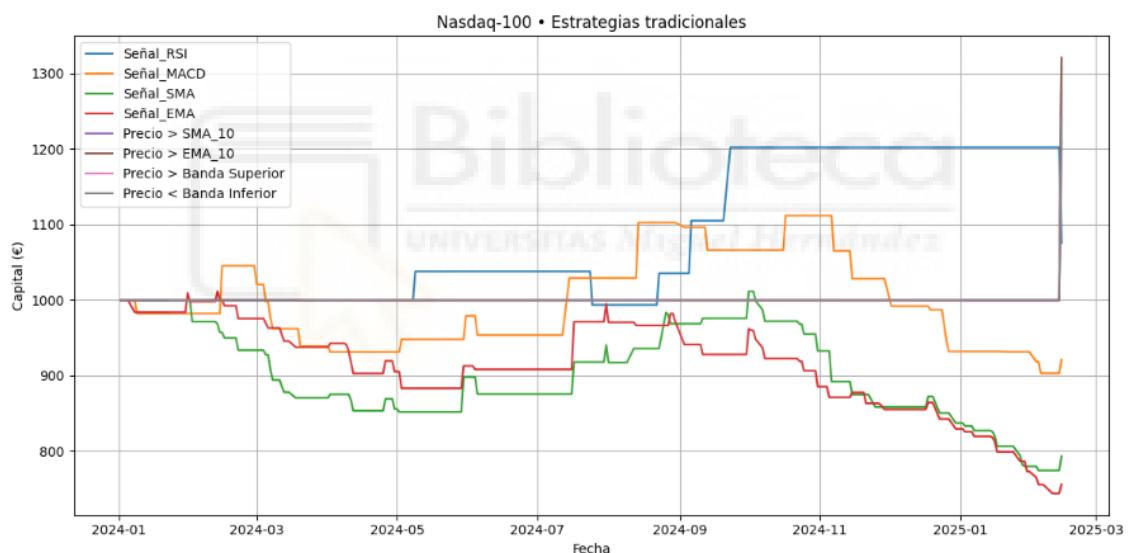


Gráfico 4.3.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con posición siempre abierta con Window Size 20. Fuente: Elaboración Propia

Tabla 4.4: Tabla comparativa Nasdaq100 con posición siempre abierta con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-8.25 %	917.46
Señal_RSI	+7.56 %	1,075.59

Señal_MACD	-7.88 %	921.17
Señal_SMA	-20.66 %	793.41
Señal_EMA	-24.39 %	756.12
Precio > SMA_10	+30.82 %	1,308.24
Precio > EMA_10	+32.09 %	1,320.91
Precio > Banda Superior	+21.04 %	1,210.38
Precio < Banda Inferior	+24.17 %	1,241.66

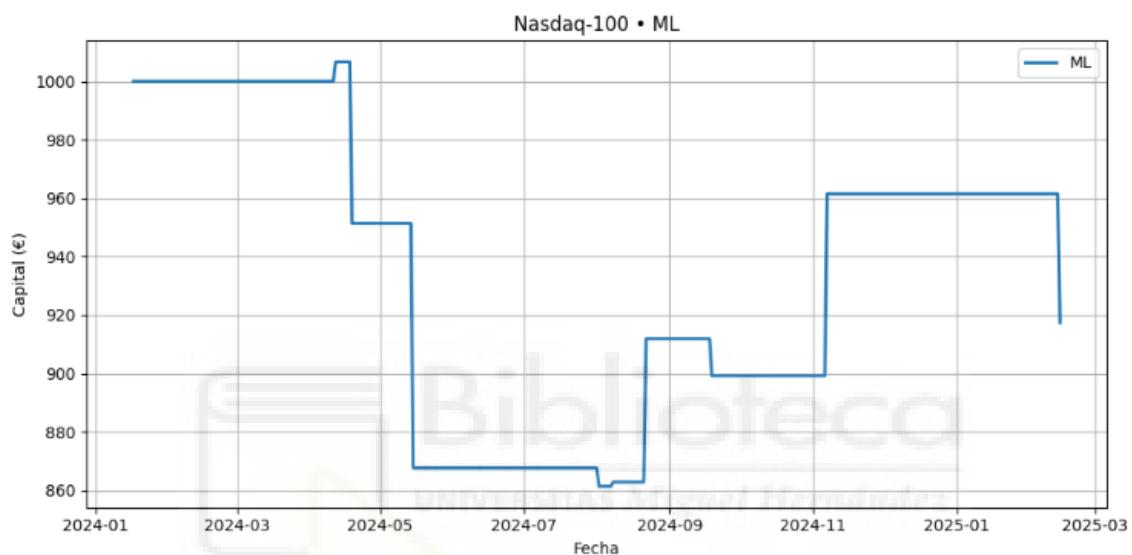


Gráfico 4.4.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con posición siempre abierta con Window Size 10. Fuente: Elaboración Propia

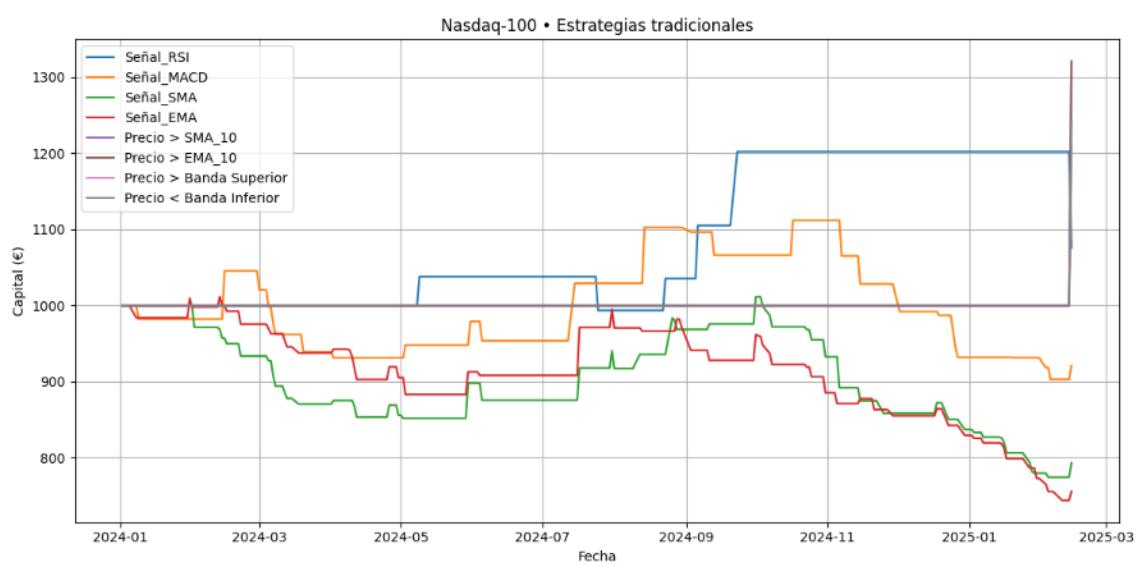


Gráfico 4.4.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con posición siempre abierta con Window Size 10. Fuente: Elaboración Propia

14.5. Bitcoin, Machine Learning con apalancamiento y Stop-loss

El análisis de BTC en un entorno de backtesting con apalancamiento ($\times 5$ y $\times 10$) y un Stop-loss del 10 % revela una clara variación en el rendimiento de las estrategias dependiendo del tamaño de ventana empleado en el modelo de Machine Learning (ML). El parámetro tamaño de ventana indica la cantidad de días previos considerados como historial para entrenar el modelo y hacer la predicción.

Por ejemplo, utilizando apalancamiento $\times 10$, la estrategia "Precio < Banda Inferior" asegura un rendimiento positivo y constante (+67.49 %, capital final: 1,674.95 €) como se observa en la tabla 5.1, aunque con un apalancamiento * 10 se vuelve negativa la rentabilidad dando un -11.79%. Por otro lado, "Precio > EMA_10" logra rendimientos acumulados que superan el 250 % en diversos escenarios como se aprecia en las tablas 5.3, 5.5 y 5.7. y en los gráficos 5.3.2, 5.5.2 y 5.7.2. Concluimos que a mayor tamaño de ventana mayor rentabilidad.

En contraste, el modelo de Machine Learning ha mostrado un desempeño muy desfavorable, especialmente con tamaños de ventana pequeños (10–20), donde el capital final puede descender por debajo de los 20 € o incluso mostrar valores negativos (por el apalancamiento y la activación del Stop-loss en situaciones adversas), tal y como dicen las tablas 5.5, 5.6, 5.7 y 5.8.

También, la "Señal_MACD" y "Señal_SMA", obtienen resultados razonables con determinados tamaños de ventana y bajo apalancamiento, pero su rendimiento disminuye rápidamente al aumentar el riesgo.

En resumen, las estrategias tradicionales que responden a situaciones de precios relativos, como aquellas fundamentadas en el cruce con bandas de Bollinger o promedios móviles, demuestran ser más efectivas y consistentes en condiciones de mercado desfavorables con apalancamiento, en comparación con un modelo de Machine Learning que no se adapta correctamente. Esto indica que, en contextos inestables como BTC, las estrategias

simples pero bien estructuradas pueden ser más efectivas que modelos complicados que no estén ajustados adecuadamente.

Tabla 5.1: Tabla comparativa BTC con apalancamiento *5 con StopLoss del 10% con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-90.37 %	96.27
Señal_RSI	-76.90 %	231.01
Señal_MACD	-75.44 %	245.64
Señal_SMA	-78.49 %	215.06
Señal_EMA	-70.43 %	295.74
Precio > SMA_10	+31.81 %	1,318.07
Precio > EMA_10	+44.27 %	1,442.68
Precio > Banda Superior	-58.36 %	416.39
Precio < Banda Inferior	+67.50 %	1,674.95



Gráfico 5.1.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

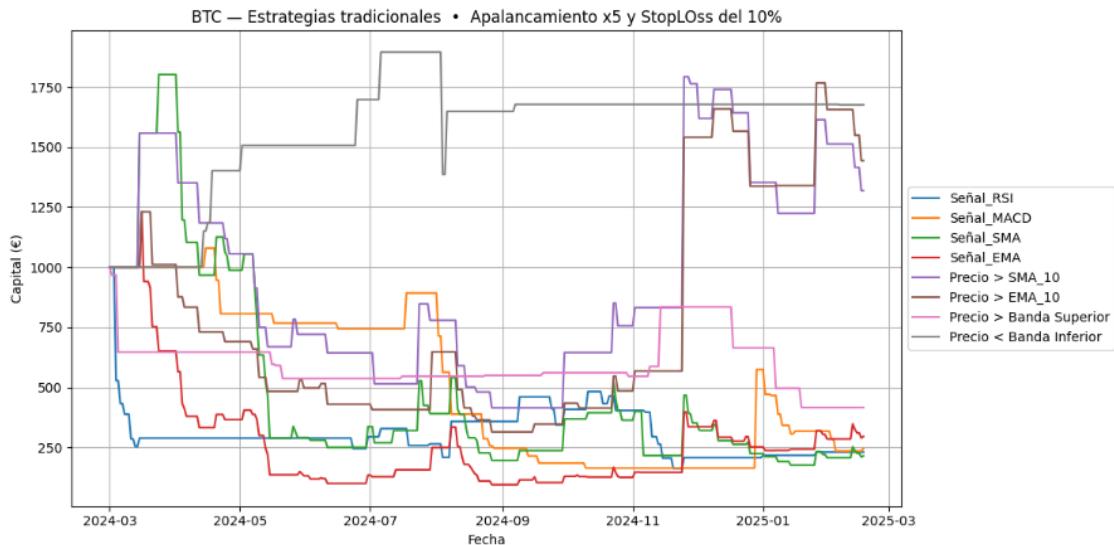


Gráfico 5.1.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

Tabla 5.2: Tabla comparativa BTC con apalancamiento *10 con StopLoss del 10% con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-99.47 %	5.35
Señal_RSI	-85.10 %	148.97
Señal_MACD	-99.44 %	5.60
Señal_SMA	-99.94 %	0.60
Señal_EMA	-99.71 %	2.87
Precio > SMA_10	-75.65 %	243.46
Precio > EMA_10	-82.62 %	173.82
Precio > Banda Superior	-90.73 %	92.72
Precio < Banda Inferior	-11.79 %	882.10

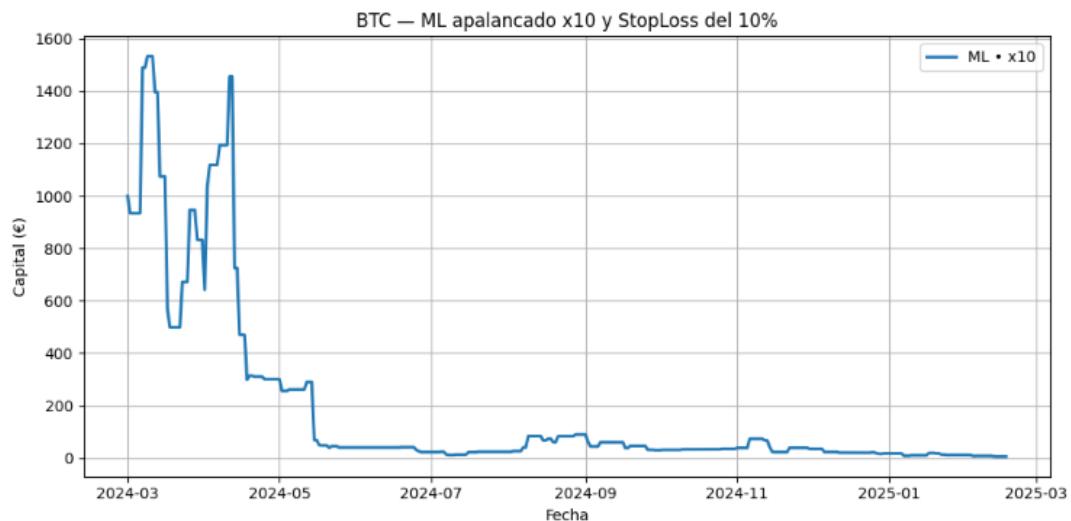


Gráfico 5.2.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

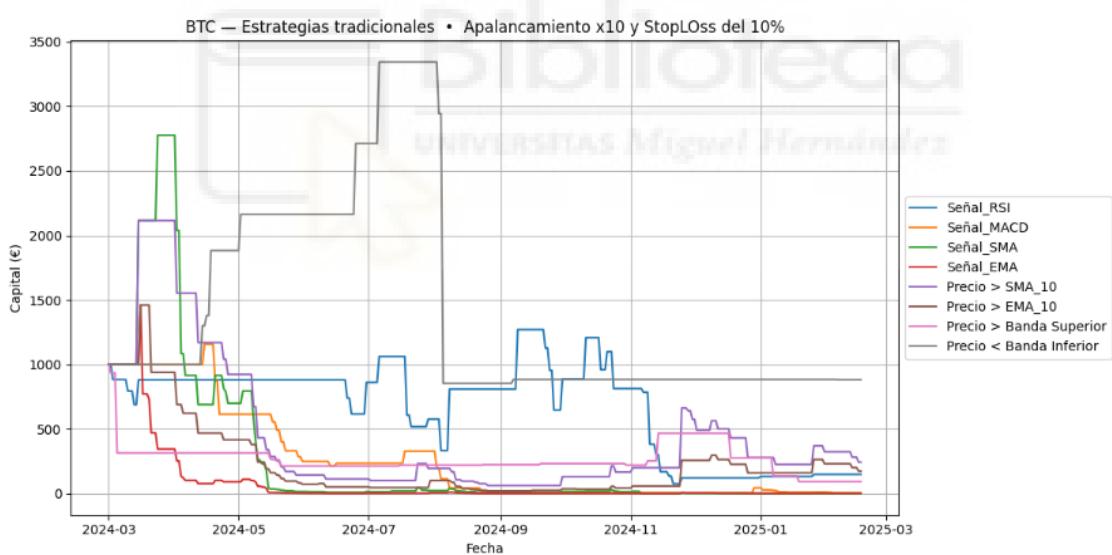


Gráfico 5.2.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

Tabla 5.3: Tabla comparativa BTC con apalancamiento *5 con StopLoss del 10% con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-85.51 %	144.91
Señal_RSI	-88.90 %	111.01
Señal_MACD	-39.93 %	600.72
Señal_SMA	-50.92 %	490.78
Señal_EMA	-32.62 %	673.78
Precio > SMA_10	+213.90 %	3,138.97
Precio > EMA_10	+258.31 %	3,583.12
Precio > Banda Superior	-24.76 %	752.35
Precio < Banda Inferior	+67.49 %	1,674.95



*Gráfico 5.3.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración*

Propia

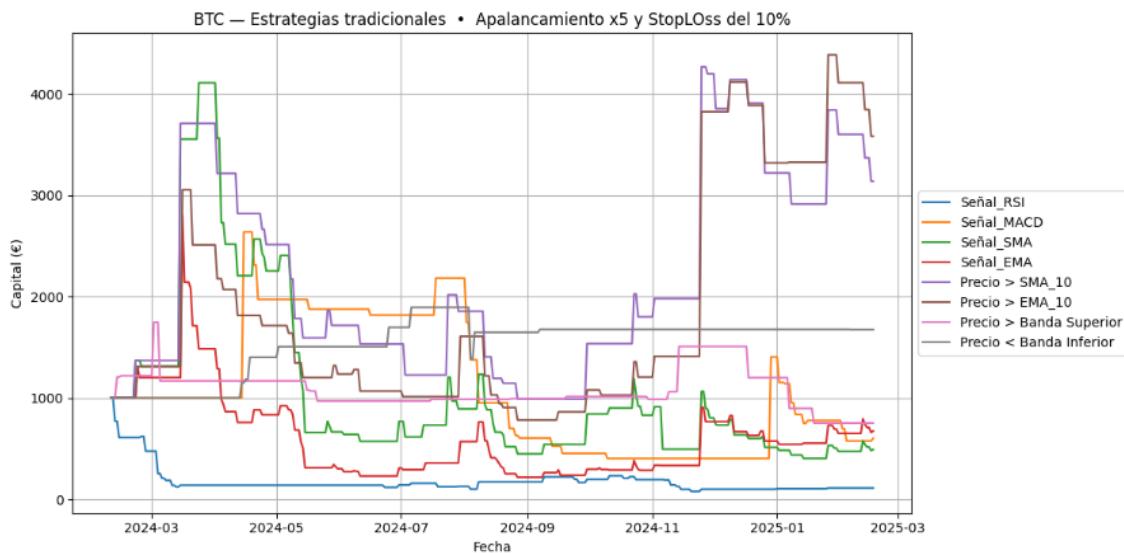


Gráfico 5.3.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

Tabla 5.4: Tabla comparativa BTC con apalancamiento *10 con StopLoss del 10% con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-97.95 %	20.54
Señal_RSI	-95.37 %	46.29
Señal_MACD	-97.93 %	20.68
Señal_SMA	-99.80 %	1.99
Señal_EMA	-99.03 %	9.74
Precio > SMA_10	-11.58 %	884.21
Precio > EMA_10	-29.34 %	706.59
Precio > Banda Superior	-73.27 %	267.29
Precio < Banda Inferior	-11.79 %	882.10

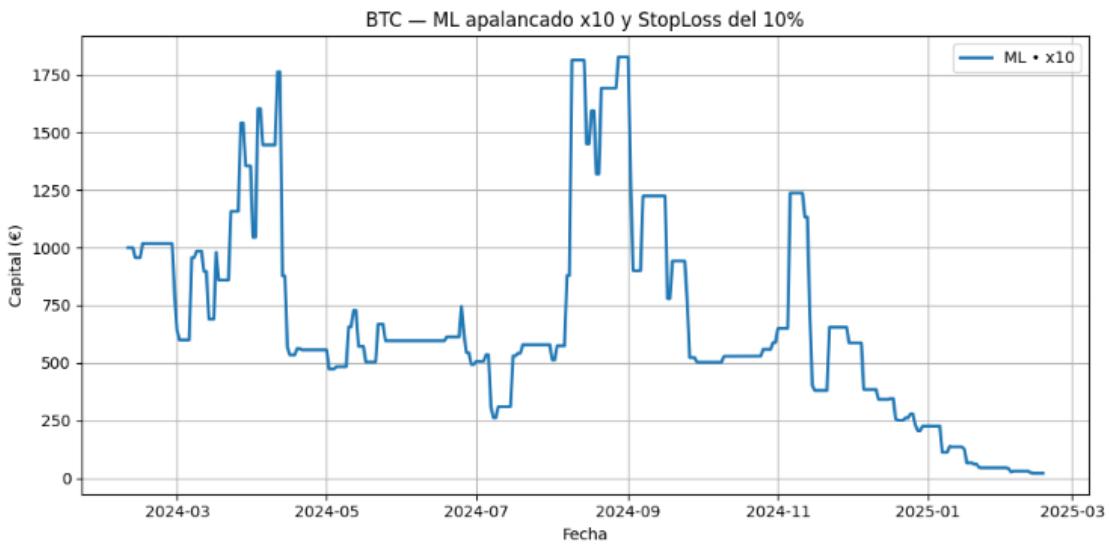


Gráfico 5.4.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

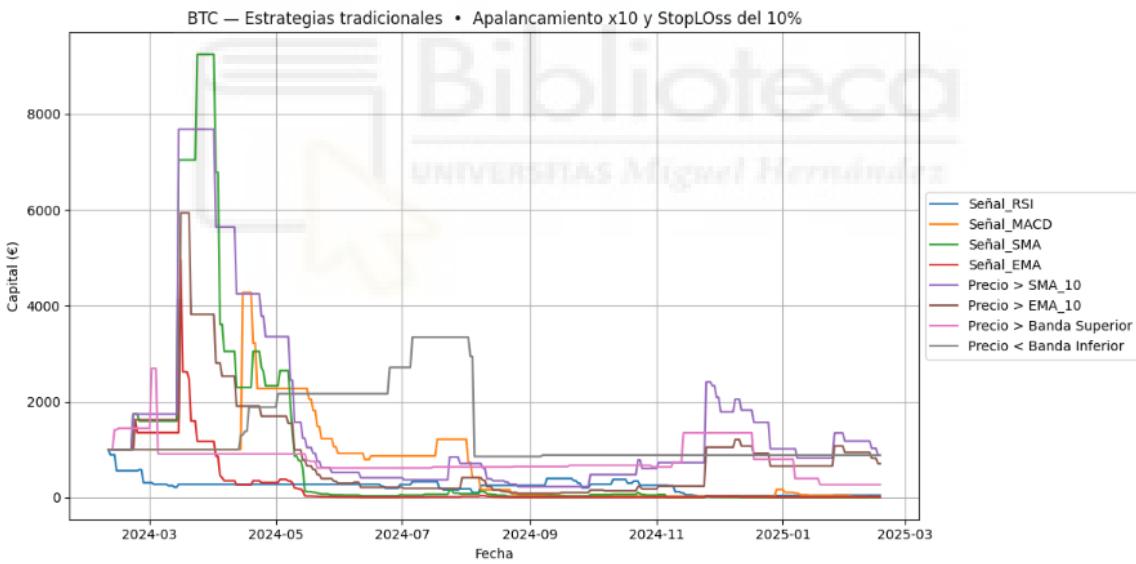


Gráfico 5.4.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

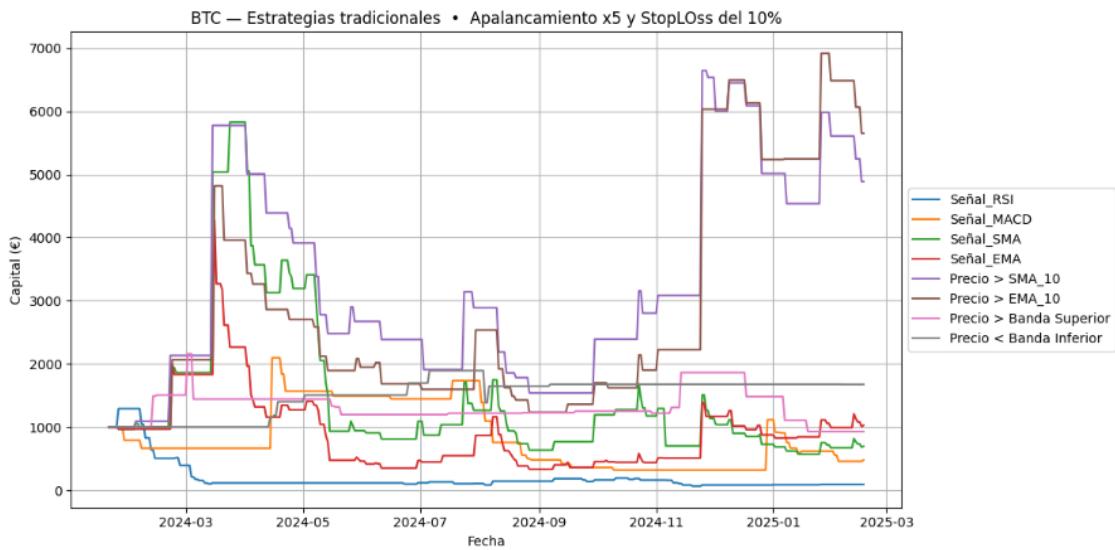
Tabla 5.5: Tabla comparativa BTC con apalancamiento *5 con StopLoss del 10% con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
------------	-----------------	-------------------

ML	-96.36 %	36.39
Señal_RSI	-90.80 %	92.02
Señal_MACD	-52.25 %	477.55
Señal_SMA	-30.42 %	695.80
Señal_EMA	+2.80 %	1,028.03
Precio > SMA_10	+388.79 %	4,887.91
Precio > EMA_10	+465.14 %	5,651.37
Precio > Banda Superior	-7.06 %	929.37
Precio < Banda Inferior	+67.49 %	1,674.95



Gráfico 5.5.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia



. *Gráfico 5.5.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia*

Tabla 5.6: Tabla comparativa BTC con apalancamiento *10 con StopLoss del 10% con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-100.85 %	-8.52
Señal_RSI	-97.34 %	26.56
Señal_MACD	-98.94 %	10.60
Señal_SMA	-99.68 %	3.24
Señal_EMA	-98.24 %	17.60
Precio > SMA_10	+74.90 %	1,748.95
Precio > EMA_10	+36.62 %	1,366.24
Precio > Banda Superior	-62.53 %	374.70
Precio < Banda Inferior	-11.79 %	882.10

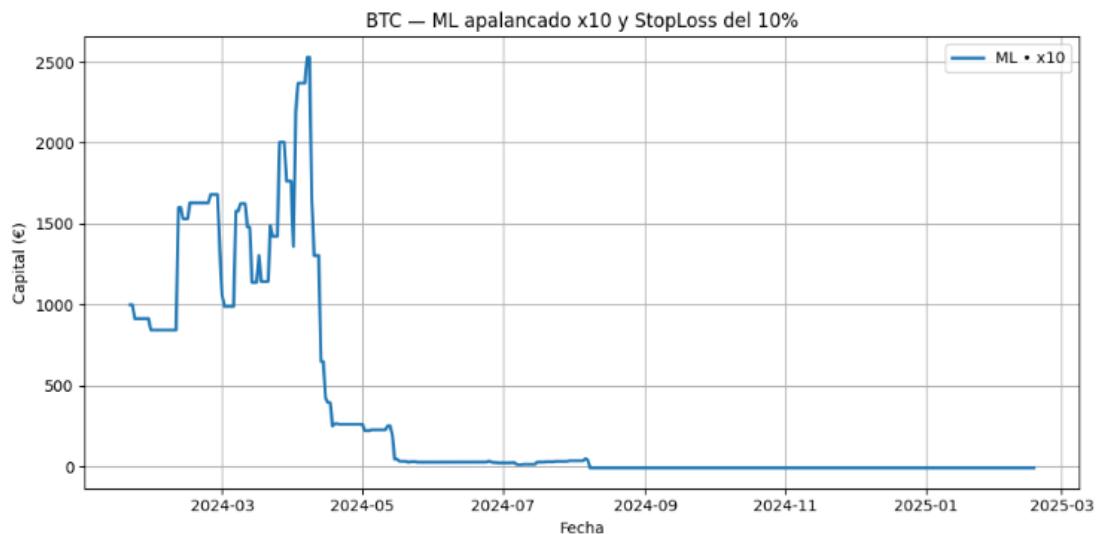


Gráfico 5.6.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

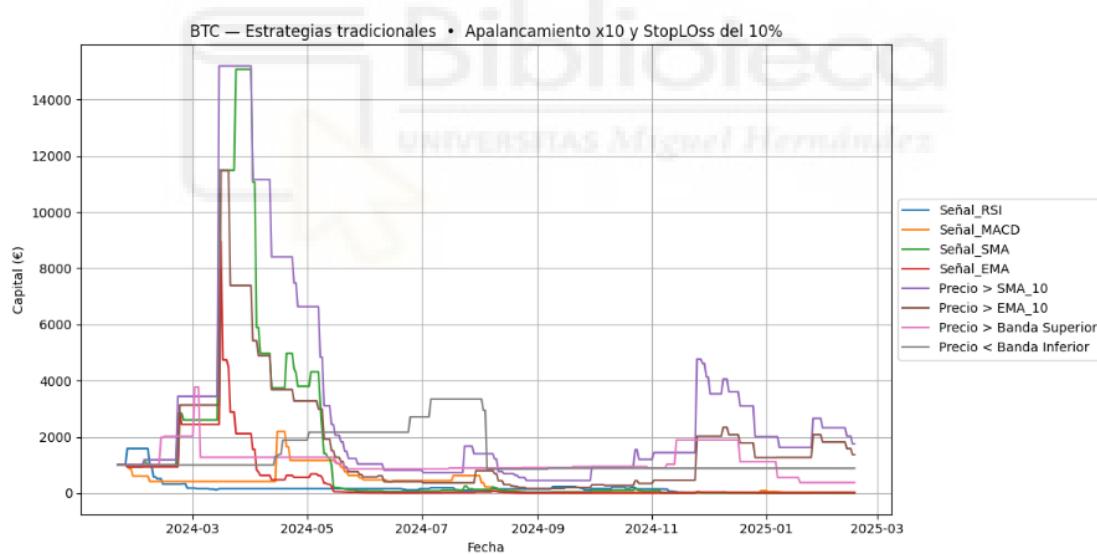


Gráfico 5.6.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

Tabla 5.7: Tabla comparativa BTC con apalancamiento *5 con StopLoss del 10% con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-98.67 %	13.32
Señal_RSI	-90.80 %	92.02
Señal_MACD	-73.24 %	267.55
Señal_SMA	-49.94 %	500.61
Señal_EMA	-26.04 %	739.64
Precio > SMA_10	+203.50 %	3,035.03
Precio > EMA_10	+250.91 %	3,509.08
Precio > Banda Superior	-7.06 %	929.37
Precio < Banda Inferior	+67.49 %	1,674.95



*Gráfico 5.7.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración*

Propia

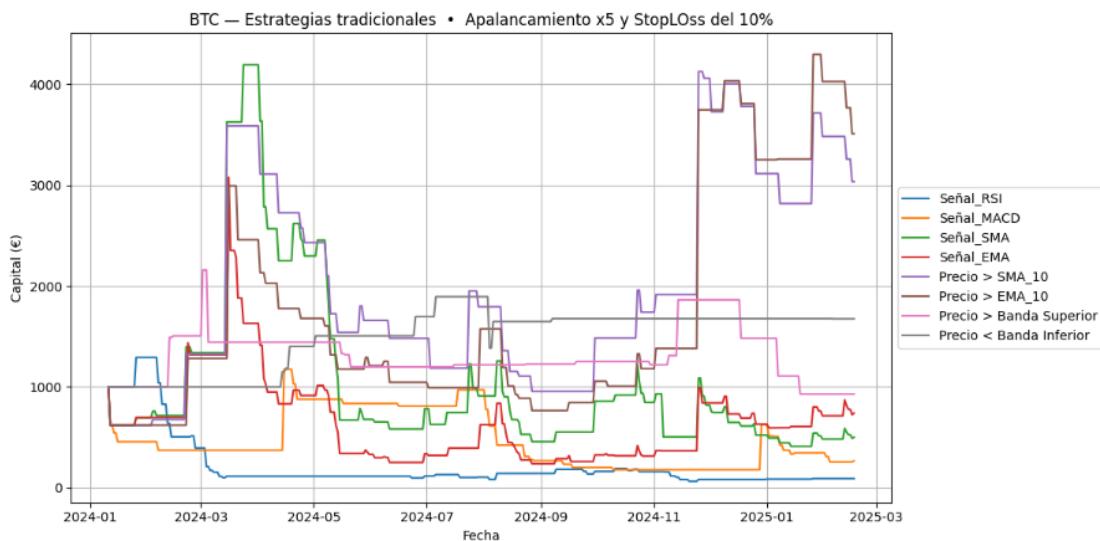


Gráfico 5.7.2: Gráfico de Líneas Bitcoin con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

Tabla 5.8: Tabla comparativa BTC con apalancamiento *10 con StopLoss del 10% con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-98.67 %	13.32
Señal_RSI	-90.80 %	92.02
Señal_MACD	-73.24 %	267.55
Señal_SMA	-49.94 %	500.61
Señal_EMA	-26.04 %	739.64
Precio > SMA_10	+203.50 %	3,035.03
Precio > EMA_10	+250.91 %	3,509.08
Precio > Banda Superior	-7.06 %	929.37
Precio < Banda Inferior	+67.49 %	1,674.95

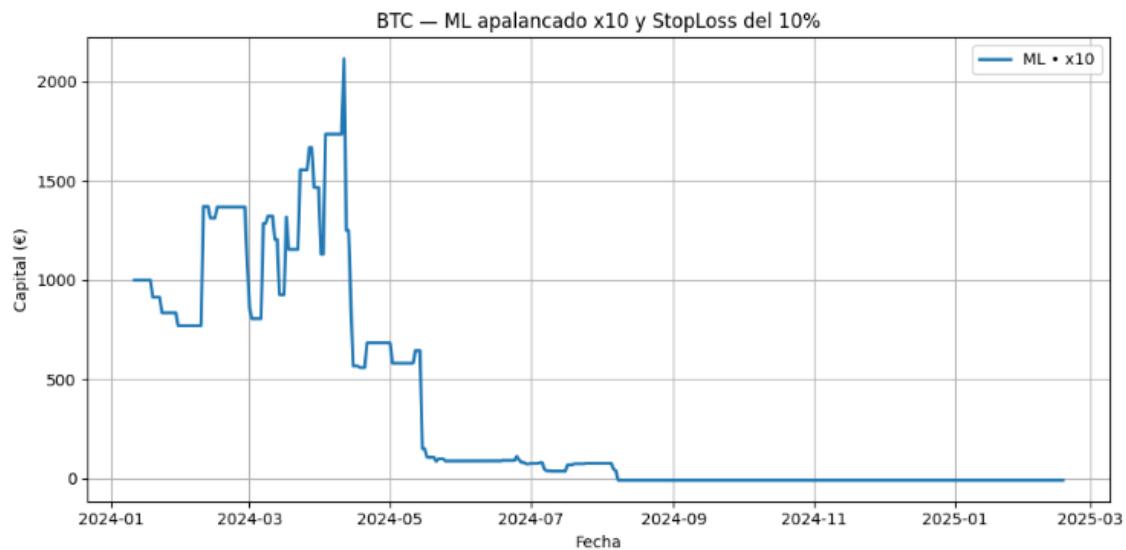


Gráfico 5.8.1: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

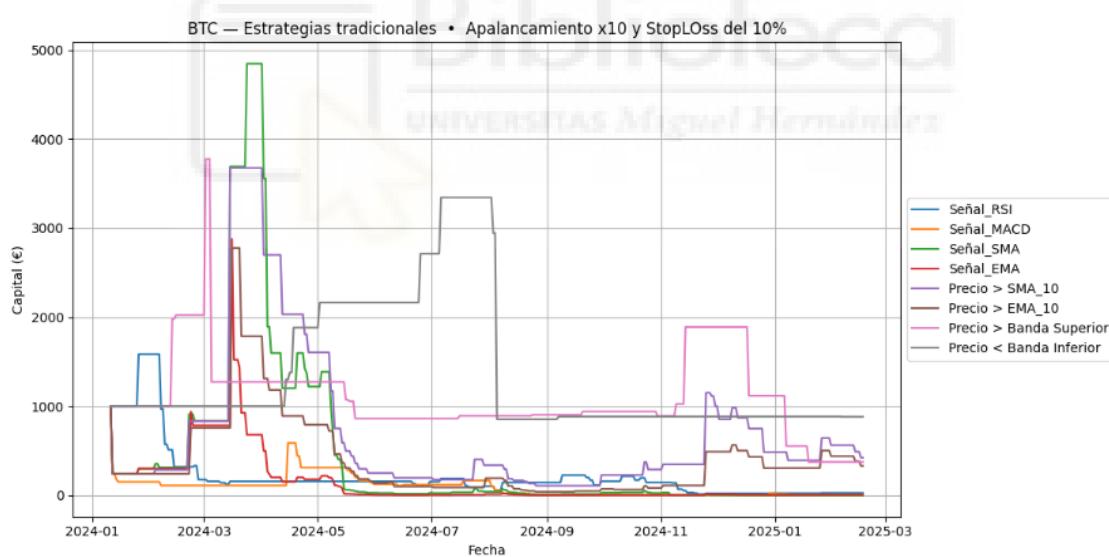


Gráfico 5.8.2: Gráfico de Líneas Bitcoin con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

14.6. Nasdaq100, Machine Learning con apalancamiento y Stop-loss

Los resultados obtenidos al implementar diversas estrategias en el índice Nasdaq100 evidencian una notable variabilidad según el enfoque adoptado. Con un tamaño de ventana de 60 días, un apalancamiento de x10 y un stop-loss dinámico del 10 %, se nota que varias estrategias convencionales superan significativamente al modelo fundamentado en Machine Learning (ML) como se aprecia en la tabla 6.2

La estrategia de Machine Learning muestra un rendimiento negativo de -32.13 % con una ventana de 60 días como muestra la tabla 6.1, lo que resulta en una considerable pérdida de capital, concluyendo en 678.74 €. Pero no solo con una ventana de 60, con los demás tamaños de ventana como 40, 20 y 10 no mejora como se ven en las tablas 6.2, 6.3, 6.4, 6.5, 6.6, 6.7 y 6.8. Esto indica que, el modelo no logró ajustarse adecuadamente.

En contraste, diversas estrategias tradicionales han obtenido resultados favorables. La táctica "Precio > Banda Superior" resultó ser la más provechosa, con un +22.15 % con un apalancamiento *5 con ventana 60 como se ve en la tabla 6.1, logrando un capital final de 1,221.45 €. Pero en la tabla 6.4 se logró la mayor rentabilidad con un 46.11%. Asimismo, resaltan las estrategias fundamentadas en indicadores de tendencia, como MACD (+13.54 %) y SMA (+11.10 %) con una ventana de 60 como dice la tabla 6.1, lo que sugiere que observar señales de cruce o comportamiento en relación con medias móviles puede resultar más eficiente en contextos volátiles y apalancados que los modelos de Machine Learning formados en intervalos móviles restringidos.

En cambio, ciertas estrategias tradicionales también presentan pérdidas, como EMA (-25.93 %) o Precio > EMA_10 (-18.64 %) como nos dicen la mayoría de las tablas como por ejemplo la 6.1, 6.2..., aunque estas bajas han sido inferiores a las del modelo ML.

En conclusión, en situaciones de apalancamiento y control de riesgo a través de Stop-loss, ciertas estrategias tradicionales pueden superar notablemente al modelo de Machine Learning en cuanto a beneficios. Esto indica que la sencillez de entrada/salida puede ser más eficaz que modelos complicados cuando el entorno de mercado no se adapta adecuadamente a los patrones establecidos.

Tabla 6.1: Tabla comparativa Nasdaq100 con apalancamiento *5 con StopLoss del 10% con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-32.13 %	678.74
Señal_RSI	-15.14 %	848.63
Señal_MACD	+13.54 %	1,135.38
Señal_SMA	+11.10 %	1,110.95
Señal_EMA	-25.93 %	740.75
Precio > SMA_10	+9.61 %	1,096.10
Precio > EMA_10	-18.64 %	813.63
Precio > Banda Superior	+22.15 %	1,221.45
Precio < Banda Inferior	-13.63 %	863.68



Gráfico 6.1.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

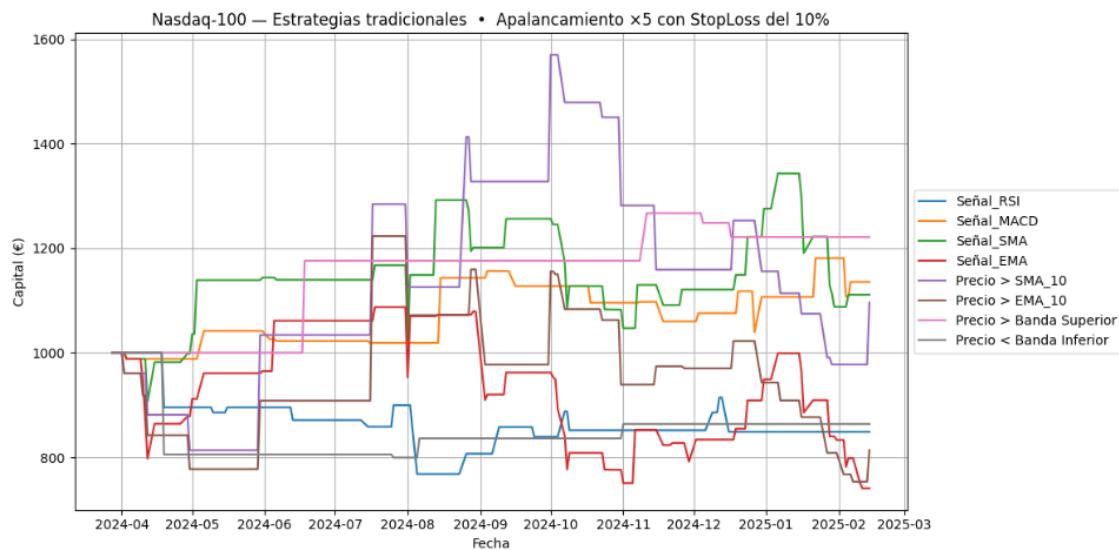


Gráfico 6.1.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

Tabla 6.2: Tabla comparativa Nasdaq100 con apalancamiento *10 con StopLoss del 10% con Window Size 60. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-32.13 %	678.74
Señal_RSI	-15.14 %	848.63
Señal_MACD	+13.54 %	1,135.38
Señal_SMA	+11.10 %	1,110.95
Señal_EMA	-25.93 %	740.75
Precio > SMA_10	+9.61 %	1,096.10
Precio > EMA_10	-18.64 %	813.63
Precio > Banda Superior	+22.15 %	1,221.45
Precio < Banda Inferior	-13.63 %	863.68

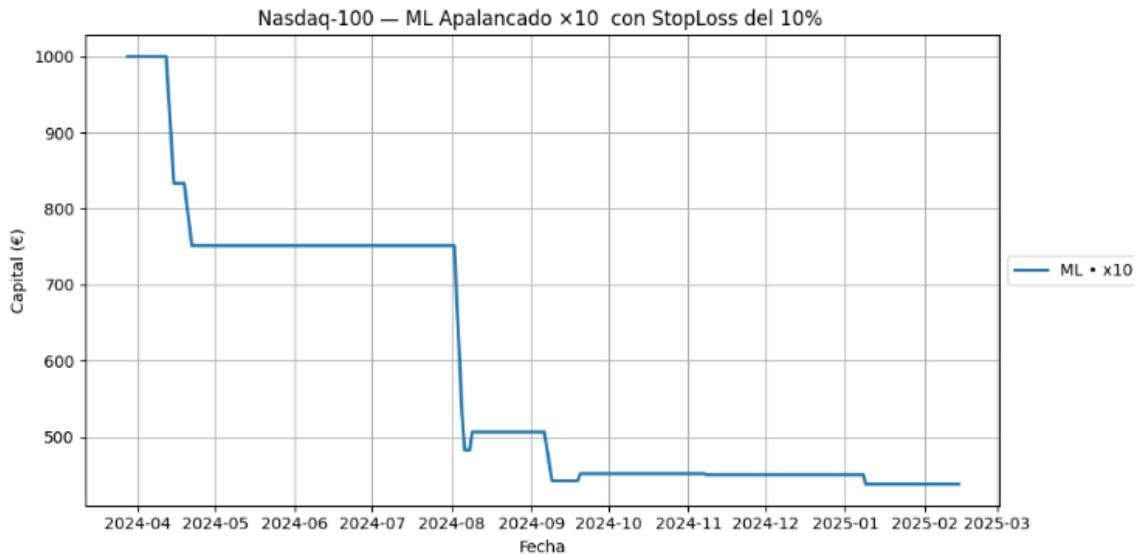


Gráfico 6.2.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

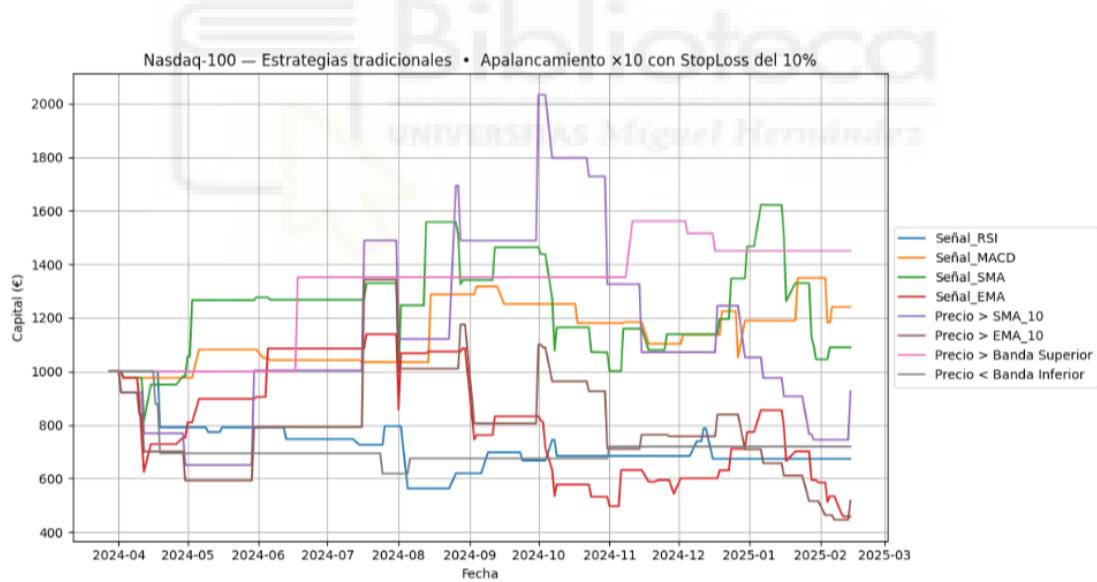


Gráfico 6.2.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 60. Fuente: Elaboración Propia

Tabla 6.3: Tabla comparativa Nasdaq100 con apalancamiento *5 con StopLoss del 10% con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-40.65 %	593.47
Señal_RSI	-15.14 %	848.63
Señal_MACD	+4.08 %	1,040.76
Señal_SMA	+0.08 %	1,000.79
Señal_EMA	-22.54 %	774.58
Precio > SMA_10	-0.29 %	997.15
Precio > EMA_10	-20.05 %	799.51
Precio > Banda Superior	+22.68 %	1,226.80
Precio < Banda Inferior	-13.63 %	863.68



*Gráfico 6.3.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración*

Propia

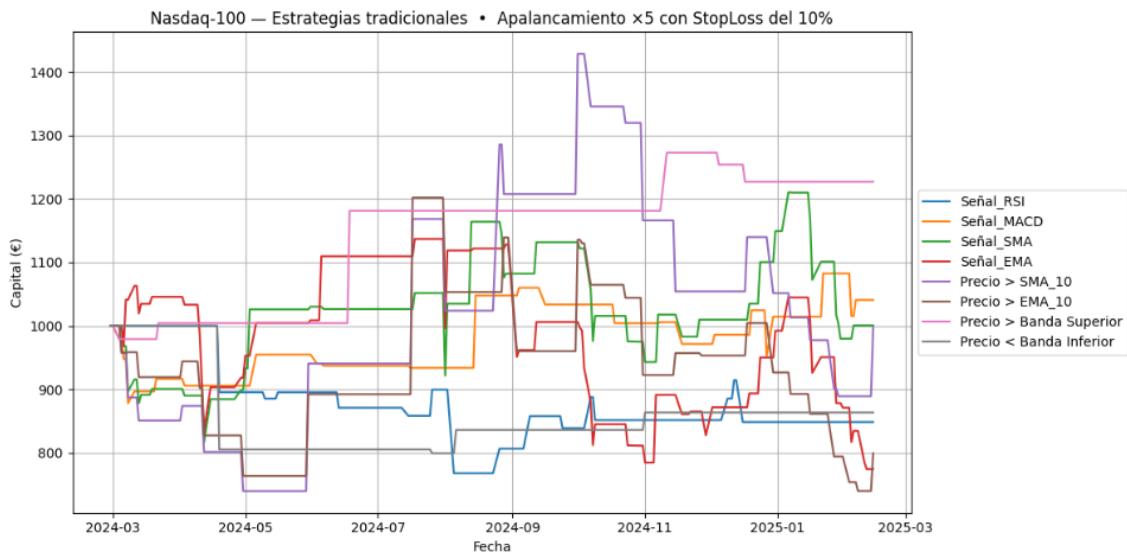


Gráfico 6.3.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

Tabla 6.4: Tabla comparativa Nasdaq100 con apalancamiento *10 con StopLoss del 10% con Window Size 40. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-66.74 %	332.63
Señal_RSI	-32.62 %	673.81
Señal_MACD	+3.39 %	1,033.89
Señal_SMA	-12.42 %	875.76
Señal_EMA	-50.32 %	496.85
Precio > SMA_10	-24.05 %	759.55
Precio > EMA_10	-50.11 %	498.90
Precio > Banda Superior	+46.11 %	1,461.14
Precio < Banda Inferior	-28.03 %	719.69

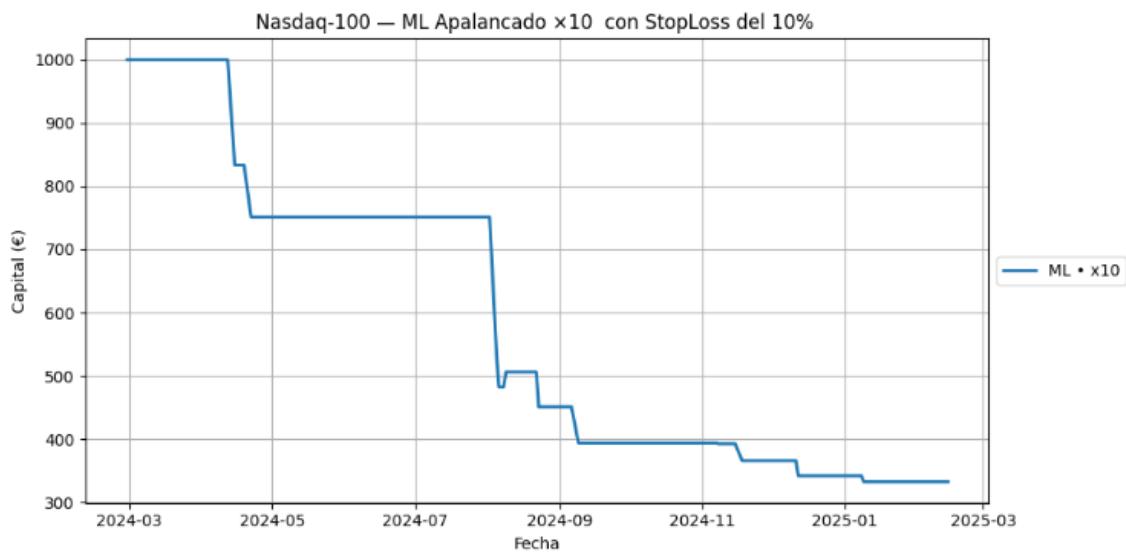


Gráfico 6.4.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

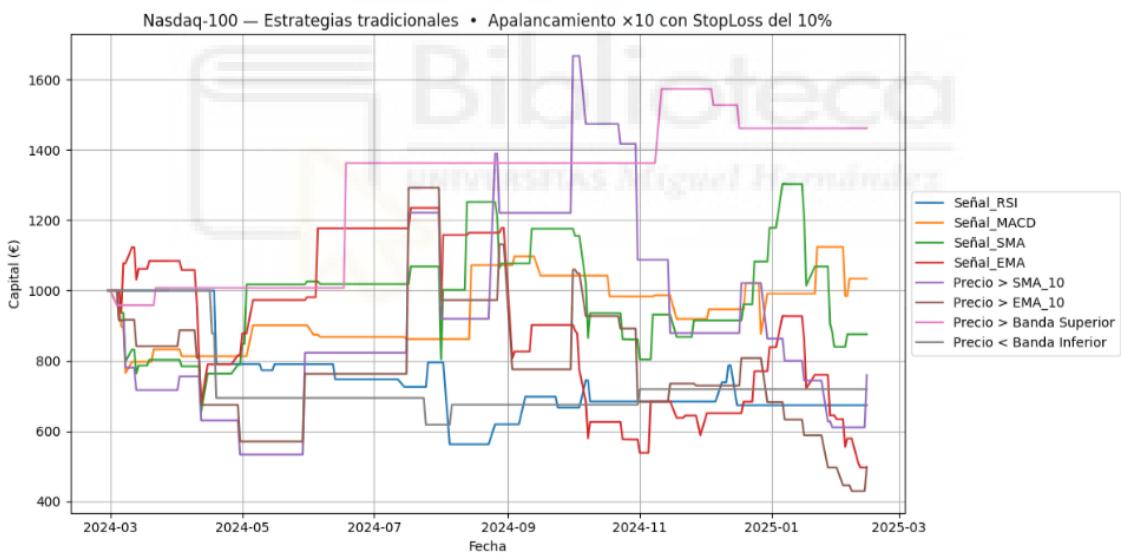


Gráfico 6.4.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 40. Fuente: Elaboración Propia

Tabla 6.5: Tabla comparativa Nasdaq100 con apalancamiento *5 con StopLoss del 10% con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
------------	-----------------	-------------------

ML	-9.18 %	908.21
Señal_RSI	-15.14 %	848.63
Señal_MACD	+8.98 %	1,089.78
Señal_SMA	-9.08 %	909.21
Señal_EMA	-22.95 %	770.54
Precio > SMA_10	-4.67 %	953.35
Precio > EMA_10	-16.99 %	830.10
Precio > Banda Superior	+22.68 %	1,226.80
Precio < Banda Inferior	-13.63 %	863.68



Gráfico 6.5.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

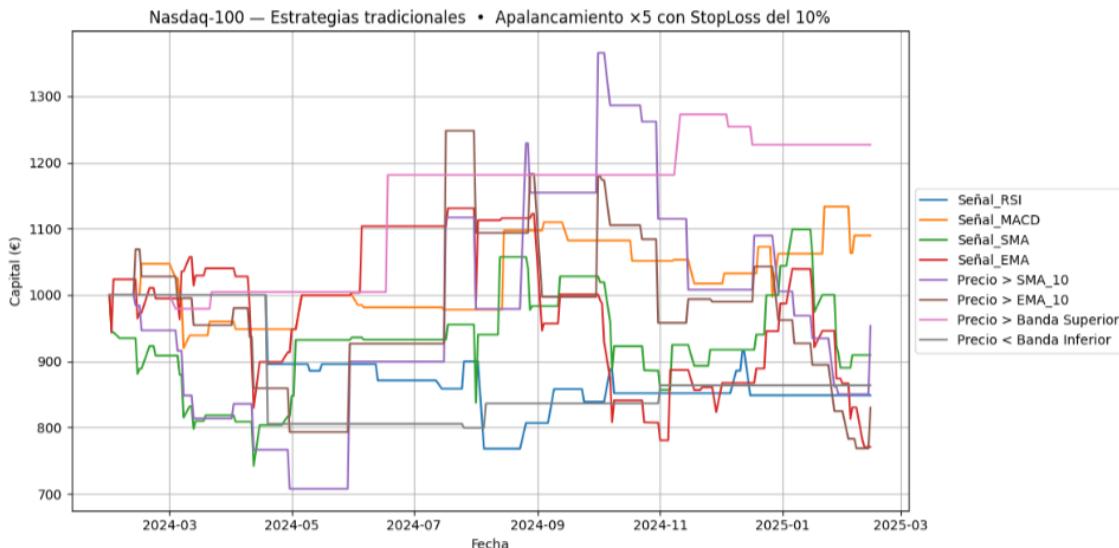


Gráfico 6.5.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

Tabla 6.6: Tabla comparativa Nasdaq100 con apalancamiento *10 con StopLoss del 10% con Window Size 20. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-20.73 %	792.74
Señal_RSI	-32.62 %	673.81
Señal_MACD	+13.13 %	1,131.29
Señal_SMA	-28.38 %	716.24
Señal_EMA	-51.57 %	484.27
Precio > SMA_10	-30.64 %	693.57
Precio > EMA_10	-46.48 %	535.16
Precio > Banda Superior	+46.11 %	1,461.14
Precio < Banda Inferior	-28.03 %	719.69

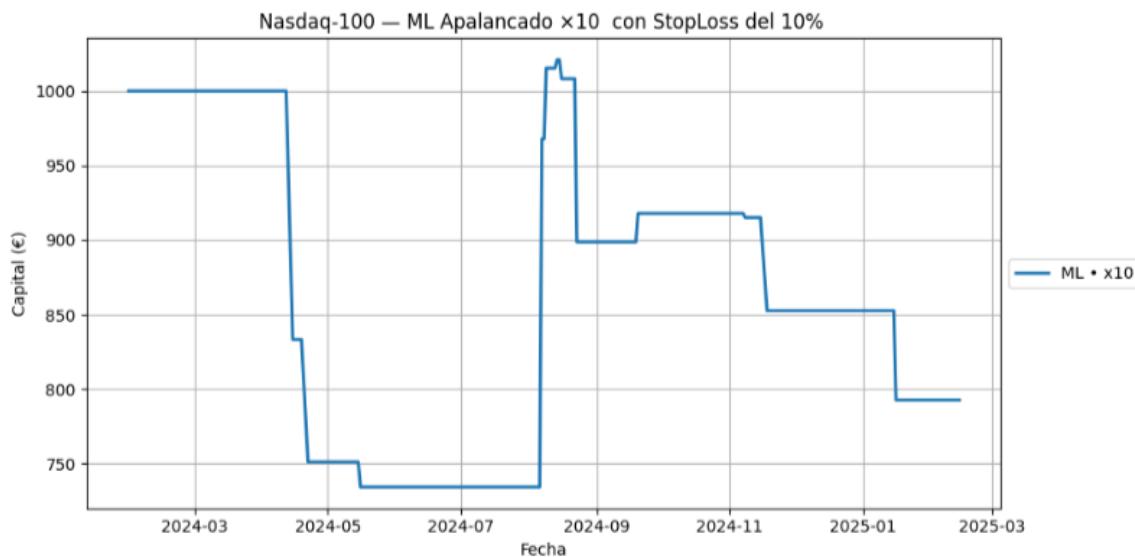


Gráfico 6.6.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

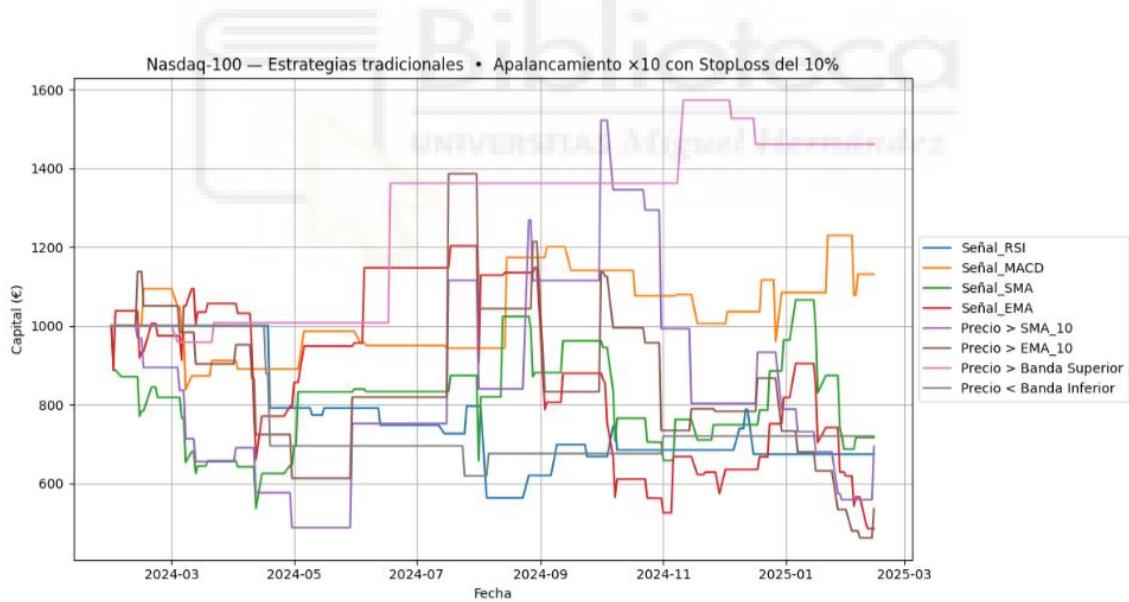


Gráfico 6.6.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 20. Fuente: Elaboración Propia

Tabla 6.7: Tabla comparativa Nasdaq100 con apalancamiento *5 con StopLoss del 10% con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
------------	-----------------	-------------------

ML	-24.89 %	751.10
Señal_RSI	-15.14 %	848.63
Señal_MACD	+8.98 %	1,089.78
Señal_SMA	-9.08 %	909.21
Señal_EMA	-22.95 %	770.54
Precio > SMA_10	+5.86 %	1,058.60
Precio > EMA_10	-7.83 %	921.75
Precio > Banda Superior	+22.68 %	1,226.80
Precio < Banda Inferior	-13.63 %	863.68



Gráfico 6.7.1: Gráfico de Líneas Nasdaq100 con Estrategia Machine Learning con apalancamiento *5 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

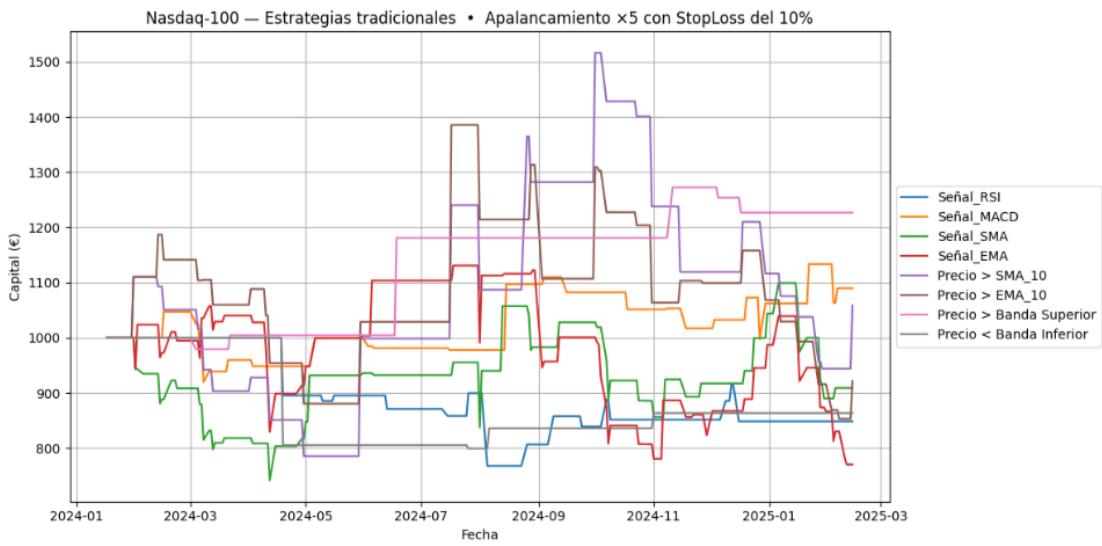


Gráfico 6.7.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *5 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

Tabla 6.8: Tabla comparativa Nasdaq100 con apalancamiento *10 con StopLoss del 10% con Window Size 10. Fuente: Elaboración Propia

Estrategia	Rendimiento (%)	Capital final (€)
ML	-46.99 %	530.01
Señal_RSI	-32.62 %	673.81
Señal_MACD	+13.13 %	1,131.29
Señal_SMA	-28.38 %	716.24
Señal_EMA	-51.57 %	484.27
Precio > SMA_10	-15.33 %	846.71
Precio > EMA_10	-34.67 %	653.33
Precio > Banda Superior	+46.11 %	1,461.14
Precio < Banda Inferior	-28.03 %	719.69

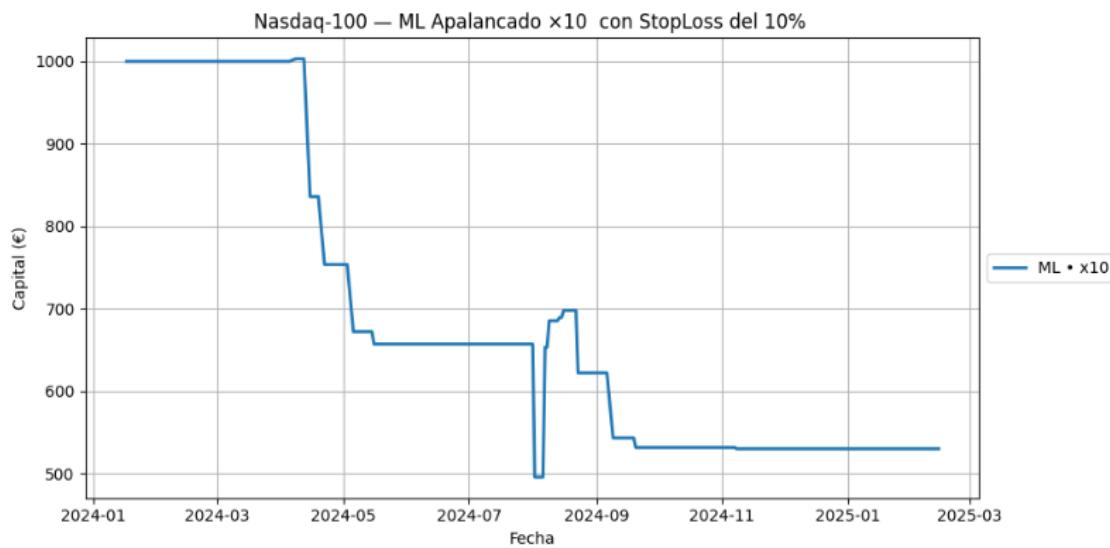


Gráfico 6.8.1: Gráfico de Lineas Nasdaq100 con Estrategia Machine Learning con apalancamiento *10 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

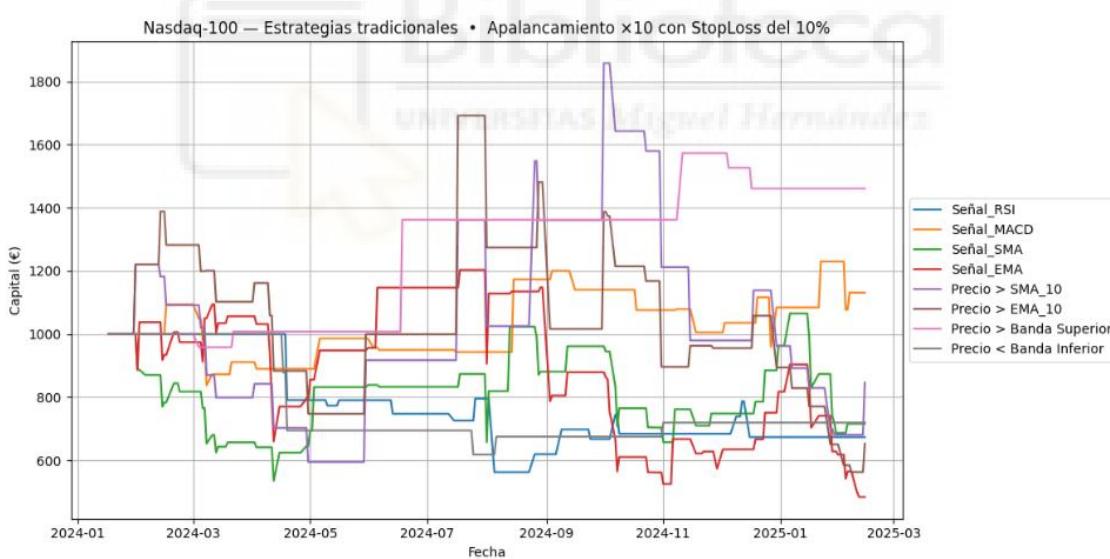


Gráfico 6.8.2: Gráfico de Líneas Nasdaq100 con Estrategias Tradicionales con apalancamiento *10 y Stop-loss del 10% con Window Size 10. Fuente: Elaboración Propia

15. Proyecto a futuro

Tras este análisis a fondo de la estrategia de Machine Learning en dos activos de naturaleza muy distinta como Bitcoin y el Nasdaq 100, se ha descubierto que el uso de modelos automatizados tiene un gran potencial, pero también algunas limitaciones si no se utiliza una gestión de riesgo adecuada y un conocimiento del mercado adecuado.

El próximo paso que se haría, sería explorar el ajuste dinámico del modelo, que quiere decir que, aunque se han utilizado diez indicadores técnicos tradicionales de los más comunes en el mercado, no todos han aportado el mismo valor predictivo en cada entorno. Sería interesante implementar técnicas de selección de características para saber qué variables son realmente significativas en función de cada activo y del mercado.

Finalmente, se seguiría explorando en la gestión de capital y control del riesgo. Los resultados del modelo con apalancamiento han demostrado que incluso un buen modelo puede quedar por los suelos en cuanto a beneficios.

Para concluir, este Trabajo de Fin de Grado ha sentado unas bases sólidas demostrando la importancia de aplicar inteligencia artificial en una situación realista, pero también ha dejado puertas abiertas en cuanto al desarrollo y mejora de este. La relación entre tecnología, análisis técnico, gestión del riesgo, conocimiento del activo y del mercado será clave para lograr estrategias consistentes, sostenibles y rentables a largo plazo.

16. Agradecimientos

Este apartado es con el que se termina todo, un ciclo de 4 años con subidas y bajadas, pero sobre todo con recuerdos. Esos recuerdos que no se te olvidan en la vida. En estas memorias siempre te aparecen las mismas personas a la mente que quieras que se mantengan en tu vida el máximo tiempo posible. Por ello, voy a dedicar unas palabras a esas personas que me han apoyado en todo este camino.

Fernando y Manoli, o como me gusta llamarlos Papá y Mamá. Si soy el hombre que soy ahora, es gracias a vosotros (Aunque siempre seré vuestro niño). Me habéis dado todo lo que le podéis dar y más a un hijo único para que no se sienta solo y se sienta como realmente se tiene que sentir, ÚNICO. Unos padres que se complementan de 10 y siempre están ahí cuando más lo necesito. Mis dos ángeles de la guarda que siempre me guían en

cada paso que doy. Para terminar, decir que mi mayor triunfo no es haber completado la carrera, es haberos tenido de padres. Os quiero hasta el infinito y más allá.

Nadia, mi compañera de un viaje interminable. Aún recuerdo la primera vez que te vi y aunque no estaba seguro sabía que algo tenía que pasar, y pasó. Ya casi 3 años a tu lado y me parecen pocos. Gracias por ser tú y apoyarme en todo lo que me proponga. Espero seguir a tu lado y que compartamos recuerdos inolvidables juntos. Te quise y te quiero y te querré. PD: Ahora solo faltas tú por graduarte.

Mi tutor de TFG José Luis, gracias por enseñarme el mundo de la inversión y haber confiado en mí para hacer el Trabajo final de Grado, porque gracias a ti he aprendido mucho sobre las inversiones que cada día me apasionan más y más. Y gracias a ti, estoy haciendo las prácticas en un sitio increíble. Te aprecio un montón

Mis amigos y familia, muchas gracias a todos los que en algún momento han formado parte de mi vida. Mis amigos de Novelda y de Elche, por más risas y fiestas juntos. Y a mis familias de Aspe y Novelda por darme siempre un cariño especial os quiero.

Por último, dar las gracias a mi profesora de Economía de Bachiller, Mamen. Nunca me he olvidado de cuando no sabía que hacer y sacaste tiempo para ver que me gustaba realmente y al final gracias a ti pude decantarme por este camino de la estadística. Un Efecto Mariposa que ha llegado hasta este punto conociendo a personas maravillosas. Contigo comenzó esta aventura y contigo acaba. Muchas gracias de corazón Mamen.

17. Bibliografía

(1) Apuntes de la asignatura Gestión de Carteras e Inversiones (Grado: Estadística Empresarial en la Universidad Miguel Hernández).

(2) Investopedia. (s. f.). *Relative Strength Index* (*RSI*).

<https://www.investopedia.com/terms/r/rsi.asp>

- (3) Investopedia. (s. f.). *Moving Average Convergence Divergence – MACD*.
<https://www.investopedia.com/terms/m/macd.asp>
- (4) Investopedia. (s. f.). *Bollinger Bands*.
<https://www.investopedia.com/terms/b/bollingerbands.asp>
- (5) Investopedia. (s. f.). *Exponential Moving Average (EMA)*.
<https://www.investopedia.com/terms/e/ema.asp>
- (6) Investopedia. (s. f.). *What is Leverage?*.
<https://www.investopedia.com/terms/l/leverage.asp>
- (7) Marginology. (2023). *What is a Stop Loss Order and How to Use It*.
<https://www.marginology.com/stop-loss-order-guide>
- (8) Yahoo Finance. (s. f.). *Bitcoin USD (BTC-USD) Historical Data*.
<https://finance.yahoo.com/quote/BTC-USD/history>
- (9) Yahoo Finance. (s. f.). *NASDAQ 100 (^NDX) Historical Data*.
<https://finance.yahoo.com/quote/%5ENDX/history>
- (10) Random Forest: Bosque aleatorio. Definición y funcionamiento. (25 de enero de 2025). DataScientest. <https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento>

18. Anexos

A continuación, el código fuente de Python empleado para desarrollar y obtener los resultados expuestos en el presente informe.

18.1. BTC Machine Learning y Estrategias Tradicionales

```
# -----
```

```
# BTC – Estrategia Machine Learning + estrategias tradicionales
```

```
# Entradas / salidas al cierre del mismo día
```

```
# -----
```

```

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier


btc_df = pd.read_excel("BTC.xlsx")

btc_df['DÍA'] = pd.to_datetime(btc_df['DÍA'])

btc_df = btc_df.sort_values('DÍA').reset_index(drop=True)

#Target

btc_df['Rendimiento_diario'] = btc_df['Precio de cierre'].pct_change().fillna(0) * 100

btc_df['Target'] = btc_df['Rendimiento_diario'].shift(-1).apply(
    lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0)
)

features = [
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',
    'Precio > SMA_10', 'Precio > EMA_10',
    'Precio > Banda Superior', 'Precio < Banda Inferior',
    'Precio > SMA_20', 'Precio > STD_20',
    'High', 'Low'
]

# ----- Machine Learning (Random Forest) -----

```

```

capital_ml    = 1000.0
capital_hist_ml = []
position      = 0.0
short_position = 0.0
entry_price   = 0.0
window_size   = 60

for i in range(window_size, len(btc_df)):
    train = btc_df.iloc[i - window_size:i].dropna(subset=features + ['Target'])
    if len(train) < window_size // 2 or train['Target'].nunique() < 2:
        capital_hist_ml.append(capital_ml)
        continue
    X_train, y_train = train[features], train['Target']
    model = RandomForestClassifier(
        n_estimators=100, class_weight='balanced', random_state=42
    )
    model.fit(X_train, y_train)

    test_day = btc_df.iloc[i]
    close_px = test_day['Precio de cierre']
    prediction = model.predict(pd.DataFrame([test_day[features]]))[0]

    # ----- (Todo al CIERRE) -----

```

```

if prediction == 1:

    # 1) Cerrar venta si lo hubiera

    if short_position:

        capital_ml *= 1 + (entry_price - close_px) / entry_price

        short_position = 0

    # 2) Abrir compra si no existe

    if position == 0:

        position = capital_ml

        entry_price = close_px


elif prediction == -1:

    # 1) Cerrar compra si lo hubiera

    if position:

        capital_ml *= 1 + (close_px - entry_price) / entry_price

        position = 0

    # 2) Abrir venta si no existe

    if short_position == 0:

        short_position = capital_ml

        entry_price = close_px


else: # Señal 0 ⇒ cerrar todo

    if position:

        capital_ml *= 1 + (close_px - entry_price) / entry_price

        position = 0

```

```
elif short_position:  
    capital_ml *= 1 + (entry_price - close_px) / entry_price  
    short_position = 0
```

```
capital_hist_ml.append(capital_ml)
```

```
last_close = btc_df.iloc[-1]['Precio de cierre']
```

```
if position:
```

```
    capital_ml *= 1 + (last_close - entry_price) / entry_price
```

```
if short_position:
```

```
    capital_ml *= 1 + (entry_price - last_close) / entry_price
```

```
capital_hist_ml[-1] = capital_ml
```

```
# ----- Estrategias tradicionales -----
```

```
estrategias = [  
    'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',  
    'Precio > SMA_10', 'Precio > EMA_10',  
    'Precio > Banda Superior', 'Precio < Banda Inferior'  
]
```

```
capitales_trad = {}
```

```
resultados = [('ML', capital_ml)]
```

```
for strat in estrategias:
```

```
capital, pos, short, entry = 1000.0, 0.0, 0.0, 0.0
```

```
hist = []
```

```
is_bin = btc_df[strat].isin([0, 1]).all()
```

```
for i in range(len(btc_df)):
```

```
    signal = btc_df.loc[i, strat]
```

```
    close_p = btc_df.loc[i, 'Precio de cierre']
```

```
    if signal == 1:
```

```
        if short:
```

```
            capital *= 1 + (entry - close_p) / entry
```

```
            short = 0
```

```
        if pos == 0:
```

```
            pos, entry = capital, close_p
```

```
    elif signal == -1:
```

```
        if pos:
```

```
            capital *= 1 + (close_p - entry) / entry
```

```
            pos = 0
```

```
        if short == 0:
```

```
            short, entry = capital, close_p
```

```
    else: # signal 0
```

```
        if pos:
```

```
            capital *= 1 + (close_p - entry) / entry
```

```
            pos = 0
```

```

        elif short:

            capital *= 1 + (entry - close_p) / entry

            short = 0

hist.append(capital)

# Cierre final

if pos:

    capital *= 1 + (btc_df.iloc[-1]['Precio de cierre'] - entry) / entry

if short:

    capital *= 1 + (entry - btc_df.iloc[-1]['Precio de cierre']) / entry

hist[-1] = capital

resultados.append((strat, capital))

capitales_trad[strat] = hist

# ----- Gráficos -----

plt.figure(figsize=(10, 6))

plt.plot(btc_df['DÍA'].iloc[window_size:], capital_hist_ml, label='ML')

plt.title('BTC – Estrategia ML')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

plt.figure(figsize=(12, 7))

for strat, hist in capitales_trad.items():

```

```

plt.plot(btc_df['DÍA'][:len(hist)], hist, label=strat)

plt.title('BTC – Estrategias tradicionales')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

# ----- Resultados finales -----

for name, cap in resultados:

    print(f'{name}: Capital final = {cap:.2f} €')

```

18.2. BTC Machine Learning y Estrategias Tradicionales con posición siempre abierta

```

## Código con posición siempre abierta

# -----
# BTC – ML + Estrategias tradicionales

# Posición SIEMPRE abierta: sólo se cambia al cierre

# cuando la señal da la dirección contraria

# -----
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.ensemble import RandomForestClassifier

# 1) Datos -----

```

```

btc_df = pd.read_excel("BTC.xlsx")

btc_df['DÍA'] = pd.to_datetime(btc_df['DÍA'])

btc_df.sort_values('DÍA', inplace=True, ignore_index=True)

# Target ±1,5 %
-----#
btc_df['Rendimiento_diario'] = btc_df['Precio de cierre'].pct_change().fillna(0)*100

btc_df['Target'] = btc_df['Rendimiento_diario'].shift(-1) \
    .apply(lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0))

# Variables predictoras -----
features = [
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',
    'Precio > SMA_10', 'Precio > EMA_10',
    'Precio > Banda Superior', 'Precio < Banda Inferior',
    'Precio > SMA_20', 'Precio > STD_20',
    'High', 'Low'
]

# -----
# 2) Estrategia Machine-Learning con posición siempre abierta
# -----
capital_ml = 1000.0

hist_ml = []

side = 0      # 1 = largo, -1 = corto, 0 = sin posición (solo al inicio)

```

```
entry_price = 0.0  
window_size = 60  
  
rf      = RandomForestClassifier(n_estimators=100,      class_weight='balanced',  
random_state=42)
```

```
for i in range(window_size, len(btc_df)):  
  
    train = btc_df.iloc[i-window_size:i].dropna(subset=features + ['Target'])  
  
    if train['Target'].nunique() < 2:  
  
        hist_ml.append(capital_ml)  
  
    continue
```

```
rf.fit(train[features], train['Target'])
```

```
today    = btc_df.iloc[i]
```

```
close_px = today['Precio de cierre']
```

```
prediction = rf.predict(pd.DataFrame([today[features]]))[0]
```

```
# 1) si no hay posición (sólo la primera vez) -> abre con la predicción
```

```
if side == 0:
```

```
    side, entry_price = prediction if prediction != 0 else 1, close_px
```

```
# 2) si la señal es contraria → cerrar y abrir al instante
```

```
elif prediction != 0 and prediction != side:
```

```
    # cerrar posición anterior
```

```
if side == 1: # largo  
    capital_ml *= 1 + (close_px - entry_price) / entry_price  
  
else: # corto  
    capital_ml *= 1 + (entry_price - close_px) / entry_price
```

```
# abrir nueva posición en la otra dirección
```

```
side, entry_price = prediction, close_px
```

```
hist_ml.append(capital_ml)
```

```
# Cierre final
```

```
last_close = btc_df.iloc[-1]['Precio de cierre']
```

```
if side == 1:
```

```
    capital_ml *= 1 + (last_close - entry_price) / entry_price
```

```
else:
```

```
    capital_ml *= 1 + (entry_price - last_close) / entry_price
```

```
hist_ml[-1] = capital_ml
```

```
# -----
```

```
# 3) Estrategias tradicionales con posición abierta siempre
```

```
# -----
```

```
estrategias = [
```

```
'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',
'Precio > SMA_10', 'Precio > EMA_10',
'Precio > Banda Superior', 'Precio < Banda Inferior'

]
```

```
hists_trad = {}

resultados = [('ML', capital_ml)]
```

```
for col in estrategias:
```

```
    cap, hist = 1000.0, []
    side, entry = 0, 0.0          # 1 = largo, -1 = corto
    is_binary = btc_df[col].isin([0, 1]).all()
```

```
for i in range(len(btc_df)):
```

```
    sig      = btc_df.loc[i, col]
    close_px = btc_df.loc[i, 'Precio de cierre']
```

```
# Señales normalizadas: binario -> 1 largo / -1 corto (inexistente) / 0 mantener
```

```
if is_binary:
```

```
    sig_norm = 1 if sig == 1 else (-1 if sig == 0 and side == 1 else side)
```

```
else:
```

```
    sig_norm = sig
```

```
# primera entrada
```

```

if side == 0 and sig_norm != 0:
    side, entry = sig_norm, close_px

# cambio de dirección

elif sig_norm != 0 and sig_norm != side:
    if side == 1:
        cap *= 1 + (close_px - entry) / entry
    else:
        cap *= 1 + (entry - close_px) / entry

side, entry = sig_norm, close_px

```

hist.append(cap)

cierre final

last = btc_df.iloc[-1]['Precio de cierre']

if side == 1:

cap *= 1 + (last - entry) / entry

elif side == -1:

cap *= 1 + (entry - last) / entry

hist[-1] = cap

hists_trad[col] = hist

resultados.append((col, cap))

```

# -----
# 4) GRÁFICOS

plt.figure(figsize=(10, 5))

plt.plot(btc_df['DÍA'].iloc[window_size:], hist_ml, label='ML', lw=2)

plt.title('BTC – Estrategia ML')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

```

```

plt.figure(figsize=(12, 6))

for name, h in hists_trad.items():

    plt.plot(btc_df['DÍA'].iloc[:len(h)], h, label=name)

    plt.title('BTC – Estrategias tradicionales')

    plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

    plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

```

```

# -----
# 5) RESULTADOS

print('\n==== Capital final ====')

for n, c in resultados:

    print(f'{n:24s} : {c:.2f} €')

```

18.3. Nasdaq100 Machine Learning y Estrategias Tradicionales

```
# -----  
  
# Nasdaq100 – ML + estrategias tradicionales  
  
# -----  
  
import pandas as pd  
  
import numpy as np  
  
import matplotlib.pyplot as plt  
  
from sklearn.ensemble import RandomForestClassifier  
  
  
  
nasdaq_df = pd.read_excel("Nasdaq100.xlsx")  
  
nasdaq_df['DÍA'] = pd.to_datetime(nasdaq_df['DÍA'])  
  
nasdaq_df = nasdaq_df.sort_values('DÍA').reset_index(drop=True)  
  
  
  
nasdaq_df['Rendimiento_diario'] = nasdaq_df['Precio de cierre'].pct_change().fillna(0) *  
100  
  
nasdaq_df['Target'] = nasdaq_df['Rendimiento_diario'].shift(-1).apply(  
  
lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0)  
  
)
```

```
features = [
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',
    'Precio > SMA_10', 'Precio > EMA_10',
    'Precio > Banda Superior', 'Precio < Banda Inferior',
    'Precio > SMA_20', 'Precio > STD_20',
    'High', 'Low'
]
```

```
# ----- Machine Learning -----
```

```
capital_ml = 1000.0
```

```
capital_hist_ml = []
```

```
position = 0.0
```

```
short_position = 0.0
```

```
entry_price = 0.0
```

```
window_size = 60
```

```
for i in range(window_size, len(nasdaq_df)):
```

```
    train = nasdaq_df.iloc[i - window_size:i].dropna(subset=features + ['Target'])
```

```
    if len(train) < window_size // 2 or train['Target'].nunique() < 2:
```

```
        capital_hist_ml.append(capital_ml)
```

```
        continue
```

```
X_train, y_train = train[features], train['Target']
```

```
model = RandomForestClassifier(
```

```

n_estimators=100, class_weight='balanced', random_state=42
)
model.fit(X_train, y_train)

test_day = nasdaq_df.iloc[i]
close_px = test_day['Precio de cierre']
prediction = model.predict(pd.DataFrame([test_day[features]]))[0]

# ---- (Todo al CIERRE) ----

if prediction == 1:
    if short_position:           # cerrar venta
        capital_ml *= 1 + (entry_price - close_px) / entry_price
        short_position = 0
    if position == 0:            # abrir compra
        position = capital_ml
        entry_price = close_px

elif prediction == -1:
    if position:                # cerrar compra
        capital_ml *= 1 + (close_px - entry_price) / entry_price
        position = 0
    if short_position == 0:      # abrir venta
        short_position = capital_ml
        entry_price = close_px

```

```

else: # predicción 0 → cerrar todo

if position:

    capital_ml *= 1 + (close_px - entry_price) / entry_price

    position = 0

elif short_position:

    capital_ml *= 1 + (entry_price - close_px) / entry_price

    short_position = 0

capital_hist_ml.append(capital_ml)

```

```

# Cierre forzoso último día

last_close = nasdaq_df.iloc[-1]['Precio de cierre']

if position:

    capital_ml *= 1 + (last_close - entry_price) / entry_price

if short_position:

    capital_ml *= 1 + (entry_price - last_close) / entry_price

capital_hist_ml[-1] = capital_ml # actualizar

```

```

# ----- Estrategias tradicionales -----

estrategias = [

'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',

'Precio > SMA_10', 'Precio > EMA_10',

'Precio > Banda Superior', 'Precio < Banda Inferior'

```

]

```
capitales_trad = {}  
resultados = [('ML', capital_ml)]
```

for strat in estrategias:

```
    cap = 1000.0
```

```
    pos = short = 0.0
```

```
    entry = 0.0
```

```
    hist = []
```

```
    is_bin = nasdaq_df[strat].isin([0, 1]).all()
```

```
    for i in range(len(nasdaq_df)):
```

```
        signal = nasdaq_df.at[i, strat]
```

```
        close_p = nasdaq_df.at[i, 'Precio de cierre']
```

```
        if signal == 1:
```

```
            if short:
```

```
                cap *= 1 + (entry - close_p) / entry
```

```
                short = 0
```

```
            if pos == 0:
```

```
                pos, entry = cap, close_p
```

```
        elif signal == -1:
```

```
if pos:
```

```
    cap *= 1 + (close_p - entry) / entry
```

```
    pos = 0
```

```
if short == 0:
```

```
    short, entry = cap, close_p
```

```
else: # señal 0
```

```
if pos:
```

```
    cap *= 1 + (close_p - entry) / entry
```

```
    pos = 0
```

```
elif short:
```

```
    cap *= 1 + (entry - close_p) / entry
```

```
    short = 0
```

```
hist.append(cap)
```

```
# Cerrar posiciones restantes
```

```
if pos:
```

```
    cap *= 1 + (nasdaq_df.iloc[-1]['Precio de cierre'] - entry) / entry
```

```
if short:
```

```
    cap *= 1 + (entry - nasdaq_df.iloc[-1]['Precio de cierre']) / entry
```

```
hist[-1] = cap
```

```
resultados.append((strat, cap))
```

```
capitales_trad[strat] = hist
```

```
# ----- Gráficos -----
plt.figure(figsize=(10, 6))

plt.plot(nasdaq_df['DÍA'].iloc[window_size:], capital_hist_ml, label='ML')

plt.title('Nasdaq 100 – Estrategia ML')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()
```

```
plt.figure(figsize=(12, 7))

for strat, hist in capitales_trad.items():

    plt.plot(nasdaq_df['DÍA'][:len(hist)], hist, label=strat)

    plt.title('Nasdaq 100 – Estrategias tradicionales')

    plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

    plt.grid(); plt.legend(); plt.tight_layout(); plt.show()
```

```
# ----- Resultados -----
```

```
for nombre, cap_final in resultados:

    print(f'{nombre}: Capital final = {cap_final:.2f} €")
```

18.4. Nasdaq100 Machine Learning y Estrategias Tradicionales con posición siempre abierta

```
#Código con posición abierta siempre

# -----
# Nasdaq-100 • ML + Estrategias tradicionales

# Operativa: la posición sólo cambia cuando la señal cambia
```

```

# -----
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier

# ----- 1) Datos -----
nasdaq_df = pd.read_excel("Nasdaq100.xlsx")
nasdaq_df['DÍA'] = pd.to_datetime(nasdaq_df['DÍA'])
nasdaq_df = nasdaq_df.sort_values('DÍA').reset_index(drop=True)

# Target ±1,5 % para el ML
nasdaq_df['Rendimiento_diario'] = nasdaq_df['Precio de cierre'].pct_change().fillna(0)*100
nasdaq_df['Target'] = nasdaq_df['Rendimiento_diario'].shift(-1).apply(
    lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0)
)

features = [
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',
    'Precio > SMA_10', 'Precio > EMA_10',
    'Precio > Banda Superior', 'Precio < Banda Inferior',
    'Precio > SMA_20', 'Precio > STD_20',
    'High', 'Low'
]

```

```

# ----- 2) Modelo ML (Random Forest) -----

window_size    = 60
capital_ml     = 1000.0
capital_hist_ml = []

pos, short, en  = 0.0, 0.0, 0.0    # posición larga / corta / precio entrada

for i in range(window_size, len(nasdaq_df)):

    train = nasdaq_df.iloc[i-window_size:i].dropna(subset=features+['Target'])

    if train['Target'].nunique() < 2:      # muy pocos ejemplos

        capital_hist_ml.append(capital_ml)

    continue

    X_train, y_train = train[features], train['Target']

    rf = RandomForestClassifier(n_estimators=100,
                                class_weight='balanced',
                                random_state=42)

    rf.fit(X_train, y_train)

    row    = nasdaq_df.iloc[i]
    close_px = row['Precio de cierre']
    pred    = rf.predict(pd.DataFrame([row[features]]))[0]

# ----- gestión: sólo se cierra / abre si la señal CAMBIA -----

```

```

# Señal de COMPRA

if pred == 1:

    if short:                      # cerrar corto y abrir largo

        capital_ml *= 1 + (en - close_px)/en

        short = 0

    pos, en = capital_ml, close_px

    elif pos == 0:                  # no había posición

        pos, en = capital_ml, close_px

```

Señal de VENTA (corto)

```

elif pred == -1:

    if pos:                      # cerrar largo y abrir corto

        capital_ml *= 1 + (close_px - en)/en

        pos = 0

    short, en = capital_ml, close_px

    elif short == 0:               # no había posición

        short, en = capital_ml, close_px

```

Señal NEUTRA ⇒ mantener lo que hubiera (no se cierra nada)

```
capital_hist_ml.append(capital_ml)
```

Cerrar posición pendiente al final

```
last_close = nasdaq_df.iloc[-1]['Precio de cierre']
```

if pos:

```
capital_ml *= 1 + (last_close - en)/en
```

if short:

```
capital_ml *= 1 + (en - last_close)/en
```

```
capital_hist_ml[-1] = capital_ml
```

----- 3) Estrategias tradicionales -----

```
estrategias = [
```

```
'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',
```

```
'Precio > SMA_10', 'Precio > EMA_10',
```

```
'Precio > Banda Superior', 'Precio < Banda Inferior'
```

```
]
```

```
hists_trad = {}
```

```
resultados = [('ML', capital_ml)]
```

for col in estrategias:

```
cap = 1000.0
```

```
pos = short = en = 0.0
```

```
hist = []
```

```
for i in range(len(nasdaq_df)):
```

```
sig = nasdaq_df.at[i, col]
```

```
close_px = nasdaq_df.at[i, 'Precio de cierre']
```

```

# compra (o mantener larga)

if sig == 1:

    if short:                      # cerrar corto y abrir largo

        cap *= 1 + (en - close_px)/en

        short = 0

        pos, en = cap, close_px

    elif pos == 0:                  # abrir largo

        pos, en = cap, close_px


# venta (o mantener corta)

elif sig == -1:

    if pos:                      # cerrar largo y abrir corto

        cap *= 1 + (close_px - en)/en

        pos = 0

        short, en = cap, close_px

    elif short == 0:               # abrir corto

        short, en = cap, close_px


# señal 0 → mantener lo que hubiera

hist.append(cap)

# cierre final

if pos:

```

```

cap *= 1 + (nasdaq_df.iloc[-1]['Precio de cierre'] - en)/en

if short:

    cap *= 1 + (en - nasdaq_df.iloc[-1]['Precio de cierre'])/en

hist[-1] = cap

hists_trad[col] = hist

resultados.append((col, cap))

```

----- 4) Gráficos -----

```

plt.figure(figsize=(10,5))

plt.plot(nasdaq_df['DÍA'].iloc[window_size:], capital_hist_ml,
         label='ML', linewidth=2)

plt.title('Nasdaq-100 • ML')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

```

```
plt.figure(figsize=(12,6))
```

```

for nombre, hist in hists_trad.items():

    plt.plot(nasdaq_df['DÍA'].iloc[:len(hist)], hist, label=nombre)

plt.title('Nasdaq-100 • Estrategias tradicionales')

plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

```

----- 5) Resultados finales -----

```
print("\nCapital final (1 000 € inicial):")
```

for n, c in resultados:

```
print(f" {n:25s} → {c:,.2f} €")
```

18.5. BTC Machine Learning con apalancamiento y Stop-loss

```
# -----
```

```
# BTC • ML + Estrategias tradicionales
```

```
# Entradas / salidas al CIERRE • Apalancamiento (×5 / ×10)
```

```
# Stop-loss: -10 % del capital disponible en el momento de abrir
```

```
# -----
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
# ----- 1) DATOS -----
```

```
btc_df = pd.read_excel("BTC.xlsx")
```

```
btc_df['DÍA'] = pd.to_datetime(btc_df['DÍA'])
```

```
btc_df = btc_df.sort_values('DÍA').reset_index(drop=True)
```

```
# Target (±1,5 %)
```

```
btc_df['Rendimiento_diario'] = btc_df['Precio de cierre'].pct_change().fillna(0) * 100
```

```
btc_df['Target'] = btc_df['Rendimiento_diario'].shift(-1).apply(
```

```
lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0)
```

)

----- 2) VARIABLES -----

```
features = [  
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',  
    'Precio > SMA_10', 'Precio > EMA_10',  
    'Precio > Banda Superior', 'Precio < Banda Inferior',  
    'Precio > SMA_20', 'Precio > STD_20',  
    'High', 'Low'  
]
```

estrategias = [

```
'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',  
'Precio > SMA_10', 'Precio > EMA_10',  
'Precio > Banda Superior', 'Precio < Banda Inferior'
```

]

----- 3) PARÁMETROS -----

apalancamientos = [5, 10]

window_size = 60

stop_loss_pct = 0.10 # 10 % del capital disponible al abrir posición

----- 4) FUNCIONES DE BACKTEST -----

```
def stop_trigger(entry_capital, pnl_capital):
```

"""Devuelve True si la pérdida supera el 10 % del capital ENTRADO."""

```
return pnl_capital < -entry_capital * stop_loss_pct
```

```
def backtest_trad(col, lev):  
    is_bin = btc_df[col].isin([0, 1]).all()  
    cap, hist = 1000.0, []  
    pos = short = entry_px = entry_cap = 0.0
```

```
for i in range(window_size, len(btc_df)):  
    sig, close_px = btc_df.loc[i, col], btc_df.loc[i, 'Precio de cierre']
```

```
# ----- gestión señales -----
```

```
if sig == 1: # LARGO
```

```
    if short:
```

```
        cap += short * (entry_px - close_px) / entry_px * lev
```

```
        short = 0
```

```
    if pos == 0:
```

```
        pos, entry_px, entry_cap = cap, close_px, cap
```

```
elif sig == -1 and not is_bin: # CORTO
```

```
    if pos:
```

```
        cap += pos * (close_px - entry_px) / entry_px * lev
```

```
        pos = 0
```

```
    if short == 0:
```

```
        short, entry_px, entry_cap = cap, close_px, cap
```

```

else:                                # NEUTRA

    if pos:

        cap += pos * (close_px - entry_px) / entry_px * lev

        pos = 0

    elif short:

        cap += short * (entry_px - close_px) / entry_px * lev

        short = 0

# ----- stop-loss dinámico -----

if pos:

    pnl_eur = pos * (close_px - entry_px) / entry_px * lev

    if stop_trigger(entry_cap, pnl_eur):

        cap += pnl_eur; pos = 0

elif short:

    pnl_eur = short * (entry_px - close_px) / entry_px * lev

    if stop_trigger(entry_cap, pnl_eur):

        cap += pnl_eur; short = 0

hist.append(cap)

# ----- cierre último día -----

last_px = btc_df.iloc[-1]['Precio de cierre']

if pos:

    cap += pos * (last_px - entry_px) / entry_px * lev

```

```
if short:
```

```
    cap += short * (entry_px - last_px) / entry_px * lev
```

```
hist[-1] = cap
```

```
return hist, cap
```

```
def backtest_ml(lev):
```

```
    rf = RandomForestClassifier(n_estimators=100,  
                               class_weight='balanced',  
                               random_state=42)
```

```
    cap, hist = 1000.0, []
```

```
    pos = short = entry_px = entry_cap = 0.0
```

```
    for i in range(window_size, len(btc_df)):
```

```
        train = btc_df.iloc[i-window_size:i].copy()
```

```
        train = train.dropna(subset=features + ['Target'])
```

```
        if len(train) < 2 or train['Target'].nunique() < 2:
```

```
            hist.append(cap); continue
```

```
X_train = train[features].astype(float)
```

```
y_train = train['Target'].astype(int)
```

```
rf.fit(X_train, y_train)
```

```
#predicción
```

```

day = btc_df.iloc[i]

close_px = day['Precio de cierre']

pred = rf.predict(pd.DataFrame([day[features]]))[0]

if pred == 1:                                # LARGO

    if short:

        cap += short * (entry_px - close_px) / entry_px * lev

        short = 0

    if pos == 0:

        pos, entry_px, entry_cap = cap, close_px, cap

elif pred == -1:                             # CORTO

    if pos:

        cap += pos * (close_px - entry_px) / entry_px * lev

        pos = 0

    if short == 0:

        short, entry_px, entry_cap = cap, close_px, cap

else:                                       # NEUTRO

    if pos:

        cap += pos * (close_px - entry_px) / entry_px * lev

        pos = 0

    elif short:

        cap += short * (entry_px - close_px) / entry_px * lev

        short = 0

```

```

# --- stop-loss ---

if pos:

    pnl_eur = pos * (close_px - entry_px) / entry_px * lev

    if stop_trigger(entry_cap, pnl_eur):

        cap += pnl_eur; pos = 0

elif short:

    pnl_eur = short * (entry_px - close_px) / entry_px * lev

    if stop_trigger(entry_cap, pnl_eur):

        cap += pnl_eur; short = 0

hist.append(cap)

```

last_px = btc_df.iloc[-1]['Precio de cierre']

```

if pos:

    cap += pos * (last_px - entry_px) / entry_px * lev

if short:

    cap += short * (entry_px - last_px) / entry_px * lev

hist[-1] = cap

return hist, cap

```

----- 5) SIMULACIÓN -----

for lev in apalancamientos:

----- ML -----

ml_hist, ml_cap = backtest_ml(lev)

```

# ----- Tradicionales -----

trad_hists, results = {}, [('ML', ml_cap)]

for strat in estrategias:
    h, c = backtest_trad(strat, lev)

    trad_hists[strat], results = h, results + [(strat, c)]


# ----- GRÁFICOS -----

plt.figure(figsize=(10, 5))

plt.plot(btc_df['DÍA'].iloc[window_size:], ml_hist,
         label=f'ML • x{lev}', linewidth=2)

plt.title(f'BTC — ML apalancado x{lev} y StopLoss del 10%')
plt.xlabel('Fecha'); plt.ylabel('Capital (€)')
plt.grid(); plt.legend(); plt.tight_layout(); plt.show()

plt.figure(figsize=(12, 6))

for strat, hist in trad_hists.items():

    plt.plot(btc_df['DÍA'].iloc[window_size:], hist, label=strat)

    plt.title(f'BTC — Estrategias tradicionales • Apalancamiento x{lev} y StopLoss del 10%')
    plt.xlabel('Fecha'); plt.ylabel('Capital (€)')
    plt.grid()

    plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

    plt.tight_layout()
    plt.show()

```

```

# ----- RESULTADOS -----

print(f"\n==== BTC • Apalancamiento x{lev} y StopLoss del 10% ===")

for name, cap_end in results:

    print(f"{name:25s} → {cap_end:.2f} €")

print()

```

18.6. Nasdaq 100 Machine Learning con apalancamiento y Stop-loss

```

# -----
# NASDAQ-100 • ML + Estrategias tradicionales
# Apalancamiento & stop-loss dinámico (%) • Entradas/salidas al CIERRE
# -----
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier

# ----- 1) DATOS -----

nasdaq_df = pd.read_excel("Nasdaq100.xlsx")
nasdaq_df['DÍA'] = pd.to_datetime(nasdaq_df['DÍA'])
nasdaq_df = nasdaq_df.sort_values('DÍA').reset_index(drop=True)

# Target ±1,5 % para ML

nasdaq_df['Rendimiento_diario'] = nasdaq_df['Precio de cierre'].pct_change().fillna(0)*100

```

```
nasdaq_df['Target'] = nasdaq_df['Rendimiento_diario'].shift(-1).apply(  
    lambda x: 1 if x > 1.5 else (-1 if x < -1.5 else 0)  
)
```

```
# ----- 2) VARIABLES -----
```

```
features = [  
    'RSI_14', 'MACD', 'EMA_10', 'SMA_10',  
    'Precio > SMA_10', 'Precio > EMA_10',  
    'Precio > Banda Superior', 'Precio < Banda Inferior',  
    'Precio > SMA_20', 'Precio > STD_20',  
    'High', 'Low'  
]
```

```
estrategias = [  
    'Señal_RSI', 'Señal_MACD', 'Señal_SMA', 'Señal_EMA',  
    'Precio > SMA_10', 'Precio > EMA_10',  
    'Precio > Banda Superior', 'Precio < Banda Inferior'  
]
```

```
# ----- 3) PARÁMETROS -----
```

```
apalancamientos = [5, 10] # ×5 y ×10  
window_size = 60  
stop_loss_pct = 0.10 # pérdida máx. diaria = 10 % del capital
```

```

# -----
# 4-A) Estrategias tradicionales
# -----

def backtest_tradicional(col, lev):

    is_binary = nasdaq_df[col].isin([0, 1]).all()

    cap = 1000.0

    hist = []

    pos = short = entry = 0.0


    for i in range(window_size, len(nasdaq_df)):

        sig      = nasdaq_df.loc[i, col]
        close_px = nasdaq_df.loc[i, 'Precio de cierre']

        # ----- gestión de señales -----

        if sig == 1:

            if short:                      # cerrar corto

                cap += short * (entry - close_px)/entry * lev

                short = 0

            if pos == 0:                   # abrir largo

                pos, entry = cap, close_px


        elif sig == -1 and not is_binary:

            if pos:                      # cerrar largo

                cap += pos * (close_px - entry)/entry * lev

```

```

pos = 0

if short == 0:                      # abrir corto

    short, entry = cap, close_px

else: # sig - neutra

    if pos:

        cap += pos * (close_px - entry)/entry * lev

        pos = 0

    elif short:

        cap += short * (entry - close_px)/entry * lev

        short = 0

# ----- stop-loss dinámico -----

sl_eur = cap * stop_loss_pct

if pos:

    pnl = pos * (close_px - entry)/entry * lev

    if pnl < -sl_eur:

        cap += pnl; pos = 0

    elif short:

        pnl = short * (entry - close_px)/entry * lev

        if pnl < -sl_eur:

            cap += pnl; short = 0

hist.append(cap)

```

```

# cierre forzoso último día

last_close = nasdaq_df.iloc[-1]['Precio de cierre']

if pos:
    cap += pos * (last_close - entry)/entry * lev

if short:
    cap += short * (entry - last_close)/entry * lev

hist[-1] = cap

return hist, cap

```

```

# -----
# 4-B) - Modelo ML Random Forest
# -----

def backtest_ml(lev):

    cap = 1000.0

    hist = []

    pos = short = entry = 0.0

    rf = RandomForestClassifier(n_estimators=100,
                               class_weight='balanced',
                               random_state=42)

    for i in range(window_size, len(nasdaq_df)):

        train = nasdaq_df.iloc[i-window_size:i].dropna(subset=features+['Target'])

        if len(train) < window_size//2 or train['Target'].nunique() < 2:

```

```

hist.append(cap); continue

rf.fit(train[features], train['Target'])

day      = nasdaq_df.iloc[i]
close_px = day['Precio de cierre']

pred     = rf.predict(pd.DataFrame([day[features]]))[0]

# ----- señales ML -----

if pred == 1:
    if short:
        cap += short * (entry - close_px)/entry * lev
        short = 0
    if pos == 0:
        pos, entry = cap, close_px
    elif pred == -1:
        if pos:
            cap += pos * (close_px - entry)/entry * lev
            pos = 0
        if short == 0:
            short, entry = cap, close_px
    else:
        if pos:
            cap += pos * (close_px - entry)/entry * lev
            pos = 0

```

```
elif short:
```

```
    cap += short * (entry - close_px)/entry * lev
```

```
    short = 0
```

```
# ----- stop-loss -----
```

```
sl_eur = cap * stop_loss_pct
```

```
if pos:
```

```
    pnl = pos * (close_px - entry)/entry * lev
```

```
    if pnl < -sl_eur:
```

```
        cap += pnl; pos = 0
```

```
elif short:
```

```
    pnl = short * (entry - close_px)/entry * lev
```

```
    if pnl < -sl_eur:
```

```
        cap += pnl; short = 0
```

```
hist.append(cap)
```

```
last = nasdaq_df.iloc[-1]['Precio de cierre']
```

```
if pos:
```

```
    cap += pos * (last - entry)/entry * lev
```

```
if short:
```

```
    cap += short * (entry - last)/entry * lev
```

```
hist[-1] = cap
```

```
return hist, cap
```

```
# -----  
# 5) SIMULACIÓN COMPLETA por apalancamiento -----
```

for lev in apalancamientos:

```
# ----- ML -----
```

```
ml_hist, ml_cap = backtest_ml(lev)
```

```
# ----- Tradicionales -----
```

```
trad_hists = {}
```

```
resultados = [('ML', ml_cap)]
```

for strat in estrategias:

```
h, c = backtest_tradicional(strat, lev)
```

```
trad_hists[strat] = h
```

```
resultados.append((strat, c))
```

```
# ----- GRÁFICOS -----
```

1) ML

```
plt.figure(figsize=(10, 5))
```

```
plt.plot(nasdaq_df['DÍA'].iloc[window_size:], ml_hist,
```

```
label=f'ML • x{lev}', linewidth=2)
```

```
plt.title(f'Nasdaq-100 — ML Apalancado ×{lev} con StopLoss del 10% ')
```

```
plt.xlabel('Fecha'); plt.ylabel('Capital (€)')
```

```
plt.grid()
```

```
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
```

```

plt.tight_layout()
plt.show()

# 2) Estrategias tradicionales

plt.figure(figsize=(12, 6))

for strat, h in trad_hists.items():

    plt.plot(nasdaq_df['DÍA'].iloc[window_size:], h, label=strat)

    plt.title(f'Nasdaq-100 — Estrategias tradicionales • Apalancamiento ×{lev} con
StopLoss del 10%')

    plt.xlabel('Fecha'); plt.ylabel('Capital (€)')

    plt.grid()

    plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))

    plt.tight_layout()
    plt.show()

# ----- RESULTADOS -----

print(f"\n==== APALANCIAMIENTO ×{lev} con StopLoss del 10% ====")

for name, cap in resultados:

    print(f'{name:25s} → Capital final: {cap:.2f} €")

    print()

```