



FACULTAD DE CIENCIAS ECONÓMICAS Y
EMPRESARIALES

**EL *TRADING* ALGORÍTMICO EN LOS
MERCADOS FINANCIEROS**
**Estrategia Basada en la Volatilidad de
los Precios de las Opciones**

Autor: Isabel Martín Hinojosa

4^o, E-4 Boston

Director: María Coronado Vaca

Madrid

Mayo 2018



Isabel
Martín
Hinojosa

EL TRADING ALGORÍTMICO EN LOS MERCADOS FINANCIEROS – Estrategia Basada en la Volatilidad de los Precios de las Opciones

Título: El *trading* Algorítmico en los Mercados Financieros – Estrategia basada en la volatilidad de los precios de las opciones

Resumen:

El *trading* algorítmico está creciendo considerablemente con la evolución de las tecnologías y la facilidad de acceso a internet. Con ayuda de libros, lenguajes de programación y sus tutoriales y brókeres online podemos, como inversores individuales, desarrollar nuestras propias estrategias de *trading*.

El objetivo principal de este trabajo es diseñar, implementar y validar una estrategia propia de *trading* algorítmico basada en la volatilidad de los precios de las opciones mediante el uso del programa Python. Por otro lado, se va a analizar el marco teórico de las estrategias de *trading* algorítmico y comparar distintos métodos de validación de estrategias, entre los que se incluyen el *Backtesting*, *Walk Forward* y *Paper Trading*. Atendiendo a los resultados obtenidos a través de una cuenta simulada de *Paper Trading*, podremos afirmar que se trata de una estrategia con una base muy buena y con gran potencial, pero no perfecta pues se trata de un algoritmo sencillo sin ningún tipo de filtros que muestra operaciones tanto positivas como negativas. Con todo ello, concluiremos que sería muy interesante profundizar más en el modelo de *trading* algorítmico para conseguir un sistema beneficioso en su totalidad.

Palabras Clave: *trading*, *trading* algorítmico, opción de compra, opción de venta, mercado, volatilidad, programa, sistema, estrategia

Abstract:

Algorithmic trading is growing considerably with the evolution of technologies and the ease of access to the internet. With the help of books, programming languages and their online tutorials and online brokers we can, as individual investors, develop our own trading strategies.

The main objective of this document is to design, implement and validate a self-developed algorithmic trading strategy based on the volatility of option prices using the Python program. On the other hand, we will analyze the theoretical framework of algorithmic trading strategies and compare different strategies validation methods, including *Backtesting*, *Walk Forward* and *Paper Trading*. Looking at the results obtained through a simulated *Paper Trading* account, we can say that it is a strategy with a very good base and great potential, but still not perfect as it is a simple algorithm

without any type of filters that shows both positive and negative operations. With all this in mind, we will conclude that it would be very interesting to go deeper into the algorithmic trading model to achieve a beneficial system in its entirety.

Key words: trading, algorithmic trading, call option, put option, market, volatility, program, system, strategy

Tabla de contenido

1. OBJETIVOS	7
1.1 OBJETIVOS DEL TRABAJO	7
1.2 JUSTIFICACIÓN DEL TEMA OBJETO DE ESTUDIO	8
1.3 METODOLOGÍA	8
1.4 ESTRUCTURA	8
2. INTRODUCCION AL <i>TRADING</i> ALGORITMICO	10
2.1 EL <i>TRADING</i> ALGORÍTMICO	10
2.2 ENTORNO CAMBIANTE	10
2.3 CRECIMIENTO RECIENTE.....	11
2.4 VENTAJAS Y DESVENTAJAS DEL <i>TRADING</i> ALGORÍTMICO	12
3. CÓMO DESARROLLAR UNA ESTRATEGIA DE <i>TRADING</i>	15
3.1 ESTABLECER OBJETIVOS ADECUADOS.....	16
3.2 EN QUÉ MERCADOS OPERAR Y CON QUÉ ACTIVO	18
3.3 QUÉ TIPO DE SISTEMA ELEGIR	18
3.4 CUÁL ES EL MARCO TEMPORAL.....	19
3.5 CÓMO ENTRAR EN EL MERCADO.....	20
3.6 CÓMO SALIR DEL MERCADO.....	20
3.7 CÓMO PROGRAMAR LA ESTRATEGIA.....	21
4. EL MERCADO DE LAS OPCIONES	22
4.1 INTRODUCCIÓN	22
4.2 HISTORIA DE LAS OPCIONES	22
4.3 EL NACIMIENTO Y CRECIMIENTO DE LA INDUSTRIA DE OPCIONES ACTUAL	22
4.4 CONCEPTOS BÁSICOS DE LAS OPCIONES	24
4.5 PRINCIPALES DIFERENCIAS ENTRE ACCIONES Y OPCIONES	26
5. ESTRATEGIA BASADA EN LA VOLATILIDAD DE LOS PRECIOS DE LAS OPCIONES: <i>STRADDLE/ STRANGLE</i>	27
5.1 CONCEPTO	27
5.2 DIFERENCIA ENTRE <i>STRADDLE</i> Y <i>STRANGLE</i>	28
5.3 DIFERENCIA ENTRE <i>LONG STRADDLE</i> Y <i>SHORT STRADDLE</i>	29
5.4 CONDICIONES DE MERCADO ADECUADAS PARA IMPLEMENTAR LA ESTRATEGIA.....	29
5.5 FUNCIONAMIENTO DE LA ESTRATEGIA	30
5.6 EJEMPLO DEL FUNCIONAMIENTO DE LA ESTRATEGIA BASADA EN LA VOLATILIDAD DE LOS PRECIOS DE LAS OPCIONES CON UN <i>LONG STRADDLE</i>	30
5.7 REQUISITOS QUE DEBE CUMPLIR UNA BUENA ESTRATEGIA DE <i>STRADDLE</i> O <i>STRANGLE</i>	31
6. ELEMENTOS NECESARIOS PARA DESARROLLAR LA ESTRATEGIA DE <i>TRADING</i>	32
6.1 BRÓKER	33
6.2 RED DE COMUNICACIONES	34
6.3 INFRAESTRUCTURA INFORMÁTICA.....	34
7. ESTRUCTURA DE UN PROGRAMA DE <i>TRADING</i>	36
8. IMPLEMENTACION DE LA ESTRATEGIA DE <i>TRADING</i> CON <i>PYTHON</i>	38
8.1 INTRODUCCIÓN	38
8.2 MÓDULO PRINCIPAL	39
8.3 MÓDULO I	40
8.4 MÓDULO II	42

8.5 MÓDULO III	43
8.6 MÓDULO IV	46
8.7 MÓDULO V	46
8.8 MÓDULO VI	48
9. MÉTODOS DE VALIDACIÓN DE ESTRATEGIAS	49
9.1 MÉTODO DE <i>BACKTESTING</i>	50
9.2 MÉTODO DE <i>WALK FORWARD</i>	51
9.3 MÉTODO DE <i>PAPER TRADING</i>	51
10. CONCLUSIONES	54
11. BIBLIOGRAFÍA.....	56
12. ANEXOS.....	59

1. OBJETIVOS

1.1 Objetivos del trabajo

Los avances tecnológicos producidos en los últimos años han dado lugar a una serie de nuevas formas de entender y acercarse a los mercados financieros. El sector financiero se ha visto completamente transformado y ha permitido a los mercados ganar en transparencia y eficacia. Una persona que se dedica al *trading*, es una persona que busca rentabilizar sus activos en los mercados financieros principalmente a través de la especulación. Hasta hace unos años, estas personas también conocidas como *traders*, tomaban decisiones en base a su propio criterio, en momentos críticos en los que la presión del tiempo era un factor a tener muy en cuenta, sin la posibilidad de comprobar la viabilidad de sus movimientos. La prueba final era en el mercado a tiempo real. Si se había tomado la decisión equivocada, ya no había forma de corregirla.

Con la aparición de estas nuevas tecnologías, cuyo auge tuvo lugar en la década de los ochenta, las posibilidades han aumentado y podríamos decir que son infinitas. Un sistema de *trading* automático permite la programación de una estrategia a través de algoritmos matemáticos capaces de generar numerosas operaciones de compra y venta de forma automática en un periodo de tiempo muy corto. Al tratarse de un programa, puede ser probado de antemano para ver su viabilidad y probabilidad de éxito en el mercado real. Estos complicados modelos estadísticos han minimizado el error y el riesgo aunque cabe destacar que no son perfectos.

El objetivo principal del presente documento es diseñar, implementar y validar una estrategia propia de *trading* algorítmico basada en la volatilidad de los precios de las opciones mediante el uso del programa Python. Como objetivos secundarios, se plantean los siguientes: por un lado, analizar el marco teórico de las estrategias de *trading* algorítmico y por otro comparar distintos métodos de validación de estrategias, entre los que se incluyen el *Backtesting*, *Walk Forward* y *Paper Trading*. En último lugar, extraeremos conclusiones del estudio realizado.

1.2 Justificación del tema objeto de estudio

El *trading* algorítmico ha cambiado la forma en que se interactúa en los mercados financieros. Son muchas las operaciones de compra y venta de instrumentos financieros que se hacen a través de programas automáticos, lo que despierta un gran interés en conocer cómo funcionan estos programas. A través de este trabajo, se pretende dar a conocer una alternativa que se presenta al inversor cuando éste opera en los mercados. Este trabajo nos permitirá obtener más conocimiento y experiencia en el funcionamiento del *trading* algorítmico de acuerdo a las tendencias de negociación tanto actuales como futuras.

1.3 Metodología

Para conseguir los objetivos propuestos, la metodología que se emplea en el presente documento es una metodología deductiva. Una metodología deductiva es aquella que irá de lo más general a lo más específico. En primer lugar se va a exponer el marco teórico : qué es el *trading* algorítmico, ventajas y desventajas, cómo desarrollar una estrategia, el mercado de las opciones, estrategia basada en la volatilidad de los precios de las opciones y se termina por exponer un ejemplo de un programa de *trading* algorítmico que sigue esa estrategia y, por supuesto en el mercado de las opciones. Por último se establecerán conclusiones. Otra metodología utilizada es la observación. Para comprobar el funcionamiento del programa hay que observar cómo se comporta en una cuenta simulada. Con ello, podrán establecerse conclusiones.

En cuanto a los tipos de información empleada en el trabajo, hay dos tipos. Por un lado, se ha recurrido en gran parte a libros para la parte más general o teórica del trabajo. Para la parte del programa de *trading* algorítmico, la información es el propio programa. Al ponerlo en funcionamiento en una cuenta simulada, se obtienen datos sobre cómo funciona el programa en el mercado real y se establecen conclusiones a partir de ello.

1.4 Estructura

La estructura del trabajo se divide en tres bloques diferenciados. El primer bloque recoge la exposición de la teoría que, posteriormente se aplicará al segundo

bloque. Este último es de carácter íntegramente práctico. El último bloque recoge la bibliografía y los anexos. En el primer bloque teórico, se analiza, en primer lugar, el trading algorítmico: su nacimiento, entorno cambiante y ventajas y desventajas. También recoge los pasos a seguir para desarrollar una estrategia de *trading* (y menciona cómo será la estrategia en cuestión). A continuación se analiza en detalle cómo funciona el mercado en el que se va a negociar y la estrategia que se va a emplear junto con los elementos necesarios para desarrollar la estrategia. El segundo bloque, de carácter práctico, recoge en primer lugar una estructura que explica cómo el programa de trading se relaciona con el mercado. A continuación viene la implementación de la estrategia de trading utilizando el programa Python y la explicación de cada uno de los módulos que permiten el funcionamiento del programa y, por último, los métodos de validación de programas, analizando en detalle el método de validación empleado en esta estrategia e incluyendo ejemplos que comprueban su funcionamiento. En último lugar, se establecen conclusiones.

2. INTRODUCCION AL *TRADING* ALGORITMICO

2.1 El *Trading* Algorítmico

El *trade* o comercio, es un concepto económico básico que implica la compra y venta de bienes y servicios, con una compensación pagada por un comprador a un vendedor. Esto se entiende como el intercambio de bienes o servicios entre las partes (Bernstein, 2008). El medio de intercambio más común para estas transacciones es el dinero, pero también existen otros como el intercambio de bienes o servicios entre ambas partes, (trueque) o pago con moneda virtual, la más popular de las cuales es el *bitcoin*. La red que permite este intercambio se denomina mercado. De esta definición, se entiende como *trading* la realización de la actividad del *trade*. Gracias a las nuevas oportunidades que han ido brindando las nuevas tecnologías en los últimos años, el *trading* puede realizarse de forma automática.

En los mercados financieros, *trading* se refiere a la compra y venta de valores, como por ejemplo la compra de acciones en la Bolsa de Nueva York. Por tanto, el *trading* algorítmico es una forma de operar en los mercados financieros cuya función principal es ejecutar funciones de compra y/o venta de instrumentos financieros de forma automática con el uso de programas, procesos y algoritmos automatizados (Kissell, 2014).

2.2 Entorno Cambiante

Como ya se ha establecido, el *trading* algorítmico representa la ejecución automatizada de instrumentos financieros incluyendo: acciones, bonos y una gran variedad de derivados financieros. Estos algoritmos son también fundamentales en estrategias de inversión y objetivos de *trading*. Esta nueva era ofrece a los inversores ejecuciones más eficientes y menores costes de transacción, lo que a su vez favorecerá un mejor rendimiento de las carteras.

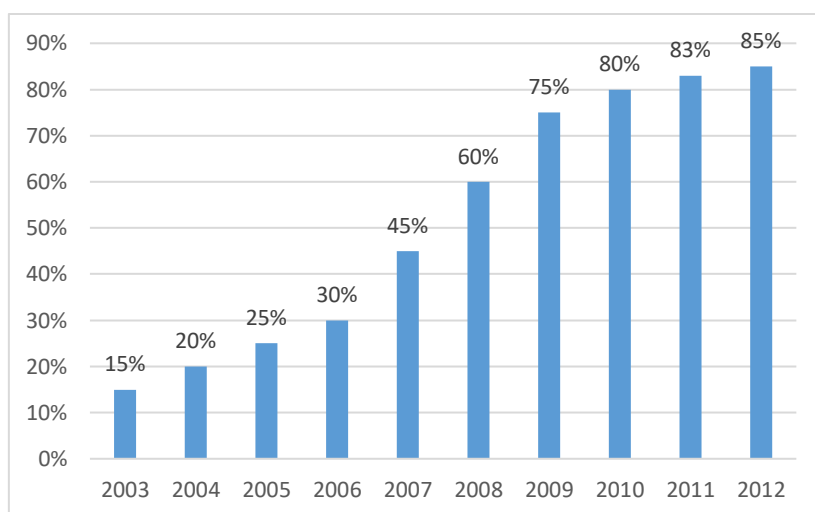
El mercado norteamericano de acciones ha experimentado grandes cambios en la microestructura de mercado, con un gran crecimiento del *trading* algorítmico. En el año 2001, el *New York Stock Exchange* (NYSE) y el NASDAQ cambiaron de un sistema en el que las acciones se cotizaban en fracciones (ej. 1/16 de dólar) a un sistema en el que se cotizan acciones en decimales (ej. céntimos por acción). Como

consecuencia, el incremento mínimo de cotización se redujo de 0.0265 dólares por acción a 0.01 dólares por acción. A pesar de que esto proporciona una gama mucho mayor de posibles precios de mercado, también ha sido criticado por interferir con el papel principal de los mercados financieros (Kissell, 2014). Más adelante, veremos cuáles son las ventajas y desventajas que ofrece el *trading* algorítmico.

2.3 Crecimiento reciente

Para posicionarse mejor a la hora de abordar el entorno cambiante del mercado, los inversores han recurrido al *trading* algorítmico. Dado que los ordenadores son más eficientes a la hora de procesar grandes cantidades de datos e información, más hábiles en la realización de cálculos complejos y más capaces de reaccionar a las cambiantes condiciones de mercado, son los más adecuados para el *trading* en tiempo real en el mercado actual (Kissell, 2014). El *trading* algorítmico se hizo popular a principios de los años 20. Ya en 2005, representaba el 25% del volumen total. En 2012 ya representaba un 85% de las transacciones totales y, a día de hoy, el *trading* algorítmico representa más del 85 por ciento de las operaciones bursátiles estadounidenses y el 30-40 por ciento de las operaciones bursátiles europeas y japonesas en conjunto (Buckey, 2016). Si miramos el gráfico 1, podemos observar que el crecimiento del *trading* algorítmico ha sido muy grande a partir del año 2000.

Gráfico 1. Trading algorítmico. Porcentaje de volumen de mercado entre el año 2003 y el 2012 en Estados Unidos.



Fuente: Kissell, 2014

2.4 Ventajas y desventajas del *Trading* Algorítmico

El *trading* algorítmico ofrece ventajas pero también desventajas. De hecho, en esta era de la economía digital caracterizada por la evolución del análisis técnico y el *trading*, existe un continuo debate al respecto. Por un lado encontramos a los *traders* tradicionales, aquellos que toman decisiones basándose en su inteligencia y emociones y por otro, encontramos el trading algorítmico. Se estima que actualmente el 70% de las operaciones a nivel global se realizan de forma automática (Cofnas, 2016). No está claro quién ganará esta batalla puesto que no se sabe si los sistemas automáticos podrán, algún día, sustituir al *trader* tradicional, pero desde luego actualmente el *trader* humano va un paso por detrás. A continuación, es importante que prestemos atención a las ventajas y desventajas de estos nuevos sistemas frente a los tradicionales (Cofnas, 2016). Para analizar las ventajas y desventajas del *trading* algorítmico en profundidad, seguimos a Kissell, 2014.

Ventajas

- **Comisiones más bajas.** las comisiones suelen ser más bajas que las tarifas de comisión tradicionales, ya que el *trading* algorítmico solo proporciona a los inversores servicios de ejecución y relacionados con la ejecución (como la gestión de riesgos o la gestión de órdenes). Las comisiones algorítmicas generalmente no compensan a los intermediarios por las actividades de investigación. Aunque algunos fondos pagan una tasa más alta por el acceso a la investigación, no suele ser el caso.
- **Anonimato.** las órdenes se ingresan en el sistema y son intercambiadas de forma automática por el ordenador allá donde se ejecuten. El *trader* de la parte compradora administra la orden desde dentro de su empresa o solicita que la orden sea administrada por los *traders* de la parte vendedora.
- **Control.** Los *traders* de la parte compradora tienen control total sobre las órdenes y son ellos los que determinan en qué mercado se ejecutan, cuáles son las reglas de envío de órdenes incluyendo: los límites y precios del mercado, las cantidades de acciones que se intercambian, los tiempos de espera y actualización, así como cuándo acelerar o decelerar las operaciones en función del objetivo de inversión del

fondo y las condiciones del mercado. Los *traders* pueden cancelar y modificar las órdenes casi instantáneamente.

- **Fuga mínima de información.** La fuga de información se minimiza ya que el agente no recibe ninguna información sobre la orden o las intenciones de inversión del inversor. El *trader* de la parte compradora puede especificar sus instrucciones de *trading* y necesidades de inversión simplemente mediante la selección del algoritmo y las especificaciones de los parámetros algorítmicos, sin tener que proporcionar ningún tipo de información adicional.
- **Transparencia.** A los inversores se les proporciona un mayor grado de transparencia sobre cómo se ejecutará la orden. Dado que las reglas de ejecución que existen para cada algoritmo se facilitan a los inversores por adelantado, éstos sabrán exactamente cómo ejecutará el algoritmo las acciones en el mercado, ya que los algoritmos harán exactamente lo que están programados para hacer.
- **Acceso.** Los algoritmos pueden proporcionar acceso rápido y eficiente a diferentes mercados y los inversores se están viendo beneficiados gracias a las conexiones de alta velocidad. Gracias a esta alta velocidad, los usuarios podrán poner en práctica oportunidades tan pronto como aparezcan en el mercado.
- **Competencia.** La evolución del *trading* algorítmico ha visto la entrada de competencia de varios participantes del mercado, como proveedores independientes, empresas de software de administración de órdenes y ejecución, proveedores de terceros y equipos de desarrollo interno, además de los tradicionales agentes de bolsa. Los inversores han recibido los beneficios de esta creciente competencia en la forma de una mejor ejecución de los servicios y menores costes. Dada la facilidad y flexibilidad para elegir y cambiar de proveedores, los proveedores de algoritmos deben ser más proactivos y mejorar continuamente sus ofertas y eficiencias.
- **Costes de transacción reducidos.** Los ordenadores están mejor equipados y son más rápidos a la hora de reaccionar a las cambiantes condiciones del mercado y eventos no planeados. Tienen mayor capacidad para garantizar la coherencia entre la decisión de inversión y las instrucciones de *trading*, lo que se traduce en un menor coste de impacto en el mercado, menor riesgo y un mayor número de órdenes completadas (menor coste de oportunidad).

- **Eliminación de las emociones e inteligencia humana.** Dado que las órdenes de *trading* se ejecutan de forma automática siguiendo unas pautas o reglas, los *traders* no podrán dudar a la hora de ejecutar estas operaciones una vez son implementadas. Así, se elimina la influencia emocional del *trader* y no hay lugar a la interpretación.
- **Backtesting u otros mecanismos.** El *trading* algorítmico permite examinar cómo funcionan ciertos algoritmos o estrategias mediante simulaciones para observar el resultado y comprobar si el sistema es o no viable y si, en ese caso, supone un beneficio. En el caso de que no lo sea, el *backtesting* también permitirá optimizar parámetros para intentar obtener unos mejores resultados.

Desventajas

El *trading* algorítmico ha existido sólo desde principios de la década del 2000 y todavía está evolucionando a un ritmo sorprendente. Desafortunadamente, los algoritmos no son toda la solución a nuestras necesidades de *trading*. Algunas de las deficiencias y limitaciones incluyen:

- **Comodidad.** Los usuarios pueden volverse inversores pasivos y mantener los mismos algoritmos durante un tiempo, independientemente de las características de las órdenes y las condiciones del mercado, simplemente porque están familiarizados con el algoritmo y resulta más cómodo.
- **Continua evaluación.** Los usuarios necesitan probar y evaluar continuamente los algoritmos para asegurarse de que están siendo empleados de manera adecuada y que están haciendo lo que deben hacer, lo que supone tiempo. Además, los usuarios necesitan medir y supervisar el desempeño de los intermediarios, los algoritmos y las condiciones del mercado para comprender qué tipo de algoritmos son los más apropiados dependiendo del entorno, las condiciones de mercado y sus necesidades de inversión.
- **Especificación.** Los algoritmos funcionan exactamente como se especifica, lo que resulta beneficioso cuando el entorno de mercado es lo que se esperaba. Sin embargo, en el caso de que ocurran eventos inesperados, es posible que el algoritmo no esté adecuadamente capacitado o programado para ese mercado en particular, lo que puede generar un rendimiento inferior y mayores costes.

- **Coherencia.** Los usuarios deben garantizar la coherencia entre el algoritmo y sus necesidades de inversión. Sin embargo, garantizar la coherencia es cada vez más difícil en momentos en que la orden de *trading* no es tan transparente como podría serlo o cuando los algoritmos reciben nombres no descriptivos que no proporcionan ninguna idea de lo que intentan hacer.
- **Demasiados algoritmos y demasiados nombres.** VWAP (*volume weighted average price*), el precio ponderado promedio por volumen es un ejemplo de un nombre algorítmico bastante descriptivo y es bastante uniforme en todos los intermediarios. Se entiende por su nombre que busca el precio ponderado promedio por volumen. Sin embargo, un algoritmo como *Tarzan* no es descriptivo y no proporciona información sobre lo que intenta hacer durante la jornada. Los inversores necesitan comprender y diferenciar entre cientos de algoritmos y realizar un seguimiento de los cambios que ocurren en estas bases de código. Por ejemplo, una institución más grande puede hacer uso de veinte intermediarios diferentes con, entre cinco y diez algoritmos diferentes cada uno, y con al menos la mitad de esos nombres siendo no descriptivos, lo que hace más difícil su comprensión.
- **Descubrimiento de precios.** El crecimiento de los algoritmos y el declive de los expertos tradicionales y las funciones del creador de mercado, ha dado lugar a un proceso de descubrimiento de precios más difícil en la apertura del mercado. Aunque los algoritmos están bien versados para determinar la estrategia adecuada, aún no son capaces de determinar el precio justo de mercado de un valor.

3. CÓMO DESARROLLAR UNA ESTRATEGIA DE *TRADING*

Antes de establecer una serie de pautas acerca de cómo desarrollar una estrategia de *trading*, debemos conocer qué es realmente una estrategia de *trading*:

Una estrategia de *trading*, se define como: “Un método disciplinado de compra y venta de activos que implica trabajar dentro de un conjunto predefinido de reglas para tomar decisiones comerciales.” (Nasdaq).

Una vez conocemos lo que es una estrategia de *trading*, hay una serie de aspectos que debemos tener muy claros antes de desarrollarla. En primer lugar es importante establecer un objetivo que sea específico, que se pueda medir, alcanzable, relevante y con un límite temporal. También debemos tener claro en qué mercado operar y con qué activo, el tipo de sistema o estrategia que queremos seguir, el límite temporal y cómo programar la estrategia. Una vez conocemos todos estos aspectos de la estrategia, decidir cómo entrar y cómo salir del mercado. Desarrollar una estrategia de *trading* no es tarea fácil pero actualmente disponemos de una gran cantidad de información que puede facilitar nuestro proceso.

A continuación, vamos a ir paso por paso, explicando detalladamente cada uno de estos aspectos y adaptándolos a nuestra estrategia.

3.1 Establecer objetivos adecuados

En primer lugar, es importante que el creador de la estrategia establezca un objetivo o varios, preferiblemente siguiendo la estrategia **SMART** - *Specific, Measurable, Attainable, Relevant, Time bound* (Davey, 2014). Es decir, los objetivos deben ser específicos, medibles, alcanzables, relevantes y limitados en el tiempo.

Un objetivo **específico** es aquel que define muy claramente el fin que pretende conseguir. Por ejemplo, un objetivo que sea “quiero crear una estrategia de trading con la que ganar mucho dinero” o “quiero una estrategia sin riesgo” son ejemplos de objetivos muy imprecisos y demasiado generales.

También es importante tener en cuenta que el objetivo pueda ser **medido**. Es imposible conocer si una estrategia ha alcanzado el objetivo si éste no se puede medir. Por ello, un sistema de *trading* debe ser aquel con métricas de rendimiento del sistema que puedan ser evaluadas y comparadas con el objetivo establecido.

Además, el objetivo debe caracterizarse por ser **alcanzable**. Es decir, no se puede pensar en una estrategia que aporte un 500% de rendimiento anual, o una

estrategia cuya máxima pérdida sea de un 2%. Esto no tiene sentido y no es realista. Establecer un objetivo de este tipo, llevará al inversor o creador de la estrategia a desesperarse ya que nunca conseguirá alcanzar los objetivos propuestos.

Es importante que crear una estrategia de *trading* sea **relevante** para quien lo hace y que por tanto los objetivos también lo sean. Desarrollar una estrategia de *trading* cuyos objetivos no interesan al que la crea, tampoco tiene sentido. Entre un *trader* que es comprometido con lo que hace y uno que no lo es, ciertamente le irá mejor al que es comprometido.

Por último, crear una estrategia de *trading* requiere de mucho tiempo y establecer un **límite temporal** es una buena idea. Por ejemplo, puede establecerse un límite de tiempo para crear la estrategia o para darle vueltas a una determinada idea. De esta manera, el proceso siempre está en movimiento.

Para comprender estos conceptos mejor, vamos a analizar dos estrategias para ver cuál cumple todos estos requisitos:

“Voy a crear una estrategia de *trading* en seis meses que cumpla mis objetivos”. Esta estrategia sería *SMART* por un lado, pero no por otro. La primera parte de la frase (“voy a crear una estrategia de *trading* en seis meses”) es específica y medible pero la segunda parte (“que cumpla mis objetivos”) es muy poco precisa. El principio del objetivo es alcanzable ya que seis meses de trabajo duro pueden resultar en una estrategia decente. Sin embargo, hay que mejorar la segunda parte.

“Voy a crear una estrategia de *trading* en seis meses, que negocie el euro, con una rentabilidad media anual del 50%, una pérdida máxima del 30%, un porcentaje de éxito del 45% o mayor y que siga los pasos correspondientes en un proceso bien definido”. Esta es una estrategia que sí cumpliría con todos requisitos para ser *SMART*.

Un objetivo *SMART* puede cambiar para ser *SMARTER* si se evalúa y reevalúa adaptándolo a la situación en cuestión. Si uno se da cuenta de que el objetivo no se puede alcanzar, éste puede cambiarse (Davey, 2014).

El objetivo a conseguir en nuestra estrategia es el siguiente: Sobre un **capital de 100,000 dólares**, poder ganar un **25% anual**, con una máxima **pérdida del 12.5%** anual, en un marco temporal de **un año**.

3.2 En qué mercados operar y con qué activo

Para esta decisión, suele haber dos opciones: diseñar una estrategia que funcione en todos los mercados o desarrollar una estrategia para un determinado mercado. La primera opción viene a ser demasiado complicada ya que es muy difícil encontrar una estrategia que opere en todos los mercados. La segunda opción es más real ya que se pueden especificar los parámetros según las características del mercado. El único problema con esto es que cuando se crea una estrategia asumiendo unas características determinadas de mercado, se asume que éstas van a mantenerse para siempre y no tiene por qué ser así (Davey, 2014). Existen muchos mercados en los que operar incluyendo: mercado monetario (renta fija con un plazo inferior a 18 meses), mercado de renta fija (bonos), mercado de renta variable (acciones), mercado de derivados (futuros, opciones...) o el mercado de divisas. El creador de la estrategia debe operar en aquel mercado que más se ajuste a sus necesidades de inversión.

En el caso de la estrategia a desarrollar, se trata de una estrategia focalizada en un mercado determinado. Este mercado es el de la **bolsa de Estados Unidos**, concretamente el mercado de las **opciones**.

3.3 Qué tipo de sistema elegir

Es importante establecer las características del sistema que se va a elegir. Según Bandy (2011), generalmente se habla de tres tipos de sistemas: *Trending Systems* (sistemas de tendencia), *Reversion to the Mean Systems* (sistemas de reversión hacia la media), *Seasonality Systems* (sistemas de carácter estacional) and *Volatility Systems* (sistemas de volatilidad).

- ***Trending Systems***. Estos sistemas se basan en la premisa de que los precios generalmente siguen una determinada dirección durante un largo periodo de tiempo de forma que el *trader* es capaz de reconocer esta tendencia, y toma una posición en la dirección de esa tendencia, obteniendo un beneficio.
- ***Mean Reversion Systems***. Estos sistemas generalmente se basan en la idea de que los precios tienden a oscilar por encima y por debajo de un cierto nivel de equilibrio con cierto grado de regularidad. De esta forma, cuando los precios se mueven

demasiado con respecto a ese nivel de equilibrio, se toma una posición que se beneficiará en el momento en que el precio vuelva a ese equilibrio.

- **Seasonality Systems.** Estos sistemas se caracterizan por identificar periodos que se producen con mayor frecuencia y que afectan a un mayor número de activos negociables. Por ejemplo, hay mucha gente que piensa que las acciones tienen cierta estacionalidad relacionada con: la forma en que se negocian a finales de mes y principios del siguiente, el vencimiento de las opciones y luna nueva o luna llena.
- **Volatility Systems o sistemas de volatilidad.** Este tipo de sistema se caracteriza principalmente por esperar un cambio brusco en el precio sin importar en qué dirección. Es decir, hablamos de sistemas que se benefician en el momento en que se produce un cambio significativo en los precios de mercado, sin importar si el precio baja o sube.

Últimamente se han producido muchos eventos que han dado lugar a grandes oscilaciones en los valores de mercado. Algunos ejemplos incluyen: el *Brexit*, elecciones políticas, crisis en Corea del Norte, decisiones de los bancos centrales sobre los tipos de interés, políticas de Trump, etc. Por ello, para la estrategia que vamos a elegir vamos a seleccionar un sistema basado en volatilidad.

En el caso de la estrategia a desarrollar, vamos a seleccionar una estrategia basada en la **volatilidad de los precios de las opciones.**

3.4 Cuál es el marco temporal

Es muy importante elegir bien el marco temporal para una determinada estrategia, ya que una misma estrategia tendrá un desempeño muy distinto en diferentes marcos temporales, y por ello debe elegirse un marco temporal que se adecúe a los objetivos. Un factor importante a tener en cuenta, es que generalmente marcos temporales más cortos suponen más operaciones (Davey, 2014).

Como ya hemos establecido en nuestro objetivo, nuestro marco temporal será de **un año.**

3.5 Cómo entrar en el mercado

La entrada en el mercado suele ser una de las decisiones más fáciles para aquellos que crean una estrategia. Es importante que la orden de entrada sea simple, porque si no se puede explicar en un idioma, más difícil será convertirla en lenguaje de programación. Además, hay que limitar el número de parámetros en la entrada. Si existen varias condiciones de entrada, es fácil tener muchos parámetros a optimizar y cuantos más haya, más riesgo se corre a la hora de ajustar el modelo. También es importante ser original y pensar diferente para conseguir una entrada que otros no hayan probado antes. Por último, hay que empezar estableciendo una sola orden al principio y poco a poco ir añadiendo en caso necesario. De esta manera, uno se da cuenta de que muchas condiciones que pensaba que eran importantes, realmente no lo son (Davey, 2014). Más adelante se explicará la entrada en el mercado de forma más detallada.

La entrada en el mercado según esta estrategia, se producirá en un momento de **volatilidad mínima**, teniendo en cuenta una **ventana temporal de 1 a 3 meses**.

3.6 Cómo salir del mercado

La salida del mercado tiene un gran impacto en la rentabilidad y es importante dedicar tiempo a pensar cómo se va a salir del mercado. Según Davey (2014) las formas más comunes de salida de mercado son las siguientes:

1. **Detenerse y revertir.** La señal de entrada para una nueva posición también se convierte en la señal de salida para la posición actual. A muchas personas les gusta estar en el mercado en todo momento, y este método es precisamente lo que logra.
2. **Salidas más técnicas** como líneas de resistencia (precio por encima del actual) o soporte (precio por debajo del actual), medias variables u otros modelos. Es importante que estas reglas de salida coordinen con las reglas de entrada.
3. **Salida en el punto de umbral de rentabilidad.** Este tipo de salida suele limitar el potencial de las ganancias, ya que normalmente el mercado luego recupera su tendencia.
4. **Stop-loss** (cortar pérdida y abandonar la posición). Una orden de *stop-loss* permite tener más seguridad asumiendo más riesgo. En caso de que se entre en pérdidas, la orden venderá por sí sola evitando pérdidas mayores para el inversor.

5. **Objetivos de beneficio.** Se puede alcanzar un objetivo de beneficio y después dar salida al mercado.
6. **Trailing stop.** Un *trailing stop* lo que pretende es maximizar los beneficios cuando el precio se está moviendo a favor del inversor pero cerrará en caso de que el precio se mueva en la otra dirección por un porcentaje específico.

Más adelante se explicarán las órdenes de salida de forma más detallada.

La salida del mercado puede tener dos formas: **objetivo de beneficio o stop-loss**. En caso de que se produzca un incremento brusco de la volatilidad, y, por tanto, se genere un beneficio, se abandonará la posición. En caso de que no se produzca ningún cambio o de que se produzca un cambio pero no lo suficientemente brusco, se dará salida a la posición para minimizar pérdidas.

3.7 Cómo programar la estrategia

Una vez se han elegido las reglas de entrada y salida específicas, el mercado en el que operar, el tipo de sistema y el marco temporal, ya se puede programar la estrategia. En caso de no estar familiarizado con el lenguaje de programación, se puede contratar a un profesional que programe la estrategia. La única desventaja es que cada vez que se necesite reajustar algún parámetro o cambiar algún dato habrá que esperar a que el profesional lo haga y además cobrará por ello. Otra opción es tratar de programar la estrategia uno mismo. Actualmente disponemos de una gran cantidad de información en internet, libros y tutoriales que pueden ayudarnos a desarrollar una estrategia y a familiarizarnos con el lenguaje de programación (Davey, 2014).

Para esta estrategia, se utilizará el lenguaje **Python** soportado por *Interactive Brokers* y cuyo entorno de desarrollo es *Spyder*.

4. EL MERCADO DE LAS OPCIONES

4.1 Introducción

Para poder comprender mejor la estrategia y su funcionamiento es preciso primero conocer el mercado de las opciones. Con el paso del tiempo, gracias a las nuevas tecnologías, cada vez más inversores individuales participan activamente en tareas de inversión por su cuenta, tomando sus propias decisiones, especialmente en el área de compra y venta de acciones. Junto a este crecimiento de inversión individual, existe una creciente curiosidad por los beneficios potenciales en el mercado de las opciones sobre acciones. Sin embargo, muchos inversores individuales no comprenden cómo funciona este mercado y es por ello que le dan la espalda.

4.2 Historia de las opciones

El uso de opciones en un intento de asegurar ganancias económicas o financieras se remonta en nuestra historia mucho más allá de lo que la mayoría de la gente esperaría. En la *Política* de Aristóteles, publicada en el año 332 a.c, se explica que el creador de las opciones fue Tales de Mileto, un filósofo. Gracias a su observación de las estrellas y el comportamiento del clima, Tales de Mileto predijo una gran cosecha de aceitunas para un año determinado. Como disponía de poco dinero, decidió obtener por una pequeña cantidad y por anticipado, el derecho de uso a las prensas de aceitunas donde se extraía el aceite. Efectivamente, su predicción se cumplió y decidió vender sus derechos por una gran cantidad de dinero. Por esta razón, muchos reconocen a Tales de Mileto como el creador de las opciones (Hoffman y Williams, 2001). Con esto, entendemos que un contrato de opción es aquel que otorga el derecho pero no la obligación de comprar o vender un bien o servicio.

4.3 El nacimiento y crecimiento de la industria de opciones actual

A mediados de la década de los 90 en EE.UU., en respuesta a la Ley de Valores de 1934, se formó la PCBC (*Put and Call Brokers and Dealers Association*) que dio lugar al nacimiento de la negociación de las opciones OTC (*Over-the-counter*). En estos momentos, la aceptación del público era muy limitada y las opciones suponían un vehículo de inversión muy ilíquido. Éstas no estaban estandarizadas y su derecho no

podía ejercerse hasta su fecha de vencimiento. Con la creación *del Chicago Board of Options Exchange* (CBOE) en 1973, los contratos de opciones de compra se estandarizaron. Junto a esta estandarización vino la liquidez y las opciones se convirtieron en una herramienta a la que el público general podía acceder. La *Options Clearing Corporation* (OCC) se estableció como la responsable de los contratos de opciones negociados. Con la función de responsable, la OCC eliminó la preocupación de que la otra parte del contrato de opción pudiera incumplir su obligación a través de una garantía a cada comprador y vendedor del desempeño de las opciones en sus contratos (Hoffman and Williams, 2001).

Uno de los desarrollos más importantes fue la institucionalización del sistema de creador de mercado. Los creadores de mercado son profesionales que negocian grandes cantidades de contratos de opciones sirviendo al público general y crean liquidez en el mercado. En el día a día, los creadores de mercado son responsables de más de la mitad de volumen total de opciones negociadas aunque este número está disminuyendo con el crecimiento del *trading* algorítmico y los programas automáticos. Con la infraestructura necesaria, la única preocupación relacionada con las opciones era el precio. Un inversor podía estar preparado para negociar una opción pero no a un precio justo. Fischer Black y Myron Scholes (1973) resolvieron este problema con su modelo conocido como el modelo *Black-Scholes*.

“El modelo *Black-Scholes* calcula el valor de una opción basándose en entradas específicas relacionadas con las diversas características de una opción, como el precio de ejercicio, el precio del activo subyacente, el tiempo de expiración, el rendimiento de activos financieros sin riesgo (tasa de interés) y la volatilidad histórica del activo subyacente (desviación estándar del mismo)” (Hoffman y Williams, 2001).

Robert C. Merton fue el primer economista en colaborar con la creación de este modelo de precios de opciones expandiendo su comprensión matemática. Más tarde, este trabajo llevó a sus autores a recibir el Premio Nobel en Ciencias Económicas en el año 1997. Este modelo no fue el primero en calcular el valor de una opción pero sí fue el primero que redujo el número de cálculos necesarios para ello. Su simplicidad permitió el uso de este modelo a tiempo real en el entorno comercial. Su capacidad para identificar los riesgos asociados con la negociación de las opciones también permitió

que el inversor individual las negociara utilizando los mismos cálculos que el inversor profesional. De esta manera, los modelos informáticos han permitido el acceso de las opciones a un público mayor y ha impulsado el crecimiento de esta industria.

En 1975 y 1976, debido al creciente interés en la negociación de opciones, la bolsa de valores americana (actualmente conocida como el NYSE – *New York Stock Exchange*), la de Filadelfia y la de California comenzaron a negociar opciones de compra, también conocidas como opciones *call*. Más tarde, en 1977 se introdujo la negociación de opciones de venta, también conocidas como opciones *put*.

Con el auge de la negociación de las opciones, el interés por encontrar el precio justo también crecía. Es entonces cuando surgió otro modelo, conocido como el modelo de Cox, Ross y Rubinstein (1979). Este modelo se caracteriza por ser un modelo binomial que explica que en cualquier momento existen dos valores posibles para una opción dependiendo de si se produce un repunte o una bajada en la acción. Con esta información el modelo calcula un precio teórico (Hoffman and Williams, 2001).

En 1999 ya se negociaban un total de aproximadamente 507 millones de contratos de opciones en las cuatro bolsas de valores americanas, lo que demuestra claramente un crecimiento enorme de la industria (Hoffman y Williams, 2001). Actualmente, las opciones son una inversión segura y el desarrollo de la tecnología ha permitido que los inversores individuales negocien opciones a un coste mínimo.

4.4 Conceptos básicos de las opciones

Todo derivado financiero tiene un precio que depende del precio de otro activo. Las opciones son un tipo de derivado financiero cuyo precio depende del precio de otro activo conocido como activo subyacente. Este activo subyacente puede ser un bono, índice bursátil o una acción. En este caso, el activo subyacente será la acción, pues la estrategia que se va a implementar comprará opciones de los valores de la bolsa de Estados Unidos. Las opciones cuestan una fracción de lo que cuesta la acción subyacente pero sin embargo tienen movimientos mucho más significativos en porcentaje y es por eso que tienen una duración determinada para ser ejercidas. Si el precio de una acción cambia, también lo hará el precio de la opción, aunque por una cantidad menor. Para una opción de compra u opción *call*, si el precio de la acción sube, el precio de la opción también subirá. Si el precio de la acción baja, también lo hará el

precio de la opción. En el caso de una opción *put* o de venta, si el precio de la acción baja, el precio de la opción sube y si el precio de la acción sube, el precio de la opción de venta baja (Olmstead, 2013).

Una opción de compra o la compra de opción sobre acción otorga al comprador de ésta opción, el derecho pero no la obligación de comprar una cantidad determinada de acciones a un precio fijo pagando una prima en un tiempo determinado. A cambio de este precio, el vendedor de la opción vende ese derecho (Olmstead, 2013). La opción de compra se conoce también como opción *call*.

En la otra dirección encontramos la opción de venta o la venta de opción sobre acción. En este caso, se otorga al comprador de dicha opción, el derecho pero no la obligación de vender una cantidad de acciones a un precio fijo, en un tiempo determinado, pagando una prima. A cambio de este precio, el vendedor de la opción está obligado a comprarla si el comprador del derecho a vender decide ejercer su derecho de venta. La opción de venta se conoce también como opción *put* (Olmstead, 2013).

Las opciones de compra y venta se caracterizan por lo siguiente: fecha de vencimiento, precio de ejercicio y prima. La fecha de vencimiento indica el día de vencimiento del derecho del poseedor. Es decir, el poseedor de la opción *call* o *put* tiene hasta un determinado día para ejercer su derecho. El precio de ejercicio indica el precio al cual el poseedor del derecho puede ejercer su derecho. Es decir, es el precio al que el poseedor de la opción puede adquirir o vender las acciones. También se conoce como *strike*. Al precio de la opción se le conoce como prima. Es decir, la prima es el precio que paga el comprador de las opciones por obtener el derecho de comprar o vender esas acciones (Rascon, 2007). Hay que recordar que un contrato de opción representa cien acciones en España y generalmente también en Estados Unidos, aunque en este caso existe un parámetro en el contrato, conocido como *MULTIPLIER* que lo indica.

La opción *call* interesa comprarla en tendencia de un mercado alcista y la opción de venta en tendencia de un mercado bajista. Por ello, con la compra de ambas dos, no importa en qué dirección se mueva el mercado, el inversor puede beneficiarse. Para que esto pase, el beneficio de una de las opciones debe superar la pérdida de la otra (Olmstead, 2013).

4.5 Principales diferencias entre acciones y opciones

Olmstead (2013) establece las siguientes como las mayores diferencias entre las acciones y las opciones:

- **Apalancamiento.** Las opciones generalmente tienen un coste mucho más bajo que el coste de la acción. Por ejemplo, si creemos que el precio de la acción de XYZ va a subir y actualmente se encuentra en los 51 dólares, se pueden comprar 100 acciones por un coste de 5,100 dólares. Sin embargo, podríamos comprar un contrato de opción de compra (teniendo en cuenta que un contrato equivale a 100 acciones) pagando por ejemplo 3 dólares por acción, lo que supone un coste total de 300 dólares.
- **Tiempo limitado.** Una de las razones por las que las opciones son más baratas que las acciones es que tienen un límite de tiempo. Una posición en largo o corto en una acción, puede mantenerse indefinidamente. Sin embargo, una posición de opción puede mantenerse hasta una fecha limitada, conocida como fecha de vencimiento. Cuando se compra una opción se puede elegir entre varias fechas de vencimiento mensuales y cuanto más tiempo queramos mantener el contrato de opción más caro será. Otro aspecto importante es que el valor de una opción baja con el tiempo si el precio de la acción no sufre variaciones. Esto supone un riesgo a la hora de poseer una opción.
- **Movimiento en el precio.** Cuando el precio de la acción varía, también lo hace el precio de la opción pero por una cantidad menor. Cuánto se ajusta el cambio en el precio de la opción respecto del precio de la acción depende del precio de referencia establecido en el contrato de la opción. Este precio de referencia se conoce como precio de ejercicio o *strike price*. Cuando se decide comprar una opción, se puede elegir entre distintos precios de ejercicio. Si el precio de ejercicio de una opción ya sea *call* o *put* es similar al precio de mercado, se conoce como una opción *at-the-money*. Si el precio de ejercicio de una opción de compra (venta) está por encima (debajo) del precio de la acción, la opción está *out-of-the-money*. Si el precio de ejercicio de una opción de compra (venta) se encuentra por debajo (encima) del precio de la acción, se dice que la opción está *in-the-money*.
- **Riesgo financiero.** Cuando se compra una opción, el riesgo máximo equivale al coste de comprar esa opción. Lo peor que puede pasar es que se mantenga la

posición de la opción hasta la fecha de vencimiento, momento en el que la opción no tenga ningún valor por no haberse producido un cambio significativo en el precio de mercado. Por ejemplo, si compramos un contrato de opción por 2 dólares por acción, nuestro máximo coste será de 200 dólares. Si comparamos la máxima pérdida de este contrato con la máxima pérdida de mantener una posición en la acción en sí, podemos observar que la pérdida es mucho mayor para la acción que el coste de una opción de compra o venta.

5. ESTRATEGIA BASADA EN LA VOLATILIDAD DE LOS PRECIOS DE LAS OPCIONES: *STRADDLE/ STRANGLE*

5.1 Concepto

La estrategia basada en la volatilidad de los precios de las opciones, lo que espera y desea es que el precio de la opción sufra un cambio brusco en el mercado, sin importar en qué dirección. De esta manera, el negociador o *trader*, podrá beneficiarse de una variación en el precio sin importar si éste baja o sube.

Como ya sabemos, un componente muy importante a la hora de calcular el precio de las opciones es la volatilidad. La estrategia se basa principalmente en la compra de un combo (una opción *call* y una opción *put*) para un activo subyacente en un momento de mínima volatilidad. Una vez compradas, se esperará a la evolución de esa combinación de opción *call* y opción *put*. Pueden suceder tres cosas:

1. El combo va perdiendo valor en caso de que no se produzca ningún movimiento del subyacente. Para minimizar pérdidas, se dará salida a la posición en el mercado.
2. Se produce una variación en el precio de mercado, pero no lo suficientemente grande como para generar beneficio. Esto generará una pérdida y también se dará salida al mercado.
3. Se produce una variación significativa en el precio del subyacente, lo que da lugar a una variación en el precio de las opciones lo suficientemente significativo para que se genere un beneficio.

El combo puede ser una *call* y una *put* con mismo precio de ejercicio o con precio de ejercicio diferente. A pesar de que la diferencia es mínima, es importante conocerla pues da lugar a nombres diferentes: *straddle* y *strangle*.

5.2 Diferencia entre *straddle* y *strangle*

Es importante conocer las diferencias entre la estrategia conocida como *straddle*, y la conocida como *strangle* a pesar de que la diferencia es mínima.

Un *straddle* es una estrategia de opciones en la que el inversor mantiene una posición tanto en una opción de compra como en una opción de venta con el mismo precio de ejercicio y fecha de vencimiento (Olmstead, 2013). Un *strangle* es muy similar al *straddle*, la única diferencia a destacar es que el precio de ejercicio no será el mismo para la opción de compra y la opción de venta. Ambas estrategias permiten al inversor obtener ganancias independientemente de si el precio de la acción sube o baja, suponiendo que el precio de las acciones varía de manera significativa.

5.3 Diferencia entre *long straddle* y *short straddle*

Para comprender mejor cómo beneficiarse de la estrategia de *straddle*, cabe distinguir entre *long straddle* y *short straddle*. La diferencia se aplica de la misma manera a la estrategia de *strangle*.

En el caso de una estrategia de *long straddle*, el inversor lo que hace es comprar, al mismo tiempo, tanto una opción *call* como una opción *put* con el mismo precio de ejercicio, y la misma fecha de vencimiento. De esta forma, si el precio de la acción se mueve lo suficiente con respecto al precio de ejercicio, el inversor ganará dinero sin importar la dirección en la que se mueva el mercado. En caso de que baje el precio, el inversor ganará dinero con la opción *put*, y si la tendencia de mercado es a la alza, el inversor ganará dinero con la opción *call*.

La estrategia de *short straddle* plantea justo lo contrario. En un momento determinado, se venden tanto una opción *call* como una *put* con la misma fecha de vencimiento y el mismo precio de ejercicio. Se obtendrá un beneficio por las primas obtenidas al vender estos derechos de compra y venta. Sin embargo, si el mercado se mueve lo suficiente en cualquiera de las direcciones, entonces el riesgo para el vendedor de las opciones es ilimitado y las pérdidas pueden ser grandes. Es decir, en este caso lo que se espera es una variación muy pequeña en el precio de la acción mientras se tenga una posición en las opciones. Es por ello que la estrategia de *short straddle* es recomendada para profesionales con más experiencia que sepan moverse mejor en situaciones de riesgo (Beagles, 2009).

Por lo explicado anteriormente, es la estrategia de *long straddle* la que se va a emplear en el modelo de *trading* algorítmico.

5.4 Condiciones de mercado adecuadas para implementar la estrategia

Es importante saber que esta estrategia es muy rentable cuando se espera un cambio muy significativo en el mercado en los próximos días, semanas o meses. Esta estrategia no es una buena opción si ha tenido lugar un evento que ha afectado a los mercados anteriormente o si se ha producido un movimiento muy brusco en el mercado recientemente. Por tanto, es importante que el mercado se encuentre en mínimos de volatilidad (Olmstead, 2013).

5.5 Funcionamiento de la estrategia

Ambas estrategias suponen, básicamente la compra de dos opciones:

- Posición larga de una opción de compra con un precio de ejercicio similar al precio de mercado.
- Posición larga de una opción de venta con un precio de ejercicio igual (*straddle*) o distinto (*strangle*) al de la opción de compra.

5.6 Ejemplo del funcionamiento de la estrategia basada en la volatilidad de los precios de las opciones con un *long straddle*

A finales de Mayo, la acción de la empresa XYZ cotiza alrededor de los 30 dólares y es una época en la que existen noticias contradictorias sobre la compañía. Algunos informes dicen que XYZ va a publicar un aviso de que el siguiente trimestre los beneficios van a quedar muy por debajo de las expectativas. Otros informes dicen que un competidor plantea comprar XYZ a una prima considerable.

Para intentar atender a cualquiera de estos dos eventos tan dramáticos, decidimos emplear una operación de straddle comprando tanto una opción de compra como una opción de venta con el mismo precio de ejercicio: Se compra tanto una opción call por 2.10 dólares por acción como una opción put por 1.90 dólares por acción con un precio de ejercicio de 30 dólares (recordar que cada contrato de opciones equivale a 100 acciones).

El coste total será de 400 dólares: $[(2.10 + 1.90) \times 100 = 400]$. Esto significa que el riesgo máximo será de 400 dólares. Ahora, no importa lo que suceda en el mercado, puesto que si se produce un cambio lo suficientemente representativo, una de las opciones dará un buen beneficio.

A continuación vamos a observar qué sucede en un escenario ideal:

A mediados de junio, el aviso sobre los beneficios de la compañía se publica y el precio en el mercado baja rápidamente a 25 dólares. Debido a esta rápida bajada en el precio, el precio de la opción de venta sube a 6.20 dólares por acción. Por otro lado, la opción de compra consigue mantenerse en un precio de 0.50 dólares por acción. A la hora de vender tanto la opción de compra como la opción de venta, se genera un beneficio de 270 dólares: $[(6.20 + 0.50) \times 100 = 670]$. Si a lo que generamos con la

venta le restamos lo que nos costaron las opciones a finales de mayo ($670 - 400$), observamos que se ha producido un beneficio de 270 dólares, 68%.

Debido a que se ha producido una variación en el precio de mercado suficientemente brusca, el beneficio generado por la venta de la opción de venta es mayor que la pérdida generada con la venta de la opción de compra y por ello se genera un beneficio.

A continuación veremos qué sucede en una situación no tan ideal:

La noticia sobre la bajada de los beneficios no se revela hasta finales de julio y el precio de mercado baja sólo hasta los 27.50 dólares. Con pocas semanas de margen hasta que las opciones venzan, hay poco valor de tiempo restante en la opción de venta, que está a un precio de 2.80 dólares por acción. El precio de la opción de compra se mantiene en 0.20 dólares por acción. Se decide dar salida a estas dos opciones, y se hace por 300 dólares: $[(2.80 + 0.20) \times 100 = 300]$. Si a lo que generamos con la venta le restamos lo que nos costaron las opciones a finales de mayo ($300 - 400$), observamos que se ha producido una pérdida de 100 dólares, 25%.

Si analizamos ambas situaciones, nos damos cuenta de que no vale sólo con que se produzca una variación en el precio, sino que esta variación debe ser lo suficientemente representativa como para que se produzca un incremento grande en el precio de opción apropiada. Además, el evento que provoque la variación en el precio de mercado debe producirse relativamente pronto después de haber iniciado la estrategia de straddle para evitar la pérdida de demasiado valor de tiempo.

5.7 Requisitos que debe cumplir una buena estrategia de *straddle* o *strangle*

Olmstead (2013) destaca cuatro requisitos que debe cumplir una buena estrategia ya sea de *straddle* o *strangle*:

- El precio de mercado debe encontrarse en un rango de precios ni muy altos ni muy bajos. Es decir, por un lado, debe haber margen para que el precio pueda bajar lo suficiente (por ejemplo, una acción con un precio de 5 dólares tiene poco margen para bajar mucho). Por otro lado, si el precio de la acción es demasiado caro, las opciones pueden tener precios demasiado altos como para crear una estrategia de *straddle* con buenos beneficios potenciales.

- Debe identificarse un evento próximo que vaya a tener un impacto en el precio de la acción. En una situación ideal, se esperaría que el precio suba con noticias positivas y baje con noticias negativas. Ejemplos de eventos que pueden afectar son la publicación de resultados, decisiones judiciales o nuevas normas que afecten a la compañía.
- Además, este nuevo evento debe tener lugar en el corto plazo y las opciones compradas no deben vencer hasta por lo menos un mes después de que se haya producido la noticia.
- Hay que tener en cuenta que si el evento ha generado ya expectativas en el mercado, esto va a provocar que los precios de las opciones se encuentren por encima de lo habitual por la creciente demanda y esto puede dar lugar a beneficios más bajos puesto que el coste de compra de las opciones es mayor (Olmstead, 2013).

Una vez conocemos el funcionamiento de la estrategia, es importante disponer de los elementos necesarios para ponerla en marcha.

6. ELEMENTOS NECESARIOS PARA DESARROLLAR LA ESTRATEGIA DE *TRADING*

Existen una serie de elementos que son necesarios para poder poner una estrategia de *trading* en marcha. Los más importantes son los siguientes:

- **Bróker.** Un bróker proporciona datos de mercado y ejecuta órdenes de compra y venta.
- **Red de comunicaciones.** Una buena red de comunicaciones requiere de acceso a internet de forma segura y rápida.
- **Infraestructura informática.** Esto incluye todos los programas necesarios y las comunicaciones entre ellos para el desarrollo de la estrategia.

6.1 Bróker

Los brókeres tienen una misión importante en el buen funcionamiento de las estrategias de *trading*. Los brókeres online han crecido mucho en los últimos años y han permitido el acceso de usuarios profesionales y no profesionales a datos de mercado reales y a plataformas de *trading*. Son responsables de que el *trading* algorítmico pueda llevarse a cabo de forma tan sencilla. Éstos se encargan de suministrar aplicaciones a través de las cuales los usuarios pueden enviar órdenes a ejecutar en el mercado. Algunos de los más avanzados proporcionan interfaces para que sean programas automáticos los que envíen las órdenes al mercado. Dos de los brókeres más conocidos son *TD Ameritrade* e *Interactive Brokers*.

- ***TD Ameritrade*.** Este bróker es conocido por proporcionar acceso a plataformas de *trading* intuitivas y sencillas a la hora de utilizarlas. Además ofrece distintos niveles de servicio según el nivel de conocimientos del inversor así como ayuda a través de una app móvil. También da acceso a datos reales de mercado y ofrece todo tipo de productos incluyendo la bolsa (acciones y opciones), futuros, bonos y renta fija, *ETFs* (fondos cotizados), fondos de inversión etc. Este bróker ha recibido el premio a mejor bróker en inversiones a largo plazo varios años consecutivos (*TD Ameritrade*, 2018).
- ***Interactive Brokers*.** Este bróker es conocido no sólo por proporcionar servicios muy intuitivos para no profesionales sino por ofrecer servicios a un bajo coste. Es el bróker interactivo más grande del mundo por número de transacciones y es líder en el mercado de *Forex* (mercado de divisas). También da acceso a plataformas de *trading* online, y ofrece distintos tipos de producto incluyendo divisas, acciones, opciones, futuros, *ETFs*, metales, índices, bonos etc. (*Interactive Brokers*, 2018).

6.2 Red de comunicaciones

Como ya hemos dicho, una buena red de comunicaciones debe ser rápida y segura. Hasta hace unos años, tener acceso a internet de forma constante era difícil. Sin embargo, hoy en día, un buen acceso a internet es muy fácil pues las conexiones de alta velocidad están muy logradas y casi todo el mundo dispone de acceso a la red. Acceso a la red es crucial a la hora de poder llevar a cabo una estrategia de *trading* y es muy importante tenerlo en cuenta.

6.3 Infraestructura informática

Por último es importante disponer de una adecuada infraestructura informática. En los últimos años, debido al auge de la tecnología y al crecimiento del *trading* online gracias a las numerosas facilidades que lo permiten, se han ido interponiendo **plataformas** para llevar a cabo el *trading*. Estas plataformas son cruciales en la vida de un inversor pues proporcionan datos de mercado, precios de los activos financieros y permiten abrir, cerrar y controlar las posiciones según las necesidades de inversión. La mayor parte de los *traders* seleccionan como plataforma aquella que les proporciona su bróker online. Por ejemplo, *Interactive Brokers* dispone también de una plataforma para *trading* conocida como *Trader Workstation* (TWS). Existen otros inversores que prefieren seleccionar otras plataformas individuales que les proporcionan otras herramientas más avanzadas. Algunas de las más conocidas son: *Ninjatrader*, *Ctrader* o *Metastock*. A su vez, han ido apareciendo **lenguajes para la programación** de algoritmos que están al alcance de todos los usuarios a través de tutoriales o libros. Chan (2017) habla de los lenguajes de programación más conocidos entre los que encontramos: *Matlab*, *R* y *Python*.

- **Matlab.** *Matlab* es un lenguaje de programación muy conocido por disponer de un interfaz muy completo y fácil de utilizar. También dispone de una enorme caja de herramientas que habla de casi todas las técnicas matemáticas que puedan necesitarse a la hora de desarrollar una estrategia de *trading*. Una de estas herramientas, permite que *Matlab* conecte con numerosos brókeres para recibir datos de mercado y enviar y recibir órdenes de mercado. La única desventaja es que no ofrece sus servicios de forma gratuita.

- **R.** *R* es muy similar a *Matlab*. Es un lenguaje de programación que también ofrece paquetes especializados a los inversores. Muchos de ellos son más complejos que los de *Matlab* debido a que hay más investigadores académicos que lo utilizan. La mayor desventaja del lenguaje *R* es que es más lento.
- **Python.** *Python* es muy similar a los dos anteriores. Es especialmente conocido por ser fácil de utilizar y por disponer de librerías de cálculos matemáticos y temas científicos muy avanzados. Es más rápida que *R* pero más lenta que *Matlab*.

Tabla 1. Comparación de los programas Matlab, R y Python. Escala 1-3 siendo 3 mejor y 1 peor.

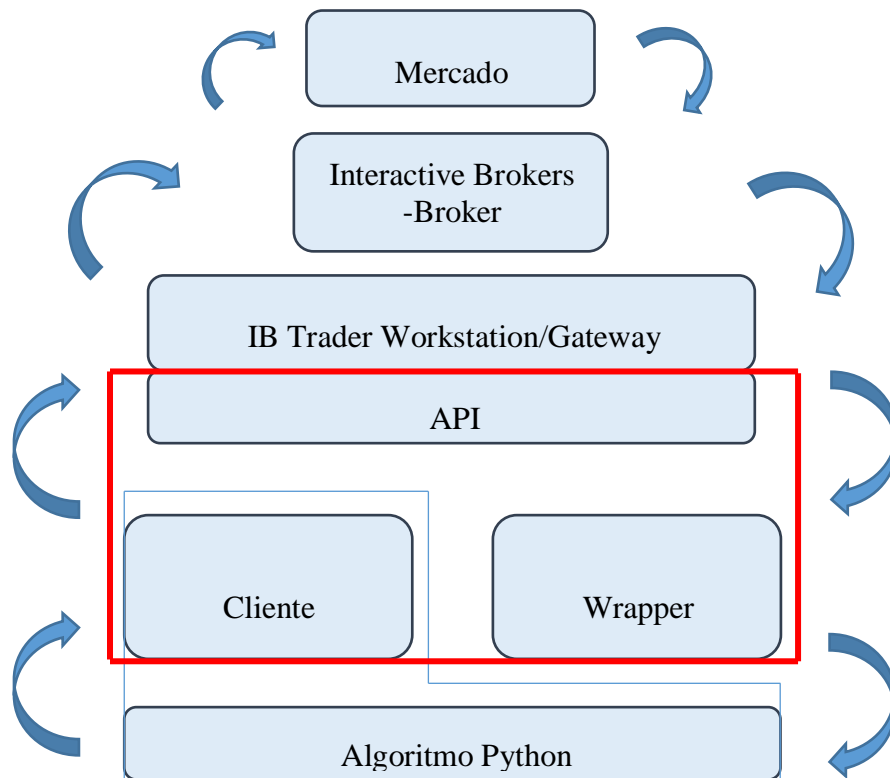
Característica	<i>Matlab</i>	<i>R</i>	<i>Python</i>
Facilidad de uso	1	2	2
Velocidad	1	3	2
Cajas de herramientas	2	1	1
Conectividad a brókeres	1	2	2
Atención al cliente	1	N/A	N/A
Precio	2	1	1

Fuente: Chan, 2017

Python es el lenguaje de programación utilizado para el desarrollo de esta estrategia. En concreto, en la implementación de esta estrategia se hace uso de una librería que se llama Pandas que permite la manipulación de series temporales de datos, por ejemplo cotizaciones bursátiles de acciones u opciones. Está soportado por *Interactive Brokers* como bróker, lo que nos permitirá tener acceso a datos de mercado y a enviar órdenes y recibirlas y además ofrece una plataforma de trading anteriormente mencionada que se conoce como *IB Trader Workstation*. El entorno de desarrollo que utiliza *Python* se llama *Spyder (Scientific Python Development Environment)* y se trata de un entorno muy potente. Este entorno se utiliza para poner en práctica los programas que se han creado con el lenguaje *Python* y además dispone de herramientas de edición y de inspección de datos y depuración del programa.

7. ESTRUCTURA DE UN PROGRAMA DE *TRADING*

En este apartado vamos a ver un esquema que explica cómo se relacionan *Python* e *Interactive Brokers* a través del interfaz para el flujo de información y conectar con el mercado.



Fuente: elaboración propia

El programa realiza cálculos sobre información de activos o del mercado que le tiene que proporcionar el bróker. Para eso, llama a funciones (agrupadas en el bloque denominado Cliente) que le permiten, por ejemplo, recuperar cotizaciones históricas de un valor, cotizaciones en tiempo real u otro tipo de datos de mercado similares. Estas funciones envían la petición de los datos a las plataformas a través del interfaz (API, por sus siglas en inglés *Application Programming Interface*) que a su vez transmite estas solicitudes a los mercados. Los mercados suministran los datos a la plataforma, y ésta los devuelve al programa que los solicitó nuevamente a través de un bloque del interfaz conocido como *Wrapper* que está continuamente monitorizando las respuestas obtenidas.

Una vez conocemos cómo funciona el flujo de información, procedemos a analizar la estrategia en detalle.

8. IMPLEMENTACION DE LA ESTRATEGIA DE *TRADING* CON *PYTHON*

8.1 Introducción

En este apartado se llevará a cabo la implementación de la estrategia. Para ello, hay un conjunto de módulos, cada uno de los cuales realiza una misión específica. La estrategia ejecuta diariamente un proceso que realiza las siguientes funciones:

Hay un primer módulo que calcula si la volatilidad histórica está en un mínimo mensual, bimensual o trimestral para un conjunto de valores definidos y si está en un mínimo, los selecciona. El segundo módulo, para los valores seleccionados, identifica el precio de ejercicio de una opción de compra y de una opción de venta y el periodo de vencimiento de ambas. El periodo de vencimiento lo selecciona según si el mínimo de volatilidad histórica sea mensual (vencimiento más corto) o trimestral (vencimiento más largo). El tercer módulo lo que hace es verificar el precio de mercado para el combo seleccionado y, en función de eso y el periodo de vencimiento determina la cantidad que se compra y el precio de salida para el beneficio objetivo. El cuarto módulo solicita la orden de compra del combo y monitoriza su ejecución. Existe un quinto módulo que para las órdenes emitidas comprueba si se ha ejecutado la cantidad solicitada, si se ha ejecutado solo parcialmente o si no se ha ejecutado. Según cómo haya sido la ejecución las etiqueta como COMPLETADA, PARCIAL o CANCELADA. Por último, el sexto módulo, para las ordenes que se han completado total o parcialmente solicita una orden de venta a un precio límite (más alto que tiene en el mercado) con el que se lograría el beneficio objetivo. Esta orden de venta es GTC (*Good-till-cancel*, por sus siglas en inglés). Esto quiere decir que se mantiene vigente hasta su cancelación explícita. Por otro lado, existe un módulo (conocido como el módulo principal) que va lanzando cada uno de los anteriores de manera secuencial y otro que programa el día y la hora (siempre en días de actividad de la bolsa) del módulo principal.

A continuación se irá módulo por módulo explicando más en detalle la función de cada uno de ellos. Aunque no es posible mostrar todas las líneas de código de cada uno de los módulos, sí que se mostrarán y explicarán las principales.

8.2 Módulo Principal

Como ya se ha establecido antes, el módulo principal va lanzando cada uno de los módulos anteriores de forma secuencial. Ver **Anexo I** para las líneas de código. En líneas generales, el módulo principal se encarga, en primer lugar, de importar las librerías que se van a necesitar. Entre ellas, se importa la lista de valores seleccionada para la estrategia. Ver **Anexo II** para observar esta lista de valores. En segundo lugar, define una tarea, que se va a ejecutar de manera independiente del programa (conocida como *thread* en las líneas de código). Esta tarea es la tarea que se encarga de recibir los

datos de mercado y es una tarea que se va a ejecutar de manera continua. Es decir, esta tarea va a estar en continuo contacto con el mercado para recibir los datos necesarios. Así, después de definir la tarea, el programa se conecta al interfaz (API). Por último, el programa llama a cada uno de los módulos que van a ejecutarse a través de la función *def main*

```
def main():  
  
    import historical_volatility  
    import option_chain  
    import combo_entry_conditions  
    import entry_orders_from_contract_file  
    import entry_orders_completed_from_contract_file  
    import exit_orders_from_contract_file
```

Una vez se han ejecutado cada uno de los módulos (I, II, III, IV, V), finaliza la tarea y se desconecta del API.

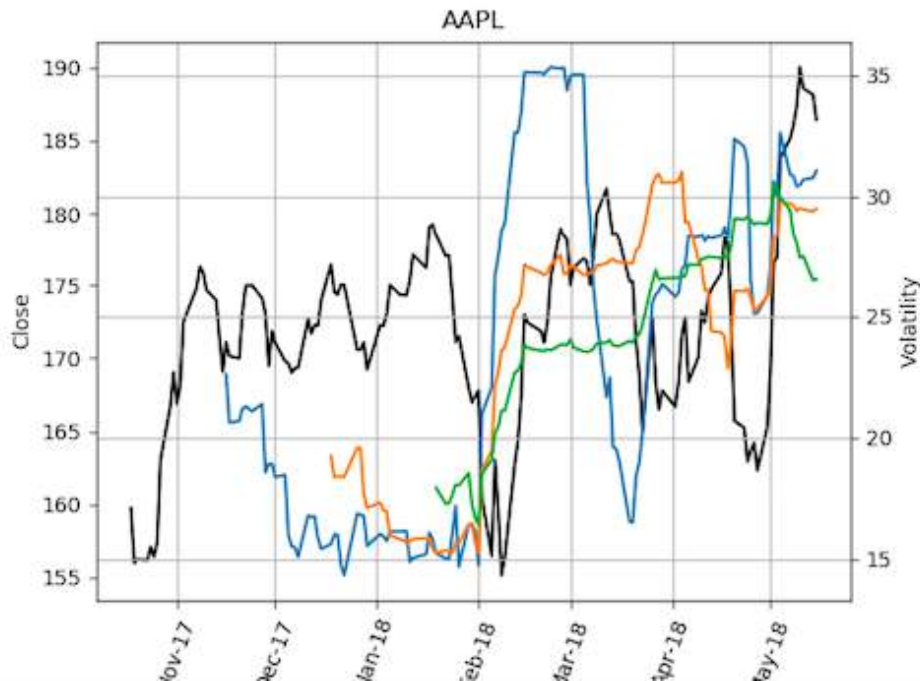
8.3 Módulo I

El primer módulo es aquel que se encarga de calcular si la volatilidad está en un mínimo mensual, bimensual o trimestral. En caso de lo esté, selecciona aquellos valores. En primer lugar, el módulo se encarga de definir cómo se calcula la volatilidad a uno, dos tres meses. Además, se encarga de definir el mínimo de volatilidad:

```
def df_volatility(df, period):  
  
    # Calculate anualized historical volatility for the period (1M, 2M or 3M)  
    return (df.rolling(period).std() * np.sqrt(gv.period_1Y) * 100)  
  
def df_minimum(df, period):  
  
    return df.rolling(period).min()
```

A continuación, el módulo pide los datos históricos para cada valor y los mete en una serie de Pandas *data frame*, que va a permitir, a continuación dos cosas: por un lado, va a calcular volatilidad histórica de cada uno de los valores y por otro, va a permitir que se vea, en un gráfico cada valor y su volatilidad histórica. Aquí vemos un ejemplo:

Gráfico 2. Volatilidad Histórica de la acción de Apple.



Fuente: elaboración propia

El Gráfico 2 representa la volatilidad histórica de la acción de Apple. Para poder analizar el gráfico debemos conocer que la línea negra indica el cierre de la cotización de la acción en el día, la línea azul representa la volatilidad a un mes, la línea naranja volatilidad a dos meses y la verde volatilidad a tres meses (puesto que para un mes necesito menos datos que para dos o tres meses, la línea empieza más a la izquierda en el gráfico). El eje izquierdo (*close*) representa unidades en dólares (línea negra) y el eje derecho representa la volatilidad anualizada en porcentaje para las líneas azul, naranja y verde. Si miramos, por ejemplo, la bajada en el precio de la acción en el mes de febrero (acción baja desde aproximadamente 180 dólares hasta 155), esto provoca un salto en la volatilidad de un mes enorme. También produce un salto en la volatilidad a dos y tres meses pero no tan significativo. Esto se debe a que la volatilidad a un mes cuenta con menos días, y por ello, un impacto muy grande se verá más reflejado en la volatilidad a un mes que en la volatilidad a dos y tres meses.

Una vez a la calculado la volatilidad histórica, identifica aquellos que se encuentren en un mínimo:

```
date_2M = datetime.now().date() + relativedelta(months=2)
### Identify stocks at historical volatility minimum @ 1M
if (df.iloc[-1].name in
```

```

df[(df['1M_Vol'] <= df_minimum(df['1M_Vol'],
gv.period_1M))].index.tolist():
    stock_selected.append([contract.symbol, datetime.now().date(), '1M',
                           contract, df['Close'].iloc[-1],
                           df['1M_Vol'].iloc[-1], df['2M_Vol'].iloc[-1],
                           df['3M_Vol'].iloc[-1], date_2M])

```

y posteriormente los selecciona.

8.4 Módulo II

El módulo segundo lo que hace es, para aquellos valores que han sido seleccionados con mínimos de volatilidad mensuales, bimensuales o trimestrales, establece el combo, es decir, la opción de compra, la opción de venta, el precio de ejercicio y la fecha de vencimiento de ambas. En primer lugar, define el contrato de opción, y selecciona la fecha de vencimiento y los precios de ejercicio.

```

### Select maturity date
    gv.taskDone.clear()
    optOrderId = app.nextOrderId()
    app.reqContractDetails(optOrderId, opt_contract)
    gv.taskDone.wait()
    gv.optMaturityDateList.sort()
    dateList = intersect(gv.optMaturityDateList, opt_exp_cal)
    dateList.sort()
    optMaturityDate = min(x for x in dateList if x >= minMaturityDate)
    opt_contract.lastTradeDateOrContractMonth = optMaturityDate

### Select strikes
    gv.optStrikesList = []
    gv.taskDone.clear()
    optOrderId = app.nextOrderId()
    app.reqContractDetails(optOrderId, opt_contract)
    gv.taskDone.wait()
    gv.optStrikesList.sort()
    strike_low = max(x for x in gv.optStrikesList if x <= stockPrice)

```

```

strike_high = min(x for x in gv.optStrikesList if x >= stockPrice)
if strike_low == strike_high:
    comboDescription = 'Straddle'
    putStrike = strike_low
    callStrike = strike_high
elif ((stockPrice-strike_low) / (strike_high-strike_low)) <= 0.25:
    comboDescription = 'Straddle'
    putStrike = strike_low
    callStrike = strike_low
elif ((strike_high-stockPrice) / (strike_high-strike_low)) <= 0.25:
    comboDescription = 'Straddle'
    putStrike = strike_high
    callStrike = strike_high
else:
    comboDescription = 'Strangle'
    putStrike = strike_low
    callStrike = strike_high

```

A continuación, construye el contrato de opción de venta y el contrato de opción de compra que van a formar el combo junto con la fecha de vencimiento y el precio de ejercicio. Estos combos son seleccionados en una lista.

8.5 Módulo III

El tercer módulo lo que hace es verificar el precio de mercado para el combo seleccionado y, en función de eso y el periodo de vencimiento determina la cantidad que se compra y el precio de salida para el beneficio objetivo. Para ello, solicita la primera fecha para la que existen cotizaciones del *put contract* y del *call contract*. A continuación solicita tres meses de cotizaciones para el combo (*put + call*) e identifica los precios mínimos en tres meses, dos meses y un mes.

```

### Find 3 months, 2 months and 1 month minimum for combo
date_2M = comboPrices.iloc[-1].name - relativedelta(months=2)
date_1M = comboPrices.iloc[-1].name - relativedelta(months=1)
minComboPrice_3M = min(comboPrices['Close'])

```

```
minComboPrice_2M = min(comboPrices['Close'].loc[date_2M:])
```

```
minComboPrice_1M = min(comboPrices['Close'].loc[date_1M:])
```

Una vez ha identificado los precios mínimos, calcula la cantidad de contratos a comprar en función del tipo de mínimo y de la última cotización del combo.

```
### Calculate quantity, profit target price and stop loss price
```

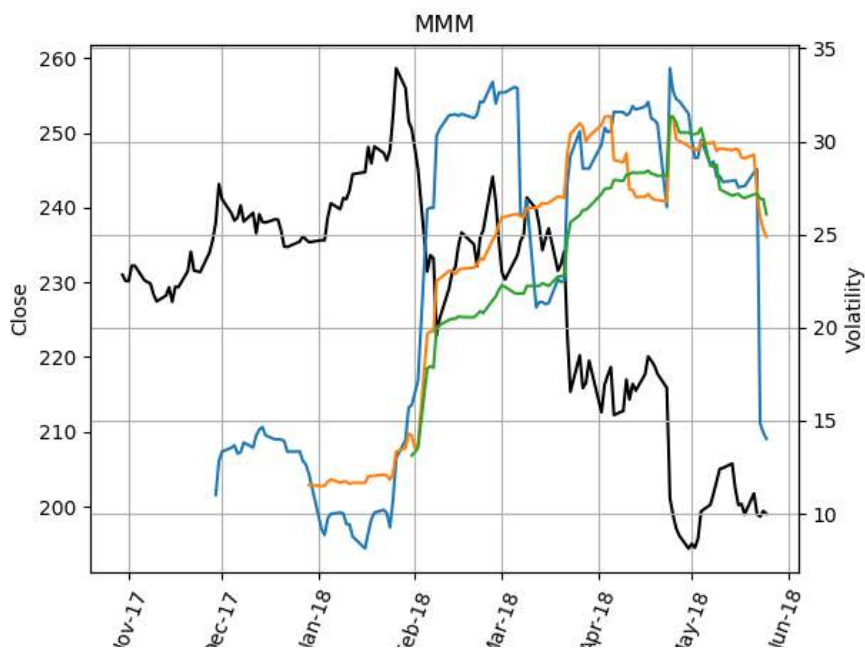
```
comboPeriod = combo[2]
```

```
comboQuantity = round(entry_parameters[comboPeriod][ep.Amount] /
```

```
(comboPrices['Close'].iloc[-1] * 100))
```

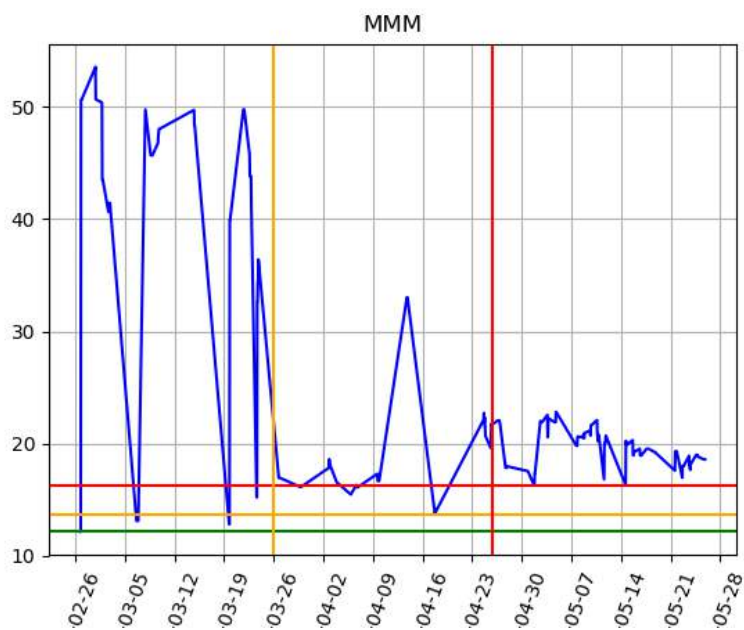
Por último realiza una representación gráfica de la cotización del combo (*put + call*) y de los mínimos a tres meses, dos meses y un mes.

Gráfico 3. Volatilidad Histórica de la acción de 3M.



Fuente: elaboración propia

Gráfico 4. Precio del combo de la acción de 3M.



Fuente: elaboración propia

El **gráfico 3** es muy similar al **gráfico 1** presentado en el módulo I. La línea negra indica el cierre de la cotización de la acción en el día, la línea azul representa la

volatilidad a un mes, la línea naranja volatilidad a dos meses y la verde volatilidad a tres meses (puesto que para un mes necesito menos datos que para dos o tres meses, la línea empieza más a la izquierda en el gráfico).

El **gráfico 4** presenta el precio del combo de la acción de 3M en este caso (línea azul). El eje de fechas (eje horizontal) marca con la línea roja un mes atrás y con la línea naranja dos meses atrás. En el eje de precios (eje vertical), la línea roja marca el mínimo de un mes, la línea naranja marca el mínimo a dos meses y la línea verde marca el mínimo a tres meses. De esta manera, podemos ver si el precio del combo está cerca de los mínimos a uno, dos o/y tres meses.

8.6 Módulo IV

El cuarto módulo solicita la orden de compra del combo y monitoriza su ejecución. Para ello, para todos los contratos y cantidades seleccionados en el módulo anterior envía una orden de compra a precio actual de mercado (*MARKET*) y válida durante la sesión del día (*DAY*).

El programa, una vez que ha ejecutado los módulos anteriores para cada uno de los valores, ejecuta dos módulos cuya función es comprobar qué órdenes de compra se han completado y enviar para las mismas las órdenes de venta condicionadas para conseguir el beneficio objetivo.

8.7 Módulo V

Este módulo, comprueba tres cosas para las órdenes de compra emitidas. En primer lugar, comprueba si se ha ejecutado la cantidad total solicitada. Si es así, etiqueta la orden como COMPLETADA.

```
if gv.order_status[ordId][2] == 0.0:
```

```
    combo_entry_order_submitted[0].extend(['COMPLETED - BUY MKT',
```

```
                                             gv.order_status[ordId][1],
```

```
                                             gv.order_status[ordId][2],
```

```
                                             gv.order_status[ordId][3]))
```

```
    combo_entry_order_completed.append(combo_entry_order_submitted[0])
```

```
    with open('file_combo_entry_order_completed', 'wb') as f:
```

```

        pickle.dump(combo_entry_order_completed, f)
    f.close()
    combo_entry_order_submitted.remove(combo_entry_order_submitted[0])
    with open('file_combo_entry_order_submitted', 'wb') as f:
        pickle.dump(combo_entry_order_submitted, f)
    f.close()
elif gv.order_status[ordId][2] != 0.0:

```

En segundo lugar, si la orden se ha ejecutado solo parcialmente, etiqueta la orden como PARCIAL y cancela la cantidad restante.

```

if gv.order_status[ordId][1] != 0.0:
    app.cancelOrder(ordId)
    combo_entry_order_submitted[0].extend(['PARTIAL - BUY MKT',
                                           gv.order_status[ordId][1],
                                           gv.order_status[ordId][2],
                                           gv.order_status[ordId][3]])
    combo_entry_order_completed.append(combo_entry_order_submitted[0])
    with open('file_combo_entry_order_completed', 'wb') as f:
        pickle.dump(combo_entry_order_completed, f)
    f.close()
    combo_entry_order_submitted.remove(combo_entry_order_submitted[0])
    with open('file_combo_entry_order_submitted', 'wb') as f:
        pickle.dump(combo_entry_order_submitted, f)
    f.close()

```

Por último, si la orden no se ha ejecutado, la cancela y la etiqueta como CANCELADA.

```

else:
    app.cancelOrder(ordId)
    combo_entry_order_submitted[0].extend(['CANCELLED - BUY MKT',
                                           gv.order_status[ordId][1],
                                           gv.order_status[ordId][2],

```

```

        gv.order_status[ordId][3]])
    combo_entry_order_cancelled.append(combo_entry_order_submitted[0])
    with open('file_combo_entry_order_cancelled', 'wb') as f:
        pickle.dump(combo_entry_order_cancelled, f)
    f.close()
    combo_entry_order_submitted.remove(combo_entry_order_submitted[0])
    with open('file_combo_entry_order_submitted', 'wb') as f:
        pickle.dump(combo_entry_order_submitted, f)
    f.close()

```

8.8 Módulo VI

Para cada orden etiquetada como COMPLETADA o PARCIAL, calcula el precio de venta en función del precio al que se ha realizado la compra y el objetivo de beneficio determinado. A continuación envía una orden de venta condicionada (*LIMIT*) al precio de venta para obtener el objetivo de beneficio e ilimitada en el tiempo (*GTC – Good Till Cancelled*).

```

### Recover entry order contract, quantity and average fill price
    contract = combo[cp.ComboContract]
    periodMin = combo[cp.PeriodMin]
    quantity = combo[cp.Filled]
    avgFillPrice = combo[cp.AverageFillPrice]

### Calculate stop loss and profit target
    pt_price = round(avgFillPrice * (1.0+entry_parameters[periodMin][ep.Target]), 2)

### Send PT order
    pt_order = app.LimitOrder("SELL", quantity, pt_price)
    pt_order.tif = 'GTC'
    app.simplePlaceOid = app.nextOrderId()
    combo_entry_order_completed[0].extend([app.simplePlaceOid])
    app.placeOrder(app.simplePlaceOid, contract, pt_order)

```



```
combo_entry_order_completed[0].extend(['SUBMITTED - SELL LMT', quantity,  
pt_price])
```

Para comunicar los datos entre los distintos módulos (por ejemplo los valores en un mínimo mensual de volatilidad, los contratos *put* y *call* seleccionados, las órdenes de compra enviadas y las órdenes de compra realizadas) se utiliza un conjunto de archivos en los que la información de salida de un módulo constituye la información de entrada del siguiente.

Es un modo bastante primitivo que, aunque garantiza la persistencia de los datos entre sesiones en caso de errores, se puede mejorar definiendo una nueva "clase" que incluya toda la información que se va anotando en los ficheros. Esto además mejoraría la legibilidad del código y es la próxima mejora planificada.

9. MÉTODOS DE VALIDACIÓN DE ESTRATEGIAS

Una vez la estrategia está lista para operar, es importante comprobarla a través de distintos métodos de validación antes de operar con dinero real en el mercado. Debido a que los mercados son impredecibles, puede que una estrategia funcione en un momento determinado y sin embargo, puede ser que más tarde no sea una estrategia viable. Según Davey (2014) Los métodos más conocidos para validar estrategias son los siguientes: *Backtest*, *Paper Trading* y *Walkforward*.

9.1 Método de *Backtesting*

El *backtest* es uno de los métodos de validación de estrategias más conocidos y a menudo se utiliza para comparar y evaluar diferentes modelos estadísticos. Lo que el *backtest* pretende, es comprobar la viabilidad de nuestro sistema en el pasado. Así, el *backtest* se lleva a cabo mediante la reconstrucción, con datos históricos de transacciones que habrían ocurrido en el pasado empleando reglas definidas por una estrategia determinada. De esta manera, cuantas más variables a optimizar, más *backtests* habrá que realizar para comprobar el sistema. Este tipo de método se basa en la idea de que si el sistema ha funcionado con datos en el pasado, también lo hará en el futuro puesto que la historia se repite. Como consecuencia, los resultados de un *backtest* dependen en gran medida de las condiciones de mercado que hayan tenido lugar durante ese periodo de tiempo y, por tanto, es mejor realizar un *backtest* en un marco de tiempo largo para que se puedan apreciar diferentes condiciones de mercado (Würtz, Chalabi et.al, 2009).

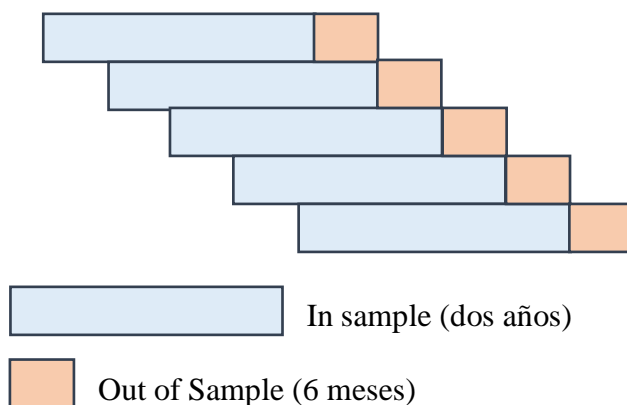
Con este tipo de test, el inversor puede hacerse a la idea de cuál será su rentabilidad o su máximo *drawdown* en el periodo de inversión (*drawdown* se refiere a la caída que ha sufrido el sistema entre un máximo y el mínimo que se produce después).

Además, es importante optimizar el sistema. Esto quiere decir que hay que tratar de probar distintas combinaciones de los parámetros de la estrategia para encontrar el que más se adecúe a nuestros objetivos prestando atención a la posible sobre optimización. A veces, al incluir determinados filtros o restricciones nos queda un sistema perfecto, que nos dice lo que queremos oír pero que al probar en el mercado es un fracaso puesto que solo funcionaba en el pasado. Por ello, hay que tener cuidado y es mejor seleccionar un menor número de parámetros para dejar más margen al modelo.

Por último, hay que tener en cuenta que un resultado positivo en un *backtest* no significa que el modelo vaya a ser válido para siempre, puesto que los mercados están continuamente cambiando y es impredecible pensar que la historia vaya a repetirse.

9.2 Método de *Walk Forward*.

El método de *walk forward* consiste en realizar sucesivas optimizaciones en sucesivos periodos. Para comprender mejor cómo funciona, observamos un ejemplo. Se optimiza un sistema durante dos años de historial de datos (*in sample*) y se aplica a los datos no vistos de los siguientes seis meses (*out of sample*). A continuación, la ventana de optimización se mueve en seis meses y se realiza una nueva optimización para esos nuevos seis meses de datos. De nuevo, la ventana de optimización de datos se mueve en seis meses y se realiza una nueva optimización para esos nuevos seis meses de datos. Este proceso se repite cuantas veces sea necesario hasta llegar al periodo más reciente (Jaekle y Tomasini, 2011).



Fuente: elaboración propia basada en Jaekle y Tomasini (2011).

9.3 Método de *Paper Trading*

El método de *Paper trading* es un método que permite probar una estrategia de trading en el mercado real sin arriesgar capital. Esto es, muchos inversores que utilizan un bróker online también pueden acceder a crearse una cuenta de *paper trading* para simular las estrategias de forma gratuita. El *paper trading* tiene, como ventajas, que no se pone en riesgo dinero real y que puede servir de aprendizaje para nuevos *traders*. Es decir, una persona que no está tan familiarizada con esta forma de invertir podrá ir

aprendiendo y coger experiencia antes de invertir dinero real. Sin embargo, al no arriesgar nada de dinero, los *traders* pierden la tensión que es tomar decisiones para minimizar riesgo con el fin de perder la menor cantidad de dinero posible y pueden tomar decisiones muy arriesgadas que no tomarían en caso de operar con su dinero. *Paper trading* es una buena forma de validación de estrategias siempre que se sepa utilizar correctamente.

En el caso de nuestra estrategia, la forma de comprobar su viabilidad es a través del *paper trading*. Para un subyacente, hay muchas opciones debido los distintos vencimientos y precios de ejercicio y en consecuencia, obtener las cotizaciones para todas es más complicado. Es por ello que realizar un *backtest* o un *walkforward* es demasiado complejo y por eso el *paper trading* resulta la mejor opción para validar la estrategia de *trading* implementada.

En la **tabla 2**, se presentan los resultados de *paper trading* de la estrategia descrita entre el 30 de enero y el 9 de marzo.

Tabla 2. Resumen de ejecución de la estrategia entre el 30 de Enero y 9 de Marzo.

DEMO ACCOUNT - SIMULATED TRADING

Interactive Brokers

Realized Summary

January 30, 2018 - March 9, 2018

Symbol	Realized				Unrealized			
	S/T Profit	S/T Loss	Total	Difference	S/T Profit	S/T Loss	Total	Difference
Equity and Index Options								
AAPL 15JUN18 165.0 P	1.877,95	0	1.877,95		0	0	0	
AAPL 15JUN18 170.0 C	0	-1.091,98	-1.091,98	785,97	0	0	0	
ABBV 18MAY18 110.0 P	0	0	0		0	-564,66	-564,66	
ABBV 18MAY18 115.0 C	0	0	0		446,66	0	446,66	-118
DIS 20APR18 110.0 C	0	-379,97	-379,97		0	0	0	
DIS 20APR18 110.0 P	548,02	0	548,02	168,05	0	0	0	
GE 15JUN18 14.0 P	0	0	0		0	-357,34	-357,34	
GE 15JUN18 15.0 C	0	0	0		114,56	0	114,56	-242,78
JPM 15JUN18 115.0 C	0	-544,57	-544,57		0	0	0	
JPM 15JUN18 115.0 P	1.081,09	0	1.081,09	536,52	0	0	0	
MDT 17AUG18 80.0 C	0	0	0		604,41	0	604,41	
MDT 17AUG18 80.0 P	0	0	0		0	-548,35	-548,35	56,06
MRK 15JUN18 57.5 P	1.945,26	0	1.945,26		0	0	0	
MRK 15JUN18 60.0 C	0	-1.155,67	-1.155,67	789,59	0	0	0	
TXN 20JUL18 105.0 P	0	0	0		0	-465,24	-465,24	
TXN 20JUL18 110.0 C	0	0	0		390,76	0	390,76	-74,48
UTX 18MAY18 135.0 P	1.076,11	0	1.076,11		0	0	0	
UTX 18MAY18 140.0 C	0	-693,85	-693,85	382,26	0	0	0	
VZ 15JUN18 52.5 P	1.087,69	0	1.087,69		0	0	0	
VZ 15JUN18 55.0 C	0	-644,27	-644,27	443,42	0	0	0	
WMT 21SEP18 105.0 C	0	-383,74	-383,74		0	0	0	
WMT 21SEP18 105.0 P	1.034,08	0	1.034,08	650,34	0	0	0	
Total	8.650,19	-4.894,05	3.756,14		1.556,40	-1.935,58	-379,19	
Total (All Assets)	8.650,19	-4.894,05	3.756,14		1.556,40	-1.935,58	-379,19	

Fuente: elaboración propia

Antes de entender los resultados es preciso conocer qué información nos aporta el gráfico. La primera columna denominada *Symbol* nos indica el valor al que se refiere. Por ejemplo *AAPL* hace referencia a *Apple* y *WMT* a *Walmart*. Si el valor termina con el símbolo P, se trata de la opción *put* y si lo hace con C, se trata de la opción *call*. Por otro lado, existen dos grandes columnas diferenciadas, *Realized*, que se refiere a aquellos valores que ya han vendido, y *Unrealized* que se refiere a aquellos valores que todavía no han vendido, que podrán hacerlo en beneficio o pérdida. Dentro de cada una

de estas columnas, existen otras cuatro columnas diferenciadas. La primera indica el beneficio (de la opción *call* o *put*), la segunda indica la pérdida (de la opción *call* o *put*), la tercera muestra el total y la última muestra la diferencia entre el beneficio y la pérdida que se han producido.

Los resultados muestran que se trata de una estrategia generalmente beneficiosa. En el tiempo mostrado, se produce un beneficio total de 3.756,14 euros frente a 379,19 de pérdidas. Además si nos fijamos, la estrategia en este periodo de tiempo ha ganado con las opciones *put*, es decir cuando se ha producido una bajada lo suficientemente significativa del valor en el mercado. Por ello, aquí se demuestra que no importa en qué dirección se mueva el mercado, puede generarse un beneficio. En este caso con una bajada en el valor de mercado se ha producido un beneficio. Más adelante, se harán conclusiones más específicas acerca de cómo podría mejorarse el programa.

10. CONCLUSIONES

El objetivo principal de este trabajo ha sido diseñar, implementar y validar una estrategia propia de *trading* algorítmico basada en la volatilidad de los precios de las opciones mediante el uso del programa Python.

El *trading* algorítmico está creciendo tanto en instituciones como en inversores individuales. Hoy en día, disponemos de gran cantidad de información en internet y libros que nos permiten poder desarrollar una estrategia de *trading*. Para ello, es importante tener en cuenta una serie de cosas. En primer lugar, es importante disponer del tiempo y la voluntad necesarios para dedicar el tiempo que se necesita en desarrollar y probar una estrategia de *trading*. También es importante establecer objetivos claros y adecuados. Para ello, estos objetivos deben ser específicos, medibles, alcanzables, relevantes y limitados en el tiempo. Una vez realizada la investigación necesaria, es importante determinar en qué mercado operar y con qué activo, el tipo de sistema a elegir, el marco temporal de la estrategia, cómo entrar y salir del mercado y por último, cómo programar la estrategia. Todos y cada uno de estos elementos son importantes y necesarios antes de comenzar el desarrollo del sistema. Además, para desarrollar una estrategia de *trading*, es muy importante disponer de un bróker (que hoy en día puede ser online), una buena red de comunicaciones y una infraestructura informática adecuada. Por último, una vez desarrollado el sistema de *trading* también debemos comprobar su viabilidad y probabilidad de éxito antes de probarlo en el mercado real. Esta comprobación puede realizarse a través de cualquier tipo de método de validación entre los que se incluyen: el *backtest*, *walk forward* y *paper trading*.

En el caso de la estrategia propia implementada, atendiendo a los resultados obtenidos, podemos afirmar que se trata de una estrategia con una base muy buena y con gran potencial, pero no perfecta. Se trata de un algoritmo sencillo sin ningún tipo de filtros que muestra operaciones tanto positivas como negativas. Concretamente, se trata de un sistema que solo exige como condición que el subyacente cumpla el mínimo de volatilidad y a pesar de ello muestra operaciones beneficiosas. En este caso, para asegurar buenas compras de opciones (*call* y *put*), asegurar que se escogen bien los mínimos y asegurar una determinada rentabilidad se podrían implementar una serie de filtros que ayudarían a obtener mejores resultados. Así, con un periodo de evaluación mayor y que permita también probar filtros como: que el precio del combo de opciones esté también en valores mínimos, que se establezcan rangos de precios para el

subyacente o para los combos, se podría lograr una estrategia más segura. También se puede recurrir al ajuste de parámetros, como por ejemplo el objetivo de beneficio. Si se ajusta el beneficio, por ejemplo de un 10% a un 7%, puede suceder que todas las operaciones ganen pero en menor medida y en ocasiones es mejor opción ganar menos pero en todas las operaciones.

En definitiva, sería muy interesante profundizar más en el modelo de *trading* algorítmico para conseguir un sistema beneficioso en su totalidad además de investigar otro tipo de sistemas que puedan resultar en grandes beneficios para un riesgo determinado.

11. BIBLIOGRAFÍA

- Bandy, H. B. (2011). *Quantitative trading systems: Practical methods for design, testing, and validation*. Sioux Falls: Blue Owl Press.
- Beagles, W. A. (2009). *Equity and index options explained*. Chichester: Wiley.
- Bernstein, W. J. (2009). *A splendid exchange: How trade shaped the world*. London: Distributed by Group West.
- Black, F & Scholes, M (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, 81(3), 637-654.
- Buckey, R.P. (2016). Reconceptualizing the Regulation of Global Finance. *Oxford Journal of Legal Studies*, 36(2), 242-271.
- Chan, E. P. (2017). *Machine trading: Deploying computer algorithms to conquer the markets*. Hoboken, NJ: John Wiley & Sons.
- Cofnas, A. (2016). *Trading binary options: Strategies and tactics*. Hoboken, NJ: John Wiley & Sons.
- Cox, J.C, Ross, S.A, Rubinstein, M (1979). Option Pricing: A Simplified Approach. *Journal of Financial Economics*, 7(3), 229-263.
- Davey, K. J. (2014). *Building winning algorithmic trading systems: A traders journey from data mining to Monte Carlo simulation to live trading*. Hoboken, NJ: Wiley.
- NASDAQ Financial Glossary. (n.d.). Definition of "Trading strategy " - Recuperado el 15 de Mayo, 2018, de <https://www.nasdaq.com/investing/glossary/t/trading-strategy>
- Hoffman, A & Williams, M. (2001). *Fundamentals of the options market*. New York: McGraw-Hill.

Integrated Investment Management. (n.d.). Recuperado el 19 de Mayo, 2018, de <https://www.interactivebrokers.com/>

Jaekle, U & Tomasini, E. (2011). *Trading Systems. A new approach to system development and portfolio optimization*. Great Britain: Harriman House,

Kissell, R. (2014). *Science of Algorithmic Trading and Portfolio Management*. Elsevier.

Olmstead, W. E. (2013). *Options for the beginner and beyond: Unlock the opportunities and minimize the risks*. Upper Saddle River, NJ: FT Press.

Online Broker | TD Ameritrade. (n.d.). Online Stock Trading, Investing, Recuperado el 20 de Mayo, 2018, de

<https://www.bing.com/cr?IG=67AB5091AA344F3E85134F150E7ECFB6&CID=0B830A90707E68D120BF016E7183696F&rd=1&h=7cc2SNP9TKWPIO2IM-NUrQ0B1ymd5ED1dKrLcgsHHuQ&v=1&r=https://www.tdameritrade.com/home.page&p=DevEx.LB.1,5544.1>

Rascon, P.J. (2007). *Manual Práctico de Opciones Financieras. Una Forma Equilibrada de Invertir en Bolsa*. LibrosEnRed

Würt, D., Chalabi, Y. et.al. (2009). *Portfolio Optimization with R/Rmetrics*. Zurich: Finance Online GmbH. Recuperado de https://books.google.es/books?id=keDLpq86DU4C&printsec=frontcover&hl=es&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false

12. ANEXOS

Anexo I: Líneas de código del módulo principal

```
from datetime import datetime
from ibapi import wrapper
from ibapi.utils import iswrapper
from ibapi.client import EClient
from ibapi.contract import Contract, ContractDetails
from ibapi.order import Order
from ibapi.order_state import OrderState
from ibapi.execution import Execution
from ibapi.tag_value import TagValue
from ibapi.common import TickerId, TickAttrib, OrderId
from ibapi.ticktype import TickType, TickTypeEnum
import threading
import pickle

import gv
from gv import stock_list as stock_list

class myThread (threading.Thread):
    def __init__(self, app):
        threading.Thread.__init__(self)
        self.app = app
    def run(self):
        self.app.run()

class TestApp(wrapper.EWrapper, EClient):
    def __init__(self):
        wrapper.EWrapper.__init__(self)
        EClient.__init__(self, wrapper=self)
        self.started = False
        self.nKeybInt = 0
        self.nextValidOrderId = None
    @iswrapper
    def nextValidId(self, orderId: int):
        super().nextValidId(orderId)
        self.nextValidOrderId = orderId
        gv.taskDone.set()
    @iswrapper
    def error(self, reqId: TickerId, errorCode: int, errorString: str):
        super().error(reqId, errorCode, errorString)
        #print("Error. Id: ", reqId, " Code: ", errorCode, " Msg: ", errorString)
    @iswrapper
    def historicalData(self, reqId: TickerId, date: str, open: float, high: float,
                       low: float, close: float, volume: int, barCount: int,
                       WAP: float, hasGaps: int):
```

```

        super().historicalData(reqId, date, open, high, low, close, volume,
                               barCount, WAP, hasGaps)
        gv.rows_list.append([date, open, high, low, close, volume, barCount, WAP,
hasGaps])
    @iswrapper
    def historicalDataEnd(self, reqId: int, start: str, end: str):
        super().historicalDataEnd(reqId, start, end)
        gv.taskDone.set()
    def nextOrderId(self):
        oid = self.nextValidOrderId
        self.nextValidOrderId += 1
        return oid

```

```
def main():
```

```

import historical_volatility
import option_chain
import combo_entry_conditions
import entry_orders_from_contract_file
import entry_orders_completed_from_contract_file
import exit_orders_from_contract_file

```

```

### Connection to IB API ###
app = TestApp()
app.connect(gv.host, gv.port, gv.clientId)
print("serverVersion:%s connectionTime:%s" %
      (app.serverVersion(), app.twsConnectionTime()))
thread = myThread(app)
gv.taskDone.clear()
thread.start()
gv.taskDone.wait()

```

```

### Main task ###
combo_entry_order_submitted = []
with open('file_combo_entry_order_submitted', 'wb') as f:
    pickle.dump(combo_entry_order_submitted, f)
f.close()

```

```

for stock in stock_list:
    historical_volatility.main(app, stock)
    option_chain.main(app)
    combo_entry_conditions.main(app)
    entry_orders_from_contract_file.main(app)

```

```

entry_orders_completed_from_contract_file.main(app)
exit_orders_from_contract_file.main(app)
### End of Main task ###

```

```
### Disconnect from IB API ###  
app.done = True
```

```
if __name__ == "__main__":  
    main()
```

Anexo II: Lista de valores para la estrategia

```
stock_list = (('AAPL', 'STK','SMART','NASDAQ','USD'),  
              ('AXP', 'STK','SMART','NYSE', 'USD'),  
              ('BA', 'STK','SMART','NYSE', 'USD'),  
              ('CAT', 'STK','SMART','NYSE', 'USD'),  
              ('CSCO', 'STK','SMART','NASDAQ','USD'),  
              ('CVX', 'STK','SMART','NYSE', 'USD'),  
              ('DIS', 'STK','SMART','NYSE', 'USD'),  
              ('GE', 'STK','SMART','NYSE', 'USD'),  
              ('GS', 'STK','SMART','NYSE', 'USD'),  
              ('HD', 'STK','SMART','NYSE', 'USD'),  
              ('IBM', 'STK','SMART','NYSE', 'USD'),  
              ('INTC', 'STK','SMART','NASDAQ','USD'),  
              ('JNJ', 'STK','SMART','NYSE', 'USD'),  
              ('JPM', 'STK','SMART','NYSE', 'USD'),  
              ('KO', 'STK','SMART','NYSE', 'USD'),  
              ('MCD', 'STK','SMART','NYSE', 'USD'),  
              ('MMM', 'STK','SMART','NYSE', 'USD'),  
              ('MRK', 'STK','SMART','NYSE', 'USD'),  
              ('MSFT', 'STK','SMART','NASDAQ','USD'),  
              ('NKE', 'STK','SMART','NYSE', 'USD'),  
              ('PFE', 'STK','SMART','NYSE', 'USD'),  
              ('PG', 'STK','SMART','NYSE', 'USD'),  
              ('TRV', 'STK','SMART','NYSE', 'USD'),  
              ('UNH', 'STK','SMART','NYSE', 'USD'),  
              ('UTX', 'STK','SMART','NYSE', 'USD'),  
              ('V', 'STK','SMART','NYSE', 'USD'),  
              ('VZ', 'STK','SMART','NYSE', 'USD'),  
              ('WMT', 'STK','SMART','NYSE', 'USD'),  
              ('XOM', 'STK','SMART','NYSE', 'USD'),  
              ### Other Financial  
              ('BAC', 'STK','SMART','NYSE', 'USD'),  
              ('WFC', 'STK','SMART','NYSE', 'USD'),  
              ('HSBC', 'STK','SMART','NYSE', 'USD'),  
              ('C', 'STK','SMART','NYSE', 'USD'),  
              ('MA', 'STK','SMART','NYSE', 'USD'),  
              #('RY', 'STK','SMART','NYSE', 'USD'),  
              #('LFC', 'STK','SMART','NYSE', 'USD'),  
              ('TD', 'STK','SMART','NYSE', 'USD'),  
              ('SAN', 'STK','SMART','NYSE', 'USD'),
```

('MUGF', 'STK','SMART','NYSE', 'USD'),
 ### Other Healthcare
 ('NVS', 'STK','SMART','NYSE', 'USD'),
 ('ABBV', 'STK','SMART','NYSE', 'USD'),
 ('NVO', 'STK','SMART','NYSE', 'USD'),
 ('AMGN', 'STK','SMART','NASDAQ','USD'),
 ('MDT', 'STK','SMART','NYSE', 'USD'),
 ('SNY', 'STK','SMART','NYSE', 'USD'),
 ('ABT', 'STK','SMART','NYSE', 'USD'),
 ('BMY', 'STK','SMART','NYSE', 'USD'),
 ### Other Services
 ('AMZN', 'STK','SMART','NASDAQ','USD'),
 ('BABA', 'STK','SMART','NYSE', 'USD'),
 ('CMCSA', 'STK','SMART','NASDAQ','USD'),
 ('UPS', 'STK','SMART','NYSE', 'USD'),
 ('UNP', 'STK','SMART','NYSE', 'USD'),
 ('CHTR', 'STK','SMART','NYSE', 'USD'),
 ### Other Industrial Goods
 ('HON', 'STK','SMART','NYSE', 'USD'),
 ### Other Basic Materials
 ('RDS B', 'STK','SMART','NYSE', 'USD'),
 ('PTR', 'STK','SMART','NYSE', 'USD'),
 ('BHP', 'STK','SMART','NYSE', 'USD'),
 ('BP', 'STK','SMART','NYSE', 'USD'),
 ('TOT', 'STK','SMART','NYSE', 'USD'),
 ('BBL', 'STK','SMART','NYSE', 'USD'),
 ('SNP', 'STK','SMART','NYSE', 'USD'),
 ('SLB', 'STK','SMART','NYSE', 'USD'),
 ### Other Consumer Goods
 ('BUD', 'STK','SMART','NYSE', 'USD'),
 ('TM', 'STK','SMART','NYSE', 'USD'),
 ('PEP', 'STK','SMART','NASDAQ','USD'),
 ('PM', 'STK','SMART','NYSE', 'USD'),
 ('UN', 'STK','SMART','NYSE', 'USD'),
 ('BTI', 'STK','SMART','NYSE', 'USD'),
 ('MO', 'STK','SMART','NYSE', 'USD'),
 ### Other Technology
 ('GOOG', 'STK','SMART','NASDAQ','USD'),
 ('FB', 'STK','SMART','NASDAQ','USD'),
 ('T', 'STK','SMART','NYSE', 'USD'),
 ('CHL', 'STK','SMART','NYSE', 'USD'),
 ('TSM', 'STK','SMART','NYSE', 'USD'),
 ('ORCL', 'STK','SMART','NYSE', 'USD'),
 ('SAP', 'STK','SMART','NYSE', 'USD'),
 ('NVDA', 'STK','SMART','NASDAQ','USD'),
 ('AVGO', 'STK','SMART','NASDAQ','USD'),
 ('TXN', 'STK','SMART','NASDAQ','USD'),)