UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN

"DISEÑO E IMPLEMENTACIÓN DE ESTRATEGIAS DE ANÁLISIS TÉCNICO PARA TRADING ALGORÍTMICO"

TRABAJO FIN DE GRADO

Junio - 2023

AUTOR: Joel Sánchez Jódar
DIRECTOR: Jesús Javier Rodríguez Sala

RESUMEN

Este proyecto trata sobre la realización de un programa de trading algorítmico que es capaz de operar en los mercados financieros de forma automática. Para ello, el programa recibe datos, los procesa, realiza los cálculos determinados y en base a los resultados genera una señal de compra o de venta. Además, también se encarga de controlar el riesgo para evitar pérdidas descontroladas. Gestiona el riesgo mediante cálculos de volatilidad de un determinado momento, siendo estos cálculos realizados a mano previamente a la incorporación del programa. El estudio del análisis de volatilidad a mano también es explicado.

En la última parte también se expone a prueba el programa durante un mes, para poder ver los resultados que obtiene en el mercado. Estas pruebas también son explicadas en detalle para comprender el funcionamiento general.

AGRADECIMIENTOS

Agradecimientos a mi director del trabajo Jesus Javier por su atención y ayuda durante todo el desarrollo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ALGORITMOS

Capítulo 1
Introducción

1.1.- INTRODUCCIÓN A LOS MERCADOS FINANCIEROS

En los mercados financieros desde hace años se puede operar de muchas maneras. Gracias al desarrollo de nuevas tecnologías se han abierto las posibilidades de operar de formas que hace años no se podían, o sólo unas pocas personas podían permitirse[4].

Una de las formas modernas para comprar y vender activos financieros es a través de programas informáticos automáticos, los cuales mediante algoritmos, son capaces de tomar la decisión de comprar o vender un activo financiero. Un ejemplo de este tipo de software es el algoritmo 'Artemisa' de Sersan Sistemas. Cabe mencionar que es complicado encontrar algoritmos de este tipo públicos, ya que la mayoría están dentro de instituciones de forma privada.

Antes de entrar en detalle sobre esta forma de comercio, se explicará cómo funciona el trading en general, así como el mercado.

1.1.1.- Forex

Forex, también conocido como el mercado de divisas, es un mercado mundial donde se encuentran las divisas del mundo. En este mercado se compran y venden las divisas, siendo el mercado financiero más grande del mundo y el que más dinero mueve diariamente. Se estima que se negocian diariamente más de 5 trillones de dólares americanos[5,6].

Este mercado está abierto 24 horas todos los días, salvo los fines de semana que permanece cerrado. Esto permite que las cotizaciones de las divisas tengan mucha liquidez, lo que significa que la mayoría de veces, cuando queramos comprar, va a existir un vendedor, y cuando queramos vender, va a existir un comprador. Hay que tener en cuenta que no todas las divisas presentan la misma liquidez, ya que no todas están siendo comercializadas por la misma cantidad de personas. Por ejemplo, el dólar estadounidense o el euro son las divisas que tienen la mayor liquidez en este mercado, ya que son el par más negociado con un porcentaje total del 22,7%[5,6].

Este mercado comenzó sus inicios en el periodo babilónico, cuando se realizaban operaciones mediante el sistema de trueques como medio de intercambio. Este sistema se remonta al año 6000 a.C, el cual se considera como la base de la formación del comercio de divisas. En estos tiempos, se intercambiaban bienes por otros bienes, hasta que la demanda aumentó y años después se produjo la moneda de oro[8].

Las monedas de oro comenzaron a usarse como medio de pago ya que permitían portabilidad, divisibilidad, uniformidad y ponía límite a la oferta. Más adelante, a finales del siglo XVIII, se instauró el patrón oro. Con el patrón oro los gobiernos garantizaban que el valor del papel moneda estaba respaldado al 100% respecto al oro, es decir, que el oro y la moneda tenían el mismo precio. El papel moneda se impuso debido a que era muy pesado transportar grandes cantidades de oro.

Esto funcionó bien hasta la primera guerra mundial, donde los países europeos se vieron en la necesidad de suspender el patrón oro debido a que no podían financiar la guerra, y necesitaban hacerlo mediante la impresión del papel moneda, lo que supuso el fin de la relación 1:1 entre el oro y papel moneda.

En ese momento, cada país desarrolló su moneda fiduciaria, por lo que ya había en este punto más de un tipo de moneda en el mundo. Estas nuevas monedas permitieron los tipos de cambio que se utilizan a día de hoy entre divisas y su comercialización, donde cada

divisa tiene un precio diferente al resto y los tipos de cambio varían dependiendo de las situaciones económicas, políticas y sociales del territorio.

1.1.2.- Participantes en el mercado

Hay que tener en cuenta quiénes son los que participan en la comercialización de Forex, ya que para operar en este mercado, es necesario entender bien por qué se mueve el precio y quién lo mueve. Al ser un mercado que mueve tanto dinero, tiene que haber muchos actores operando detrás. A continuación, se explica de forma general quiénes son los participantes del mercado de divisas[7].

Bancos comerciales y de inversión

Los bancos comerciales y de inversión son los participantes que más dinero mueven en este mercado. El mercado interbancario es donde más volumen se crea, debido a que las transacciones de divisas entre bancos son de cantidades muy grandes. Estos bancos realizan transacciones de divisas para sus clientes, y además, ellos mismos realizan operaciones especulativas para su beneficio.

Bancos centrales

Los bancos centrales son los que se encargan de controlar el precio de su moneda a través de las políticas financieras. Esto lo hacen de diferentes maneras. La más conocida es a través de los tipos de interés.

Los tipos de interés sirven para dar más o menos valor al dinero. Estos son aumentados o disminuidos por los bancos centrales dependiendo de la situación económica, siendo utilizados como herramienta para frenar la inflación. Esto hace que el precio de la divisa correspondiente se vea muy afectado dependiendo lo que hagan, por lo que la mayoría de operadores están muy atentos a estas políticas.

Otra forma que tienen para controlar el precio es comprar o vender grandes cantidades de divisa, en caso de que sea necesario. Por ejemplo, si una divisa está cayendo en el mercado Forex, los bancos centrales acuden a comprar esa moneda para que no pierda tanto su valor y no cause graves problemas a la economía.

Gestores y fondos de inversión

Los gestores y fondos de inversión son los terceros participantes más importantes en el mercado. Este grupo comercializa divisas para fondos de pensiones, fundaciones, etc. Estos también operan con el propósito de un beneficio propio.

Empresas multinacionales

Aunque estos participantes no operan de forma directa, sí que afectan al mercado de forma indirecta. Las empresas cuando venden o compran productos al exterior, deben realizar intercambios entre las divisas correspondientes. Además, si es el caso de que venden productos a varios países extranjeros, la empresa tiene en su poder diferentes divisas, por lo que pueden ir cambiando de unas a otras dependiendo de sus necesidades.

Inversores individuales o 'retails'

Los inversores individuales, también llamados 'retails', son las personas que operan con poca cantidad de dinero en el mercado. Este participante por sí solo no tiene efecto apenas en el mercado, ya que no manejan tanto dinero como el resto de los participantes.

En este grupo se encuentran las personas que intentan obtener beneficios mediante técnicas para operar las divisas, ya sea con análisis fundamental (noticias, políticas, etc) o análisis técnico (análisis de gráficas).

1.1.3.- Trading

El trading es la especulación sobre instrumentos financieros con el objetivo de obtener beneficios a largo plazo. Hay que destacar la importancia de la temporalidad en la que se hace trading, ya que no todas las personas buscan beneficios en las mismas temporalidades. Hay diferentes tipos de traders como los scalpers, day traders o swing traders. Los scalpers buscan beneficios a muy corto plazo, siendo sus operaciones de una duración en el tiempo muy corta, incluso menor al minuto. Los day traders también buscan operaciones con beneficios de corto plazo, pero de mayor tiempo que los scalpers. Estos pueden mantener sus operaciones con una duración de hasta un día. Por último, los swing traders mantienen sus operaciones durante muchos días, e incluso semanas, buscando beneficios a largo plazo. En este caso, con el fin de obtener beneficios de los tipos de cambios en el mercado Forex, pero se pueden operar todo tipo de instrumentos financieros como acciones, índices, futuros, etc[2].

El trading se suele realizar aplicando alguno de los siguientes tipos de análisis: el análisis técnico y el análisis fundamental. Cada trader (así se denomina a la persona que ejerce el negocio de trading) elige su tipo de análisis para operar. Hay traders que utilizan solo un tipo de análisis y otros que prefieren utilizar una combinación de ambos. No hay ninguno mejor ni peor, hay todo tipo de traders y de ambas formas se pueden obtener beneficios. A continuación se comenzará explicando qué es el análisis técnico.

1.1.3.1- Análisis técnico

El análisis técnico es el estudio de los movimientos del mercado, generalmente mediante el uso de gráficas con el objetivo de intentar pronosticar movimientos futuros. Este análisis se realiza mediante información como la que se muestra en la figura 1.1, donde se muestra la propia acción del precio, además del volumen y de otros indicadores técnicos basados en datos del pasado. Este enfoque se basa en tres principios básicos: los movimientos del mercado lo descuentan todo, los precios se mueven por tendencias y la historia se repite[1].

Figura 1.1: Ejemplo de Gráfica para Análisis técnico.

El significado de que los movimientos del mercado lo descuentan todo, quiere decir que para las personas enfocadas únicamente en este tipo de análisis, en el gráfico está todo lo necesario por saber, ya que todos los impactos de noticias y políticas, entre otros, se van a ver reflejados directamente en el gráfico.

Los analistas técnicos hablan sobre la oferta y la demanda, ya que si la oferta es superior a la demanda, el precio debería de bajar, y esto se ve reflejado en el gráfico. Siendo esta la base de todos los pronósticos económicos. Por ello, para los técnicos, si el precio está subiendo, independientemente de las razones de por qué está sucediendo la subida, la demanda está superando a la oferta y por ende los fundamentos deben ser alcistas.

1.1.3.2- Análisis fundamental

Por otra parte, el análisis fundamental se centra en las fuerzas económicas de la oferta y la demanda, y entiende el por qué de los movimientos. Dicho de otra forma, se basan en las

noticias económicas, así como en datos que publican de forma periódica países y gobiernos. Estas noticias afectan directamente al precio, ya sea de una forma positiva o negativa, dependiendo de los resultados. Por ejemplo, en el caso de las acciones, el día en el que una empresa muestra sus resultados, si estos son mejor de lo esperados, el precio subirá, en cambio si son peor de lo esperado, el precio bajará[1].

El tipo de noticias que afectan a las divisas son las que afectan tanto al entorno social como político del país o países de la moneda específica (figura 1.2), como por ejemplo los resultados de empleo o el PIB.

08:30	CHF	IPP (Mensual) (Oct)	0,2%	0,2%
11:00	EUR	Comparecencia de Panetta, del BCE		
11:00	EUR	Producción industrial en la zona euro (Mensual) (Sep)	0,3%	1,5%
12:00	EUR	Comparecencia de Wuermeling, del Buba alemán		
12:00	EUR	Comparecencia de Enria del BCE		
12:25	BRL	Informe del mercado objetivo del BCB		
13:00	USD	Informe mensual de la OPEP		
13:00	INR	IPC (Anual) (Oct)	6,73%	7,41%
14:00	INR	IPC (Anual) (Oct)	7,30%	7,41%

Figura 1.2: Calendario económico con las noticias que afectan a las divisas en un día concreto.

Ambos enfoques, tanto el técnico como el fundamental, tienen el mismo objetivo, determinar la dirección en la que los precios se moverán en un futuro. El analista fundamental estudia la causa del movimiento mientras que el técnico estudia el efecto. Lo mejor es unir ambos análisis, de esta forma se tendrán más confirmaciones y un mejor entendimiento del mercado general.

Una vez entendidas ambas formas de análisis, se describe de forma detallada la información que se puede obtener mediante el análisis técnico concretamente, debido a que para realizar el trading algorítmico, nos centramos principalmente en este tipo de análisis. Los patrones o indicadores técnicos serán comentados más adelante.

1.1.3.3- Velas japonesas

Para el estudio del precio se utilizan los gráficos de velas japonesas, también llamados gráficos OHLC. La unidad básica de información es una vela, cada vela contiene cuatro

datos de información: el dato de apertura (*Open*), de cierre (*Closed*), el precio mínimo (*Low*) y el precio máximo (*High*).

Cada vela recopila esta información en la temporalidad que el analista configure, y con estos datos son con los que se trabaja. La temporalidad significa que si se elige 1 hora, cada vela va a representar la información obtenida en 1 hora, en otro caso igual, si se elige una temporalidad de 1 minuto, la información será recopilada en 1 minuto. Visualmente se ve como se indica en la figura 1.3, pero hay que tener en cuenta que de forma algorítmica únicamente se trabaja con los datos en crudo.

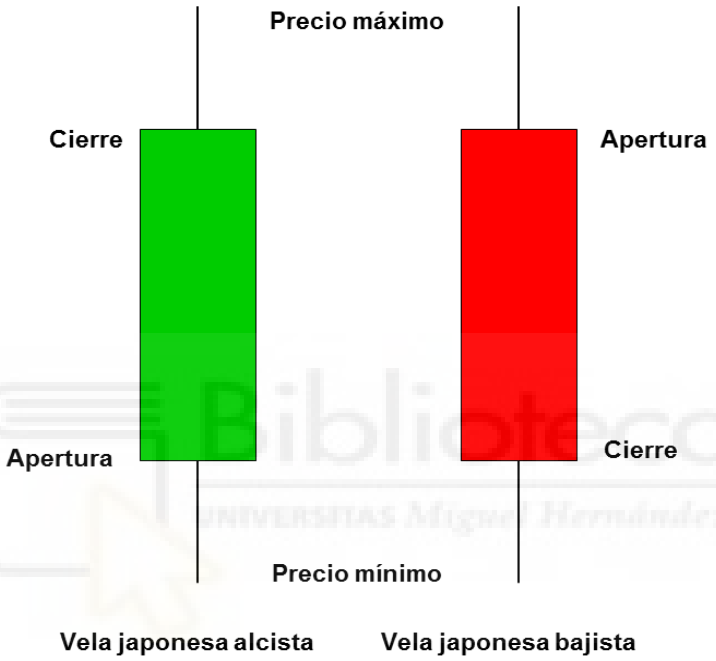


Figura 1.3: Representación visual de velas japonesas.

1.1.3.4- Lotaje en Forex

El lotaje es otro factor muy importante a la hora de comprar o vender en Forex. Esto es básicamente el número de unidades que se van a comprar o vender, es decir, un lote es una unidad de medida para la transacción. Un ejemplo para entenderlo, reflejando a la vida cotidiana, si se compra leche en un supermercado, es posible comprar un “lote”, el cual contiene 6 botes de leche.

Este número hay que tenerlo muy en cuenta para operar tanto con trading manual como con trading algorítmico, ya que hay que saber qué comprar y qué vender. El lote estándar es de 100.000 monedas, pero hay otros tipos de lotes, dependiendo las necesidades (ver Tabla 1.1).

Tabla 1.1 - Ejemplos de lotes

Lote	Unidades
Estándar	100.000
Mini	10.000
Micro	1.000
Nano	100

Ya se ha comentado lo que significa un lote, ahora se va a explicar cuánto dinero es cada cosa. Primero, los cambios en los precios de las divisas son muy pequeños (se suelen usar hasta 4 decimales), y por ello se miden en la unidad llamada 'pips'. Un pip es una unidad de medida muy pequeña de las divisas, y se obtienen beneficios cuando las monedas se mueven 'pips' a nuestro favor. Por esta razón, se opera con lotes de grandes cantidades de monedas para obtener beneficios, ya que para estas variaciones de precio con poca cantidad de monedas no se obtiene prácticamente nada[9].

Un ejemplo, si operamos el par de divisas EUR/USD (euro/dólar estadounidense), a un precio de 1.2130$, la fórmula para calcular el precio de un pip con un lote estándar sería la siguiente:

$$(0.0001 / 1.2130)*100.000 = 8.24\$$$

Hay que tener en cuenta que el valor del pip (0.0001) varía dependiendo del activo que se vaya a operar. Aún explicado esto, hoy en día existen calculadoras de pips que calculan el lote automáticamente sin necesidad de realizar estos cálculos a mano, por lo que es mucho más fácil para operar.

1.1.3.5- Apalancamiento

Una duda que puede surgir al ver los cálculos del apartado anterior es que parece que se requiere una cantidad muy grande de dinero para operar en Forex, ya que, debido a la baja volatilidad, se necesita comprar o vender cantidades muy grandes de divisas para poder obtener un beneficio interesante, lo cual es cierto, pero con el apalancamiento es posible resolver este problema.

El apalancamiento consiste en utilizar dinero 'prestado' por el broker (entidad intermediaria entre el mercado y el trader) para aumentar la cantidad de dinero que podemos utilizar para una operación de compra o venta. Es la relación entre el capital propio que arriesga el inversor y el que realmente se utiliza en una operación financiera gracias a dicho préstamo.

El apalancamiento financiero debe llevarse a cabo con mucho cuidado, ya que se pueden obtener más beneficios pero también más pérdidas. Sin el suficiente conocimiento en el tema, y sin una gestión de riesgo correcta, se podría llegar a perder todo el capital propio muy rápidamente.

En trading se usa el apalancamiento por las razones mencionadas anteriormente, con una gestión de riesgo que se comentará más adelante. Los tipos habituales de apalancamiento son: 1:10, 1:30, 1:50, 1:100 y 1:500. Un ejemplo práctico de esto, es que si se quiere comprar una acción que cuesta 1.000$ y se dispone de 100$, se puede utilizar un apalancamiento de 1:10 y así comprar esa acción con solamente 100$ de capital propio. En la tabla 1.2 se muestra un ejemplo comparativo de beneficios y pérdidas en operaciones con apalancamiento y sin apalancamiento.

Tabla 1.2 - Comparativa de apalancamiento

	Trading con Apalancamiento	**Trading sin apalancamiento**
Ratio de apalancamiento	1:20	1:1
Inversión (capital propio)	5.000 €	5.000 €
Exposición	100.000 €	5.000 €
EURUSD aumenta un 5%	+5.000 €	+250 €
EURUSD disminuye un 5%	-5.000 €	-250 €
Capital propio en caso pérdida del 5%	**0 €**	**4.750 €**

1.1.3.6- Costes de operación

Por último, otro factor importante a la hora de hacer trading es la cuestión de los costes por operar. Tener en cuenta estos gastos puede suponer la diferencia entre ganar o perder dinero. Hay diferentes tipos de costes, y cada uno merece la pena entenderlo para tenerlo en cuenta a la hora de comprar o vender.

El más común es el *spread* que es la diferencia entre el precio de compra y el precio de venta. Un activo no tiene el mismo precio a la hora de comprar que al venderlo. Esta diferencia entre ambos precios, es el beneficio que obtiene el bróker (intermediario) por hacer uso de su plataforma.

El otro tipo de coste es la comisión por operación, que consiste en un porcentaje fijo que cobra el bróker al abrir una operación de compra o venta. Hay otra comisión que solo afecta si se dejan operaciones abiertas durante la noche. Si se cierran antes no hay que

pagar nada, pero si se dejan durante la noche, se cobrará una pequeña comisión por cada día que dure la operación.

Existen otras comisiones, pero son más generales y que no afectan directamente a la hora de operar, como comisiones por inactividad o retiro de dinero. En trading algorítmico la comisión que más afecta es la del *spread*, aunque todo depende del tipo de estrategia utilizada o el tipo de cuenta para operar.

1.1.4.- Trading algorítmico

El trading algorítmico es la realización de operaciones de compraventa mediante un programa automatizado, desarrollado en un lenguaje de programación, y con una estrategia basada en algoritmos[11].

Estos programas permiten mediante software informático abrir y cerrar operaciones de acuerdo con las reglas establecidas por el algoritmo desarrollado, teniendo en cuenta todos los datos mencionados anteriormente. Una vez se dan los criterios programados para abrir operaciones, el algoritmo las abrirá automáticamente, de esta manera se elimina la necesidad de realizar el trabajo manualmente.

Estos programas informáticos funcionan mediante la unión de diferentes herramientas, como por ejemplo las API's a plataformas de mercados, cuentas de brokers, algoritmos, estadística, etc. Todo el funcionamiento será comentado con detalle más adelante.

1.1.4.1- Ventajas

Realizar trading de esta forma tiene sus ventajas. Algunas de estas son eliminar uno de los puntos más críticos de este negocio, que es el factor psicológico.

La psicología juega un papel muy importante en el trading, y teniendo un programa que opera por nosotros, la psicología se puede llevar mucho mejor, ya que no van a haber errores debidos a factores humanos, como podría ser la avaricia o pérdidas llevadas por emociones. El programa sólo operará en base a sus reglas y condiciones establecidas.

Otra ventaja, la que más llama la atención, es la automatización y la autonomía. Al realizar un programa que opera por nosotros, no es necesario estar pendiente de los gráficos o noticias, ya que de eso se encarga el software desarrollado. Además, de que puede trabajar durante mucho tiempo analizando el mercado continuamente, mientras que para un trader humano no sería posible.

1.1.4.2- Responsabilidad

Aún teniendo en cuenta las ventajas, no significa que una vez desarrollado un software haya que olvidarlo y vaya a estar obteniendo beneficios continuamente. Un programa de trading, como en todo desarrollo software, necesita procesos iterativos para ir comprobando que todo funciona correctamente, ir haciendo pruebas, mejoras, etc.

No se puede dejar a la suerte una vez finalizado el desarrollo. Además, aunque el programa opere de manera automática, el desarrollador debe estar revisando constantemente que todo está funcionando correctamente, ya sea el algoritmo, la conexión, el riesgo, etc.

Por esto, para que todo funcione correctamente, hay que tener en cuenta los siguientes pasos. Lo primero es desarrollar un plan. Se debe validar y comprobar mediante prueba y error, que el algoritmo y plan general están funcionando correctamente. Si se da el caso de que no se están obteniendo buenos resultados hay que revisar el programa y mejorarlo. Hoy en día se pueden realizar muchas pruebas en trading gracias a las nuevas tecnologías antes de utilizar dinero real, esto se conoce como *'backtesting'*. De esta forma, estaremos realizando pruebas con dinero ficticio y simulando las condiciones del mercado real.

Otro punto importante es considerar el trading algorítmico como un negocio. Se tiene que entender que aunque sea un software que opera automáticamente, no se puede enfocar la idea como si fuese un pasatiempo.

Por último, utilizar la tecnología como ventaja competitiva. El objetivo es utilizar todas las herramientas que sea posible para mejorar el beneficio, así como programas para recopilar datos históricos, pruebas con dinero virtual, intermediarios con altas velocidades de operaciones, etc. Todo esto puede ahorrarnos dinero, además de evitar estrés y frustración.

1.1.4.3- Lenguaje de programación

Para desarrollar el software, se necesita un lenguaje de programación. Los programas de trading algorítmico se pueden desarrollar con varios lenguajes de programación, pero en este caso, se va a utilizar Python.

Python es un lenguaje de programación muy versátil, y dispone de muchas librerías que permiten implementar algoritmos (p.e.: Tensor Flow), o para procesamiento de datos (p.e.: Pandas), lo cual ayuda mucho a la hora de recopilar y procesar datos.

Además, ya existen librerías para Python que permiten la comunicación directa entre el mercado, el bróker y el software de trading a través de código.

1.2.- JUSTIFICACIÓN DEL PROYECTO

Este proyecto se va a realizar por motivación personal. Personalmente, me encantan las finanzas y la programación, por lo que este proyecto es una intersección de ambas ramas. En el desarrollo del mismo se utilizarán conocimientos de finanzas, ingeniería, matemáticas y estadística. En concreto, las finanzas para entender trading, así como los precios, noticias, técnicas, etc, para obtener beneficios en los mercados financieros.

Por otra parte, los conocimientos de ingeniería son necesarios para todo el proceso de desarrollo del software, así como los procesos de ingeniería del software, el diseño de algoritmos, desarrollo, testing, etc.

Las matemáticas y estadísticas son aplicadas tanto para el algoritmo desarrollado, que funciona en base a datos, y para tener en cuenta la importancia del riesgo. Mantener una gestión de riesgo correcta para que estadísticamente a largo plazo se obtengan beneficios. Esto quiere decir que aunque el programa no gane todas las operaciones, tendrá en cuenta los porcentajes de pérdida y ganancias para calcular a largo plazo las probabilidades de acierto y fallo que se necesitan para ser rentable.

Otra razón del proyecto, es que a mi me gusta el trading manual, llevo operando los mercados financieros un tiempo, y desde hace un año he querido realizar un programa de trading algorítmico, por lo que este trabajo fin de grado supone una buena oportunidad para enfocarse en ello.

1.3.- OBJETIVOS

El objetivo principal es conseguir desarrollar un programa que opere en los mercados financieros de forma automatizada con una gestión de riesgo correcta y un algoritmo bien diseñado. Conseguir un programa hecho en Python que sea eficiente, obtenga datos, realice su preprocesamiento, haga las llamadas correctamente al intermediario financiero, y que consiga operar de manera decente.

Un objetivo más personal sería conseguir beneficios a largo plazo, aunque no sea un porcentaje grande. En el caso de que no, al menos que no pierda mucho dinero, y las pérdidas estén bien controladas, es decir, que no queme dinero. En el futuro, el software irá mejorando mediante pruebas, para conseguir un programa que opere correctamente en los mercados financieros.

1.4.- LÍMITES DEL PROYECTO

Obtener beneficio no es el objetivo final del proyecto, pero sí se intenta llegar a ese punto.

No es objetivo desarrollar un programa que opere en todos los mercados financieros, me centro únicamente en el mercado de divisas. Por lo que puede ser que el algoritmo desarrollado, con la gestión de riesgo concreta, no sea válido para otros mercados.

Además, no es objetivo desarrollar ninguna interfaz para el programa, ya que los bots de trading no utilizan ninguna, suelen ser puestos en servidores y simplemente se ponen en ejecución.

Capítulo 2 Antecedentes y estado de la cuestión

2.1.- SITUACIÓN ACTUAL DEL TRADING ALGORÍTMICO

Desde el inicio de Internet y su posterior evolución, el trading algorítmico comenzó a ser un pilar muy importante en el mundo financiero. Actualmente se estima que el 60-75% del volumen total de trading proviene de sistemas automatizados según Select USA[12].

En 2003, el porcentaje de trading algorítmico respecto del total del volumen generado en los mercados financieros era de apenas un 15%, pero su uso ha ido creciendo sin parar, dando lugar a más de un 70% en los mercados actuales. En algunos mercados, como por ejemplo, el mercado Forex, se llega a estimar que más del 80% del volumen proviene de estos sistemas automatizados. Por ello mismo, según los datos, podemos observar que es un factor muy importante y en gran uso para todos los mercados actuales. Como dato

curioso, comentar que Asia, en concreto Japón, presenta el mayor porcentaje de uso de trading algorítmico del mundo, siendo los siguientes el mercado Americano y el Europeo.

2.1.1.- Tendencia actual

Actualmente, el trading algorítmico es usado mayoritariamente por instituciones que manejan grandes cantidades de dinero, y este método les permite poder operar en los mercados de una forma en la que manualmente sería muy costoso y difícil. Estas instituciones utilizan estrategias algorítmicas basadas en cálculos computacionales, dependiendo de los requerimientos que tengan para ejecutar y gestionar las operaciones.

Estas instituciones operan con grandes cantidades de dinero, por lo que los costes asociados a las operaciones también son muy grandes, por ello, al utilizar estos sistemas de trading algorítmico pueden ahorrarse grandes costes, así permitiendo mejorar sus beneficios. Al requerir de operaciones tan grandes, utilizan estrategias para realizar operaciones poco a poco de forma gradual, por ejemplo, si quieren comprar 1.000.000 de acciones o cualquier otro activo, si lo hiciesen de golpe podrían provocar grandes desplazamientos en los precios, por ello van comprando cada X segundos ciertas cantidades dependiendo de su estrategia para que sea lo más óptimo posible.

Otro beneficio para las instituciones, es que con ciertos algoritmos pueden operar a velocidades muy rápidas, permitiéndoles ganar beneficios de movimientos en el mercado que un humano no podría.

2.1.2.- Estadísticas por mercados

A continuación, se explicarán algunos datos relacionados con el crecimiento del uso del trading algorítmico por mercados, como el mercado de acciones o el mercado Forex. Según los datos proporcionados por Goldman Sachs, en el 2016, el 70% del mercado de acciones era operado mediante algoritmos, el de futuros por el 50%, el de opciones un 40% y el de Forex un 30%.

Acciones

En 2018, el 60-73% de todas las operaciones del mercado de acciones de EE.UU. fueron realizadas mediante algoritmos programados. Además, en Europa, los fondos de cobertura usaron algoritmos para operar con más del 80% de su volumen total. Por esta razón, se espera que entre 2021 y 2026 las acciones sean operadas con un volumen de 8.61 billones de dólares mediante algoritmos.

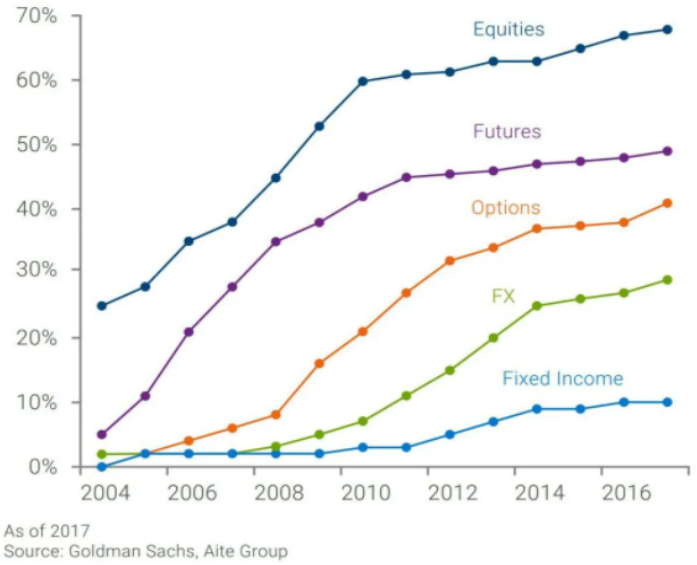


Figura 2.1: Cuota de mercado por negociación algorítmica

Forex

Este es el mercado donde mejor se implementó el trading algorítmico, permitiendo transferencias de dinero de una forma mucho más eficaz. En este mercado, incluso personas individuales (retails traders), han conseguido desarrollar algoritmos por sí mismos para operar.

Se calcula que en 2019 el 70% de todas las transacciones en el mundo de este mercado fueron de forma algorítmica. Además, instituciones han comentado que esperan que en los próximos años aumente otro 15% respecto del total.

Fondos de cobertura

Estos fondos tienen bajo su propiedad grandes cantidades de activos financieros, y con el paso del tiempo están usando cada vez más algoritmos para gestionar su portafolio de activos. De esta manera, además de ganar velocidad en las operaciones a realizar, obtienen un mejor precio de compra o de venta, ya que el algoritmo está programado para ello, proporcionándoles un mejor rendimiento.

De forma general, un 46% de los fondos de cobertura en Europa y Estados Unidos han utilizado algoritmos en 2020, aumentando un 33% respecto a 2019 [12].

Trabajos relacionados

Como curiosidad, el aumento del uso de estos algoritmos también ha traído un aumento en la demanda de trabajo para este tipo de personal con conocimientos en trading,

programación, análisis y matemáticas. En Reino Unido, en el año 2021, se solicitaron 87.560 puestos de trabajo para estos desarrolladores [12].

2.1.3.- Expectativas y pronósticos de crecimiento

Según un informe sobre el estudio del crecimiento de este sector, se espera que tenga un mayor crecimiento en un futuro [13]. Según dicho estudio, algunos de los factores que justifican esta conclusión son, que las empresas del sector están invirtiendo en nuevas tecnologías para aplicarlas al negocio del trading, el aumento de la presencia de trading algorítmico y el apoyo de instituciones para lograrlo, así como la disminución de los costes para realizar este tipo de transacciones.

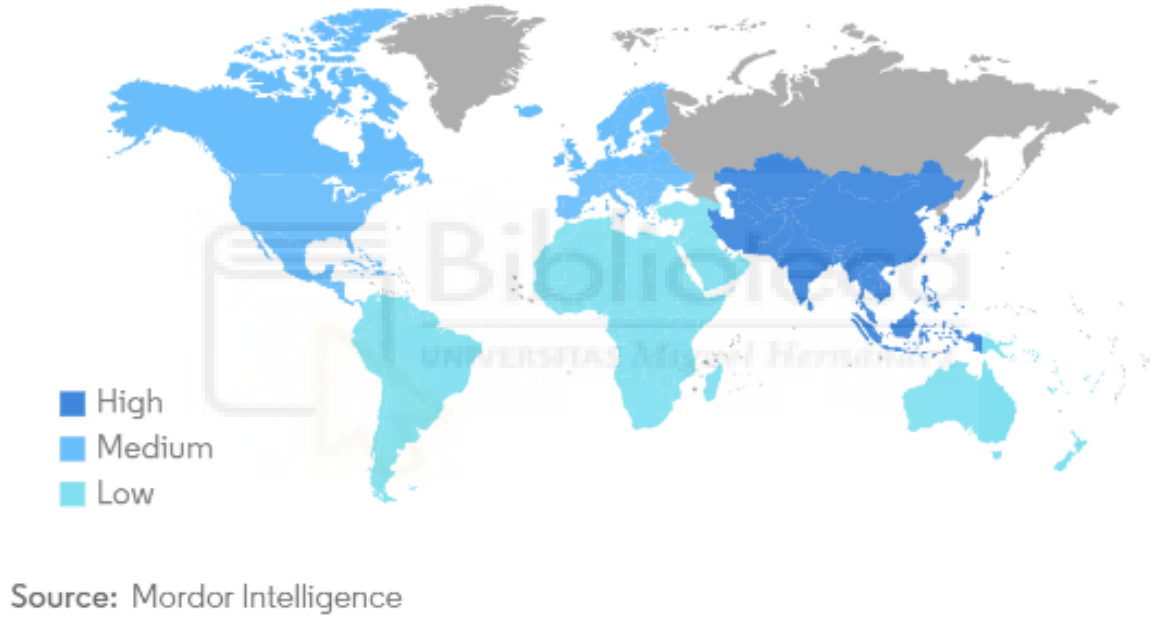


Figura 2.2: Crecimiento del trading algorítmico por región.

Por todo ello, se espera que el mercado del trading algorítmico crezca aún más, desde los 11.1 billones de dólares que había en 2019, hasta una cantidad estimada de unos 18.8 billones de dólares para 2024.

Hay que tener en cuenta, que el crecimiento que ha tenido este sector en los últimos años y el crecimiento que se espera del mismo, viene asociado a una serie de riesgos, donde todas las estrategias de negociación algorítmica, como las estrategias de alta frecuencia donde se opera a grandes velocidades por segundo, intentando obtener beneficios a muy corto plazo de tiempo, están sujetas a que se pueda perder dinero más rápido también, por lo que hay que tener mucho cuidado con el algoritmo desarrollado e ir revisándolo para asegurarse de que todo funcione correctamente.

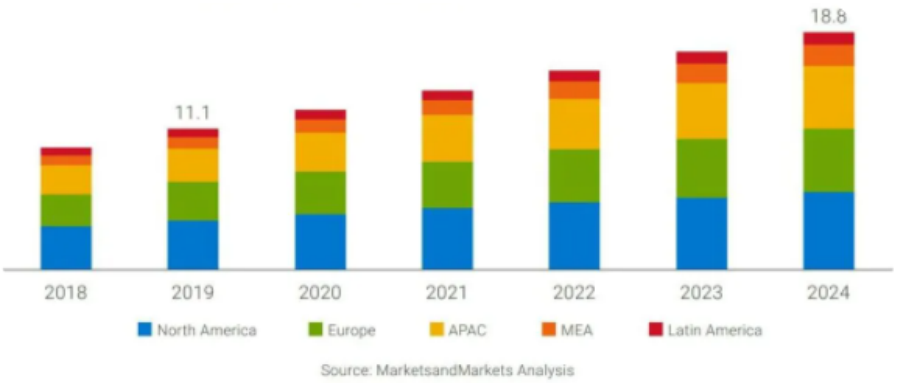


Figura 2.3: Crecimiento esperado por región entre 2018-2024.

Algunos ejemplos para tener consciencia de lo que puede suponer el uso de estos algoritmos podría ser el Crash ocurrido en Mayo de 2010 en el Dow Jones, donde una venta supuso una caída muy fuerte de 1000 puntos en un solo día, otro ejemplo problemático debido al uso de estos algoritmos fue el caso de Knight Capital, una firma que perdió en menos de una hora 440 millones debido al trading de alta frecuencia. También, en el caso de los traders retail, se estima que pierden en total al año 5 billones de dólares [14].

2.2.- HERRAMIENTAS DISPONIBLES EN EL MERCADO

Actualmente hay una gran cantidad de herramientas disponibles para trabajar con trading algorítmico, siendo imposible trabajar con todas, y a la vez poco eficiente. El objetivo es encontrar las herramientas que mejor se adapten a nuestras necesidades y a partir de ello buscar una optimización con la unión de las que necesitemos.

2.2.1.- Indicadores

Los indicadores son cálculos matemáticos que muestran información que puede ayudar a la hora de decidir si comprar o vender un activo. Son unos algoritmos cada uno desarrollado con un objetivo concreto. A continuación se explican algunos de ellos.

2.2.1.1.- Medias móviles

La media móvil (MA por sus siglas en inglés) es uno de los indicadores más utilizados y extendidos[1]. Es utilizado por una gran cantidad de personas para su operativa debido a que puede cuantificarse y verificarse muy fácilmente. La MA es un promedio de un cierto

bloque de información, con el objetivo de seguir una tendencia. El funcionamiento de esta consiste en calcular una media de los precios de cierre de los últimos días, pudiendo seleccionar los períodos de tiempo que mejor resultado nos de. Por ejemplo, si queremos calcular la media de los precios de los últimos 10 días, el cálculo consistiría en sumar todos los precios de cierre de los últimos 10 días y dividirlo entre 10, siendo un algoritmo el que cada día va recalculando la media. La fórmula es:

$$\mathrm{MA} = \frac{1}{n} \sum_{i=t-n}^{t} P_i$$

Siendo:

n: Cantidad de periodos con los que calcular la media móvil
t: Día (o instante de tiempo) actual
P_i: Precio del activo en el día 'i' (o instante de tiempo 'i')

A continuación se muestra un ejemplo del resultado visual del cálculo en un gráfico, en concreto del activo Libra esterlina/Dólar estadounidense(GBP/USD).

Figura 2.4: Media móvil de 9 días.

En este ejemplo, la línea de color azul es el resultado del cálculo de la MA con 9 días, pero para desarrollar una estrategia con este indicador, lo recomendable es probar diferentes cantidades de días y ver cual MA ofrece mejores resultados para la estrategia que se esté desarrollando. A continuación se muestra una variación con el cálculo con 20 días.

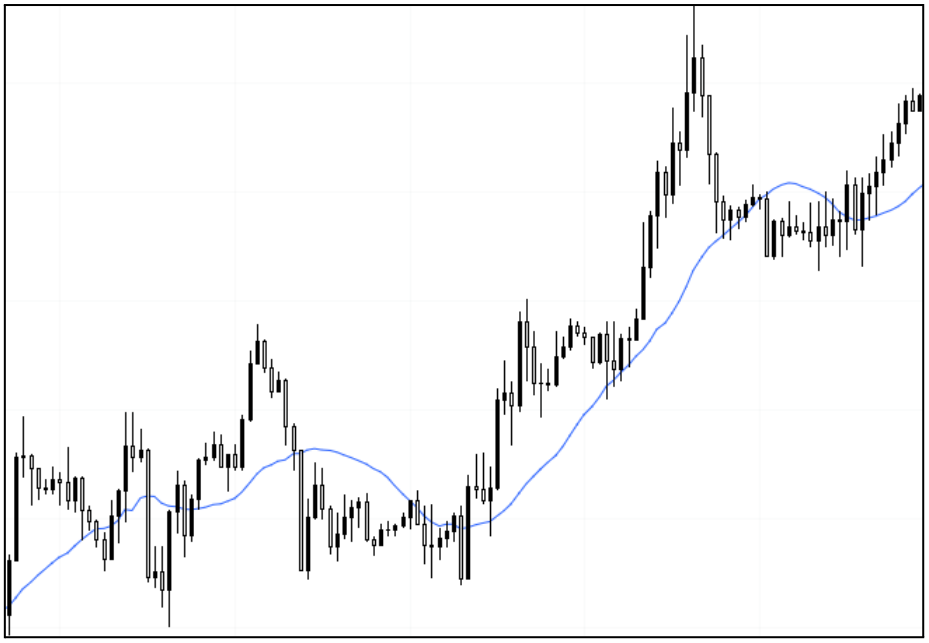


Figura 2.5: Media móvil de 20 días.

El uso de la MA en algoritmos puede variar dependiendo del programa. Algunos de sus usos pueden ser simplemente detectar la tendencia general del precio, viendo que si el precio está por encima de la media la tendencia es alcista, y si está por debajo es bajista. Otra técnica muy utilizada también, es el cruce de dos medias móviles con diferente número de parámetros, significando un cruce, una señal de compra o de venta.

2.2.1.2.- Medias móviles exponenciales

La media móvil simple tiene la crítica de que todos los días tienen el mismo peso para el cálculo, es decir el primer día tiene el mismo peso que el último. La media móvil exponencial (EMA) permite dar mayor importancia o peso a los días más recientes, la idea subyacente es que, para predecir el precio de mañana es más importante lo que ha pasado hoy o ayer que lo que pasó hace 10, 15 o 20 días. La EMA asigna una carga mayor a la información más reciente y una carga menor a la información sobre precios más antiguos, pero incluyéndose de todos modos a estos últimos en los cálculos. Esto se consigue asignando un valor porcentual al precio del último día, que se suma al porcentaje del valor del día anterior. La fórmula para el cálculo de la EMA es [15]:

$$EMA(T) = EMA(T-1) + K \star ((Precio(T) - EMA(T-a)))$$

Siendo:

T = Valor actual
T - 1 = Valor previo

K = 2 / (n - 1)
n = periodo elegido
a = dato previo del precio del activo

Figura 2.6: Ejemplo media móvil exponencial de 20 días.

2.2.1.3.- Estocástico

El estocástico (STC) fue desarrollado por J. Welles Wilder y publicado en 1978 [1]. Consiste en una banda horizontal en la que el precio oscila entre un rango de 0 a 100, donde se indica si el activo está sobrecomprado o está sobrevendido. Cuando los movimientos están por encima de 70, se consideran sobrecomprados, mientras que un movimiento por debajo de 30 sería una condición de que está sobrevendido. En otros casos estos niveles se pueden considerar 80 y 20 respectivamente, siendo está una condición más estricta que la anterior. Como se ha comentado anteriormente, el ajuste de los parámetros puede variar dependiendo de la estrategia que se utilice en cada momento. Para su cálculo se requieren dos líneas %K y %D (en la figura 2.7, %K es la línea azul y %D es la línea roja), que se calculan con las siguientes fórmulas [16]:

$$\%K = 100\left[\frac{(C - Ln)}{(Hn - Ln)}\right]$$
$$\%D = MA(\%K)$$

Siendo:

C = precio de cierre actual
Ln = precio más bajo durante las últimas 'n' sesiones
Hn = precio más alto durante las últimas 'n' sesiones

Figura 2.7: Estocástico aplicado a GBP/USD.

En la figura '2.7' se observa como cuando el precio llega a un nivel de sobrecompra en el estocástico (círculo de color rojo), el precio está sobrecomprado y tiende a reequilibrase, ocurriendo el caso contrario cuando el movimiento está por debajo del valor 30 en el rango del estocástico (círculo de color verde). Esta herramienta puede utilizarse también para detectar divergencias entre los precios reales y los datos del estocástico, todo dependiendo de su uso. Una divergencia ocurre cuando en el precio se da una estructura determinada mientras que en el STC se da otra, es por ello que se consideraría una divergencia. Un ejemplo de esto, sería encontrar en el precio un alto más alto, y al mismo tiempo coincidiera con un valor en el STC de alto más bajo, como ocurre una diferencia entre ambos, esto sería una divergencia. Estas divergencias se pueden operar, buscando entradas de compra o venta dependiendo del caso.

2.2.1.4.- Índice de fuerza relativa

El índice de fuerza relativa (RSI) mide la fuerza de los movimientos en los precios de un activo para evaluar si el precio de dicho activo está sobrevendido o sobrecomprado. Por definición, se puede observar que es muy parecido al estocástico, la diferencia con este, es que el RSI representa la relación porcentual entre las subidas y las bajadas.

En cuanto a la forma de usarlo, es muy parecido al estocástico, representando los mismos conceptos (con diferentes cálculos). Por tanto, ambos son representados en una escala porcentual en la que hay dos niveles, uno de sobrecompra y otro de sobreventa, en un rango entre 0 y 100. La principal diferencia es que el estocástico es más sensible a los

cambios de tendencia, por lo que sus oscilaciones son más bruscas. El RSI se calcula como indica la siguiente fórmula [18]:

$$RSI = 100 - \frac{100}{\left(1 + \left(\frac{AvgU}{AvgD}\right)\right)}$$

Siendo:

AvgU = promedio de las variaciones al alza del precio en el período N.
AvgD = promedio de las variaciones a la baja del precio en el período N.

Figura 2.8: Representación del RSI en gráfico

Se puede observar la diferencia entre el RSI y el estocástico en la figura 2.9, donde se observa que el estocástico (el inferior), es más sensible a los cambios de tendencia, existiendo más ruido en él.

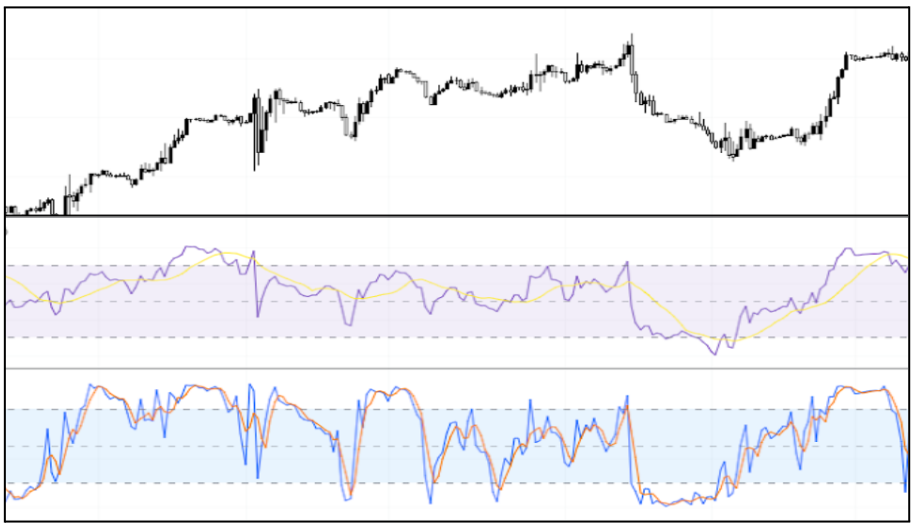


Figura 2.9: Diferencia RSI y Estocástico

2.2.2.- Volumen

El volumen es la cantidad de dinero que se introduce en el mercado durante un periodo de tiempo concreto y se representa mediante una barra vertical (figura 2.10). El análisis del volumen es muy importante para ciertos traders, ya que les sirve como confirmación de la dirección del precio, indicando si hay un gran capital por detrás del movimiento o es un movimiento sin fuerza.

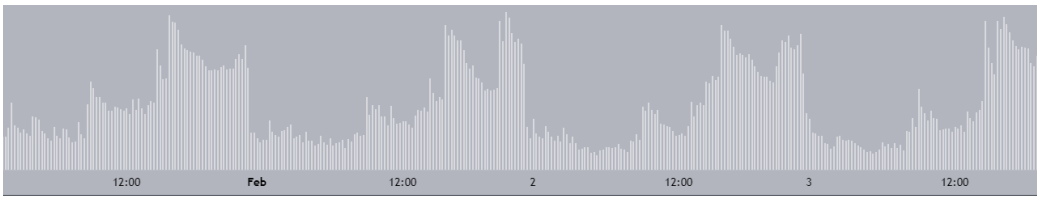


Figura 2.10: Representación del volumen.

El volumen mide la intensidad o urgencia que hay detrás del movimiento de precios [1]. Un volumen mayor significa un grado más alto de intensidad o presión, permitiendo dar una mayor confirmación al movimiento del precio. El volumen debe aumentar cuando hay un movimiento fuerte a favor de la tendencia.

Esta magnitud es fácilmente cuantificable, por lo que resulta muy fácil de aplicar en trading algorítmico, utilizándose conjuntamente con otros indicadores según la estrategia que se quiera implementar.

2.2.3.- Precio

El precio de un activo financiero se considera la mejor información sobre el mismo. Con estos datos, se puede sacar una gran cantidad de conclusiones sobre su dirección o tendencia, y combinándolos con otros factores como indicadores o volumen, se puede conseguir una buena estrategia. Con los datos del precio (apertura, cierre, máximo y mínimo), se pueden calcular de forma matemática ciertos elementos que pueden ser de gran utilidad, algunos de estos son las líneas de tendencia o soportes y resistencias, siendo estos algunos de los más utilizados por los traders. A continuación se explican tres de estos elementos que utilizan el precio del activo para su representación.

2.2.3.1.- Soportes y Resistencias

Los elementos más utilizados son los soportes y resistencias, tanto por su facilidad de entendimiento como por su efectividad si se usan correctamente. Los soportes son zonas

donde hay demanda, por lo que el precio se mantiene sobre ese nivel, en este punto el interés por comprar es mayor que el interés por vender, por lo que cuando el precio llega a esta zona el precio tiende a subir (figura 2.11 (a)).

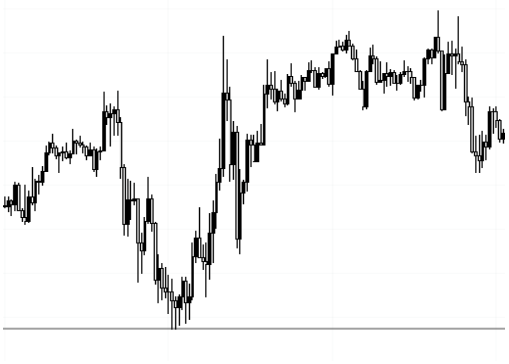


Figura 2.11 (a): Ejemplo Soporte

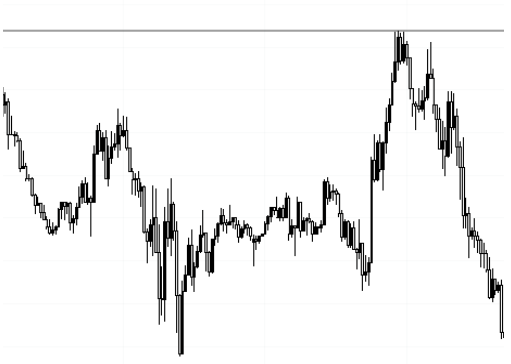


Figura 2.11 (b): Ejemplo Resistencia

Por otro lado, la resistencia es lo contrario al soporte. Una resistencia es una zona donde hay oferta, por lo que el precio se mantiene por debajo de ese nivel. En ese punto el interés por vender es mayor que el interés por comprar, por lo que cuando el precio llega a esa zona el precio tiende a bajar (figura 2.11 (b)).

Estos elementos son muy fáciles de visualizar en un gráfico, son ideales para hacer trading manual, el trader puede ver con claridad en el gráfico donde queda cada nivel de soporte o resistencia y tomar decisiones según su estrategia. Sin embargo, no es sencillo aplicarlos en un algoritmo. Para el cálculo de estos niveles con un programa, se puede desarrollar un algoritmo que use un número de velas determinado, que en este caso son un patrón de cinco velas donde la tercera tiene el dato con el mínimo de vela más bajo y las siguientes velas tienen mínimos más altos. Considerando el mínimo de la tercera vela como el soporte. En caso de resistencias sería el caso contrario, donde la tercera vela tiene el máximo más alto, y las siguientes máximos más bajos a esta (figura 2.12). En este caso, el algoritmo ha sido explicado con cinco velas, pero podría utilizarse con ‘n’ número de velas. Podrían hacerse pruebas para probar con diferentes números buscando mejores resultados.

Figura 2.12: Ejemplo algoritmo Soporte/Resistencia.

Este algoritmo codificado en Python quedaría de la siguiente manera [19]

Algoritmo 2.1: Función para detectar soportes.

```
def isSupport(df, i):
    support = df['Low'][i] < df['Low'][i-1] and
              df['Low'][i] < df['Low'][i+1] and
              df['Low'][i+1] < df['Low'][i+2] and
              df['Low'][i-1] < df['Low'][i-2]
    return support
```

Algoritmo 2.2: Función para detectar resistencias.

```
def isResistance(df,i):
    resistance = df['High'][i] > df['High'][i-1] and
                 df['High'][i] > df['High'][i+1] and
                 df['High'][i+1] > df['High'][i+2] and
                 df['High'][i-1] > df['High'][i-2]
    return resistance
```

Siendo ‘df’ el datasheet con los datos, y ‘df[‘Low’]’ la columna de datos de los valores mínimos de cada vela, y ‘df[‘High’]’, la columna de los datos de los valores máximos de cada vela.

Después de definir las funciones, se guardan todos los niveles encontrados de soporte y resistencia, pero aparece el problema de que se guardan demasiados, siendo poco óptimo. Por esto, se procede a filtrar los niveles para eliminar muchos niveles donde se considera que hay ruido, es decir, que no son óptimos. Para ello, se eliminan los niveles que están muy cerca de otros, dejando únicamente los niveles más importantes (ver figura 2.13).

Figura 2.13: Resultado de la codificación del algoritmo [19]

2.2.3.2.- Líneas de tendencia

La línea de tendencia representa el mismo concepto que los soportes y resistencias, la única diferencia con estos, es que la línea de tendencia no es horizontal sino que presentará una pendiente positiva (tendencia alcista) o negativa (tendencia bajista). La representación visual es una línea recta ascendente donde el precio se mantiene por encima de ella, en el caso de una tendencia alcista (figura 2.14), o una línea descendente con el precio por debajo de ella en el caso de una tendencia bajista.

Figura 2.14: Representación visual de la línea de tendencia

El problema de cuantificar esto para un algoritmo es el mismo que en el caso anterior. En este caso, como la línea de tendencia presenta una pendiente, se utilizan las ecuaciones matemáticas convenientes para identificar estas líneas [20]:

$$y = mx + c$$

Siendo:

c: La intersección con los puntos de los precios
x: el día del cual se quiere obtener el precio (válido para otras temporalidades)
m: pendiente de la recta ($m > 0 \Rightarrow$ tendencia alcista, $m < 0 \Rightarrow$ tendencia bajista)
y: el precio para el día x

Para calcular los parámetros de gradiente e intersección en la ecuación, lo primero es realizar una suposición simplificada, en la que queremos que la línea recta atraviesa los puntos máximos o mínimos del precio, dependiendo de si la tendencia es alcista o bajista, durante un periodo de tiempo determinado, lo que genera la ecuación:

$$C\ resistencia = Y\ máximo - (M\ resistencia * X\ máximo)$$

Para encontrar el gradiente, se consigue mediante la minimización de la distancia entre la línea y cada precio máximo o mínimo sobre el tiempo que estemos considerando, permitiendo de esta manera, que la línea solo pase por los máximos o mínimos. La ecuación resultante sería la siguiente:

$$\sum_{n=1}^{N} [y\,max\ -\ y\,n\ +\ m\,resistencia(x\,n\ -\ x\,max)]^{2}$$

Donde N es el intervalo de tiempo para los cálculos.
El resultado gráfico de los cálculos ya codificados queda de la siguiente manera:

Figura 2.15: Resultado de la codificación de la línea de tendencia [20]

Esto se puede utilizar en un programa mediante diferentes estrategias, una podría ser para operar a favor de la tendencia, o para operar cuando se rompe la estructura creada por un movimiento en estas zonas de precios, todo dependerá del algoritmo o estrategia desarrollada.

2.2.3.3.- Fibonacci

Fibonacci también es una herramienta muy utilizada en los mercados financieros. Se utiliza para calcular retrocesos, mostrando niveles importantes en los que el retroceso del precio puede finalizar y continuar su tendencia. El nivel más utilizado es el 1.618 [1], significando un retroceso del precio entre el máximo y el mínimo de un 61,8%. Este suele ser un nivel muy utilizado para entrar en operaciones en compra si la tendencia es alcista (Figura 2.16) o en venta si la tendencia es bajista.

Para el cálculo de los niveles de Fibonacci en un programa de trading, se busca el mínimo y el máximo en el precio, para poder trazar el rango en el que se encuentra el precio, y así poder posteriormente calcular los niveles importantes de retroceso, con prioridad sobre el 61,8.

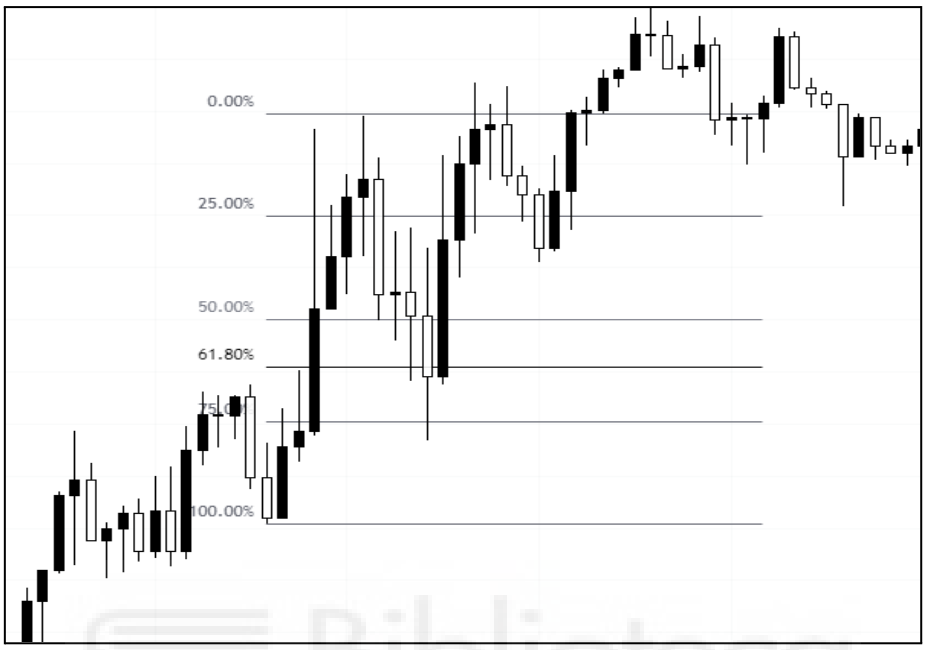


Figura 2.16: Ejemplo Fibonacci.

Esta herramienta es muy utilizada en la operativa especulativa diaria, tanto a corto como a medio y largo plazo [2]. La técnica de los retrocesos de Fibonacci funciona indiferentemente de que el precio esté a la baja o al alza.

2.2.4.- Estrategias

En los apartados anteriores se han descrito algunas de las muchas herramientas que pueden ser utilizadas para desarrollar un programa de trading algorítmico. El objetivo de las herramientas es juntar unas con otras, variando los parámetros de cada una de ellas para conseguir una buena optimización con el fin de obtener beneficios. Esto se consigue con una estrategia. Hay muchas estrategias generales usadas en trading algorítmico [17]. A continuación se explican algunas de ellas, pero teniendo en cuenta de que cada persona o institución desarrollará su propia estrategia.

2.2.4.1.- Seguimiento de tendencia

La estrategia más habitual consiste en el seguimiento de la tendencia actual del precio. El precio puede encontrarse en tres situaciones, alcista, bajista o en rango. Alcista es cuando

el activo está aumentando su precio, bajista cuando está disminuyendo y en rango cuando no está haciendo ni una ni otra cosa, simplemente está en un rango de precios.

Esta estrategia se basa, mediante herramientas como las medias móviles o incluso el propio precio del activo, en continuar la dirección del precio, así teniendo una ventaja estadística ya que se compra o vende a favor de la tendencia. Esta es la estrategia más sencilla de implementar en trading algorítmico [17] debido a que no se necesita hacer ninguna predicción del precio, simplemente se opera a favor del movimiento. Una forma fácil de implementar esta estrategia podría ser utilizando varias medias móviles con diferentes parámetros, con el fin de detectar la tendencia y operar en ella. Aún así, esto también presenta problemas, por ejemplo en épocas en las que el precio se encuentra en un estado de rango, donde el programa podría perder bastante dinero si no se programa cuidando esta parte.

2.2.4.2.- Oportunidades de arbitraje

Para realizar las operaciones de compra o venta se hace a través de mercados. Cada mercado puede tener en algunos momentos unos precios diferentes respecto a los otros, pero con variaciones muy pequeñas. Por ejemplo un activo que en un mercado tenga un precio de 1,053 y otro mercado tenga el mismo activo con un precio de compra de 1,052. Los programas que utilizan esta estrategia se centran en encontrar estas pequeñas diferencias de precio y operarlas a su favor, para ganar esa diferencia de precios, conocida como arbitraje. Estas diferencias de precios solo pueden ser operadas por programas, ya que para una persona, sería extremadamente complicado conseguir realizar operaciones de este modo de forma consistente. Hay que tener cuidado con este tipo de programas porque hay muchas empresas que consideran un problema beneficiarse de esta situación y pueden llegar a sancionar.

2.2.4.3.- Reequilibrio de fondos indexados

Los fondos indexados han crecido en popularidad debido a la simplicidad de una inversión pasiva, donde las personas compran y mantienen las posiciones a lo largo del tiempo, sin un seguimiento diario. Aún así, deben controlar su riesgo. Para obtener mejores resultados y reducir el riesgo potencial de pérdida, los inversores realizan un reequilibrio periódico de los activos que poseen en su cartera de inversión.

En base a esto, los programas de trading que se dedican a esto, buscan obtener beneficios durante estos reequilibrios de las carteras de inversores, con el objetivo de abrir operaciones antes o después del mismo.

2.2.4.4.- Modelos matemáticos

Los modelos matemáticos son empleados por los programas para obtener señales de compra o de venta dependiendo el caso. Estos modelos son elaborados mediante ecuaciones y datos cuantitativos en su totalidad, analizando datos del pasado para intentar obtener información de lo que pueda ocurrir en el futuro. Entre algunos de los modelos utilizados, podemos encontrar la estrategia delta neutral, que consiste en construir una cartera de opciones revisando que esté cubierta frente a variaciones en los precios de sus valores subyacentes.

2.2.4.5.- Reversión a la media

Esta estrategia sigue la premisa de que sea cual sea la fluctuación del precio de un activo, su precio eventualmente se equilibrará y volverá a un valor promedio después de haber alcanzado valores extremos. Por lo tanto, con esta estrategia, se comprará cuando el precio esté en límites inferiores, considerando que está en un valor extremo, y venderá cuando el precio esté en límites superiores, siendo el valor extremo opuesto a la compra.

Cuando el precio está en un límite es probable que retroceda hacia el centro. El programa de trading algorítmico se centrará en obtener beneficios de esos movimientos. El problema que tienen este tipo de algoritmos es que hay que gestionar muy bien el riesgo debido a que no se puede saber cuánto tiempo el precio va a estar en la zona de valor extremo, ni cuánto movimiento queda en esa zona, por lo que no se puede saber el momento exacto de compra o venta.

También hay otra forma de reversión de la media, que consiste en operar basado en la correlación de dos activos. La correlación consiste en la similitud o diferencia de los movimientos de dos activos, por ejemplo, si tienen una correlación positiva ambos activos van a fluctuar de la misma manera o muy parecida, mientras que si presentan correlación negativa, los activos se van a mover totalmente opuesto. Se pueden obtener beneficios con la correlación si el programa sigue dos activos, y por ejemplo, presentan correlación negativa, si uno de ellos comienza a subir su precio, podemos entender que en el otro activo el precio comenzará a bajar.

2.2.4.6.- Precio medio ponderado por volumen (VWAP)

Esta estrategia consiste en realizar compras o ventas utilizando el precio promedio ponderado para iniciar órdenes divididas en lotes pequeños. Utilizando esta estrategia con un programa, permite al trader obtener un precio de operación lo más cercano posible al

precio ponderado, tanto en tiempo como en volumen, ya que las transacciones son más rápidas y precisas.

2.2.5.- Resumen

En resumen se ha podido ver que el trading algorítmico tiene una gran cuota de mercado, siendo usado por una gran cantidad de instituciones en todos los lugares del mundo. Presenta actualmente un crecimiento de su uso, además de pronósticos de que seguirá aumentando en los próximos años.

Se han mostrado algunas de las muchas herramientas que pueden utilizarse para desarrollar un algoritmo para operar en los mercados financieros, entre los que se encuentran indicadores, el volumen o el mismo precio del activo. Entre estos, se ha podido observar que no hay ninguno mejor que otro, todo depende de la estrategia que se utilice para operar, y eso varía dependiendo de las necesidades y objetivos.

2.3.- VALORACIÓN

Tras el estudio de todas las herramientas mencionadas anteriormente, se puede llegar a la conclusión de que el buen uso de ellas depende de cada estrategia. No hay ninguna manera preestablecida para desarrollar un algoritmo que opere en los mercados financieros, es una cuestión de buscar una optimización con algunas de las herramientas, sin necesidad de combinarlas todas. Una vez desarrollado un algoritmo, se probaría con datos históricos para comprobar su funcionamiento, si opera bien o hay que seguir buscando y cambiando parámetros.

Capítulo 3
Hipótesis de trabajo

3.1.- Python

Python es el lenguaje de programación con el que se va a desarrollar el programa para operar en los mercados financieros. Es un lenguaje ampliamente utilizado tanto para aplicaciones web, como desarrollo de software, ciencia de datos y machine learning. Python presenta las características de ser un lenguaje interpretado, fácil de utilizar, tipeado dinámicamente, de alto nivel y orientado a objetos.

Python permite que los desarrolladores sean más productivos, debido a que se necesitan menos líneas de código para escribir un programa en comparación con otros lenguajes. Además, tiene una sintaxis sencilla, ya que es muy parecida al inglés. Otro de los beneficios de Python, es que cuenta con una gran biblioteca estándar que contiene códigos reutilizables para una enorme variedad de tareas [22], resultando así más fácil desarrollar programas ya que no es necesario escribir todo el código de todas las tareas desde cero. En

este tema, incorpora varias librerías que pueden ayudar con el análisis de datos y manipulación de los mismos, permitiendo para el desarrollo del programa de trading algorítmico un mayor entendimiento del mismo.

Algunas de las tareas importantes para el desarrollo del programa son extraer información a partir de los datos recolectados y sacar conclusiones de ellos. Con Python es posible realizar estas tareas; además de poder manipular los datos, como corregir y eliminar datos incorrectos o que no sean necesarios para el algoritmo, es decir, una limpieza de datos. También permite extraer y seleccionar características concretas de los datos, buscar estadísticas a partir de los datos o visualizarlos mediante diferentes formas de representación como gráficos.

3.1.1.- Mpl Finance

Un ejemplo de una biblioteca que se ha utilizado para el desarrollo del programa es Mpl Finance. Esta se ha utilizado para generar gráficos con los datos dibujados en ellos, que sirven para tener una idea visual de lo que el programa está realizando [32].

En este caso, se ha realizado una función llamada 'plot_all' que permite mostrar en el entorno de desarrollo la visualización gráfica.

```
#Para ver representación gráfica
def plot_all(data, levels):
    # Convertir los datos a un objeto 'DataFrame' compatible con 'mplfinance'
    data = data.set_index('time')
    data.index = pd.to_datetime(data.index)

    # Configurar los parámetros para las velas
    kwargs = dict(type='candle', volume=False, figratio=(12,8), title='Trading Chart')

    # Configurar los parámetros para las líneas horizontales
    hlines = dict(hlines=levels, linecolor='r')

    stoch_k_plot = mpf.make_addplot(data['%K'], color='b', alpha=0.5, linestyle='-', panel=1)
    stoch_d_plot = mpf.make_addplot(data['%D'], color='g', alpha=0.5, linestyle='-', panel=1)

    # Dibujar las velas y las líneas horizontales en la misma imagen
    mpf.plot(data, **kwargs, **hlines, addplot=[stoch_k_plot, stoch_d_plot])
```

Figura 3.1: Función para mostrar gráficos con datos.

Como se puede observar en el código de la función (Figura 3.1), se configura todo para introducir los niveles de soporte y resistencia que se pasan por parámetro, y posteriormente los niveles de sobrecompra y sobreventa. Una vez configurado, se ejecuta la instrucción 'mpf.plot' que muestra el gráfico con todos los datos anteriores.

3.2.- APIs

El término API es un acrónimo del inglés que significa *Application Programming Interfaces*. Esto es un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones, lo que permite la comunicación entre dos aplicaciones. Esta tecnología se utiliza para el desarrollo del programa de trading algorítmico, debido a que el programa debe de comunicarse con otra aplicación desde donde se realizan las operaciones de compra y venta. Además, también se utilizan APIs para extraer información, que permite posteriormente su análisis y procesamiento, y por otro lado también para conectarnos con nuestra cuenta en la plataforma de comercio. Todos estos usos serán comentados más adelante con más detalle [23].

3.2.1.- Metatrader 5

Metatrader 5 (MT5) es el software que se va a utilizar para desarrollar el programa de trading. MT5 es una plataforma de trading desarrollada por la empresa MetaQuotes. Esta es una plataforma de comercialización de activos múltiples para los mercados de divisas, acciones y futuros. Se encarga de conectar a traders individuales con diferentes mercados, como los índices bursátiles, por ejemplo el S&P 500 o incluso con commodities como el oro. Ofrece una gran cantidad de herramientas muy útiles para operar en el comercio financiero, ya que desde esta plataforma es donde se realizan las operaciones de compra y venta del activo. También cuenta con herramientas para la investigación técnica, el trading social, datos de una gran cantidad de activos, etc. Además, una de las herramientas que tiene es poder habilitar el trading algorítmico, gracias a unas APIs que nos permite conectarnos desde nuestro programa en Python con la plataforma MT5 [24].

Figura 3.2: Software Metatrader 5.

Desde las API de MT5 podemos ejecutar todas las órdenes comerciales posibles, entre ellas las órdenes pendientes, las órdenes de stop, que sirven para cerrar la posición en un precio determinado para gestionar el riesgo, y las de take profit, que es el precio en el que la operación se cierra automáticamente en ganancias. Todo esto permite que el programa desarrollado pueda operar de forma automática.

3.2.1.1.- Funciones

A continuación se describen algunas de las muchas funciones que tiene la integración de MT5 junto con Python, y que han resultado de utilidad para el desarrollo del programa.

Initialize

Esta función permite desde Python conectarse con la terminal de MT5, este es el primer paso para poder utilizar esta plataforma.

Login

El login permite conectarse con la cuenta de trading a partir de unos parámetros. La cuenta de trading se crea desde otra plataforma, en este caso, se utiliza IC Markets (se describe más adelante en el apartado 3.4.1). El usuario y contraseña será el de esta última plataforma. Un ejemplo de uso sería:

mt5.login(account=”usuario”, password=”contraseña”)

symbol_info

Con esta función se puede solicitar información de un activo concreto pasado por parámetro. Esto puede ser muy útil cuando queremos obtener la información de un instrumento financiero, devolviendo su spread, precio de compra, precio de venta, etc. La estructura de esta llamada es:

symbol_info(‘instrumento financiero’)

copy_rates_from

Esta llamada devuelve un datasheet con la información de un instrumento financiero concreto. Devuelve información desde una fecha que pasemos por parámetro. Por ejemplo, si ponemos desde el 01/02 y estamos en la fecha 03/02, devolverá la información de los últimos dos días. Los parámetros que acepta son:

```
copy_rates_from(
    symbol,      //Nombre del activo
    timeframe,   //Temporalidad de la vela
    date_from,   //Fecha inicial. Desde donde inician los datos
    count        //Número de datos máximo a recibir (opcional)
)
```

Para aclarar el parámetro de "*timeframe*" (temporalidad de la vela) se refiere a la información que se quiere recolectar dependiendo de si la vela es de 1 minuto, 5 minutos, etc. Cada vela tiene diferente información dependiendo de la temporalidad que represente, ya que una vela de un minuto va a representar la información de 1 minuto en el tiempo, mientras que una vela de cinco minutos, representa la información de 5 minutos en el tiempo. Los parámetros que se deben colocar en este lugar están en la documentación dependiendo la temporalidad, por ejemplo, para obtener la información de una vela de un minuto se pondría el parámetro 'TIMEFRAME_M1'.

La información que devuelve esta función es la fecha de la vela, que se puede pasar a hora/día/mes u otra forma de representación, el precio de apertura, el precio de cierre, el precio máximo, el precio mínimo, el volumen y el spread de la vela. La figura 3.3 muestra un ejemplo de los datos que se pueden obtener.

```
Display dataframe with data
                  time      open      high       low     close  tick_volume  spread
real_volume
0  2020-01-08 12:00:00   1.11382   1.11385   1.11110   1.11199         9354       1
0
1  2020-01-08 16:00:00   1.11199   1.11308   1.11086   1.11179        10641       1
0
2  2020-01-08 20:00:00   1.11178   1.11178   1.11016   1.11053         4806       1
0
3  2020-01-09 00:00:00   1.11053   1.11193   1.11033   1.11173         3480       1
0
4  2020-01-09 04:00:00   1.11173   1.11189   1.11126   1.11182         2236       1
0
5  2020-01-09 08:00:00   1.11181   1.11203   1.10983   1.10993         7984       1
0
```

Figura 3.3: Ejemplo datos obtenido por MT5

order_send

Esta función es la que permite realizar las operaciones de compra y venta. Tiene un único parámetro que es una estructura con toda la información para poder abrir la operación en el mercado. Esta función permite al programa de trading algorítmico automatizar las operaciones de entrada y salida.

La estructura para la operación cuenta con 17 parámetros. Algunos de los más importantes son *'action'*, *'symbol'*, *'sl'*, *'tp'* y *'type'*. El valor de *'action'* debe ser alguno de los encontrados en la documentación, son diferentes tipos de órdenes que se pueden solicitar. Puede haber confusión entre *'action'* y *'type'*, este último es el tipo de operación refiriéndose a si es una compra o una venta, donde por ejemplo una compra sería *'ORDER_TYPE_BUY'*, mientas que una venta sería *'ORDER_TYPE_SELL'*. En el caso de 'action', el tipo se refiere a si la solicitud es para una modificación de una posición ya abierta, que sería *'TRADE_ACTION_MODIFY'*, o si es para ejecutar una operación en el precio actual del activo, que sería con *'TRADE_ACTION_DEAL'*. Por otro lado, los parámetros *'sl'* y *'tp'*, son numéricos. Aquí se introduce el precio en el que cerrar la operación en caso de pérdida y el precio para cerrar la operación en caso de ganancia, respectivamente. Por último, el parámetro *'symbol',* es el nombre del activo del cual se va a realizar la operación.

3.2.1.2.- MQL 5

En este proyecto, el programa se desarrolla con el lenguaje de programación Python, pero hay que mencionar que existe un lenguaje de programación especializado en el comercio en MT5, llamado MQL5. Es un lenguaje de programación de alto nivel de programación orientada a objetos, que permite crear robots que operan en los mercados financieros e indicadores técnicos. Es un lenguaje muy parecido a C++, pero con la diferencia de que este está orientado exclusivamente a los mercados [26].

MQL5 contiene una gran cantidad de funciones que permiten el análisis de las cotizaciones, además de indicadores técnicos y medios de gestión y control de posiciones comerciales.

Tras ver este lenguaje de programación especializado, se puede llegar a la conclusión que desarrollar un programa de trading algorítmico es posible de varias maneras gracias a la posibilidad de elegir el lenguaje de programación que mejor se adapte a nuestras necesidades o requisitos.

3.3.- Herramientas de desarrollo (IDE)

Un entorno de desarrollo integrado (IDE) es un software que permite a los desarrolladores poder escribir sus programas gracias a las herramientas que les son proporcionadas. Los IDE permiten escribir, editar, probar y corregir código en un único lugar, facilitando el desarrollo de aplicaciones.

3.3.1.- Spyder

Spyder es un entorno de desarrollo integrado (IDE) de código abierto utilizado por una gran cantidad de científicos y analistas de datos. Este entorno de desarrollo es muy utilizado debido a que presenta características que son de gran ayuda para el análisis de datos, su visualización y posterior depuración [33].

Este IDE cuenta con varias zonas en la pantalla (Figura 3.4) desde donde se puede acceder a diferente información. En la parte izquierda de la pantalla es donde se escribe el código, es decir, donde se desarrolla el programa con Python. Este editor completo de código admite varios lenguajes de programación, no es necesario que sea Python, pero para este caso es el que se utilizará.

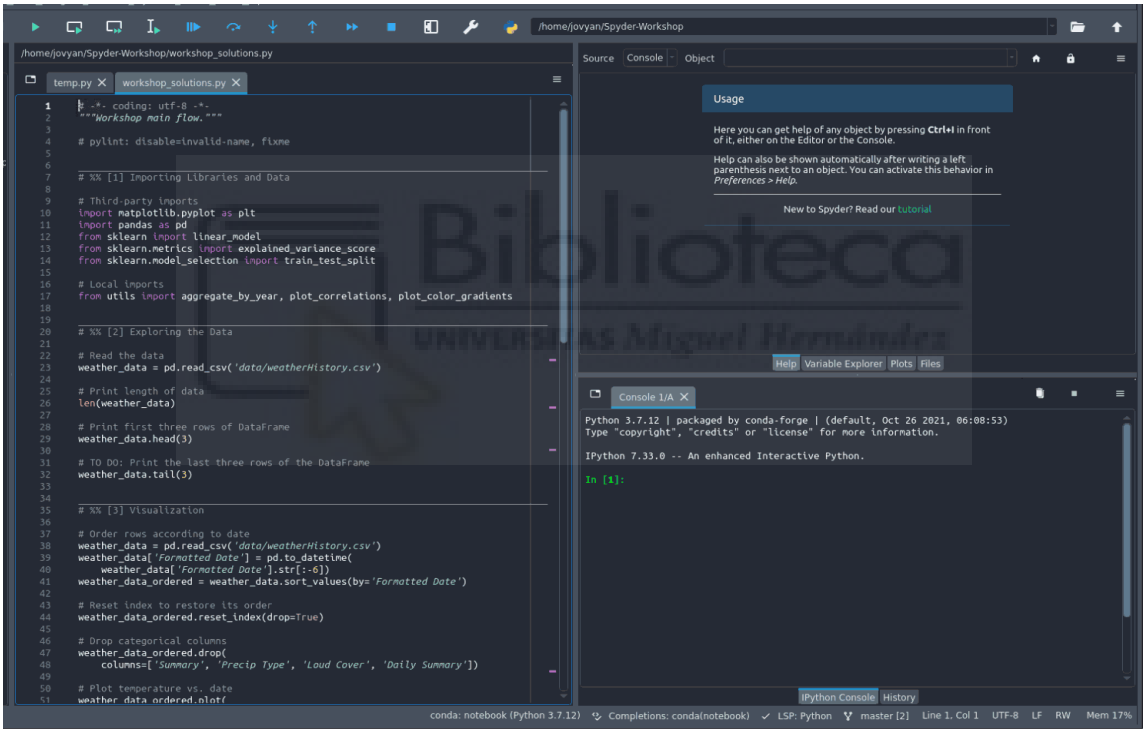


Figura 3.4: IDE Spyder.

En la parte inferior derecha se encuentra la consola interactiva de Python, que es una consola normal de comandos, que permite poder ver los resultados de la consola sin tener que movernos de la misma pantalla. En la parte superior derecha, se encuentra una herramienta muy útil para visualizar y analizar los datos con los que nos encontramos. En esta sección aparecen todos los datos recolectados y procesados, además de todas las variables que se utilizan en el programa con sus respectivos valores. Esto es una gran ayuda para el desarrollo del programa para poder conocer el estado de las variables del mismo de una forma visual y rápida. Además, en el desarrollo de un programa de trading algorítmico, puede darse el caso de necesitar ver un gráfico con las señales de compra o de

venta, o con otra información, y en esta sección se puede visualizar el gráfico generado (Figura 3.5). Por último, comentar que también cuenta con herramientas para la depuración del código.

Figura 3.5: Ejemplo visualización gráfico generado con Python en Spyder.

Hay varias formas de usar esta herramienta. Hay una manera que permite desarrollar desde la propia web con Spyder Notebook. La manera más utilizada es mediante su descarga. Para descargar Spyder, se puede hacer directamente desde su página web oficial, o desde otra plataforma llamada Anaconda [34], que reúne en ella diferentes aplicaciones y permite la descarga de Spyder desde ella. En mi caso, he utilizado esta última opción.

3.4.- Herramientas de uso externo

Además de todas las herramientas mencionadas anteriormente, se han utilizado algunas que no están directamente relacionadas con el funcionamiento del programa, pero que son necesarias para el correcto funcionamiento del mismo.

3.4.1.- IC Markets

IC Markets es una plataforma de agentes de cambio y bolsa al por menor. Es un broker, es decir es una institución encargada de las transacciones entre un comprador y un vendedor a cambio de las comisiones que se aplican por las operaciones cuando se ejecutan. En este broker es donde se crea una cuenta y se introduce el dinero con el que se va a operar, y en MT5 se introduce el usuario y contraseña de IC Markets para poder efectuar las operaciones.

Hay una gran cantidad de brókers, pero este en concreto es de los más conocidos y utilizados por varios factores. Presenta una gran flexibilidad y además unas comisiones muy bajas en este sector. También está regulado, por lo que es seguro (no una estafa, hay muchos "actores" ilegítimos en este sector), esto es algo muy importante y que da tranquilidad al inversor a la hora de introducir su dinero en este tipo de sistemas. Otro factor muy importante para su uso, es que cuenta con la opción de poder utilizar una cuenta demo con la que poder hacer pruebas sin coste.

Las cuentas demo son cuentas que contienen dinero ficticio pero te permiten operar como si fuese dinero real, es decir, puedes realizar las mismas operaciones y seguir los mismos pasos que si fuese dinero real. Esto permite poder realizar pruebas de los programas desarrollados en entornos reales sin la necesidad de utilizar dinero real, evitando así pérdidas de capital propio. Con el caso del dinero virtual, si ocurren pérdidas durante las pruebas, es posible volver a añadir nuevas cantidades de este "*dinero*", para poder seguir haciendo pruebas sin problemas.

Otra ventaja que ofrece este bróker, es la posibilidad de elegir diferentes tipos de cuenta dependiendo de la estrategia u optimización que el usuario quiera implementar. Encontramos tres tipos de cuentas, la cTrader Raw Spread, Raw Spread y Estándar. La cuenta cTrader Raw Spread es una cuenta exclusiva para otra plataforma del estilo MT5. Como en este proyecto se utiliza la plataforma MT5, esta cuenta en concreto carece de interés. Las otras dos cuentas son para MT5, por lo que sí que se podrían utilizar dependiendo del caso.

La cuenta Raw Spread es una cuenta en la que no hay spread, es decir, no existe la comisión que se aplica al abrir una operación que tiene en cuenta la diferencia entre el precio de compra y el precio de venta. En cambio, se aplica una comisión fija de 3.5$ por cada lote abierto. Por otro lado, en la cuenta estándar, no se aplica ninguna comisión fija, pero sí el spread, por lo que cada vez que se abra una operación se pagará esa diferencia de precio.

Tabla 3.1 - Comparativa de cuentas.

Tipo de cuenta	**Raw Spread**	**Estándar**
Comisión fija (por lote)	3.5 $	0.0 $
Márgenes desde (spread, en pips)	0.0	0.6
Depósito inicial (USD)	200 $	200 $
Apalancamiento máximo	1:500	1:500
Localización Servidor	Nueva York	Nueva York

En conclusión, cada tipo de cuenta tiene sus ventajas y sus inconvenientes, y la elección de una u otra depende de las pruebas realizadas y la que mejor se adapte a la estrategia desarrollada.

3.4.2.- TradingView

TradingView es la plataforma más completa para los traders. Permite visualizar todo tipo de activos, como divisas, acciones, criptomonedas, índices, etc. Sirve para realizar un análisis de los diferentes mercados, gracias a sus gráficos y herramientas [35].

Entre sus funciones encontramos todo lo necesario para realizar un análisis, como los soportes y resistencias, Fibonacci, una enorme cantidad de indicadores, etc. Además, podemos dibujar en el propio gráfico sin ninguna limitación, permitiendo a cada trader poder utilizar las herramientas que más le convengan.

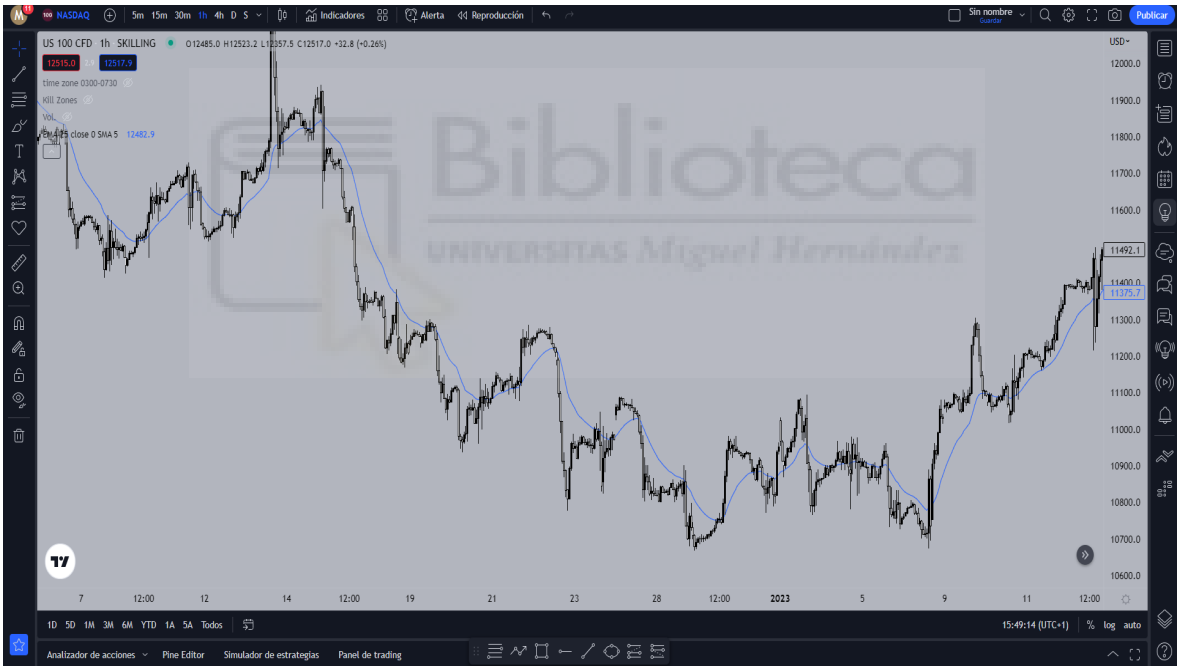


Figura 3.6: Plataforma TradingView.

Esta plataforma se usa en el proyecto de forma personal. No afecta directamente al programa desarrollado, pero sí se utiliza para poder visualizar gráficos y elegir herramientas para posteriormente, trasladar ciertas estrategias al trading algorítmico. También sirve para hacer pruebas visuales de gráficos entre el programa desarrollado y los gráficos que muestra el propio TradingView.

Capítulo 4
Metodología y resultados

4.1.- Planificación del proyecto

Para el desarrollo de este proyecto, se emplea la metodología del ciclo de vida iterativo. Esto es debido a que para la realización del bot de trading es necesario ir revisando y mejorando todos los aspectos que sean necesarios, como el riesgo o los propios parámetros de la estrategia.

El ciclo de vida iterativo es una metodología ágil para gestionar un proyecto. Esta técnica busca dividir el cronograma del proyecto en pequeñas fases relativamente independientes de la anterior. Estas fases son las iteraciones o ciclos del proyecto. Este ciclo se utiliza cuando en el proyecto no se puede definir de antemano todas las características que va a tener, como es este caso, ya que el bot se irá mejorando en base a las pruebas, recopilación de resultados y posterior optimización para la mejora.

Algunas de las ventajas de utilizar esta metodología son la gestión del riesgo del proyecto y el control del proyecto. En cuanto a la gestión del riesgo, las metodologías iterativas están pensadas para adaptarse rápidamente a los cambios, por lo que disminuye el riesgo, ya que está contemplada la revisión constante y reestructuración del proyecto a desarrollar, y en este caso, es un factor muy importante ya que no se puede realizar de una vez el proyecto esperando resultados positivos, ya que la revisión de los resultados y su futura mejora es un paso clave.

En cuanto al control del proyecto, como el proyecto está dividido en elementos independientes, como la gestión del riesgo, estrategia, gestión del capital, etc, es posible avanzar teniendo el control de donde se está actualmente y avanzar mediante metas cortas en determinadas áreas. Por ejemplo, si queremos mejorar la gestión del riesgo podemos centrarnos únicamente en ese aspecto.

Por otro lado, también se cuenta con aspectos negativos, como la falta de planificación inicial. Esto es debido a que en el inicio se puede establecer una estrategia pero con el paso del tiempo y su implementación, ésta cambie completamente en resultados finales debido a la iteración.

A continuación se muestra un diagrama con las etapas e iteraciones que se realizan continuamente para obtener mejores resultados, teniendo en cuenta que se basa en un ciclo de requerimientos, implementación y posterior evaluación. Cuando se obtienen los datos de la evaluación se vuelve a iterar para mejorar las partes que sean necesarias.

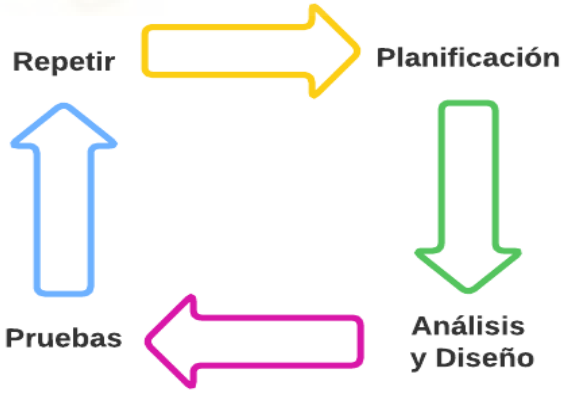


Figura 4.1: Etapas ciclo de vida iterativo.

La organización del desarrollo del programa se ha dividido en diferentes tareas, las cuales se muestran desglosadas a continuación con el tiempo de su realización. En el diagrama de Gantt las tareas se dividen para completar el proyecto en un plazo de 15 semanas.

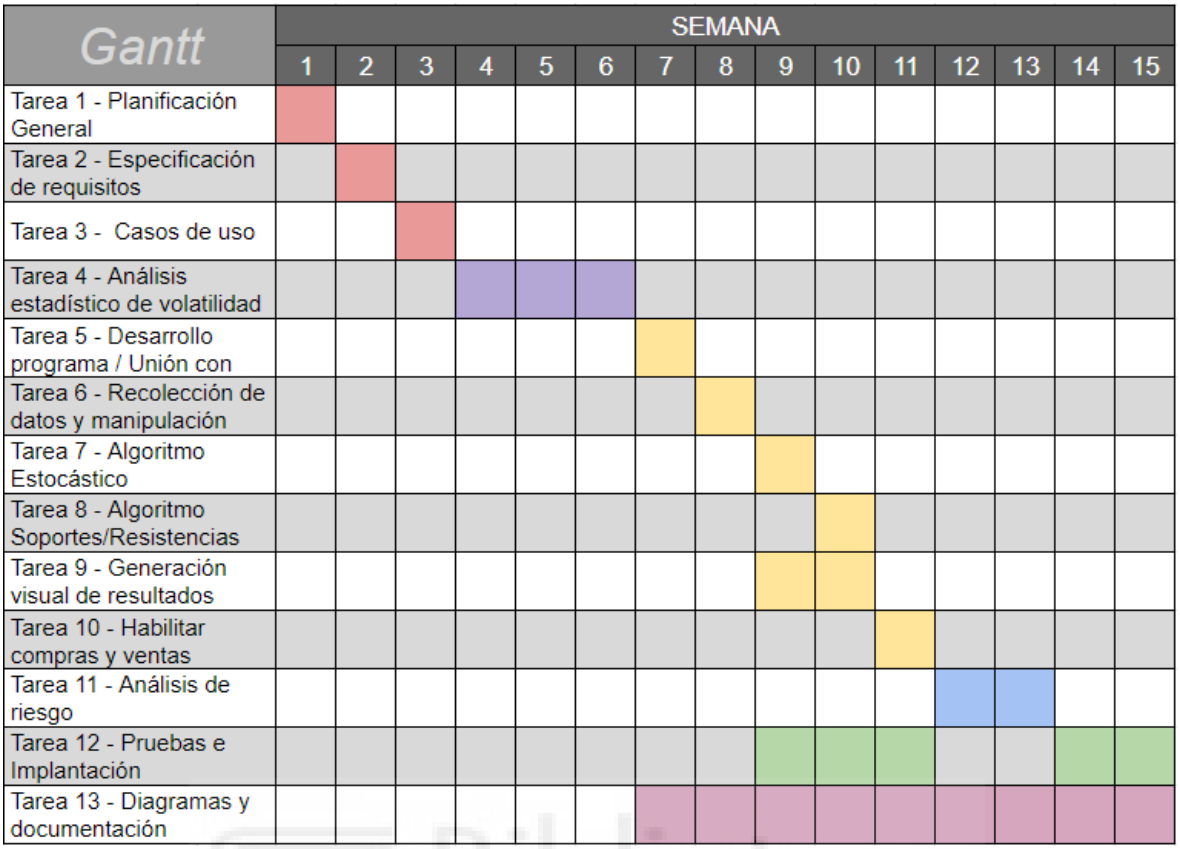


Figura 4.2: Diagrama de Gantt

4.2.- Captura de requisitos

En este proyecto no va a haber diversos tipos o roles de usuarios que vayan a acceder a una aplicación con distintas credenciales o permisos, ya que, como se ha mencionado anteriormente, se trata de desarrollar un programa para ser alojado en un servidor desde el que realiza ciertas operaciones en el mercado que sean convenientes para obtener beneficios, idealmente, no habrá ningún usuario "humano" interactuando con el programa.

4.2.1.- Requisitos funcionales

A continuación, en las tablas 4.1 a 4.6, se describen los requisitos funcionales que se han determinado para el presente proyecto.

Tabla 4.1: Requisito "Realizar compras".

RF-1	**Realizar compras**
Descripción	El programa debe poder realizar compras en un activo determinado si se da la oportunidad de compra.

Tabla 4.2: Requisito “Realizar ventas”.

RF-2	**Realizar ventas**
Descripción	El programa debe poder realizar ventas en un activo determinado si se da la oportunidad de venta.

Tabla 4.3: Requisito “Cerrar compras”.

RF-3	**Cerrar compras**
Descripción	El programa debe poder cerrar operaciones activas de compra cuando llegue a un beneficio concreto o cuando llegue a una pérdida concreta.

Tabla 4.4: Requisito “Cerrar ventas”.

RF-4	**Cerrar ventas**
Descripción	El programa debe poder cerrar operaciones activas de venta cuando llegue a un beneficio concreto o cuando llegue a una pérdida concreta.

Tabla 4.5: Requisito “Gestionar el riesgo”.

RF-5	**Gestionar el riesgo**
Descripción	El programa debe ser capaz de gestionar el riesgo y sus parámetros de forma automática sin necesidad de intervención externa.

Tabla 4.6: Requisito “Analizar el mercado en base a datos”.

RF-6	**Analizar el mercado en base a datos**
Descripción	El programa debe ser capaz en base a datos históricos de tener una estrategia determinada para analizar el mercado y decidir si realizar una compra o una venta.

4.2.2.- Casos de uso

Como ya se ha comentado, no se prevé que haya usuarios “humanos” que interactúen sobre el sistema, pero el sistema si que interactuará mediante un API con el mercado, por lo que los casos de uso de este proyecto son acciones que toma el propio sistema de forma automática. En otras palabras, el propio programa será un agente automático que realizará diferentes acciones dependiendo del momento y de los diferentes parámetros y señales que recibe. Este agente automático es el único “usuario” que se contempla en este desarrollo, y los casos de uso son las acciones que este puede realizar.

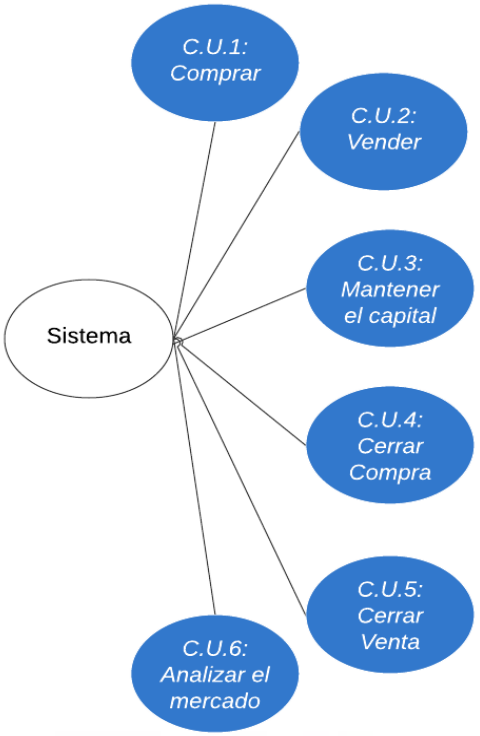


Figura 4.3: Diagrama de casos de uso.

Como se observa en la Figura 4.3, el sistema cuenta con seis casos de uso, entre los cuales están los básicos que debería tener un programa de trading automático, de entre los cuales, los principales son comprar y vender. Además, cuenta con otros casos de uso, como analizar el mercado, que es el que permite que el programa recoja los datos de la plataforma Metatrader 5 y analice la situación actual para poder generar las señales de compra o venta. En las tablas 4.7 a 4.12 se muestra con más detalle cada caso de uso.

Tabla 4.7: Caso de uso “Comprar”.

C.U. 1	Comprar
Actores	Sistema
Descripción	El programa realiza una compra en un activo determinado cuando según los datos ve una oportunidad.
Dependencias	-
Precondición	Debe poder comprar siempre que tenga suficiente capital y gestionando el riesgo.
Secuencia normal	P1 - Analizar el mercado P2 - Tras el análisis recibe señal de compra P3 - Enviar orden de compra a la plataforma
Poscondición	

Tabla 4.8: Caso de uso “Vender”.

C.U. 2	Vender
Actores	Sistema
Descripción	El programa realiza una venta en un activo determinado cuando según los datos ve una oportunidad.
Dependencias	-
Precondición	Debe poder comprar siempre que tenga suficiente capital y gestionando el riesgo.
Secuencia normal	P1 - Analizar el mercado P2 - Tras el análisis recibe señal de venta P3 - Enviar orden de venta a la plataforma
Poscondición	

Tabla 4.9: Caso de uso “Mantener el capital”.

C.U. 3	Mantener el capital
Actores	Sistema
Descripción	Si tras analizar el estado del mercado no recibe ninguna señal de compra o venta, se mantiene con el capital sin realizar ninguna acción y pasado un tiempo vuelve a iterar en busca de señales. De esta forma se evita tomar operaciones sin sentido que puedan llevar a la pérdida de dinero.
Dependencias	-
Precondición	-
Secuencia normal	P1 - Analizar el mercado P2 - No recibe señales de compra o venta P3 - Vuelve al paso 1
Poscondición	-

Tabla 4.10: Caso de uso “Cerrar compra”.

C.U. 4	Cerrar Compra
Actores	Sistema
Descripción	Cierra una operación activa de compra, con beneficio o pérdida.
Dependencias	C.U.1
Precondición	Debe haber una operación activa de compra para poder cerrarla.
Secuencia normal	P1 - Enviar señal de cierre de operación al bróker P2 - Cierre de operación
Poscondición	

Tabla 4.11: Caso de uso “Cerrar venta”.

C.U. 5	Cerrar venta
Actores	Sistema
Descripción	Cierra una operación activa de venta, con beneficio o pérdida.
Dependencias	C.U.2
Precondición	Debe haber una operación activa de venta para poder cerrarla.
Secuencia normal	P1 - Enviar señal de cierre de operación al bróker P2 - Cierre de operación
Poscondición	

Tabla 4.12: Caso de uso “Analizar el mercado”.

C.U. 6	Analizar el mercado
Actores	Sistema
Descripción	El sistema mediante una estrategia definida analiza el mercado mediante datos históricos con el objetivo de generar señales de compra o de venta.
Dependencias	-
Precondición	Se debe de haber programado una estrategia clara a seguir.
Secuencia normal	P1 - Analizar datos históricos P2 - Seguir los pasos de la estrategia P3 - Generar señales de compra, venta o ninguna
Poscondición	Dependiendo de la señal generada se activarán los casos de uso anteriores.

4.3.- Análisis estadístico de volatilidad

Antes de comenzar con el desarrollo del código del programa de trading, se realiza un estudio estadístico de la volatilidad del activo que se va a operar, en este caso del EUR/USD. Esto se realiza con el objetivo de entender los movimientos de este par de divisas y a partir de los resultados obtenidos, poder conocer características tales como su retorno medio cuando su precio aumenta o disminuye. Es de utilidad incluso para poder colocar nuestros precios de salida tanto en beneficio como en pérdida, ya que la volatilidad es lo que permite obtener beneficios ya que es cuando el mercado se mueve.

La volatilidad y el riesgo están asociados (se podría decir que son lo mismo). Cuando el activo se mueve más, podemos generar más beneficios, pero también podemos generar más

pérdidas, por ello es necesario conocer la media habitual de volatilidad de un activo para tomar decisiones en base a esta información. Por ejemplo, supongamos que el EUR/USD se mueve de forma diaria un 0.3% según los cálculos con datos históricos. Conocer este dato puede ayudar para saber que no se puede (o no se debe) abrir una operación de una duración de un día con un objetivo de beneficio de un 1% debido a que según los datos históricos estamos tratando de obtener más de 3 veces del movimiento diario de este par de divisas.

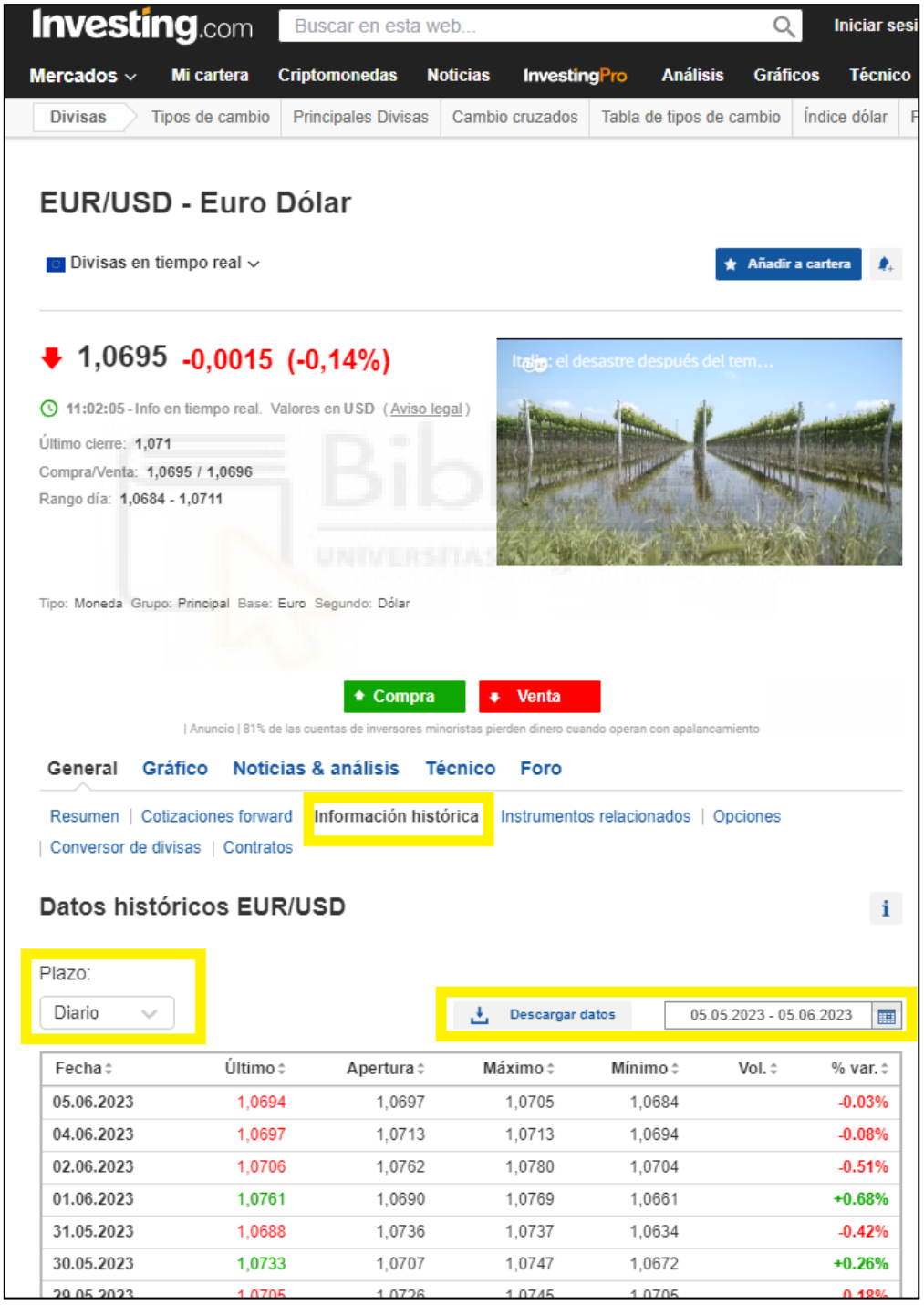


Figura 4.4: Pantalla de descarga de datos históricos para el par EUR/USD

Para comenzar con el análisis, lo primero es obtener los datos históricos. Para este proyecto, los datos se han obtenido del portal 'investing.com', este es una plataforma financiera y de noticias económicas, donde también se encuentran las cotizaciones de la gran mayoría de activos financieros. En esta web, en la sección del EUR/USD se encuentra un apartado de información histórica, donde se pueden descargar los datos en formato excel tanto diarios, semanales como mensuales (ver figura 4.4). Para este caso los datos han sido descargados en formato diario desde el año 2008.

Una vez descargados los datos (figura 4.5), estos contienen las columnas de fecha, último (último precio de ese día), apertura (primer precio de apertura), máximo, mínimo, volumen y variación del precio. Además, se han añadido dos columnas más que resultarán de utilizadas en el estudio, 'Open to Open' y 'High to Low'. 'Open to Open' se calcula como la diferencia entre el día actual y el día anterior dividido entre el día anterior, dando como resultado la diferencia entre los últimos dos precios de apertura expresado en porcentaje. La columna 'High to Low' representa el movimiento que ha habido en un día tomando el máximo y el mínimo de ese día, sería el movimiento que obtendríamos en el caso de comprar en el mínimo y vender en el máximo o viceversa, siendo algo prácticamente imposible, pero que sirve para conocer el movimiento diario. Esta se calcula con la diferencia entre el máximo y el mínimo dividido entre el mínimo, estando expresado también en porcentaje.

Fecha	Último	Apertura	Máximo	Mínimo	Vol.	% var.	Open to Oper	High to low %
26.04.2023	1,1052	1,0979	1,1095	1,0968	51,92K	0,73%	-0,607%	1,158%
25.04.2023	1,0972	1,1046	1,1068	1,0963	66,63K	-0,62%	0,500%	0,958%
24.04.2023	1,1041	1,0991	1,1051	1,0965	62,38K	0,49%	0,191%	0,784%
21.04.2023	1,0987	1,0970	1,0994	1,0938	52,68K	0,18%	0,128%	0,512%
20.04.2023	1,0967	1,0956	1,0991	1,0933	60,87K	0,12%	-0,146%	0,531%
19.04.2023	1,0954	1,0972	1,0984	1,0917	52,08K	-0,15%	0,439%	0,614%
18.04.2023	1,0971	1,0924	1,0984	1,0921	61,85K	0,41%	-0,601%	0,577%
17.04.2023	1,0926	1,0990	1,1000	1,0908	62,87K	-0,67%	-0,480%	0,843%
14.04.2023	1,1000	1,1043	1,1077	1,0972	70,22K	-0,40%	0,464%	0,957%
13.04.2023	1,1044	1,0992	1,1069	1,0977	80,32K	0,50%	0,724%	0,838%
12.04.2023	1,0989	1,0913	1,1001	1,0911	65,05K	0,72%	0,460%	0,825%
11.04.2023	1,0910	1,0863	1,0928	1,0857	59,99K	0,47%	-0,349%	0,654%
10.04.2023	1,0859	1,0901	1,0918	1,0831	29,79K	-0,35%	-0,201%	0,803%
07.04.2023	1,0897	1,0923	1,0926	1,0876	32,28K	-0,21%	0,183%	0,460%
06.04.2023	1,0920	1,0903	1,0939	1,0884	50,12K	0,16%	-0,456%	0,505%
05.04.2023	1,0903	1,0953	1,0970	1,0891	62,65K	-0,45%	0,477%	0,725%
04.04.2023	1,0952	1,0901	1,0974	1,0883	62,68K	0,52%	0,498%	0,836%

Figura 4.5: Muestra de datos de EUR/USD en Excel.

Tras la captura de estos datos, y añadidas las columnas 'Open to Open' y 'High to Low' que se necesitan para el estudio, se procede a realizar los cálculos correspondientes. Primero se analizará la columna de 'Open to Open'. Lo primero es realizar algunos cálculos básicos que se muestran a continuación en la tabla 4.13.

Tabla 4.13: Cálculos básicos columna ‘Open to Open’.

Cálculos	
Media	-0,005%
Error estándar	0,010%
Mediana	0,000%
Moda	0,000%
Desviación estándar	0,593%
Varianza simple	0,004%
Curtosis	-297,30
Asimetría estadística	0,10
Rango	7,141%
Mínimo	-3,268%
Máximo	3,872%
Suma	-18,51%
Total datos	3.800

Con estos cálculos aplicados a la columna ‘Open to Open’ podemos sacar algunas conclusiones, por ejemplo, el retorno medio en un día en el EUR/USD es de -0,005%, lo que nos puede decir que de forma diaria este par de divisas no se mueve en grandes porcentajes (lo cual es normal), ya que las divisas no suelen tener mucha volatilidad comparándolas con otros activos como acciones o materias primas como el petróleo.

Por otro lado, vemos que tanto la mediana como la moda es del 0%, por lo que de forma general con los datos históricos podemos decir que en temporalidad diaria los retornos serán bajos si no se usa apalancamiento, por lo que en tema de operar divisas es más recomendable buscar operaciones a más largo plazo donde se podría obtener más rentabilidad que de forma diaria, ya que como se observa, el movimiento entre precios de apertura de dos días consecutivos es el mínimo. Más adelante se realizará un análisis para ver la volatilidad en diferentes intervalos de tiempo.

Continuando con estos cálculos, se tiene también que la desviación estándar es del 0,593%. Esta dispersión de los datos se utilizará más adelante para calcular tres intervalos de datos entre los que se suele mover el precio.

A continuación, se calculan una serie de intervalos para obtener la frecuencia entre cada uno de ellos, para poder saber entre qué intervalos se mueve más el precio. Para ello se ha realizado un intervalo que va desde el 3% (el máximo) hasta el -3% (el mínimo), donde cada intervalo representa un 0,3%. (ver tabla 4.14).

Tabla 4.14: Cálculos de frecuencias y probabilidades 'Open to Open'.

Intervalos	Frecuencia	Etiquetas	Probabilidades	Probabilidades Acum.
-0,03	1	<-3%	0,03%	0,03%
-0,027	1	-3% to -2,7%	0,03%	0,05%
-0,024	1	-2,7% to -2,4%	0,03%	0,08%
-0,021	4	-2,4% to -2,1%	0,11%	0,18%
-0,018	17	-2,1% to -1,8%	0,45%	0,63%
-0,015	25	-1,8% to -1,5%	0,66%	1,29%
-0,012	56	-1,5% to -1,2%	1,47%	2,76%
-0,009	107	-1,2% to -0,9%	2,82%	5,58%
-0,006	297	-0,9% to -0,6%	7,82%	13,39%
-0,003	490	-0,6% to -0,3%	12,89%	26,29%
0	907	-0,3% to 0%	23,87%	50,16%
0,003	898	0% to 0,3%	23,63%	73,79%
0,006	565	0,3% to 0,6%	14,87%	88,66%
0,009	220	0,6% to 0,9%	5,79%	94,45%
0,012	120	0,9% to 1,2%	3,16%	97,61%
0,015	41	1,2% to 1,5%	1,08%	98,68%
0,018	25	1,5% to 1,8%	0,66%	99,34%
0,021	10	1,8% to 2,1%	0,26%	99,61%
0,024	6	2,1% to 2,4%	0,16%	99,76%
0,027	6	2,4% to 2,7%	0,16%	99,92%
0,03	1	2,7% to 3%	0,03%	99,95%
Más	2	>3%	0,05%	100,00%

También se han añadido dos columnas más para saber la probabilidad que hay de que el dato esté en un intervalo u otro y las probabilidades acumuladas. Vemos en la tabla 4.14 como la mayoría de movimientos se encuentran cerca del 0%, siendo este intervalo en concreto donde más frecuencia de datos hay, con una probabilidad del 23,87%. En cuanto a las probabilidades acumuladas vemos como cuando llega al 0% la probabilidad acumulada es del 50,16% lo que significa que los datos están repartidos casi por igual a ambos lados. Para visualizar mejor esta información, se ha realizado un gráfico de frecuencias (figura 4.6). En este gráfico se puede observar fácilmente como la distribución de datos se representa mediante una campana de Gauss, siendo esto normal, ya que como se ha mencionado es típico que en las divisas los movimientos sean pequeños de un día a otro, por lo que no se pueden esperar movimientos muy grandes la mayoría de días.

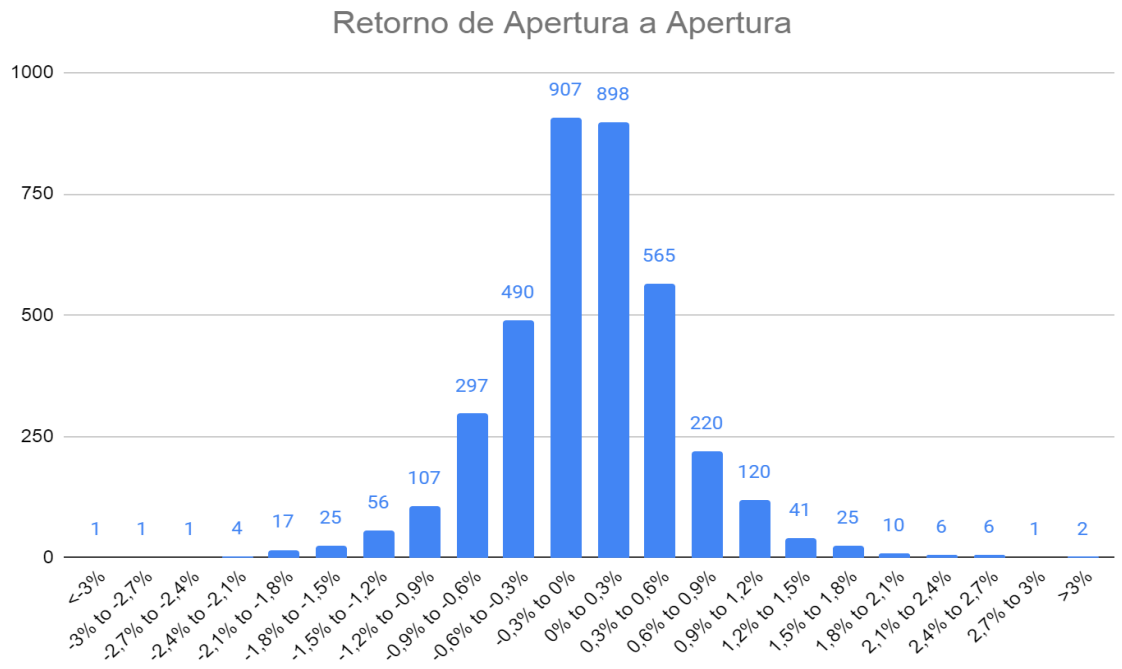


Figura 4.6: Gráfico de frecuencias del retorno de precio de apertura.

A continuación se realizan los mismos cálculos para la columna de 'High to Low' para saber el movimiento intradiario que existe entre el máximo del día y el mínimo del día, es decir, el movimiento que se podría aprovechar en un día para obtener beneficios. Lo primero es realizar los cálculos básicos que se muestran a continuación.

Tabla 4.15: Cálculos básicos columna 'High to low'.

Cálculos	
Media	0,876%
Error estándar	0,008%
Mediana	0,761%
Moda	0,791%
Desviación estándar	0,486%
Varianza simple	0,002%
Curtosis	-537,18
Asimetría estadística	2,01
Rango	4,74%
Mínimo	0,06%
Máximo	4,79%
Suma	33,275
Total datos	3.800

En este caso se observa que la media es mucho mayor que en el caso anterior siendo de 0,876%, es normal que sea mayor que en el caso anterior porque ahora lo que estamos calculando es una diferencia entre el máximo y mínimo de un día, mientras que en el caso anterior era solo la diferencia entre el precio de apertura de dos días consecutivos (en un día el precio puede subir o bajar pero posteriormente acabar en el mismo punto de donde comenzó). Con esta media se puede obtener, en el hipotético caso de realizar una entrada en el punto perfecto y una salida en el punto perfecto, un movimiento del 0,876%, pero obviamente esto no es realista, ya que es prácticamente imposible.

Por otro lado, vemos como la mediana y la moda son muy parecidas, estando más o menos de media en torno al 0,78%, por lo que en un día, el beneficio potencial no se debería esperar que fuera superior a este porcentaje, ya que no sería realista.

Tabla 4.16: Cálculos de frecuencia y probabilidades de 'High to low'.

Intervalos	Frecuencia	Etiquetas	Probabilidades	Probabilidades Acum.
0,002	19	<0,2%	0,50%	0,50%
0,004	347	0,2% to 0,4%	9,13%	9,63%
0,006	801	0,4% to 0,6%	21,08%	30,71%
0,008	887	0,6% to 0,8%	23,34%	54,05%
0,01	623	0,8% to 1%	16,39%	70,45%
0,012	424	1% to 1,2%	11,16%	81,61%
0,014	248	1,2% to 1,4%	6,53%	88,13%
0,016	156	1,4% to 1,6%	4,11%	92,24%
0,018	115	1,6% to 1,8%	3,03%	95,26%
0,02	68	1,8% to 2%	1,79%	97,05%
0,022	35	2% to 2,2%	0,92%	97,97%
0,024	23	2,2% to 2,4%	0,61%	98,58%
0,026	14	2,4% to 2,6%	0,37%	98,95%
0,028	10	2,6% to 2,8%	0,26%	99,21%
0,03	11	2,8% to 3%	0,29%	99,50%
0,032	5	3% to 3,2%	0,13%	99,63%
0,034	1	3,2% to 3,4%	0,03%	99,66%
0,036	1	3,4% to 3,6%	0,03%	99,68%
0,038	5	3,6% to 3,8%	0,13%	99,82%
0,04	2	3,8% to 4%	0,05%	99,87%
More	5	>4%	0,13%	100,00%

La tabla 4.16 y la figura 4.8 muestran un nuevo análisis de frecuencias que permite interpretar mejor los datos. En este caso, el tramo realizado está entre un 0,2% y más del 4%, a intervalos del 0,2%. En las probabilidades acumuladas se observa como el 70% de las veces el movimiento está por debajo de un 1%, siendo esto normal y ajustándose con los resultados que se han obtenido con los cálculos de la columna del 'Open to Open'. Esto es importante tenerlo en cuenta para no buscar puntos de salida fuera de lo normal, ya que debemos intentar tener las probabilidades a nuestro favor en la mayoría de casos posibles. En cuanto a la representación gráfica (figura 4.7) se observa como la mayoría de movimientos se encuentran entre el 0,8% y a partir del 1% disminuye drásticamente.

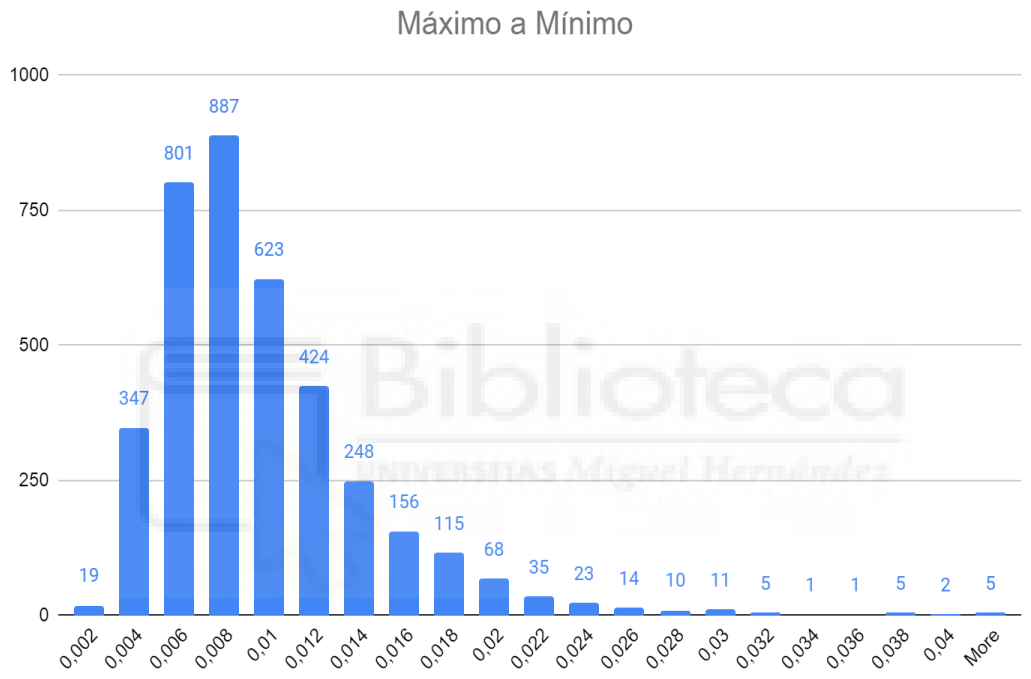


Figura 4.7: Gráfico de frecuencias de máximo a mínimo.

Para terminar esta parte, se calculan algunos datos que sirven para entender los movimientos generales de este par de divisas. Lo primero que se ha calculado es el retorno en un día en el que el precio del EUR/USD haya aumentado, siendo este de media un 0,43%. En el caso contrario, el retorno medio en un día en el que el precio haya disminuido es de -0,44%. Sabiendo esto, podemos concluir que en un día positivo para el EUR/USD podemos esperar de media un movimiento al alza de 0,43%, y en un día negativo podemos esperar de media un movimiento a la baja de -0,44%. Estos datos han sido calculados mediante la media únicamente de los días positivos y negativos respectivamente.

En cuanto al número de días que son positivos o negativos (tabla 4.17), el cálculo muestra lo siguiente. El número de días positivos y negativos es prácticamente el mismo, pero con un poco de diferencia hacia el positivo, por lo que de forma general podemos decir que hay un pequeño porcentaje de más días positivos, aunque en los días negativos hay un

movimiento mayor que los días positivos, siendo estos números con una diferencia muy pequeña.

Tabla 4.17: Total días positivos y negativos.

Total días Positivo	1894	49,84%
Total días Negativo	1882	49,53%
0	24	0,63%

A continuación se calculan los rangos entre los que es más probable que se mueva el precio mediante la desviación estándar calculada sobre la columna de 'Open to Open'. Para ello se establecen tres rangos y se obtiene la probabilidad de que se encuentre en cada uno de ellos. Para ello la fórmula en el rango 1 es la media más 1 multiplicado por la desviación estándar en el lado positivo, mientras que en el lado negativo sería la resta. El cálculo es igual para todos los intervalos pero multiplicado por el intervalo (por ejemplo, en el intervalo 2 se multiplica por 2).

Tabla 4.18: Intervalos a partir de la desviación estándar.

Desviación Estándar	**1**	**2**	**3**
Upper	0,59%	1,18%	1,77%
Lower	-0,60%	-1,19%	-1,78%

Tabla 4.19: Porcentaje de intervalos.

Std Dev 2	**Actual**	**Actual %**	**Normal**	**Normal %**
1	2847	74,92%	2592	68,20%
2	3598	94,68%	3625	95,40%
3	3748	98,63%	3792	99,80%

Con estos resultados vemos que en el intervalo 1 el movimiento máximo es del 0,59% y el mínimo del -0,60%, y en la tabla de abajo se observa como el movimiento se encuentra en este intervalo casi el 75% de las veces, por lo que nuestro enfoque estará dentro de este rango, porque es donde más probabilidades tenemos de que el precio se mueva en ese rango.

Por último, para saber la volatilidad de la divisa en diferentes marcos temporales, se han realizado unas tablas (4.20, 4.21 y 4.22) en las que se reúnen las medias de los movimientos de diferentes intervalos de tiempo. Para ello, se recogen los datos en diferentes temporalidades, diaria, semanal y mensual, y se realizan las diferentes medias de ellos para saber el movimiento medio y así poder ver en el largo plazo lo que se podría obtener de beneficio con estos movimientos.

Tabla 4.20: Cálculo de volatilidad diaria.

	1 SEMANA	1 MES	3 MESES	1 AÑO	2 AÑOS
MEDIA	5 Días	20 Días	60 Días	240 Días	480 Días
Volatilidad Diaria	0,86%	0,93%	0,84%	1,01%	0,82%

Tabla 4.21: Cálculo de volatilidad semanal.

	3 MESES	6 MESES	1 AÑO	2 AÑOS	3 AÑOS
MEDIA	12 Semanas	26 Semanas	52 Semanas	104 Semanas	156 Semanas
Volatilidad Semanal	1,74%	1,98%	2,11%	1,74%	1,67%

Tabla 4.22: Cálculo de volatilidad mensual.

	6 MESES	1 AÑO	2 AÑOS	3 AÑOS	5 AÑOS
MEDIA	6 Meses	12 Meses	24 Meses	36 Meses	60 Meses
Volatilidad Mensual	4,53%	4,69%	3,75%	3,58%	3,24%

Con este recuento de volatilidad en diferentes temporalidades se puede observar cómo a medida que la temporalidad aumenta, el movimiento es mayor y a medida que disminuye el movimiento es menor. Esto tiene sentido ya que las divisas suelen tener movimientos en un horizonte temporal de entre 1 a 3 meses debido a políticas monetarias y otros indicadores económicos, por tanto, no tienen movimientos muy volátiles en temporalidades bajas.

4.4.- Implementación

En la primera versión del desarrollo del programa de trading, se ha decidido colocar la estrategia del estocástico (la cual está explicada en más detalle en el apartado 2.2.1.3 del capítulo dos). Esta estrategia indica los niveles de sobrecompra y sobreventa de un activo determinado. Además se ha añadido a lo anterior la detección de los niveles de soportes y resistencias que se calculan de forma automática por el algoritmo desarrollado.

También, como método de ayuda visual para el desarrollo, se ha habilitado la opción de que el programa muestre una imagen con los indicadores dibujados, en este caso, del estocástico y de los soportes y resistencias. De esta forma, a modo de ayuda para el desarrollo, se puede ver de forma visual donde están los niveles en el gráfico.

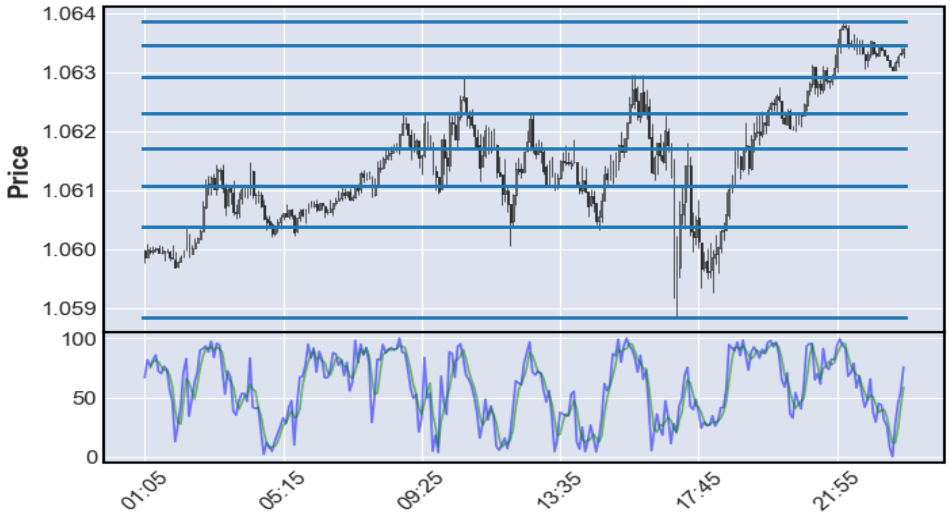


Figura 4.8: Imagen resultante de ejecutar el programa.

Para este caso, la estrategia consiste en que el programa debe vender cuando el precio esté en un nivel de sobrecompra, es decir que el estocástico esté en la parte superior de su área, y en ese punto el precio esté en un nivel de resistencia.

En este punto, ya están implementados ambos indicadores, y ahora lo que hace falta es generar la señal de entrada. Para ello, se establecen las siguientes condiciones. La primera es que para buscar una compra, en los datos del estocástico, los valores K y D sean ambos menores que 30. Cumplida esta primera condición, se busca el siguiente nivel de soporte, y este precio sería la entrada de la operación. Para el caso de una venta es al contrario, los valores K y D deben ser mayores que 70, y cuando ambos estén por encima, se busca la siguiente resistencia y ese sería el punto de venta.

Los valores K y D se calculan como se muestran a continuación [16]:

$$\%K = 100\left[\frac{(C - Ln)}{(Hn - Ln)}\right]$$
$$\%D = MA(\%K)$$

Siendo:

C: precio de cierre actual
Ln: precio más bajo durante las últimas ‘n’ sesiones
Hn: precio más alto durante las últimas ‘n’ sesiones
MA: media móvil de K

El diagrama de flujo de la figura 4.9 muestra una representación visual para la generación de una señal de compra o venta. Este diagrama únicamente muestra el motivo de la generación de la señal de una manera simplificada, ya que solo tiene en cuenta los niveles

de sobrecompra o sobreventa y los niveles de soporte y resistencia. No tiene en cuenta otros factores como los algoritmos del cálculo de K y D, los cálculos para el desarrollo de los niveles de soporte o resistencia dinámicos, la gestión del riesgo, la volatilidad calculada en el análisis estadístico del apartado anterior o los niveles de Take Profit o Stop Loss.

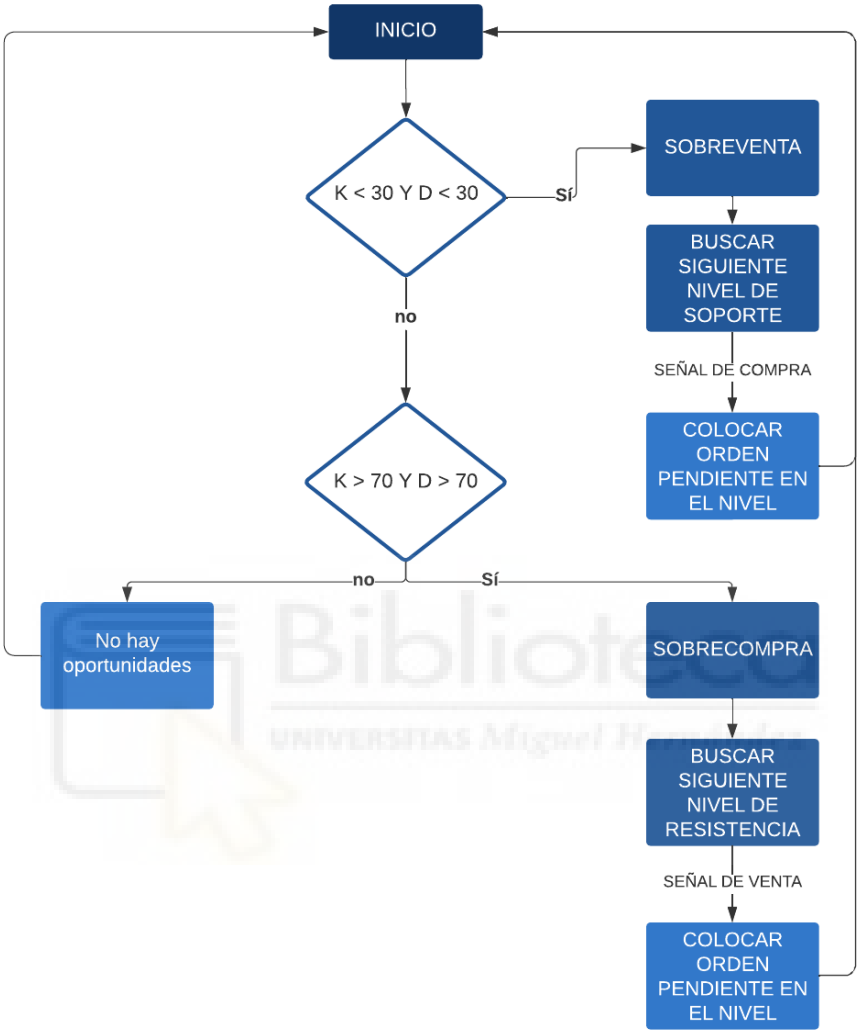


Figura 4.9: Diagrama de flujo para generación de señales.

En el programa, tanto las señales de compra como las señales de venta son representadas de forma numérica, dando como resultado un 1 una señal de compra, un -1 una señal de venta y un 0 en el caso de que no existan oportunidades (Algoritmo 4.1).

Algoritmo 4.1: Pseudocódigo decisión de señal.

```
1 if signal[-2] == 1:
2     print("señal compra")
3 elif signal[-2] == -1:
4     print("señal venta")
5 else:
6     print("no hay señal")
```

Como se puede observar en el código anterior, las señales se guardan en un array, ya que el programa va revisando los datos del data frame y decidirá si en ese momento determinado hay señal de compra o venta, y colocara su valor determinado en el array llamado 'signal'. Por esta razón, si el data frame tiene 200 filas de datos, el array signal tendrá 200 igual, ambos serán del mismo tamaño (figura 4.10).

Figura 4.10: Data frame y Array Signal.

Cabe mencionar que como se muestra en el ejemplo, la señal que tenemos en cuenta a la hora de colocar una operación no es la última señal generada, es decir 'signal[-1]', sino que se tiene en cuenta la penúltima la cual es 'signal[-2]'. Esta decisión tiene una explicación. El programa cuando se está ejecutando en tiempo real recoge los datos desde un instante tiempo 'x' hasta la actualidad, es decir, hasta la última vela que se ha generado en el gráfico. El problema que esto tiene es que esta última vela aún se está creando y no ha finalizado por lo que puede variar hasta que se acabe su temporalidad. Por ejemplo, si ejecutamos el programa en un determinado momento recolectando velas de temporalidad de 5 minutos, la penúltima vela ya ha finalizado, pero la última que aparece en los datos aún está en desarrollo. En el caso de que la penúltima vela sea de la 13:00 horas, si ejecutamos el programa a las 13:02, el último dato sólo está teniendo en cuenta esos dos minutos que han pasado desde la última vela que finalizó a la 13:00, y a esta aún le quedan otros 3 minutos de desarrollo. Por esta razón, como el último dato aún se está desarrollando puede generar señales falsas, porque puede ocurrir que en un momento dado

la vela haga que el precio disminuya mucho, pero en su último minuto de desarrollo se dé la vuelta completamente y acabe en el precio inicial, por lo que una señal generada durante el transcurso de la vela no es fiable.

4.4.1. Puntos de entrada y salida

Una vez que ya tenemos la generación de las señales de compra y venta, lo que se necesita es conocer justamente el punto de entrada de la operación y los puntos de salida, tanto en el caso de pérdidas como en el caso de ganancias.

En el caso del punto de entrada, este está definido por los niveles de soporte y resistencia que han sido comentados anteriormente. Estos niveles son generados por el algoritmo utilizado para su generación y guardados de nuevo en un array (figura 4.11).

Index	Type	Size	Value
0	float64	1	1.09187
1	float64	1	1.09146
2	float64	1	1.09225
3	float64	1	1.09306
4	float64	1	1.09344
5	float64	1	1.09046
6	float64	1	1.0899
7	float64	1	1.08814
8	float64	1	1.08944
9	float64	1	1.08486
10	float64	1	1.08562

Figura 4.11: Niveles de soporte y resistencia.

Para la generación de los niveles de soporte y resistencia, se ha utilizado un algoritmo que calcula en funciones separadas los soportes y las resistencias. Lo primero es entender que estas funciones son llamadas varias veces dentro de un bucle que recorre los datos del pasado para poder detectar tanto los niveles de soporte como los de resistencia.

Cuando encuentra un nivel de soporte o resistencia los guarda en el array de 'levels' (figura 4.12), el cual es el que permite definir los niveles de entrada que utiliza el programa para entrar a las operaciones de compra o venta. Ambas funciones **is_support** e

is_resistance cuentan con los mismos parámetros. La función **is_support** (figura 4.14) es la encargada de detectar los niveles de soporte. Su cálculo se basa en detectar estos niveles en base a un precio actual, los dos siguientes y los dos anteriores para determinar si es un nivel válido. Los parámetros que recibe son:

- **df**.- El datasheet con los datos para poder obtener los precios para el cálculo.
- **i.-** El número que aumenta de valor en el bucle para ir avanzando.

```
levels_aux = []
levels = []
for i in range(2, data.shape[0] - 2):
  if is_support(data, i):
    low = data['low'][i]
    if is_far_from_level(low, levels, data):
      levels.append((i, low))
      levels_aux.append(low)
  elif is_resistance(data, i):
    high = data['high'][i]
    if is_far_from_level(high, levels, data):
      levels.append((i, high))
      levels_aux.append(high)
```

Figura 4.12: Bucle de soporte y resistencias.

Como se observa en la figura 4.13, los cálculos para detectar los niveles de soporte se basan en los precios mínimos, ya que en estos precios es donde el precio se da la vuelta para continuar hacia arriba.

```
def is_support(df,i):
  cond1 = df['low'][i] < df['low'][i-1]
  cond2 = df['low'][i] < df['low'][i+1]
  cond3 = df['low'][i+1] < df['low'][i+2]
  cond4 = df['low'][i-1] < df['low'][i-2]
  return (cond1 and cond2 and cond3 and cond4)
```

Figura 4.13: Función para detectar soportes.

La función **is_resistance** es la encargada de calcular los niveles de resistencia. La mecánica es la misma que para los niveles de soporte, al igual que los parámetros. La única diferencia se encuentra en que aquí se toman los precios máximos, ya que son los precios en los que comienza a caer una vez llega a este punto.

```
def is_resistance(df,i):
  cond1 = df['high'][i] > df['high'][i-1]
  cond2 = df['high'][i] > df['high'][i+1]
  cond3 = df['high'][i+1] > df['high'][i+2]
  cond4 = df['high'][i-1] > df['high'][i-2]
  return (cond1 and cond2 and cond3 and cond4)
```

Figura 4.14: Función para detectar resistencias.

Por último, la función **is_far_from_level** es la encargada de controlar que durante el bucle de cálculo de los niveles de soporte y resistencia, no se repita ningún nivel, ya que esto podría traer efectos negativos en el desarrollo posterior del programa.

- **value**.- Es el valor mínimo o máximo dependiendo el caso del momento actual del bucle.
- **levels.-** Es un array con los niveles de soporte y resistencia calculados hasta el momento.
- **df.-** Es el datasheet con los datos.

```
def is_far_from_level(value, levels, df):
  ave =  np.mean(df['high'] - df['low'])
  return np.sum([abs(value-level)<ave for _,level in levels])==0
```

Figura 4.15: Función para evitar niveles repetidos.

Teniendo en cuenta lo anterior, una vez que de señal de compra, el programa va a buscar el siguiente nivel de soporte. Esto lo hace teniendo en cuenta el precio actual y buscando el siguiente precio de la lista que sea menor que el precio actual.

Para los niveles de salida en beneficio o pérdidas, se tiene en cuenta el análisis de volatilidad realizado anteriormente. Como se mencionó en ese apartado, no se debería colocar el Take Profit o el Stop Loss demasiado alejados del precio actual, debido a que históricamente se ha mostrado estadísticamente que no va a llegar a esos niveles. Si por ejemplo el precio se mueve en un día un 0.1% no tiene sentido colocar el nivel de salida en un 0,5% de distancia a menos que queramos aguantar posiciones durante más de un día. Si se quieren cerrar las operaciones de forma diaria se mira la volatilidad diaria y se colocan los niveles en base a ello. Según los datos calculados anteriormente históricamente la mayoría de días el movimiento suele oscilar entre 0% y 0,3%, por lo que personalmente, no colocaré los niveles de salida a más distancia.

La codificación de la volatilidad es muy parecida a los cálculos realizados en el estudio de la volatilidad, con pequeñas variaciones para adaptarlo a código en Python. Además, la volatilidad se ha calculado en base a los movimientos intradiarios, ya que se ha decidido que el programa debe abrir y cerrar las operaciones en el mismo día, sin dejar operaciones abiertas por la noche. En un futuro podría ser una idea para cambiarlo y probar en temporalidades mayores realizando los cálculos con otras temporalidades y ver resultados, ya que en el estudio de volatilidad se ha comprobado que las divisas a corto plazo se mueven poco en relación con los movimientos en temporalidades mayores.

Conocidos los niveles de entrada y sabiendo que los niveles de salida se calculan en base a la volatilidad de la divisa, para concretar los precios exactos de salida se necesita primero conocer la gestión del riesgo que vamos a emplear.

4.4.2. Gestión del riesgo

La gestión del riesgo es un factor muy importante ya que permite mantener un control del capital que gestionamos y no perder grandes cantidades de dinero haciendo operaciones sin sentido o arriesgando más dinero del que podemos permitirnos. Este es el factor que nos permite ganar más dinero del que se pierde en el largo plazo. Sin tener en cuenta este factor, es muy difícil poder sacar algo de rentabilidad en los mercados financieros.

Tabla 4.23 - Control de pérdidas y ganancias.

Pérdida total en la cuenta de trading	Ganancia necesaria para recuperar
10%	11%
15%	17%
25%	33%
30%	42%
50%	100%
75%	300%
90%	900%

Para entender mejor la importancia del riesgo y la importancia de controlar las pérdidas, en la tabla 4.23 se muestra el porcentaje necesario de ganancias para recuperar lo pérdido y, como se puede observar, pérdidas que pueden estar dentro de lo ‘normal’, como un 10%, se necesita una ganancia del 11% para recuperar ese dinero perdido. En cambio, cuando la pérdida llega al 25% ya se necesita una ganancia del 33% de la cuenta de trading para recuperar el dinero perdido, siendo este porcentaje muy elevado y difícil de conseguir. En este punto ya se estaría operando para recuperar el dinero y no para obtener beneficios

reales. A partir de una pérdida en la cuenta de más del 50%, ya resulta extremadamente complicado de conseguir, ya que serían unas rentabilidades de más del 100% siendo algo irreal y muy poco sustentable en el largo plazo.

Por esta razón, nunca se debe sobre operar o arriesgar más dinero de lo que cada inversor se pueda permitir, ya que si se arriesgan grandes cantidades de dinero y se pierde la operación, recuperar ese dinero de forma sana y sostenible es muy difícil. El objetivo no es solamente ganar dinero, sino también mantenerlo y protegerlo durante el proceso. Para ello, se intenta perder lo menos posible arriesgando un porcentaje pequeño del total del capital. Teniendo en cuenta todo lo anterior y la importancia del riesgo, el programa de trading va a arriesgar por operación un 0,5% del total del dinero que haya en la cuenta de trading. Para este caso se utiliza una cuenta de dinero virtual con 10 mil euros, por lo que por operación el programa arriesga 50 euros.

De esta forma, arriesgando únicamente el 0,5% por operación necesitaríamos perder 20 operaciones seguidas para perder un 10% de nuestro dinero, siendo improbable perder tantas operaciones seguidas sin ninguna ganancia de por medio. De esta forma, tenemos las probabilidades a nuestro favor ya que no nos exponemos tanto al mercado y no tenemos la posibilidad de perder mucho dinero de forma rápida. Por ejemplo, para perder el 20% de nuestro dinero se tendrán que perder 40 operaciones seguidas.

En este punto ya hemos determinado el riesgo que vamos a asumir por operación, ahora se utiliza el ratio riesgo-beneficio para tener más ventaja a la hora de tomar las operaciones y operar en el largo plazo. Este ratio riesgo-beneficio consiste en determinar cuánto más se puede ganar en relación a lo que se puede perder. Por ejemplo, con un ratio de 1:1 se estaría ganando lo mismo que se podría perder, es decir, si el riesgo potencial es de 50 euros, el beneficio potencial también sería de 50 euros. En otro caso en el que el ratio sea de 1:2, si el riesgo potencial son 50 euros, el beneficio potencial ya son 100 euros. Por lo tanto este caso es más conveniente que el anterior porque es posible ganar el doble de lo arriesgando.

En la tabla 4.24 se muestra la información sobre este ratio riesgo-beneficio, en el que se puede ver el porcentaje de operaciones ganadoras que hay que efectuar para no perder dinero en relación al ratio. Como se puede ver, a mayor ratio riesgo-beneficio menor es el número de operaciones ganadoras necesarias para ganar dinero. Cuanto menor es el ratio, más operaciones ganadoras se necesitan para mantenerse en positivo, como se puede ver, por ejemplo, en el ratio de 1:0.3, donde se necesita un porcentaje de operaciones ganadoras del 75%. Para el programa de trading, se va a utilizar un riesgo beneficio de 1:2, por lo que por cada 50 euros arriesgados se podrán ganar 100 euros. Además, necesitamos únicamente un 33% de operaciones ganadoras para no tener pérdidas, por lo que, junto a todo lo demás, se observa como el objetivo es que, en el largo plazo, las probabilidades estén a nuestro favor.

Tabla 4.24: Relación Riesgo-Beneficio

Porcentaje de ganancias	Ratio Riesgo-Beneficio
75%	1:0.3
60%	1:0.7
50%	1:1
40%	1:1.5
33%	1:2
25%	1:3

Teniendo en cuenta todos estos datos y todos los anteriores del análisis de volatilidad y el riesgo que estamos dispuestos a asumir, recopilando todo lo anterior se pueden resumir los puntos en la tabla 4.25.

Tabla 4.25: Resumen operativa

Riesgo por operación	0.5%
Ratio Riesgo-Beneficio	1:2
% necesario de operaciones ganadoras	33%
Señal de compra o venta	K y D (sobrecompra/sobreventa)
Punto de entrada	Soportes/Resistencias dinámicos
Movimiento Intradiario EUR/USD	0%-0.3%
Stop Loss	Cálculo de volatilidad reciente (0.05%-0.08%)
Take Profit	El doble que el Stop Loss (0.1%-0.16%)

Teniendo en cuenta todo lo anterior, cabe mencionar que como se ha mostrado, el programa no hace muchas operaciones por día debido a que como se ha calculado en la volatilidad de este par de divisas, no cuenta con demasiado movimiento en el corto plazo, e intentar operar en temporalidades muy pequeñas con muchas operaciones requeriría un apalancamiento demasiado grande y por ende las comisiones de apertura de operaciones serían demasiado grandes, disminuyendo los beneficios drásticamente.

4.4.3. Ejemplo Real del programa

A continuación se muestra la sentencia que sigue el programa de trading paso a paso para entender el funcionamiento del mismo y los procedimientos que sigue para efectuar las

operaciones. En este ejemplo se va a mostrar con los datos del día 12 de mayo de 2023, pero el funcionamiento es el mismo independientemente del día.

En primer lugar, obtener en tiempo real el datasheet datos proporcionado por Metatrader 5. A partir de estos datos se realizan los cálculos pertinentes para obtener los niveles de sobrecompra y sobreventa, además de calcular los niveles de soporte y resistencia dinámicos.

levels	list	12	[(4, 1.09187), (20, 1.09146), (48, 1.09225), (105, 1.09306), (118, 1.0 ...
levels_aux	list	12	[1.09187, 1.09146, 1.09225, 1.09306, 1.09344, 1.09046, 1.0899, 1.08814 ...

Figura 4.16: Niveles de soporte y resistencia.

El siguiente paso es generar la señal de compra o venta (o ninguna). En este día, se puede observar en la figura 4.17 como ha generado una señal de venta. Se observa como en la fila número 146 aparece el valor '-1' (valor para las señales de venta). Si observamos el datasheet (figura 4.18), en la fila número 146 vemos como esta señal de venta ha ocurrido a la 13:15h, con un precio de cierre de 1,09118.

Index	Type	Size	Value
140	int	1	0
141	int	1	0
142	int	1	0
143	int	1	0
144	int	1	0
145	int	1	0
146	int	1	-1
147	int	1	0

Figura 4.17: Array de señales.

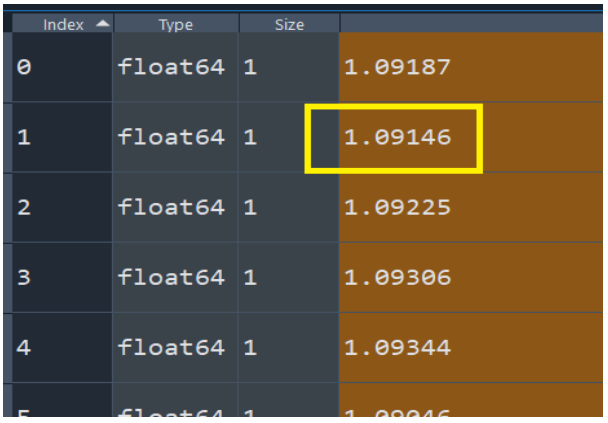
Index	time	open	high	low	close	tick volume	spread	real volume	n high
140	2023-05-12 12:45:00	1.0904	1.09086	1.0904	1.09062	214	0	0	1.09173
141	2023-05-12 12:50:00	1.0906	1.0908	1.09049	1.09075	152	0	0	1.09173
142	2023-05-12 12:55:00	1.09076	1.0911	1.09074	1.09108	154	0	0	1.09173
143	2023-05-12 13:00:00	1.0911	1.09124	1.09093	1.09102	133	0	0	1.09173
144	2023-05-12 13:05:00	1.091	1.09112	1.09087	1.09106	97	0	0	1.09173
145	2023-05-12 13:10:00	1.09101	1.09132	1.09101	1.09111	150	0	0	1.09169
146	2023-05-12 13:15:00	1.09111	1.09121	1.09096	1.09118	202	0	0	1.09132
147	2023-05-12 13:20:00	1.09118	1.09148	1.09118	1.09142	145	0	0	1.09148

Figura 4.18: Datasheet con los datos.

Tras ver la señal generada y la hora, para comprobar lo que ocurrió después de la señal, vemos la imagen que ha generado el programa para ver visualmente los niveles (figura 4.19) . La flecha amarilla indica donde se generó la señal de venta (13:15h).

Figura 4.19: Imagen generada por el programa.

En este punto se observa como el precio se encuentra en un nivel de sobrecompra, por lo que lo siguiente que va a realizar el programa es buscar la entrada. Para ello busca el siguiente nivel de resistencia. Como habíamos visto en el datasheet de forma visual, el precio con el que cerró la vela en el momento de la compra fue de 1,09118, por lo que el siguiente nivel de resistencia corresponde al nivel 1,09146 (figura 4.20).



Index	Type	Size	
0	float64	1	1.09187
1	float64	1	1.09146
2	float64	1	1.09225
3	float64	1	1.09306
4	float64	1	1.09344

Figura 4.20: Siguiente nivel de resistencia.

La entrada para venta se efectuará en este nivel. Para comprobar si el precio llegó, vemos en el datasheet (figura 4.21) como la siguiente vela en su máximo precio llega al 1,09148, por lo que nada más generada la señal, en los siguientes 5 minutos entra en venta y como se ha visto en el gráfico anterior, el precio posteriormente disminuyó siendo esta una operación ganadora en la que se obtendrían beneficios.

146	2023-05-12 13:15:00	1.09111	1.09121	1.09096	1.09118	202
147	2023-05-12 13:20:00	1.09118	1.09148	1.09118	1.09142	145
148	2023-05-12 13:25:00	1.09145	1.09145	1.09116	1.09116	84

Figura 4.21: Datasheet con precios después de señal.

En esta simulación con datos reales se ha mostrado visualmente una operación. Ahora falta mostrar cómo se envía la señal de venta para que la operación se efectúe realmente en la plataforma comercial con nuestra cuenta de trading a partir de código en Python.

4.4.4. Enviar las señales a MetaTrader 5

Para finalizar la parte de implementación, se muestra como enviar las operaciones generadas, de compra o venta, a la plataforma Metatrader 5 que está vinculada con nuestra cuenta de Trading para poder efectuar las operaciones de forma automática. Esto se realiza utilizando la integración de Mql5 en Python, donde encontramos código en la documentación de Python con Metatrader 5 que permite enviar las operaciones a sus servidores y poder comprar o vender de forma automática a partir de este código. Se ha creado una función que he denominado "operacion", que permite realizar las operaciones de forma automática. Esta función recibe 4 parámetros: "tipoOperacion", "symbol", "stoploss" y "level" (figura 4.22), a partir de los cuales se enviará una señal u otra (compra o venta) al servidor.

```
def operacion(tipoOperacion, symbol, stoploss, level):
```

Figura 4.22: Parámetros función "operacion".

La función **operación** se encarga de envíar a la plataforma de comercio (Metatrader 5) las órdenes para ejecutar las operaciones de compra y venta. Dependiendo de los parámetros que se le pasen a la función, enviará al servidor las operaciones con precios de entrada o salida diferentes. Esta función recibe los siguientes parámetros:

- **tipoOperación**.- Es una cadena de texto que indica si la operación es compra o es venta, en el caso de compra se pasa por parámetro "compra", en el caso de venta se pasa por parámetro "venta".

- **symbol**.- Es el par de divisa que se opera, en este caso se pasa "EURUSD".
- **stoploss**.- Es la volatilidad calculada que permite colocar el precio de stop en caso de pérdida, y con ello también el precio de salida en caso de beneficio.
- **level**.- Es el precio de entrada, que sería el nivel de soporte o resistencia dependiendo si es compra o venta. Para la explicación pondré manualmente estos niveles, pero en realidad serán los calculados durante la ejecución del programa.

Para la ejecución de una orden de compra, se crea la estructura 'request'. En las figuras 4.23 y 4.24 se muestra como completar esta estructura para una operación de compra y una operación de venta respectivamente, donde se indican el tipo de orden, el nivel de entrada, niveles de salida, etc.

```
#REQUEST PARA ENVIAR ORDEN DE COMPRA
if tipoOperacion=="compra":
    price = level
    request = {
        "action": mt5.TRADE_ACTION_PENDING,
        "symbol": symbol,
        "volume": lot,
        "type": mt5.ORDER_TYPE_BUY_LIMIT,
        "price": price,
        "sl": price - (stoploss*5),
        "tp": price + (stoploss*10),
        "magic": 234000,
        "comment": "python script open",
        "type_time": mt5.ORDER_TIME_GTC,
        "type_filling": mt5.ORDER_FILLING_IOC,
    }
```

Figura 4.23: Estructura para compras.

```
#REQUEST PARA ENVIAR ORDEN DE VENTA
elif tipoOperacion=="venta":
    price = level
    request = {
        "action": mt5.TRADE_ACTION_PENDING,
        "symbol": symbol,
        "volume": lot,
        "type": mt5.ORDER_TYPE_SELL_LIMIT,
        "price": price,
        "sl": price + (stoploss*5),
        "tp": price - (stoploss*10),
        "magic": 234000,
        "comment": "python script open",
        "type_time": mt5.ORDER_TIME_GTC,
        "type_filling": mt5.ORDER_FILLING_IOC,
    }
```

Figura 4.24: Estructura para ventas.

Como se observan en las dos estructuras, la diferencia más característica es en el parámetro "type", que toma el valor "TYPE_BUY_LIMIT" cuando es una compra, o "TYPE_SELL_LIMIT" cuando es una venta.

Cabe destacar el parámetro "action", que toma el valor "TRADE_ACTION_PENDING". Este es el comando para el envío de una señal pendiente, es decir, no se activa en el mercado en tiempo real, sino que se manda al servidor para que cuando el precio llegue a cierto nivel (en este caso el nivel de soporte o resistencia) se compre o venda dependiendo de la señal.

Hecho todo lo anterior, se ejecuta la instrucción que manda la órden finalmente, con otro código para comprobar que todo ha salido correctamente o que muestre el error en caso de fallo.

```
# enviamos la solicitud comercial
result = mt5.order_send(request)
# comprobamos el resultado de la ejecución
print("1. order_send():SELL by {} {} lots at {}".format(symbol,lot,price));
#Si la solicitud da error salir
if result.retcode != mt5.TRADE_RETCODE_DONE:
    print("2. order_send failed, retcode={}".format(result.retcode))
    print("shutdown() and quit")
    mt5.shutdown()
    sys.exit(0)
else:
    print("2. order_send done, ", result)
    print("   opened position with POSITION_TICKET={}".format(result.order))
return result
```

Figura 4.25: Envío de la orden.

Si todo sale correctamente y la orden se envía a la plataforma de trading, en la terminal de Spyder (el entorno de desarrollo) aparece el mensaje con la información de la señal, mostrando que ha sido correcto.

```
1. order_send():SELL by EURUSD 0.01 lots at 1.081
2. order_send done,  OrderSendResult(retcode=10009, deal=0, order=402112432,
volume=0.01, price=0.0, bid=0.0, ask=0.0, comment='Request executed',
request_id=1193211224, retcode_external=0, request=TradeRequest(action=5,
magic=234000, order=0, symbol='EURUSD', volume=0.01, price=1.081, stoplimit=0.0,
sl=1.0803071428571427, tp=1.0823857142857145, deviation=0, type=2,
type_filling=1, type_time=0, expiration=0, comment='python script open',
position=0, position_by=0))
   opened position with POSITION_TICKET=402112432

In [38]:
```

Figura 4.26: Información en la terminal.

Ahora, en Metatrader 5, se puede ver en la parte inferior como la órden está colocada como una 'buy limit' y en cuanto el precio llegue al valor de entrada se realizará la compra.

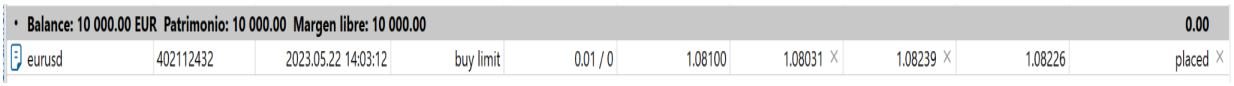


Figura 4.27: Órden de compra (vista MT5).

En el gráfico también aparece la representación del 'buy limit' donde la línea horizontal verde es el precio de entradas y la línea roja superior e inferior son el take profit y el stop loss respectivamente (figura 4.28).

Figura 4.28: Representación gráfica de órden de compra.

Por último, la llamada a la función **operacion** se realiza con varias condiciones dependiendo si es compra o venta como se muestra en la figura 4.29.

```
#---------------- ÓRDENES DE ENTRADA ---------------------

if signal[-2] == 1:
    print("señal compra")
    operacion("compra", symbol, float(atr[-1:]), 1.081)

elif signal[-2] == -1:
    print("señal venta")
    operacion("venta", symbol, float(atr[-1:]), 1.0825)

else:
    print("no hay señal")
```

Figura 4.29: Condiciones de llamada a la función

4.5.- Pruebas/Implantación

Tras desarrollar el programa de trading, se realizan las pruebas para comprobar su funcionamiento con datos históricos, este tipo de pruebas se conoce como '*backtesting*'. Para ello, se ha realizado una función llamada con ese nombre ('backtesting') para simular el comportamiento del programa con datos históricos. El programa da como resultado las operaciones que habría tomado ese día con el beneficio o pérdida correspondiente. Para estas pruebas, se ha supuesto una cuenta de trading con 10.000 euros, por lo que arriesgando un 0,5% por operación significa que una operación ganadora tendría un beneficio de 100 euros y una operación perdedora una pérdida de 50 euros.

La función **backtesting** recibe cuatro parámetros:

- **signals**.- Es el array donde se encuentran las señales de compra o venta generadas anteriormente. Si es 1 es compra y si es -1 es venta. Permite a partir de este array poder ver cuando se realiza una compra o venta (o nada).

- **levels.-** Es el array con los niveles de soportes y resistencia. Son necesarios para determinar los precios de entrada.

- **data.-** El datasheet con todos los datos. Sirve para ver los precios del activo en cada momento, así como los precios máximos y mínimos.

- **atr.-** Es el cálculo de volatilidad que permite saber cuando salir de la operación gestionando el riesgo.

Como se observa en el código de esta función, el programa revisa todos los datos de un día concreto y comprueba tanto las señales de compra (figura 4.30), como las señales de venta (figura 4.31). Si se da una señal de compra o venta en el array de señales, lo primero que hace es buscar el siguiente nivel de compra o el siguiente nivel de venta, siendo este donde se colocaría la órden pendiente.

Una vez se tiene la órden de entrada, entra en otro bucle en el que revisa a partir de la posición de entrada hasta el resto del datasheet si el precio llegó y entró a la operación. Si el precio llega al punto de entrada, entra de nuevo en otro bucle para comprobar si el resultado de esa operación fue positivo o negativo. Este resultado se guarda en la variable "realized_pnl", donde la ejecución del programa devuelve esta variable con el resultado de las operaciones diarias.

```python
#Función para pruebas de rendimiento
def backtest(signals, levels, data, atr):
    capital = 10000  # Capital inicial
    realized_pnl = 0  # Ganancia o pérdida realizada

    for i in range(len(signals)):

        #Desde las 08:30h hasta las 18:00h únicamente
        if i >= 88 and i <= 204:
            signal = signals[i]

            #Señal de compra
            if signal == 1:
                #ordeno los niveles de soporte y resistencia
                levels.sort(reverse=True)
                dataAux = data.iloc[i, 4]
                #Busco siguiente nivel de soporte
                for x in range(len(levels)):
                    if dataAux > levels[x]:
                        entrada = levels[x]
                        break
                #Busco si entra en la operación
                for index, valor in data["low"][i:].iteritems():
                    #Si entra a la operación veo si es rentable...
                    if entrada >= valor:
                        stop = data.iloc[index, 1] - (atr[index] * 5)
                        tp = data.iloc[index, 1] + (atr[index] * 10)
                        for index2, valor2 in data["high"][i:].iteritems():
                            if valor2 >= tp:
                                realized_pnl += 100
                                print("+1 compr tp")
                                break
                            elif valor2 <= stop:
                                realized_pnl -= 50
                                print("+1 compr sl")
                                break
                        break
```

Figura 4.30: Programa Backtesting señal de compra.

```python
            #Señal de venta
            elif signal == -1:
                #ordeno los niveles de soporte y resistencia
                levels.sort()
                dataAux = data.iloc[i, 4]
                #Busco siguiente nivel de resistencia
                for x in range(len(levels)):
                    if dataAux < levels[x]:
                        entrada = levels[x]
                        break
                #Busco si entra en la operación
                for index, valor in data["high"][i:].iteritems():
                    if entrada <= valor:
                        stop = data.iloc[index, 1] + (atr[index] * 5)
                        tp = data.iloc[index, 1] - (atr[index] * 10)
                        for index2, valor2 in data["low"][i:].iteritems():
                            if valor2 <= tp:
                                realized_pnl += 100
                                print("+1 vent tp")
                                break
                            elif valor2 >= stop:
                                realized_pnl -= 50
                                print("+1 vent sl")
                                break
                        break

    return capital, realized_pnl
```

Figura 4.31: Programa Backtesting señal de venta.

Un ejemplo de ejecución y los resultados que muestra son los siguientes, en este caso se realiza la prueba con los datos del día 12 de mayo de este año (2023), donde se puede observar como ha realizado cuatro operaciones las cuales son dos operaciones de compra y dos operaciones de venta. En este día el resultado fue positivo de +100 euros (figura 4.32).

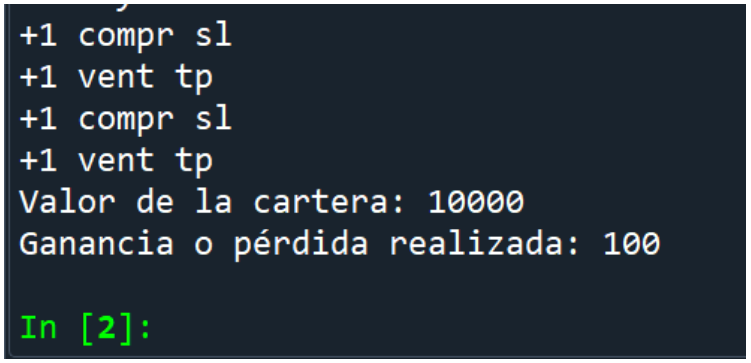


Figura 4.32: Ejemplo resultado de la prueba diaria.

Para interpretar los datos que se muestran por pantalla al ejecutar el programa, se explican a continuación los posibles resultados con su significado:

- "+1 compr sl": Se ha realizado una compra pero ha tocado el stop loss.
- "+1 compr tp": Se ha realizado una compra y ha tocado el take profit.
- "+1 vent sl": Se ha realizado una venta pero ha tocado el stop loss.
- "+1 vent tp": Se ha realizado una compra y ha tocado el take profit.

En el caso de los resultados del día 12 de mayo, vemos como ha tenido dos operaciones de compra perdedoras y otras dos de venta ganadoras, por lo que por el riesgo-beneficio, el total del día es positivo.

La tabla 4.26 muestra las pruebas realizadas por días con los resultados obtenidos en el mes de mayo. Como se puede observar en los datos recopilados de las pruebas realizadas, el programa obtuvo un beneficio final de 300 euros, pero cabe mencionar que este resultado en realidad sería algo menor debido a que en en esta simulación no se han tenido en cuenta las comisiones de apertura de las operaciones, debido a que estas comisiones son variables dependiendo de la situación actual del mercado.

Este resultado muestra un beneficio de un 3% en un mes, siendo una cifra realista y alcanzable en el largo plazo, ya que buscar rentabilidades de un 10% todos los meses es algo irreal y para nada sostenible en el largo plazo. También mencionar que no es una cifra de resultados asegurados todos los meses, ya que en meses siguientes este número podría variar dependiendo de las circunstancias del mercado.

Tabla 4.26: Pruebas realizadas el mes de mayo.

Día	Operación Compra Ganadora	Operación Compra Perdedora	Operación Venta Ganadora	Operación Venta Perdedora	Beneficio / Pérdida
Lunes - 1		2		2	-200
Martes - 2	1	1		1	0
Miércoles - 3	1			2	0
Jueves - 4		2	2		100
Viernes - 5		3	1		-50
Lunes - 8	1	4	2		100
Martes - 9		2	1		0
Miércoles - 10	1	1	1	2	50
Jueves - 11		1			-50
Viernes - 12		2	2		100
Lunes - 15	1				100
Martes - 16		2	1	1	-50
Miércoles - 17		1	1		50
Jueves - 18		2	1		0
Viernes - 19	1			3	-50
Lunes - 22					0
Martes - 23		1	1		50
Miércoles - 24		2			-100
Jueves - 25		1	2		150
Viernes - 26		2	2		100
TOTAL					**+300€**

Capítulo 5
Conclusiones y trabajo futuro

5.1.- CONCLUSIONES

Una vez desarrollado todo el proyecto, se puede llegar a la conclusión de que uno de los factores más importantes para operar y desarrollar un bot de trading algorítmico es tener muy en cuenta todos los factores relacionados con el riesgo, para procurar tener el mayor número posible de probabilidades a nuestro favor para tratar de obtener resultados positivos sostenibles en largo plazo.

También destacar la importancia de ver resultados en el largo plazo, ya que durante la realización de las pruebas de rendimiento del bot de trading, se observa como el primer día perdió 200 euros, y esto no ha supuesto un problema ya que al final del mes consiguió ser rentable. Esto es debido a lo comentado anteriormente sobre el factor del riesgo y las probabilidades en el largo plazo.

Al finalizar el desarrollo del programa y las pruebas, se ha llegado a la conclusión de que el proceso de probar el programa con la función específica para ello (backtesting) es una de las partes más importantes y complicadas. Esto es debido a que para poder probar el programa hay que intentar simular mediante esta función como se iría comportando el programa en esos días de forma real. Por lo que hay que tener en cuenta la gran mayoría del desarrollo anterior, tanto de código, como análisis estadísticos.

En cuanto a los objetivos que se establecieron al comienzo de este proyecto, se ha conseguido completar la mayoría de estos. El programa desarrollado es capaz de operar el par de divisas EURUSD de forma automatizada, asumiendo una gestión de riesgo correcta. Aunque cabe destacar que el análisis del riesgo fue desarrollado a mano previamente para poder incorporarlo en el programa, no es un algoritmo específico que calcule estos datos. Este análisis manual fue necesario para poder incorporar posteriormente con los resultados obtenidos la forma correcta de gestión del riesgo y poder automatizarlo.

También se han cumplido los objetivos de obtener los datos de la plataforma Metatrader 5, su procesamiento y realizar las llamadas correctamente al intermediario financiero, uniendo la cuenta de trading de un broker junto con la plataforma comercial. Además, se ha conseguido que los resultados del mes de mayo de 2023 fuesen positivos, pero como se mencionó en su momento, esto no indica que siempre lo vaya a ser, por lo que hay que continuar haciendo pruebas e ir viendo cómo se desarrolla con el paso del tiempo.

5.2.- POSIBLES DESARROLLOS FUTUROS

En desarrollos futuros se podría intentar que el programa opere más de un activo y no esté limitado únicamente al EURUSD. Por esta razón, en el código ya se han realizado la funciones y todos los datos pasando por parámetro el símbolo del activo a operar. En este caso, como solo había uno, solo recibían este, pero en un futuro se podría ir añadiendo más y ver el comportamiento del programa en otros activos financieros y ver sus resultados.

También ir revisando mediante las pruebas donde se podría mejorar el programa e ir iterando en mejores versiones que consigan una rentabilidad mayor con el paso del tiempo.

Bibliografía

[1] Análisis Técnico de los Mercados Financieros
Jhon J. Murphy
Ediciones Gestión 2000 / 10/2022

[2] Leones contra Gacelas - Manual completo del Especulador
José Luis Cárpatos
4º Edición / 10/2022

[3] La metodología Wyckoff en profundidad
Rubén Villahermosa Chaves
2º Edición / 10/2022

[4] The brief history of trading terminals
https://www.etnasoft.com/trading-terminals-history/
10/2022

[5] Forex Market Size: A Traders Advantage
https://finance.yahoo.com/news/forex-market-size-traders-advantage
Enero 2014 / 10/2022

[6] Forex Market Size: A Traders Advantage
https://www.dailyfx.com/education/beginner/forex-market-size.html
Enero 2019 / 10/2022

[7] Forex Market: Who Trades Currencies and Why
https://www.investopedia.com/articles/forex/11/who-trades-forex-and-why
Agosto 2018 / 10/2022

[8] ¿Cómo se creó el Forex?
https://www.atfx.com/es/analisis/estrategias-de-trading/como-se-creo-el-forex
Marzo 2022 / 10/2022

[9] What is a Lot in Forex?
https://www.babypips.com/learn/forex/lots-leverage-and-profit-and-loss
10/2022

[10] Qué comisiones tiene un Bróker y cómo afectan a las ganancias
https://admiralmarkets.com/es/education/articles/forex-basics/comisiones-broker
Noviembre 2022 / 10/2022

[11] Basics of Algorithmic Trading: Concepts and Examples
https://www.investopedia.com/articles/active-trading
Junio 2022 / 10/2022

[12] What Percentage of Trading Is Algorithmic?
https://www.quantifiedstrategies.com/what-percentage-of-trading-is-algorithmic/
Septiembre 2022 / 02/2023

[13] Algorithmic Trading Market - Growth, Trends…
https://www.mordorintelligence.com/industry-reports/algorithmic-trading-market
Estudio 2018-2028 / 02/2023

[14] Trading Statistics (2023)
https://analyzingalpha.com/algorithmic-trading-statistics
Junio 2022 / 02/2023

[15] Medias Móviles
https://www.bolsaytrading.com/2016/07/medias-moviles.html
Julio 2016 / 02/2023

[16] El indicador Estocástico en profundidad
https://admiralmarkets.com/es/education/articles/forex-indicators
Junio 2022 / 02/2023

[17] Basics of Algorithmic Trading
https://www.investopedia.com/articles
Diciembre 2022 / 02/2023

[18] Indicador RSI
https://www.avatrade.es/educacion/professional-trading-strategies/rsi
02/2023

[19] Detection of price support and resistance levels in Python
https://towardsdatascience.com/detection-of-price-support-and-resistance-levels
Julio 2020 / 02/2023

[20] Financial Market Trend Lines
https://medium.com/@dannygrovesn7/creating-stock-market-trend-lines
Julio 2022 / 02/2023

[21] Fibonacci retracements in Python
https://towardsdatascience.com/fibonacci-retracements-in-python-470eb33b6362
Marzo 2021 / 02/2023

[22] ¿Qué es Python?
https://aws.amazon.com/es/what-is/python/
02/2023

[23] API: qué es y para qué sirve
https://www.xataka.com/basics/api-que-sirve
Agosto 2019 / 02/2023

[24] Metatrader 5: ¿Qué es y cómo funciona?
https://www.revistagestion.ec/cifras/metatrader-5-que-es-y-como-funciona
Agosto 2022 / 02/2023

[25] Documentación MT5 de Python
https://www.mql5.com/en/docs/integration/python_metatrader5
02/2023

[26] Lenguaje de programación de estrategias comerciales MetaQuotes Language 5
https://www.ampfutures.com/es/metatrader/automated-trading/mql5
02/2023

[27] ¿Es IC Markets de fiar?
https://www.okbrokers.es/ic-markets/
Enero 2023 / 02/2023

[28] Resumen de cuenta
https://www.tecnicasdetrading.com/2017/06/usar-correctamente-relacion-riesgo-beneficio.html
https://www.icmarkets.com/global/es/tipo-de-cuenta/resumen
02/2023

[29] Ciclo de vida iterativo
https://blog.comparasoftware.com/ciclo-de-vida-iterativo-que-es

[30] Gestión del riesgo
https://www.cmcmarkets.com/es-es/trading-guias/riesgos-gestion-monetaria

[31] Ratio Riesgo-Beneficio
https://www.tecnicasdetrading.com/2017/relacion-riesgo-beneficio.html

[32] MPL Finance
https://github.com/matplotlib/mplfinance

[33] Spyder
https://www.spyder-ide.org/
02/2023

[34] Anaconda
https://www.anaconda.com/
02/2023

[35] TradingView
https://es.tradingview.com/
02/2023