



Escola d'Enginyeria de Telecomunicació i  
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TREBALL FINAL DE GRAU

**TÍTOL DEL TFG: Análisis de algoritmos financieros**

**TITULACIÓ: Grau en Enginyeria d'Aeroports**

**AUTOR: Iker Larrauri González**

**DIRECTOR: Toni Oller Arcas**

**DATA: 21 de octubre del 2021**



**Título:** Análisis de algoritmos financieros

**Autor:** Iker Larrauri González

**Director:** Toni Oller Arcas

**Data:** 21 de octubre del 2021

## Resumen

Debido los avances tecnológicos de los últimos años, han surgido nuevas formas de operar en los mercados financieros. En este Trabajo de Fin de Grado se le da especialmente relevancia a una de ellas, el trading algorítmico, debido a la cual es posible automatizar estrategias de trading con el fin de que operen de manera autónoma en los mercados.

Para ello, primeramente, se muestra un estudio acerca del diseño, desarrollo y validación de una estrategia de inversión en los distintos mercados financieros. Se ha llevado a cabo una profunda investigación sobre el mundo de la bolsa y de los mercados, así como sobre la creación de un algoritmo que implemente dicha estrategia.

Una vez realizado ese primer estudio y diseño, se procede a la implementación de la estrategia mediante la creación de un algoritmo que sea capaz de analizar los mercados de forma ininterrumpida. El cual se someterá a un método de validación estricto con el fin de examinar los resultados obtenidos.

Del análisis e implementación de esta nueva forma de operar se puede contemplar el uso del trading algorítmico como una herramienta válida, tanto para operadores como instituciones, ofreciendo una gran serie de ventajas, como la eliminación de los factores emocionales o la reducción de los tiempos de toma de decisión. Sin embargo, la rentabilidad y viabilidad de un modelo algorítmico va supeditada a numerosos factores, por lo que no se podrá ofrecer ninguna garantía de éxito hasta la implementación en los mercados.

**Title:** Financial algorithm analysis

**Author:** Iker Larrauri González

**Director:** Toni Oller Arcas

**Date:** October 20 rd 2021

## Overview

Due to technological advances in recent years, new ways of operating in financial markets have emerged. In this Final Degree Project, one of them is given special relevance, algorithmic trading, due to which it is possible to automate trading strategies in order for them to operate autonomously in the markets.

To do this, first, a study about the design, development and validation of an investment strategy in the different financial markets is shown. In-depth research has been carried out on the world of the stock market and markets, as well as on the creation of an algorithm that implements this strategy.

Once this first study and design has been carried out, the strategy is implemented by creating an algorithm that is capable of analyzing the markets without interruption. Which will undergo a strict validation method in order to examine the obtained results.

From the analysis and implementation of this new way of operating, the use of algorithmic trading can be contemplated as a valid tool, both for operators and institutions, offering a great series of advantages, such as the elimination of emotional factors or the reduction of the decision making times. However, the profitability and viability of an algorithmic model is dependent on many factors, so no guarantee of success can be offered until it is implemented in the markets.

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>CAPÍTULO 1. PROYECTO Y OBJETIVOS .....</b>	<b>7</b>
1.1. Estado del arte.....	7
1.2. Contexto .....	9
1.3. Robots financieros .....	10
1.4. Objetivos del trabajo .....	11
<b>CAPÍTULO 2. ESTRATEGIA .....</b>	<b>12</b>
2.1. Diseño de la estrategia .....	12
<b>CAPÍTULO 3. ARQUITECTURA DE LA SOLUCIÓN .....</b>	<b>16</b>
3.1. Arquitectura .....	16
3.2. Flujo de la arquitectura .....	17
<b>CAPÍTULO 4. IMPLEMENTACIÓN .....</b>	<b>19</b>
4.1. Lenguaje de programación.....	19
4.2. Librerías.....	20
4.3. Entorno de trabajo.....	23
4.4. GitHub.....	24
4.5. Proyecto Backtesting .....	26
<b>CAPÍTULO 5. PLANIFICACIÓN .....</b>	<b>28</b>
<b>CAPÍTULO 6. CONCLUSIONES .....</b>	<b>30</b>
6.1. Sigüientes pasos .....	31
6.2. Estudio medioambiental .....	32
<b>BIBLIOGRAFÍA .....</b>	<b>34</b>
<b>ANEXOS .....</b>	<b>36</b>

# INTRODUCCIÓN

La tecnología está cambiando la estructura de la sociedad y se están creando oportunidades nunca antes vistas. Los mercados financieros han evolucionado y una de las formas más conocidas de operar en ellos es el trading. Se trata de una forma de inversión para la que se necesitan de una experiencia y un conocimiento financiero importantes.

La propuesta de este trabajo es ofrecer un mecanismo que sea capaz de analizar el estado de la bolsa en todo momento. Para ello se creará una estrategia de inversión en bolsa, que sea capaz de identificar tendencias, que tome decisiones de inversión y generar un sistema rentable de inversión, respaldado por una serie de métodos de validación de estrategias.

En el primer capítulo, se puede encontrar un estado del arte, con el fin de analizar sistemas existentes parecidos, un contexto, una introducción a los robots financieros y unos objetivos sobre los que se trabajará todo el proyecto.

A continuación, en el segundo capítulo, se describe el diseño de la estrategia de trading que se ha realizado. Tratará de explicar de una forma dinámica y detallada los pasos que se han seguido para el diseño de la estrategia.

El capítulo tercero hablará sobre la arquitectura de la solución. Se ofrecerá una explicación del sistema y de las partes que lo conforman, así como de su funcionamiento.

Para continuar, en el cuarto capítulo, se tratará la implementación. Se explicará el lenguaje de programación escogido, las librerías que se han utilizado, el entorno de trabajo, GitHub y el proyecto de backtesting en cuestión.

El capítulo quinto trata sobre la planificación, en él se describen las tareas más importantes, así como la carga de trabajo de cada una de ellas.

Finalmente, se presentan las conclusiones del proyecto, los siguientes pasos que se pueden realizar, y un estudio medioambiental.

## CAPÍTULO 1. Proyecto y objetivos

En el primer capítulo, proyecto y objetivos, se realizará un “estado del arte”, posteriormente se definen el contexto y los robots financieros, finalmente se presentan los objetivos principales y secundarios del trabajo.

### 1.1. Estado del arte

En primer lugar, se realizará un “Estado del arte” como parte introductoria, donde se hará un repaso de proyectos que puedan llegar a tener sistemas y técnicas parecidos. El objetivo de este primer punto es ofrecer una introducción al proyecto y explicar las diferencias que pueda llegar a tener con otros proyectos similares, y lo que se quiere ofrecer con él.

Es importante destacar que muchas veces se confunde el termino de trading automático con el de trading algorítmico, sin embargo, hay algunas sutiles diferencias. Entre ellas se encuentra que el trading automatizado puede ser simplemente un software donde se automaticen las operaciones de trading, normalmente mediante el uso de robots que estudian los diferentes indicadores técnicos del mercado. Por otro lado, el trading algorítmico, no tiene que resultar obligatoriamente en la ejecución automatizada de las operaciones en el mercado. Es posible realizar trading algorítmico mediante ejecuciones manuales, cosa que no ocurre en el trading automático.

En la actualidad, cada día es más habitual encontrarnos con robots de trading que operan en los mercados financieros, gracias a los beneficios que pueden llegar a tener. Es por lo que a continuación, tras un minucioso estudio del mercado, se van a citar varios de los mejores robots de trading del 2021, así como plataformas que permitan realizar trading algorítmico.

- **Yuan Pay Group:** Se trata de un trader automatizado de origen chino, con una experiencia de más de 6 años en los diferentes mercados. Es un distribuidor autorizado de la moneda china E-Yuan, por lo que está respaldado por el gobierno chino. Tiene una serie de características que son difíciles de encontrar en otras plataformas. Es una plataforma de trading segura con un buen cifrado y con un buen servicio de atención al cliente. Es la única plataforma en la actualidad desde la que se permite comprar yuan digital, por lo que es un distribuidor exclusivo de la moneda. El proceso de registro es sencillo y tiene una fácil verificación de la cuenta. En cuanto a costes y comisiones se refiere, ya que es una plataforma relativamente nueva, no tiene costes ni comisiones.[1] La tasa de ganancia promedia es de 1:3 según su propia página.[2]

- **Bitcoin Champion:** Esta plataforma de comercio automático es impulsada por tecnologías de IA y ofrece servicios de comercio automático desde 2016. El robot permite operar mediante CDFs en Bitcoin, Bitcoin Cash, Ethereum y Ripple. Sus características son su elevada rentabilidad, algoritmos comerciales poderosos gracias a la implementación de IA y una aplicación móvil altamente intuitiva.[3]
- **RoboForex:** Se trata de un bróker que permite operar en el mercado de divisas (Forex) y otros instrumentos a través de CFDs. Se podría afirmar que es el líder en el sector de la automatización, en su plataforma es posible utilizar una amplia lista de robots de trading para Forex. Como se puede observar en su propia pagina web, cuentan con más de 800.000 clientes de 169 países diferentes y con una experiencia de más de 12 años. Las principales ventajas que destacar son la baja cantidad de deposito mínimo necesario, los spreads institucionales desde 0 pips en los pares principales, permiten retirar el capital sin comisión, y el apalancamiento que permiten en las inversiones es elevado.[4]

Es posible encontrar multitud de robots de trading automatizados en la actualidad, y cada día son más las personas y plataformas los que se dedican a desarrollarlos. Sin embargo, a la hora de indagar en la rentabilidad y en la seguridad de estos, se puede observar que existe un problema emergente en los últimos años. Aprovechando la falta de regulación, han aparecido en el Estado Español, negocios o tal y como los denomina la Comisión Nacional del Mercado de Valores (CNMV), “chiringuitos financieros”. Normalmente estos chiringuitos financieros según la CNMV, son “entidades y personas que no están autorizadas para actuar en los mercados de valores, como prestar servicios de inversión previstos en la LMV (recepción, transmisión y ejecución de órdenes de clientes, gestión de carteras o asesoramiento en materia de inversión), o realizar la actividad reservada a las instituciones de inversión colectiva”.[5]

Es por lo que los posibles esquemas piramidales y los fraudes, cada día son más comunes en todo lo concerniente a los robots de trading y las compañías que los ofrecen, por lo que las personas se pueden sentir confundidas. Tal y como indica Alberto Cáceres Sánchez, en un trabajo sobre la problemática de las estafas piramidales, “este tipo de negocios operan en un vacío legal, siendo los propios organismos reguladores de la materia los que advierten de su posible fraude piramidal, lo cual no evita que sigan incrementándose en nuestro país”.[6]

Es por todo ello, que en este proyecto, después de realizar una profunda investigación acerca de la estrategia de inversión que se debería de seguir, y automatizarla para ver sus resultados mediante un “backtesting”, se buscaría desmarcarse de todo este tipo de negocios y mostrar de forma clara y transparente los resultados de rentabilidad obtenidos y explicar la forma en la que trabajaría el robot y la estrategia que sigue.



## 1.2. Contexto

Es importante comprender en primer lugar qué es la bolsa, y como se opera en ella. La bolsa, definiéndolo sencillamente, es un mercado especializado donde se producen compra y ventas de acciones u obligaciones, por las que se fija un precio público, que se denomina cambio o cotización. Gracias a un mercado organizado, conocido como bolsa, se canalizan las ofertas y las demandas existentes de valores negociables e instrumentos financieros, donde se formaliza un precio de equilibrio o cotización de los títulos que se están negociando.[7]

En segundo lugar, hay que entender como se opera en el mundo de la bolsa y qué es una operación de bolsa. En grandes rasgos, se denomina operación de bolsa a las transacciones de valores admitidos a negociación en las bolsas de valores. Para que dichas operaciones se realicen, existirá la participación o una mediación de al menos un miembro de la bolsa y a través de los sistemas de contratación que las bolsas de valores tengan establecidos previamente. Por lo que la operación de bolsa requiere de una serie de componentes diferentes.

El ordenante, es la persona física o jurídica que da la orden de venta o de compra de valores en bolsa. Es necesario un miembro de la bolsa que ejecute la operación de compra o venta de valores, por lo que obligatoriamente ha de intervenir una agencia de valores y bolsa o sociedad miembro del mercado. Los valores admitidos a negociación en bolsa son requeridos también, es decir, el objeto de la operación. Y, por último, se necesitan los sistemas de contratación que las bolsas tengan preestablecidos para realizar estas operaciones.

En los últimos años estas operaciones de compraventa en la bolsa se han visto aumentadas por el crecimiento de operar a través de internet. La mayoría de las entidades financieras sean españolas o extranjeras, permiten a los clientes operar en tiempo real a través de internet. Además, la posibilidad de operar a través de internet ha ofrecido a grandes ventajas a los inversores como una mayor agilidad a la hora de tomar decisiones y unas comisiones por intermediación más bajas respecto al sistema tradicional.[8]

Como ya se ha comentado anteriormente, gracias a las nuevas tecnologías y al avance de las computadoras, cada día es más común el trading algorítmico. Al trading como tal se le conoce por la compraventa de instrumentos financieros en el mercado bursátil. Su objetivo es la obtención de un beneficio mediante la compra de un activo financiero que luego se pretende vender más caro o por la venta de un valor que a continuación se pretende comprar más barato.[9]

Por lo que el trading algorítmico conforma una serie de métodos diseñados para resolver problemas matemáticos, enviando así ordenes de compraventa de acciones en base a parámetros establecidos. Los profesionales del sector y los especialistas diseñan estrategias y establecen normas, que introducen en las computadoras para que los algoritmos analicen de manera casi instantánea cuando es el mejor momento para realizar las operaciones bursátiles. A esta tecnología se le denomina SOR, proveniente del inglés Smart Order Router, envío inteligente de órdenes.

Según los analistas, las ventajas que tiene esta forma de trading es que se evitan fallos muchas veces concernientes a las emociones humanas, como la esperanza, muchas veces infundada, aunque los indicadores técnicos señalen lo contrario. Por otro lado, los factores psicológicos no afectan a los robots, que no tienen estados de humor y se podía decir que son impasibles, ya que no sienten miedo o avaricia y no entran en pánico en situaciones de alta presión. La rapidez y velocidad que tienen las computadoras es otra parte con la que los humanos no pueden competir, las computadoras son capaces de realizar muchas operaciones por segundo, algo que es vital. A su vez, el ahorro en gastos de personal también es notorio, por lo que la tendencia es de comisionar cada vez en mayor medida, las operaciones financieras a modelos computacionales en lugar de a personas. Así mismo, hay que introducir los avances en el aprendizaje automático y la inteligencia artificial (IA), cosa que también conducirá en el mismo camino comentado.[10]

La inteligencia artificial no es nueva en la industria del trading ya que lleva usándose más de 10 años por empresas especializadas en el sector, sobre todo en EE. UU. Existen múltiples compañías bien posicionadas en el sector como pueden ser, Trading Technologies, GreenKey Technologies, Kavout, Auquan, etc... En los próximos años está previsto que se creen más empresas especializadas en el uso de la Inteligencia Artificial para diseñar estrategias de inversión, por lo que en los próximos 10 años se podrá observar como la IA se une completamente al mundo del trading y será indispensable para los traders, como ya ocurre en otros muchos ámbitos de nuestra vida cotidiana.[11]

### **1.3. Robots financieros**

En el mercado Forex se pueden diferenciar claramente dos tipos de traders distintos, los que operan manualmente y aquellos de utilizan el trading automático. Un robot financiero, es un software de trading automático que ejecuta ordenes en los mercados, acordes con las pautas que se les han dado al desarrollar el código. Los robots de trading de vanguardia funcionan mediante mecanismos de inteligencia artificial (IA), aunque son especialmente costosos y complicados de encontrar.

Hay una serie de cuestiones a tener en cuenta a la hora de evaluar los robots de trading. En primer lugar, el número de operaciones que realiza el robot, ya que es común que los brokers cobren comisiones por cada operación que se realiza, por lo que es un aspecto importante a tener en cuenta ya que la suma de las comisiones puede ser importante. Otro dato relevante son las ganancias y pérdidas consecutivas, así como la pérdida media y la ganancia media en un número elevado de operaciones. Es importante calcular la relación entre riesgo y ganancia que existe, ya que tiene una relación directamente proporcional con la cantidad porcentual de nuestra cartera, junto con la tasa de éxito. En las conclusiones del proyecto, este tipo de cuestiones se discutirán con los resultados obtenidos y se hará una buena evaluación del algoritmo desarrollado.

Como ya se ha comentado anteriormente las ventajas principales de los robots de trading es la rapidez de ejecución y análisis que poseen, la imparcialidad, ya que se eliminan los factores emocionales, y los horarios de trading flexibles que permiten operar cuando más beneficie al trader.

Las desventajas que podrían poseer es la fiabilidad a la hora de analizar el algoritmo que se posee, ya que el rendimiento pasado no garantiza el rendimiento futuro, por lo que debería de tener una supervisión constante. Posibles errores de código que se cometan a la hora de realizarlos. El reemplazo del razonamiento humano, ya que la intuición y conocimientos que posee un buen trader, muy difícilmente pueden ser reemplazados por un robot.

Otros aspectos importantes a tener en cuenta es que debido al auge que está sufriendo este sector del mercado y el gran desconocimiento y vacíos legales que existen, hay una gran cantidad de estafas con robots de trading. En las conclusiones del proyecto se continuará tratando la cuestión de si es beneficioso operar con robots de trading.[12]

## **1.4. Objetivos del trabajo**

Como ya se ha comentado anteriormente, el sector financiero ha sufrido una gran transformación en los últimos años gracias a las nuevas tecnologías, y los mercados han ganado eficacia y transparencia.

Los algoritmos matemáticos capaces de generar operaciones de compraventa en el mercado y que permiten la programación de una estrategia de trading, pueden ser probados de antemano y estudiar su rentabilidad y probabilidad de éxito.

El objetivo esencial de este proyecto es diseñar, programar y validar una estrategia de trading algorítmico financiera. Para ello, se realizará un profundo trabajo de investigación en cuanto al funcionamiento de la bolsa, las diferentes estrategias y procedimientos de realizar trading, y los diversos indicadores técnicos existentes.

A continuación, la estrategia se implementará y se desarrollará en un código de Python para probar así su viabilidad y operatividad en la bolsa real y realizar un "backtesting".

Como objetivos secundarios se podrían plantear los siguientes: por un lado, construir una plataforma que implemente el algoritmo con la estrategia desarrollada una vez realizado el backtesting y sea capaz de tomar decisiones por sí misma, ofrecer una interfaz sencilla e intuitiva al usuario, y por otro, añadir al algoritmo una serie de métodos de inteligencia artificial, como puede ser el "machine learning", para que el algoritmo sea capaz de aprender de sus errores o aciertos y mejorar así su operatividad.

## CAPÍTULO 2. Estrategia

En este segundo capítulo del trabajo, se explican los pasos y decisiones que se han seguido a la hora de desarrollar y diseñar todos los detalles de la estrategia de trading.

### 2.1. Diseño de la estrategia

Para comenzar, el método de trading que se va a usar será el scalping, que se trata de una manera de operar a muy corto plazo. En este caso la estrategia que se ha creado es para hacer operaciones en marcos de tiempo M5 (5 minutos), aunque se tendrá que comprobar en M3 o M1 su funcionalidad y resultados, ya que también puede trabajar bien para esos marcos de tiempo. Por otro lado, aunque la estrategia a priori pueda funcionar tanto como para índices, valores, futuros, divisas, etc..., en un primer lugar será diseñada para operar en el mercado FOREX. Se ha escogido este mercado debido a que sus activos son muy líquidos y se busca que el spread entre la oferta y la demanda sea lo más bajo posible.

Una vez establecidos estos términos, se puede continuar con la explicación de la estrategia. Dicha estrategia tratará de aplicar los niveles de Fibonacci combinándolos con dos indicadores: el estocástico y el RSI. El primer paso que hay que hacer es identificar una tendencia en un gráfico de velas japonesas en M5, como la foto que se muestra a continuación en la *Figura 2.1*.



**Fig. 2.1** Tendencia en gráfico con velas japonesas. Fuente: elaboración propia.

La forma en la que se identificará una tendencia en un primer momento es observando las velas del gráfico. Si se detectan unas cuantas velas con mínimos y máximos crecientes, estaremos en la posibilidad de una tendencia alcista. Por el contrario, si detectamos máximos decrecientes, se podría tratar de una tendencia bajista. Para un marco de tiempo de M5, se puede utilizar un marco de tiempo de M30 para poder ver las tendencias con algo más de claridad. Además, se puede observar patrones de dobles techos o suelos para determinar el inicio o el final de una nueva tendencia, por lo que la tendencia sería más clara aún. El mercado se mueve siempre haciendo correcciones, por lo que si se encuentra en una tendencia alcista, por ejemplo, siempre habrá momentos en el que el precio caiga para aplicar la corrección del valor que está subiendo y una vez hecha la corrección, seguir con la tendencia. Se intentará aprovechar en la mayor medida posible este tipo de correcciones en la tendencia para comprar o vender, aplicando los niveles o retrocesos de Fibonacci.



**Fig. 2.2** Estrategia Fibonacci. Fuente: [13]

Como se puede apreciar en la *Figura 2.2* de arriba, la estrategia nace con una relación de ganancia 2:1, ya que la distancia entre el punto que se compra y el Stop Limit es exactamente la mitad a la distancia entre ese punto y el máximo previo. Por lo que funcionando 4 de cada 10 veces la rentabilidad sería positiva.

Cuando el mercado realiza una ruptura de la tendencia y entra en los niveles de Fibonacci de entre 38,2% y 50%, se encontraría en la zona de compra en este caso, poniendo el Stop Loss (SL) en una zona de aproximadamente el 63,5%. En caso de que la tendencia fuera a la baja, se aplicará el mismo método, pero vendiendo en la zona de entre 38,2% y 50%. A la hora de buscar el objetivo de la operación, como mínimo habrá que esperar a que alcance los valores donde rompe la tendencia principal, pero se aplicaran las extensiones de Fibonacci junto a mover el SL, para ir vendiendo o comprando las acciones y sacar así el máximo beneficio a la operación.

Para saber con más exactitud cuándo entrar en una operación, ya que los niveles de entre 38,2% y 50% puede ser un gran espacio, se utilizará de acompañantes un oscilador estocástico y el RSI. En cuanto al estocástico se refiere, cuando nos encontramos en la zona de compra o venta de Fibonacci, y si además el estocástico cruza la línea de una forma ascendente, se tomará como una señal

para comprar. Si por el contrario estamos en una tendencia bajista y nos encontramos en zona de venta y el estocástico cruza la línea de una forma descendente, será una indicación para vender.



**Fig. 2.3** Oscilador estocástico. Fuente: [14]

En la *Figura 2.3* se puede observar muy bien como en una tendencia alcista (con mínimos y máximos crecientes), cuando se rompe la tendencia y entra dentro de los niveles de Fibonacci, se espera a que el oscilador estocástico corte de una forma ascendente y se compra.

Por otro lado, se encuentra el indicador RSI, que se utilizará para ver las divergencias y las señales de fin de tendencia mediante ellas. Si se aprecia que nos encontramos en una tendencia clara, después de un doble techo o suelo y el indicador estocástico indica justo el momento bueno en el que operar, todo son motivos para operar con más seguridad todavía. Por lo que en esos casos se podrían hacer operaciones con mayor cantidad de dinero.

En la *Figura 2.4* que se muestra a continuación se puede apreciar una divergencia alcista, que como se puede ver indica el final de la tendencia bajista que había, sumándole el doble suelo que se puede ver. En este caso, viendo la divergencia y el doble suelo, con gran seguridad podría haber una tendencia alcista a continuación (que es lo que ocurre), y en una de las correcciones de esa tendencia se podría aplicar la estrategia con una probabilidad de ganancia mucho mayor.



**Fig. 2.4** Divergencia alcista. Fuente: Elaboración propia.

En cuanto a gestión monetaria, cada operación tendrá aproximadamente un 2% del saldo total de la cartera, y esa operación se podrá dividir en distintos lotes de ese 2%. Se decidirá cuándo comprar y vender en un primer momento y después ajustando el Stop Loss, cuando salir de la operación cada lote.

A continuación, se presentará a modo de resumen el algoritmo.

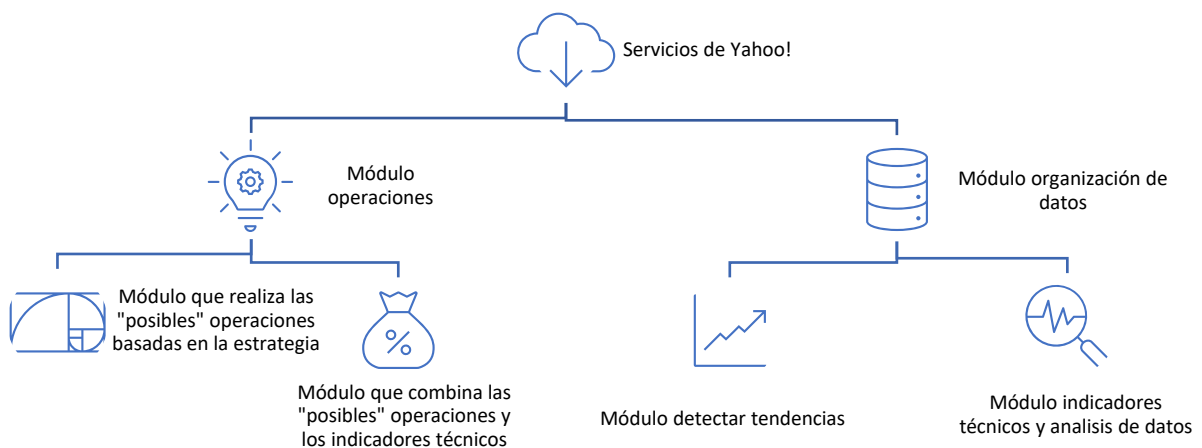
- **Precondición:** Se tienen precargados una secuencia de datos con unas determinadas características. Dicha secuencia de datos tiene un periodo de duración de 3 años.
- **Entradas:** Vector en el que cada elemento representa un instante de tiempo con la siguiente información:
  - a. Precio apertura
  - b. Precio de cierre
  - c. Volumen
  - d. Precio máximo
  - e. Precio mínimo
- **Recorrido sobre el vector:**
  - a. Calcula indicadores técnicos.
  - b. Detecta tendencias alcistas y bajistas. Utilizando los datos de precio de apertura y de cierre.
  - c. Aplicación de Fibonacci. Se generan posibles compras o ventas.
  - d. Comparación con el indicador técnico.
  - e. Se aplica la acción.

## CAPÍTULO 3. Arquitectura de la solución

Lo primero que se definirá en este capítulo es la arquitectura que tiene el algoritmo, seguidamente de una explicación del flujo de la arquitectura.

### 3.1. Arquitectura

En la siguiente *Figura 3.5* que se muestra a continuación, se puede observar la arquitectura del código, y los módulos de los que dispone.



**Fig. 3.5** Arquitectura del algoritmo. Fuente: Elaboración propia.

- **Servicios de Yahoo!:** En primer lugar, el algoritmo utiliza la API REST de Yahoo! Finance para descargar los datos históricos de la bolsa de los últimos años. Una API REST es cualquier interfaz entre sistemas que use HTTP con el fin de obtener datos o generar operaciones en todos los formatos posibles.[15] Gracias a la parte de finanzas del gigante tecnológico, es posible descargar los datos financieros de los últimos años con una gran variedad de información y fiabilidad.
- **Módulo de organización de datos:** En este módulo del algoritmo se extraen los datos descargados anteriormente y se filtran y organizan. Yahoo! Finance nos proporciona mucha información que no resulta relevante para este proyecto. Es por eso que dicha información se



elimina, filtrando así los datos, se organiza en archivos CSV y creando diccionarios, facilitando su posterior manipulación.

- **Módulo de operaciones:** Esta parte del algoritmo engloba y predefine algunas variables que se utilizan en los siguientes módulos del programa.
- **Módulo de detectar tendencias:** La detección de tendencias es una parte muy importante de la estrategia, este módulo se encarga de analizar todos los datos y localizar las posibles tendencias alcistas y bajistas que existan. Una vez localizados los puntos donde se encuentran las tendencias, habrá un módulo que se encargue de aplicar la estrategia en dichos puntos.
- **Módulo de indicadores técnicos y análisis de datos:** El siguiente módulo se encarga de analizar los datos con el fin de obtener los indicadores técnicos más utilizados en el mundo financiero. Una vez realizadas las operaciones matemáticas oportunas, el módulo es capaz de ofrecernos indicadores como pueden ser medias móviles, RSI, osciladores estocásticos y volumen entre otros. Algunos de ellos no serán utilizados inicialmente ya que la estrategia no los contempla, sin embargo, en un futuro pueden ser útiles ya que no nos encasillamos y puede que otras combinaciones de indicadores técnicos sean más fructíferas.
- **Módulo que realiza las “posibles” operaciones:** En esta parte del algoritmo es donde se desarrollan y aplican las herramientas que más peso tienen de la estrategia. Se analizan los datos en las tendencias detectadas y se aplican los retrocesos de Fibonacci con el fin de encontrar los posibles puntos de compraventa de los productos.
- **Módulo que combina las “posibles” operaciones y los indicadores:** Este módulo se encarga de estudiar y combinar los indicadores técnicos en los posibles puntos de compraventa. Una vez combinados, muestra los resultados obtenidos de cada operación y del total de operaciones.

## 3.2. Flujo de la arquitectura

El flujo del algoritmo entonces quedaría de la siguiente manera:

1. El primer paso que realiza el algoritmo es descargar mediante los servicios de Yahoo! Finance los datos históricos de cualquier compañía o divisa, en el tiempo que se le asigne.
2. Una vez los datos han sido descargados, el algoritmo los filtra mediante el módulo de organización de datos y crea unos diccionarios con los que va a trabajar. Una vez creados estos diccionarios con los datos (que contienen los precios de apertura o cierre de los productos, los máximos, mínimos, etc...), el módulo de detección de tendencias los analiza para encontrar los puntos donde habrá posibles tendencias alcistas o bajistas

y se pueda proceder a aplicar la estrategia. A su vez, el módulo de indicadores técnicos trabaja para calcular todos los indicadores en cada punto.

3. Una vez se han realizado los procesos anteriores, el algoritmo ejecuta el módulo de operaciones. Este módulo aplica los retrocesos de Fibonacci en los distintos puntos que se hayan detectado tendencias y se obtienen así los posibles puntos de compraventa. Por último, entra en acción el módulo que combina las posibles operaciones con los indicadores técnicos. Si en un cierto momento donde ocurre una posible operación de compraventa, los indicadores son favorables y se cumplen las directrices diseñadas en la estrategia, entonces el algoritmo procede a ejecutar dicha posible operación. Una vez ejecutada, va mostrando los resultados que obtiene.

## CAPÍTULO 4. Implementación

En el siguiente capítulo se explicará el lenguaje de programación empleado, las librerías, el entorno de trabajo, GitHub y, por último, se describirán los aspectos más importantes de implementación en la prueba piloto desarrollada.

### 4.1. Lenguaje de programación

En un primer momento, una vez diseñada la estrategia de inversión, la idea era desarrollar el algoritmo en una API de un bróker. Se analizó la posibilidad de conectar el algoritmo directamente a una cuenta demo en tiempo real. Sin embargo, las complicaciones que traía consigo la utilización de una cuenta demo y la falta de herramientas y recursos por parte de las plataformas, desembocó en la elección de un lenguaje de programación con su propio ecosistema para la elaboración de un backtesting del algoritmo. El backtest se encuentra entre uno de los métodos más conocidos en cuanto a validación de estrategias se refiere, a menudo es utilizado para la comprobación y la evaluación de diferentes modelos estadísticos. Su objetivo principal es comprobar la viabilidad de la estrategia en el pasado, estudiando y empleando los datos históricos de transacciones ocurridas en el pasado.[16]

En cuanto al lenguaje de programación, se decidió utilizar Python por una serie de características que se van a mostrar a continuación:

- Se trata de un lenguaje simple, legible, da soporte a conceptos potentes y se trata de un lenguaje natural y muy avanzado.
- La cobertura funcional que se obtiene con Python es muy extensa, permite hacer absolutamente todo lo que deseemos; desde accesos a bases de datos e implementación de modelos de datos de objetos avanzados para manipular entidades, generación automática de consultas SQL, acceder a servidores, puede utilizar tecnologías del mundo Java, o la generación de documentos de texto, imágenes, flujos de audio o de video entre otros.
- Otra característica importante es la homogeneidad del propio lenguaje de programación y de sus librerías. Python dispone de ciertas librerías con una gran reputación, como es el caso de la informática científica.
- Permite la realización de cálculos científicos para la generación de gráficos o cálculo distribuido, y se trata de un lenguaje cercano a las matemáticas. Python permite la gestión de una manera natural, sencilla y eficaz de operaciones matemáticas.
- Se trata del lenguaje de programación referencia en Deep learning y machine learning, por lo que de cara al futuro y a futuras mejoras es un aspecto muy importante a tener en cuenta.

- Abarca una gran comunidad. Gracias a la popularidad que ha adquirido, detrás del lenguaje se encuentra una gran comunidad dispuesta a ayudar a quien lo requiera. Es posible pedir consejo y recibir feedback con rapidez y seguridad.[17]

Es gracias a estas características y ventajas que tiene Python que se decidió usar para realizar el backtesting de la estrategia. Además, como ya se ha comentado anteriormente, ofrece un amplio abanico en cuanto a librerías se refiere, y en el mundo de las finanzas no se queda atrás. Esta característica, junto a la posibilidad de implementar inteligencia artificial para optimizar el algoritmo de una forma relativamente más sencilla que en otros lenguajes, fueron determinantes a la hora de la elección del lenguaje.

## 4.2. Librerías

En esta parte del informe se va a proceder a explicar las distintas librerías de Python que utiliza el algoritmo. En la *Figura 4.6* que se muestra a continuación se pueden observar las primeras líneas del código, donde aparecen las librerías importadas.

```
1 from pyalgotrade import strategy
2 from pyalgotrade.barfeed import yahoofeed
3 from pyalgotrade.barfeed import quandlfeed
4 import matplotlib.pyplot as plt
5 import yfinance as yf
6 import pandas as pd
7 import csv
8 from pyalgotrade.technical import ma
9 from pyalgotrade.technical import rsi
10 from pyalgotrade.technical import stoch
11 import numpy as np
12
```

**Fig. 4.6** Librerías del código. Fuente: Elaboración propia.

- **Librería PyAlgoTrade:** Es una librería de Python enfocada al trading algorítmico, más concretamente, en el backtesting y apoyo de trading en vivo. Una vez se tiene una estrategia de trading diseñada, esta librería permite evaluarla con datos históricos y ver cómo se comporta de una forma sencilla. Otra ventaja que tiene esta librería es que permite fácilmente optimizar una estrategia usando múltiples ordenadores. A su vez, depende de otras librerías como pueden ser NumPy, matplotlib, pytz

o tornado entre otras. Por lo que también se deberán tener instaladas esas librerías en el sistema si se desea utilizar los servicios que ofrece esta librería. Es una librería sumamente importante para el desarrollo del backtesting, ya que soporta órdenes de Market, Limit, Stop y StopLimit y está totalmente documentada. Es capaz de trabajar con los archivos de plataformas como Yahoo! Finance, Google Finance y NinjaTrader CSV.

Mediante esta librería que trabaja con archivos CSV, se es capaz de obtener indicadores técnicos como pueden ser el RSI, las medias móviles, bandas de Bollinger o indicadores estocásticos. Por lo que gracias a ella se elimina una importante carga de trabajo al desarrollar el código. Por otro lado, es capaz de ofrecer métricas como el drawdown, o el análisis de nuestra estrategia, por tanto, es sumamente práctica. Otras características importantes que no se pueden obviar, es que permite operar con Bitcoins mediante Bitstamp y que analiza eventos de Twitter en tiempo real, por lo que cualquier noticia significativa que pueda afectar al mercado financiero es capaz de detectarla. [18]

- **Librería Yfinance:** Esta librería de Python utiliza los datos de Yahoo! Finance y descarga los datos financieros que se solicitan en formato DataFrame. La API de Yahoo! Proporciona acceso a los resúmenes financieros, existencias de precios históricos, etc... En la *Figura 4.7* que se muestra a continuación, se puede ver cómo esta librería es utilizada en el código.[19]

```
34 def get_datos_historicos(nombre_compañia):
35     # consigue data stock
36     datos = yf.download(nombre_compañia, '2016-01-01', '2019-08-01')
37     # return datos
38     csvstring = datos.to_csv()
39     # print (csvstring)
40     returnvalue = datos.to_csv(nombre_compañia)
41     # datos['Adj Close'].plot()
42     # plt.show()
43     # lee los archivos csv y le da el nombre inicial a cada valor
44     with open(nombre_compañia) as csvfile:
45         reader = csv.DictReader(csvfile)
46         vector_datos = []
47         # para añadir cada fila del documento a un vector y trabajar con el
48         for row in reader:
49             vector_datos.append(row)
50             # print (row)
51             # print(row['Open'])
52         return (datos, vector_datos)
53     # return (vector_datos[1].__getitem__('Open'))
54     # datos1 = yf.download('AMZN', '2016-01-01', '2019-08-01')
55     # csvstring=datos1.to_csv()
56     # returnvalue1=datos1.to_csv('AMZN1619.csv')
```

**Fig. 4.7** Función obtener datos históricos. Fuente: Elaboración propia.

Mediante la función `get_datos_historicos(nombre_compañía)` que utiliza la librería `yfinance`, se es capaz de descargar los datos históricos de una compañía y analizarlos añadiéndolos a un vector vacío. En el ejemplo que se muestra a continuación se pueden observar los resultados de apertura y cierre de las acciones de Apple junto a varios indicadores técnicos.

```
C:\Users\iker1\AppData\Local\Programs\Python\Python38-32\python.exe C:/Users/iker1/PycharmProjects/backtesting/prueba.py
2015-12-31 00:00:00 strategy [INFO] 26.752500534057617 26.315000534057617 None None 0
2016-01-04 00:00:00 strategy [INFO] 25.65250015258789 26.337499618530273 None None 1
2016-01-05 00:00:00 strategy [INFO] 26.4375 25.677499771118164 None None 2
2016-01-06 00:00:00 strategy [INFO] 25.139999389648438 25.174999237060547 None None 3
2016-01-07 00:00:00 strategy [INFO] 24.67000076293945 24.112499237060547 None None 4
2016-01-08 00:00:00 strategy [INFO] 24.637500762939453 24.239999771118164 None None 5
2016-01-11 00:00:00 strategy [INFO] 24.74250030517578 24.63249969482422 None None 6
2016-01-12 00:00:00 strategy [INFO] 25.137500762939453 24.989999771118164 None None 7
2016-01-13 00:00:00 strategy [INFO] 25.07999923706055 24.34749984741211 None None 8
2016-01-14 00:00:00 strategy [INFO] 24.489999771118164 24.8799991607666 None None 9
2016-01-15 00:00:00 strategy [INFO] 24.049999237060547 24.282499313354492 None None 10
2016-01-19 00:00:00 strategy [INFO] 24.602500915527344 24.165000915527344 None None 11
2016-01-20 00:00:00 strategy [INFO] 23.774999618530273 24.197500228881836 None None 12
2016-01-21 00:00:00 strategy [INFO] 24.264999389648438 24.075000762939453 None None 13
2016-01-22 00:00:00 strategy [INFO] 24.657499313354492 25.354999542236328 42.55812612463424 None 14
2016-01-25 00:00:00 strategy [INFO] 25.3799991607666 24.860000610351562 39.3093110147715 None 15
2016-01-26 00:00:00 strategy [INFO] 24.982500076293945 24.997499465942383 40.66430704482824 None 16
2016-01-27 00:00:00 strategy [INFO] 24.010000228881836 23.354999542236328 31.59092406427291 None 17
2016-01-28 00:00:00 strategy [INFO] 23.447500228881836 23.522499084472656 33.227170068580165 None 18
2016-01-29 00:00:00 strategy [INFO] 23.697500228881836 24.334999084472656 40.64362972893234 None 19
2016-02-01 00:00:00 strategy [INFO] 24.11750030517578 24.107500076293945 39.32651636934812 None 20
2016-02-02 00:00:00 strategy [INFO] 23.854999542236328 23.6200008392334 36.590151580241226 None 21
2016-02-03 00:00:00 strategy [INFO] 23.75 24.087499618530273 40.841227804320404 None 22
2016-02-04 00:00:00 strategy [INFO] 23.96500015258789 24.149999618530273 41.406781379084386 None 23
2016-02-05 00:00:00 strategy [INFO] 24.1299991607666 23.5049991607666 37.42993875204186 None 24
```

Fig. 4.8 Datos históricos. Fuente: Elaboración propia.

- **Librería Matplotlib:** Es una de las librerías más usadas de Python para generar gráficas a partir de datos contenidos en listas, vectores, en el lenguaje de programación y en su extensión matemática NumPy, la cual hablaremos a posterior. Permite realizar gráficas complejas en 2D y 3D con diferentes tipos de líneas, colores, marcadores, textos en los ejes, etc... Además es una librería comprehensiva que es capaz de crear visualizaciones estáticas, animadas e interactivas en Python.[20]
- **Librería Pandas:** Se trata de una librería especializada en el análisis y manejo de estructuras de datos. Sus principales características entre otras son; que permite leer y escribir fácilmente ficheros de diferentes formatos (CSV, Excel y bases de datos), permite acceder a los datos mediante índices o nombres para filas y columnas, permite trabajar con series temporales, o que se ofrecen métodos en ella para dividir, reordenar y combinar conjuntos de datos. Es una herramienta rápida, flexible y potente para el análisis de datos.[21]
- **Librería Numpy:** Numpy es una librería de Python que está especializada en el cálculo numérico y el análisis de datos, especialmente cuando se trata de un gran volumen de datos. Una novedad importante es que

incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en distintas dimensiones, e incorpora también funciones potentes y eficientes para su manipulación. Su nombre proviene de “Numerical Python”, por lo que se puede intuir que es la librería principal para la informática científica.[22]

### 4.3. Entorno de trabajo

En este punto del informe, se va a comentar el entorno del trabajo que se ha utilizado para el desarrollo del código. En primer lugar, hay que explicar lo que es un Entorno de Desarrollo Integrado (IDE). Se trata de un sistema de software para el diseño de aplicaciones que es capaz de combinar herramientas del desarrollador comunes en una sola interfaz gráfica de usuario. Permite al usuario enfocarse 100% en el código, ya que el IDE se encarga de lo demás.

PyCharm es un IDE enfocado en Python de la empresa JetBrains y es usado por un gran número de desarrolladores. En este caso, se ha utilizado la versión Community del mismo, aunque hay disponible la versión Profesional, de pago y con aún más ventajas. A continuación, se van a nombrar las características más importantes de este entorno de trabajo:

- Al trabajar siempre por proyectos, por defecto se crea un entorno virtual para empezar a desarrollar. Por lo que se trata de una gran ventaja.
- El flujo de trabajo se hace mucho más dinámico, ya que se permite navegar fácilmente por el código, saltando entre las clases y métodos que se hayan creado.
- Contiene un editor inteligente con varias ventajas, identifica cualquier tipo de errores, la sintaxis en el momento que se escribe, formateo de código, completa el código, detecta el código duplicado y trae consigo documentación automática.
- Ofrece una buena ayuda a la hora de realizar refactorizaciones, de manera segura con cambios locales y globales.
- Posee una terminal incorporada, por lo que no es necesario salir del IDE.
- Contiene también, un depurador gráfico, un ejecutor de pruebas y un inspector de código.

Ofrece una infinidad más de ventajas y características. Es un entorno de trabajo que se puede personalizar al gusto de cada desarrollador y es muy intuitivo, además de ofrecer una gran ayuda en cuanto a solución de errores y autocomplección a medida que se escribe el código. Como en su propia página web indican, está diseñado por programadores y para programadores, con el fin

de proporcionar al usuario todas las herramientas que necesita para un desarrollo productivo.[23]

#### 4.4. GitHub

GitHub se trata de una de las principales plataformas para crear proyectos abiertos de aplicaciones y herramientas, por lo que gracias a sus funciones colaborativas todas las personas que lo desean pueden aportar sus ideas.

GitHub se creó con el fin de alojar el código de las aplicaciones del desarrollador que o desee, y hace 3 años fue comprada por Microsoft. Como usuario de la plataforma, además de poder compartir y ver el código de los demás desarrolladores, también se puede acceder al perfil de los creadores para colaborar con su proyecto o leer acerca de esa persona.

La plataforma, como su propio nombre indica, utiliza el sistema de control de versiones Git diseñado por Linus Torvalds. Permite a los desarrolladores tener copias de cada una de las versiones de su proyecto, por lo que no se pierden los estados anteriores cuando se actualiza. Es una herramienta sumamente útil ya que permite comparar el código de un archivo y divisar diferencias entre las versiones, fusionar cambios de distintas versiones o restaurar versiones antiguas.[24]

Para el proyecto en cuestión, la utilización de GitHub ha sido sumamente importante. Ya que ha permitido observar código de desarrolladores de trading algorítmico, además de la completa documentación de librerías de Python. Como ya se ha comentado anteriormente, una de las librerías de mayor relevancia y peso en el trabajo ha sido PyAlgoTrade, debido a las características y funcionalidades que tiene. Gracias a GitHub, es posible encontrar toda la documentación de esta librería de una forma fácilmente legible y entendible, conjuntamente con ejemplos. Como se puede observar en la *Figura 4.9* de a continuación, así se vería la tabla de contenidos de la documentación de la librería en GitHub, por lo que es una tarea fácil encontrar la información que se desea en cada momento.





The image shows a screenshot of the PyAlgoTrade documentation website. The page has a dark blue header with the text 'PyAlgoTrade 0.20 documentation' and 'PyAlgoTrade documentation'. On the left side, there is a sidebar with navigation links: 'Table Of Contents', 'Next topic', 'This Page', and 'Quick search'. The main content area is titled 'Contents:' and lists a comprehensive table of contents for the documentation, including sections for Introduction, Tutorial, Trading, Optimizing, Plotting, and various data sources and strategies.

**PyAlgoTrade 0.20 documentation**  
PyAlgoTrade documentation  
Index and labels

**Table Of Contents**

**Next topic**  
Introduction

**This Page**  
Show Source

**Quick search**  
Go

Enter search terms or a module, class or function name.

**Contents:**

- Introduction
- Tutorial
  - Trading
  - Optimizing
  - Plotting
- Documentation for the code
  - bar – Instrument prices
  - dataserries – Basic dataserries classes
  - feed – Basic feeds
    - CSV support
    - CSV support Example
  - barfeed – Bar providers
    - CSV
    - Yahoo! Finance
    - Google Finance
    - Quandl
    - Ninja Trader
  - technical – Technical indicators
    - Example
    - Moving Averages
    - Momentum Indicators
    - Other indicators
  - broker – Order management classes
    - Base module and classes
    - Backtesting module and classes
  - strategy – Basic strategy classes
    - Strategy
    - Position
  - stratanalyzer – Strategy analyzers
    - Returns
    - Sharpe Ratio
    - DrawDown
    - Trades
    - Example
  - plotter – Strategy plotter
  - optimizer – Parallel optimizers
  - marketsession – Market sessions
- Tools
  - Quandl
  - BarFeed resampling
- Event profiler
  - Example
- Bitcoin
  - Bitstamp support
    - bitstamp – Bitstamp reference
    - Bitstamp Example
  - Bitcoin Charts support
    - bitcoincharts – Bitcoin Charts reference
    - Bitcoin Charts example
- Twitter support
  - twitter – Twitter feed reference
    - Feed
    - Twitter Example
- TA-Lib integration
- Computational Investing Part I
  - Homework 1
- Sample strategies
  - Momentum
    - VWAP Momentum Trade
    - SMA Crossover
    - Market Timing Using Moving-Average Crossovers
  - Mean Reversion
    - Ernie Chan's Gold vs. Gold Miners
    - Bollinger Bands
    - RSI2
  - Others

**Fig. 4.9** Contenidos PyAlgoTrade. Fuente: [25]

Entre los contenidos, primeramente, se ofrece una pequeña introducción acerca de la librería, para a continuación proceder con un tutorial de muestra y con toda la documentación y herramientas para desarrollar el código. La información que se incluye es sumamente útil, ya que se explican las diferentes herramientas y funcionalidades que tiene la librería, por lo que a la hora de querer añadir en el código diferentes funciones, se pueden encontrar de una forma sencilla los recursos para hacerlo. En la siguiente *Figura 4.10*, se puede observar cómo se encuentra la documentación. Por ejemplo, cuando se programaron las líneas de código para la obtención de los indicadores técnicos, se utilizaron varias clases como las que se pueden apreciar.[25]

## Moving Averages

```
class pyalgotrade.technical.ma.SMA(dataSeries, period, maxLen=None)
```

```
Bases: pyalgotrade.technical.EventBasedFilter
```

Simple Moving Average filter.

**Parameters:**

- **dataSeries** (`pyalgotrade.dataseries.DataSeries.`) – The DataSeries instance being filtered.
- **period** (*int.*) – The number of values to use to calculate the SMA.
- **maxLen** (*int.*) – The maximum number of values to hold. Once a bounded length is full, when new items are added, a corresponding number of items are discarded from the opposite end. If None then `dataseries.DEFAULT_MAX_LEN` is used.

```
class pyalgotrade.technical.ma.EMA(dataSeries, period, maxLen=None)
```

```
Bases: pyalgotrade.technical.EventBasedFilter
```

Exponential Moving Average filter.

**Parameters:**

- **dataSeries** (`pyalgotrade.dataseries.DataSeries.`) – The DataSeries instance being filtered.
- **period** (*int.*) – The number of values to use to calculate the EMA. Must be an integer greater than 1.
- **maxLen** (*int.*) – The maximum number of values to hold. Once a bounded length is full, when new items are added, a corresponding number of items are discarded from the opposite end. If None then `dataseries.DEFAULT_MAX_LEN` is used.

```
class pyalgotrade.technical.ma.WMA(dataSeries, weights, maxLen=None)
```

```
Bases: pyalgotrade.technical.EventBasedFilter
```

Weighted Moving Average filter.

**Parameters:**

- **dataSeries** (`pyalgotrade.dataseries.DataSeries.`) – The DataSeries instance being filtered.
- **weights** (*list.*) – A list of int/float with the weights.
- **maxLen** (*int.*) – The maximum number of values to hold. Once a bounded length is full, when new items are added, a corresponding number of items are discarded from the opposite end. If None then `dataseries.DEFAULT_MAX_LEN` is used.

**Fig. 4.10** Clases PyAlgoTrade. Fuente: [25]

Una vez finalizado el proyecto y el código en cuestión del backtesting del algoritmo, se estudiará la posibilidad de añadir dicho código en un repositorio de GitHub, con el fin de que todas las personas que lo deseen puedan acceder a él. Como consecuencia, se espera que las ideas introducidas en él puedan servir de ayuda a posibles desarrolladores, o que el código se vea optimizado o se añadan funcionalidades que no han sido consideradas en este proyecto.

## 4.5. Proyecto Backtesting

El proyecto de Python desarrollado como prueba piloto consiste en un conjunto de scripts de Python que utiliza las librerías anteriormente descritas e implementan la estrategia y el algoritmo especificado en los puntos 2.1. y 3.1.

La primera versión del proyecto consumía los servicios del bróker Interactive Brokers (IB) en tiempo real, pero la limitación de la licencia impedía realizar un análisis completo de una serie completa que precisaba el algoritmo. La segunda versión utiliza una serie de documentos CSV que se han descargado previamente.

Date	Open	High	Low	Close	Volume	Ex-Dividend	Split Ratio	Adj. Open	Adj. High	Adj. Low	Adj. Close	Adj. Volume
2009-12-31	132.41	132.85	130.75	130.9	4223400.0	0.0	1.0	108.03834720989	108.39735991869	106.68389017214	106.80628086833	4223400.0
2009-12-30	131.23	132.68	130.68	132.57	3867000.0	0.0	1.0	107.07554039993	108.25865046302	106.62677451393	108.16889728582	3867000.0
2009-12-29	132.28	132.37	131.8	131.85	4184200.0	0.0	1.0	107.9322752732	108.00570969091	107.54062504542	107.58142194415	4184200.0
2009-12-28	130.99	132.31	130.72	132.31	5800400.0	0.0	1.0	106.87971528604	107.95675341244	106.65941203291	107.95675341244	5800400.0
2009-12-24	129.89	130.57	129.48	130.57	4265100.0	0.0	1.0	105.98218351403	106.53702133673	105.64764894447	106.53702133673	4265100.0

Fig. 4.11 Datos en formato CSV. Fuente: Elaboración propia.

Como se puede apreciar en el ejemplo *Figura 4.11* de CSV se obtiene información de las variables de los precios de apertura o de cierre, así como del volumen o de los precios máximos y mínimos entre otros. Estos son los datos entrada del algoritmo para la toma de decisiones de trading.

En el siguiente enlace de GitHub se puede descargar el proyecto desarrollado. [26] Pero se presenta una de las partes más relevantes del código desarrolladas. En este caso se muestra una parte de la función donde se buscan y analizan las distintas regresiones de Fibonacci en los puntos de tendencia detectados y decide si son posibles puntos de compra o de venta.

```

210 def aplicar_fibonacci(self):
211     a=float(0.382)
212     b=float(0.50)
213     c=float(0.64) #stoplossss
214     n = 0
215     buscador = 0
216     while n < len(self.vector_datos):
217         longitud_tendencia=(len(self.tendencia_alza) - 2)
218         if ((self.tendencia_alza[n] == buscador) and n<longitud_tendencia:
219             min = (self.vector_datos[buscador].__getitem__('Low'))
220             max_encontrado=False
221             try:
222                 i=buscador
223                 while i< len(self.vector_datos) and (max_encontrado==False):
224                     if (self.vector_datos[i+3].__getitem__('Open')) < (self.vector_datos[i+2].__getitem__('Close')):
225                         max= (self.vector_datos[i+2].__getitem__('High'))
226                         max_encontrado=True
227                     else:
228                         i += 1
229             except ValueError:
230                 print ('Hay un fallo')
231             max=float(max)
232             min=float(min)
233             # si es una tendencia alcista:
234             R1A =float (max - (a * (max - min))) # retroceso de 38,2 etc
235             R2A = float(max - (b * (max - min))) #0,5

```

Fig. 4.12 Código función Fibonacci. Fuente: Elaboración propia.

## CAPÍTULO 5. Planificación

En este capítulo se va a detallar la carga de trabajo y como se ha distribuido durante todo el proyecto. A continuación, se van a nombrar las tareas más relevantes que se han realizado durante el proyecto y una breve explicación de cada una de ellas:

- **Definición del proyecto:** Tiempo dedicado a definir la idea principal del proyecto y el alcance al que aspiraba.
- **Investigación:** Tiempo destinado en cualquier tipo de investigación; incluye investigación acerca del mercado financiero, distintas formas de operar en bolsa, plataformas de brokers, APIs, así como de los lenguajes óptimos para programar algoritmos de trading y el desarrollo del ecosistema y herramientas.
- **Diseño:** Tiempo empleado definiendo y diseñando la estrategia de inversión.
- **Desarrollo:** Esta fase incluye principalmente la parte de programación y de código.
- **Documentación:** Tiempo dedicado a escribir el informe del proyecto

En la *Tabla 5.1* que se muestra a continuación, se puede observar una estimación de tiempo y porcentaje dedicados a cada tarea.

**Tabla 5.1.** Cargas de trabajo. Fuente: Elaboración propia.

Tarea	Estimación de tiempo (horas)	Estimación de tiempo (porcentaje)
Definición del proyecto	20	3,33%
Investigación	150	25%
Diseño	100	16,66%
Desarrollo	250	41,66%
Documentación	80	13,33%
TOTAL	600	100%

Como se puede apreciar en la *Tabla 5.1* superior, la tarea que menos porcentaje de tiempo ocupó fue la definición del proyecto. A la hora de investigar los posibles temas de interés para realizar el proyecto, se obtuvo una primera idea muy aproximada a lo que terminó siendo el objetivo y el alcance del proyecto.

Por otro lado, se puede observar que el desarrollo fue la carga de trabajo que más porcentaje de tiempo experimentó. Esto tiene sentido, ya que la implementación de la estrategia y el desarrollo del algoritmo y el código es una de las partes fundamentales del trabajo. Programar el algoritmo y ser capaz de solucionar los problemas que se planteaban ocupó una extensa cantidad de horas.

Otra parte de gran importancia para el proyecto, fueron las horas empleadas en investigación y diseño de la estrategia. En un primer momento, cuando se realizó la definición del proyecto, los conocimientos previos sobre el mundo de la bolsa y las finanzas eran muy básicos, por lo que la parte de investigación fue un elemento muy importante para adoptar las herramientas necesarias que permitieron un buen diseño de la estrategia que se iba a seguir. Durante todo el proyecto la investigación ha jugado un papel crucial, tanto para el diseño de la estrategia como a la hora de desarrollar el código.

Por último, con una carga de trabajo de aproximadamente el 15% de las horas totales, estaría la parte de documentación. Subjetivamente hablando, se trató de dar más relevancia a la investigación, el diseño y el desarrollo del trabajo. Desde un primer momento se decidió así, ya que más allá de la importancia de desarrollar un buen informe, el objetivo principal era la adquisición de los conocimientos y las herramientas para lograr unos buenos resultados y poder así transmitirlos en el informe.

## CAPÍTULO 6. Conclusiones

El objetivo principal con el que se ha realizado el proyecto ha sido el diseño, desarrollo, implementación y validación de una estrategia propia de trading algorítmico, basada en los retrocesos y extensiones de Fibonacci junto a ciertos indicadores técnicos, mediante el uso de Python.

El trading algorítmico es una herramienta que está en auge hoy en día, tanto en inversores autónomos como en instituciones. Esto es debido a la variedad de ventajas que ofrece, así como de la cantidad de información que se puede encontrar en internet y libros acerca de la materia. Es por lo que a día de hoy es posible desarrollar una estrategia de trading, para ello es relevante mencionar una serie de puntos. Primeramente, es necesario poseer la voluntad y tiempo necesarios para investigar y adquirir los conocimientos precisos para el diseño de la estrategia. Otro aspecto clave a tener en cuenta es establecer los objetivos claros y adecuados, para ello se deben de tratar de objetivos específicos, alcanzables, relevantes y limitados en un periodo de tiempo. Otra parte fundamental a la hora de desarrollar la estrategia es determinar en qué mercados operar y con qué activos, el marco de tiempo en el que se va a realizar, la forma en la que se va a entrar y salir del mercado, y finalmente, cómo programar la estrategia. Todos los puntos nombrados anteriormente son aspectos claves para el diseño y desarrollo de una estrategia de trading, así como, la necesidad de utilizar un bróker, poseer una infraestructura informática y tecnológica adecuada y una buena red de comunicaciones.

En cuanto a la estrategia diseñada en el proyecto, se decidió en un primer momento implementarla en el mercado Forex, ya que es un mercado en el que la liquidez es alta y por lo tanto se reducen los spreads y costes de transacción de las operaciones. Sin embargo, esto puede afectar negativamente, ya que, al aumentar la liquidez del mercado mediante los algoritmos de trading, puede aumentarse también la volatilidad de los mercados y volverse menos eficientes. De todas formas, estos aspectos negativos no tienen la misma fuerza en todos los mercados, y el mercado Forex de divisas al tratarse de por sí un mercado muy líquido y con un gran volumen de transacciones no se ve tan afectado como otros mercados. Es por esas razones que se podría afirmar que Forex es el sitio adecuado para realizar trading algorítmico, ofreciendo la posibilidad de efectuar un gran número de operaciones en pocos segundos.

Una vez analizados los resultados obtenidos del backtest se pueden apreciar una serie de datos satisfactorios. En primer lugar, obtenemos un profit factor de 1,92. El profit factor es la relación entre las ganancias conseguidas y las pérdidas, se calcula dividiendo la suma de ganancias entre la suma de pérdidas. Como se puede observar, en nuestro caso el profit factor es superior a 1, por lo que esto querría decir que nuestro sistema es ganador. Tiene sentido, ya que a la hora de definir la estrategia, como ya se comentó, nacía con una relación 2 a

1. En siguiente lugar, está el Drawdown de nuestra estrategia, que analiza el nivel de riesgo de un sistema de trading, ya que mide el retroceso en la curva de resultados respecto al máximo anterior. En nuestro caso el Drawdown que hemos obtenido es de aproximadamente el 13%. En el apartado de siguientes pasos se comentarán ciertos aspectos interesantes del Drawdown y de cómo pueden variar los resultados del proyecto.

Por otro lado, cabe destacar que el trading algorítmico tiene grandes ventajas, como evitar los componentes emocionales y evitar la necesidad de operar bajo grandes presiones, sin embargo, se añaden posibles errores de programación o implementación de los modelos diseñados. En el caso de este proyecto, considero que puede haber ciertas partes del código que podrían ser mejoradas. A mi juicio, pueden existir partes complejas del algoritmo que deberían de ser mejoradas. Hay ciertos puntos de la estrategia que son relativamente fáciles de identificar de forma visual, cuando se obtiene práctica, que son especialmente difíciles de implementar en un código. Por ejemplo, la detección de líneas de tendencia es un aspecto clave en la estrategia, que a su vez podría decirse que es la parte más complicada de implementar en el código. Es por eso que siendo crítico con los resultados, se podrían mejorar notoriamente perfeccionando partes complejas del código.

Si echamos la vista atrás a los objetivos del trabajo, se podría decir que los principales y fundamentales se han cumplido con éxito, si bien es cierto que no todos se han realizado satisfactoriamente. Debido a lo cual, se profundiza sobre este tema en el apartado de siguientes pasos del proyecto.

En definitiva, la conclusión del proyecto es muy satisfactoria y deja la puerta abierta a futuras investigaciones y desarrollo de la idea principal. Sería muy interesante seguir profundizando más en el tema de trading algorítmico y seguir desarrollando y optimizando las herramientas existentes.

## 6.1. Siguiendo pasos

Como se ha comentado anteriormente, hay ciertos aspectos del proyecto que podrían tener una continuidad. Es un tema que abarca diferentes campos, como pueden ser el de la computación, las matemáticas, el financiero... Por lo que el proyecto puede tomar una infinidad de caminos para continuar con su desarrollo.

En primer lugar, se encuentran los objetivos secundarios del trabajo que no se han podido cumplir por la falta de tiempo. Entre estos, se halla el desarrollo de una plataforma que contenga implementado el algoritmo conectado a un bróker y ofrezca una interfaz para el usuario. La interfaz podría conformar, por ejemplo, una aplicación móvil o una página web.

Otra forma de continuar con el proyecto sería seguir con el desarrollo del código. Como ya se ha dicho hay partes que pueden ser perfeccionadas con tiempo y esfuerzo, además, sería un buen proyecto para enfocarlo a la inteligencia artificial. Una vez se ha demostrado que la estrategia es rentable, el siguiente paso es optimizarla. Una de las razones por las que se escogió el lenguaje de

programación de Python es que es referente en Deep learning y machine learning, por lo que de cara a futuras mejoras del proyecto puede ser un factor relevante a tener en cuenta.

La estrategia, aunque se ha demostrado que puede ser rentable, cabe la posibilidad de que se pueda mejorar además de optimizar. Si bien es cierto que el mercado principal para el que se ha diseñado es el de divisas ya que ciertas características principales le pueden beneficiar, cabe la posibilidad de que funcione de una mejor forma en otros mercados y con diferentes acciones. Esto se puede extrapolar también a los marcos de tiempo en los que se aplica la estrategia, así como a los indicadores con los que se combina. Hay una gran variedad de indicadores técnicos con los que la estrategia podría funcionar y la única manera de corroborar que los elegidos son los óptimos es comprobando los resultados con las demás combinaciones. Es por eso que, a la hora de desarrollar el código, se añadieron más indicadores de los que se iban a utilizar, de cara al futuro y a posibles mejoras del algoritmo. Realizando estas comprobaciones y optimizaciones del código, los resultados podrían variar de una forma notoria. La gestión monetaria y ese estudio puede ser muy importante también para la optimización del algoritmo. La gestión monetaria podría variar el Drawdown y el porcentaje total de la cuenta notoriamente. El incremento porcentual de la cuenta con un sistema con esperanza matemática positiva crece en mayor proporción que el Drawdown hasta llegar al porcentaje óptimo. Por lo que disminuyendo el porcentaje arriesgado a partir del porcentaje o fracción óptima por trade, se reduce el beneficio geoméricamente y el Drawdown aritméticamente. [27]

Como se puede observar, el proyecto tiene muchos campos en los que se puede seguir investigando y trabajando. Desde el punto de partida se ha buscado realizar el mejor trabajo posible y el mayor desarrollo del proyecto, por lo que las puertas quedan abiertas en una multitud de campos que pueden seguir aumentando la envergadura del proyecto.

## 6.2. Estudio medioambiental

En el siguiente apartado se estudiará el posible impacto que pudiera tener el proyecto en el medio ambiente. Los estudios ambientales son documentos técnicos que estudian la interacción humana con el medio ambiente. Se trata de un amplio campo de estudio y de gran relevancia ya que nos afecta a todos, y se le debería dar una gran importancia. Implica y abarca una serie de conjuntos básicos como pueden ser la ecología o la ciencia ambiental, además de temas asociados como la geografía, ética, política, antropología y demás.

En cuanto a este proyecto se refiere, se podría decir que el impacto medioambiental que tiene no es excesivo, tanto de forma positiva como negativa.

Por un lado, tenemos los aspectos que podrían ser negativos para el medio ambiente. En esta casilla se podría incluir los recursos en la nube que hay que emplear. Aunque no son especialmente significativos, es una parte del proyecto que podría afectar al medio ambiente. Estos servicios en la nube son gracias a



los más de 100 millones de servidores que están repartidos en centros de datos alrededor del mundo, así como cientos de miles de kilómetros de cables de fibra óptica, antenas o routers. Es por lo que estamos contribuyendo al uso de los distintos recursos que generan residuos y emisiones contaminantes, así como gases de efecto invernadero.[28]

Por otro lado, se encuentra la parte del proyecto que podría ser beneficiosa para el medio ambiente. Gracias a los temas que se han investigado y desarrollado, podría influir en la conducta de la gente de forma positiva. Antiguamente los servicios de inversión en bolsa se hacían desde lugares específicos y con la necesidad para la gente de tener que desplazarse. Hoy en día, este tipo de conductas está cambiando cada día más, y el proyecto podría contribuir más a eso. Es una forma de facilitar la toma de decisiones de las personas que se dediquen al mundo de las inversiones en bolsa, así como de automatizar ciertas estrategias que podría disminuir el gasto energético de este tipo de personas. Es por eso que gracias a la automatización y desarrollo de ciertas estrategias y algoritmos, se podrían reducir emisiones perjudiciales para el medio ambiente como pueden ser carburantes o electricidad.

Una vez analizados tanto los aspectos positivos como negativos para el medio ambiente del proyecto, se podría concluir que el impacto y la importancia no son significativos. Ninguna de las dos partes es suficientemente importante y de gran magnitud como para asegurar que tiene un impacto relevante.

## Bibliografía

- [1] “Prueba y revisión de YuanPay Group | Coinlist.me.” <https://coinlist.me/es/robots/yuanpay-group/> (accessed Oct. 05, 2021).
- [2] “Sitio web oficial de Yuan Pay Group 2021 | Yuanpaygroup.org™.” <https://yuanpaygroup.org/es/> (accessed Oct. 05, 2021).
- [3] “Bitcoin Champion App™ | Sitio oficial 2021.” <https://bitcoinchampion.io/es/> (accessed Oct. 05, 2021).
- [4] “Mercado de Divisas / Forex - 24/5 | Corredor de divisas - RoboForex.” <https://roboforex.com/es/> (accessed Oct. 06, 2021).
- [5] “CNMV - Entidades no autorizadas por la CNMV (‘Chiringuitos Financieros’).” <https://www.cnmv.es/Portal/advertenciaslistado.aspx?tipoAdv=1> (accessed Oct. 06, 2021).
- [6] A. Cáceres Sánchez Tutora and P. Tapia Ballesteros, “Facultad de Ciencias Sociales, Jurídicas y de la Comunicación El delito de estafa, la problemática de las estafas piramidales”.
- [7] “Manual de la inversión en bolsa”.
- [8] C. Vallejo and O. Torres, “Manual de la inversión en bolsa.” [https://www.caixabank.com/deployedfiles/caixabank/Estaticos/PDFs/aula/Manual\\_Inversion\\_Bolsa\\_\\_1.pdf](https://www.caixabank.com/deployedfiles/caixabank/Estaticos/PDFs/aula/Manual_Inversion_Bolsa__1.pdf) (accessed Oct. 06, 2021).
- [9] “View of Trading in the financial market.” <http://www.rfinanzas.com/index.php/RDF/article/view/7/10> (accessed Oct. 06, 2021).
- [10] “‘Trading algorítmico’: cómo funcionan y qué tienen que ver los algoritmos con la caída de la bolsa de Wall Street - BBC News Mundo.” <https://www.bbc.com/mundo/noticias-42960565> (accessed Oct. 06, 2021).
- [11] “Inteligencia Artificial y aprendizaje automático en trading.” <https://www.saladeinversion.com/articulos/inteligencia-artificial-y-aprendizaje-automatico-en-trading/> (accessed Oct. 06, 2021).
- [12] “¿Qué son los Robots de Trading automáticos?, ¿funcionan? - Admirals.” <https://admiralmarkets.com/es/education/articles/automated-trading/robots-forex> (accessed Oct. 21, 2021).
- [13] “¿Cómo utilizar Fibonacci en nuestra operativa ‘Pablo Gil 20.03.2019’? - YouTube.” [https://www.youtube.com/watch?v=IY9VnwSurXE&t=4s&ab\\_channel=M%C3%A9todoTrading](https://www.youtube.com/watch?v=IY9VnwSurXE&t=4s&ab_channel=M%C3%A9todoTrading) (accessed Oct. 21, 2021).
- [14] “Curso de Scalping con Pablo Gil - YouTube.” [https://www.youtube.com/watch?v=cGGUY0nrCg&t=788s&ab\\_channel=XTBEspa%C3%B1a](https://www.youtube.com/watch?v=cGGUY0nrCg&t=788s&ab_channel=XTBEspa%C3%B1a) (accessed Oct. 21, 2021).
- [15] “API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos.” <https://www.bbvaapimarket.com/es/mundo-api/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos/> (accessed Oct. 13, 2021).
- [16] “Portfolio Optimization with R/Rmetrics - Google Libros.” <https://books.google.es/books?hl=es&lr=&id=keDLpq86DU4C&oi=fnd&pg>

- =PR11&dq=W%C3%BCrtz,+Chalabi+et.al,+2009&ots=4AcZ7jBxkJ&sig=dzC7RsYAMjU-uMDGKGoGF8RAYfQ#v=onepage&q=W%C3%BCrtz%2C%20Chalabi%20et.al%2C%202009&f=false (accessed Oct. 14, 2021).
- [17] "Python 3: los fundamentos del lenguaje - Sébastien Chazallet - Google Libros."  
[https://books.google.es/books?hl=es&lr=&id=KRYyvKmZvpwC&oi=fnd&pg=PA81&dq=ventajas+python+&ots=UG68CuicL3&sig=-RalCURhg3VVKcNYVbFP\\_\\_muru8#v=onepage&q=ventajas%20python&f=false](https://books.google.es/books?hl=es&lr=&id=KRYyvKmZvpwC&oi=fnd&pg=PA81&dq=ventajas+python+&ots=UG68CuicL3&sig=-RalCURhg3VVKcNYVbFP__muru8#v=onepage&q=ventajas%20python&f=false) (accessed Oct. 14, 2021).
- [18] "PyAlgoTrade documentation — PyAlgoTrade 0.20 documentation."  
<http://gbeced.github.io/pyalgotrade/docs/v0.20/html/#indices-and-tables> (accessed Oct. 15, 2021).
- [19] "yfinance · PyPI." <https://pypi.org/project/yfinance/> (accessed Oct. 15, 2021).
- [20] "Overview — Matplotlib 3.4.3 documentation."  
<https://matplotlib.org/stable/contents.html> (accessed Oct. 15, 2021).
- [21] "pandas - Python Data Analysis Library." <https://pandas.pydata.org/> (accessed Oct. 15, 2021).
- [22] "NumPy." <https://numpy.org/> (accessed Oct. 15, 2021).
- [23] "PyCharm: el IDE de Python para desarrolladores profesionales, por JetBrains." <https://www.jetbrains.com/es-es/pycharm/> (accessed Oct. 15, 2021).
- [24] "Qué es Github y qué es lo que le ofrece a los desarrolladores."  
<https://www.xataka.com/basics/que-github-que-que-le-ofrece-a-desarrolladores> (accessed Oct. 20, 2021).
- [25] "PyAlgoTrade - Algorithmic Trading." <https://gbeced.github.io/pyalgotrade/> (accessed Oct. 20, 2021).
- [26] "ikerlarrauri17/Algoritmo-backtesting-trading: Este es el código del backtesting del algoritmo de trading desarrollado."  
<https://github.com/ikerlarrauri17/Algoritmo-backtesting-trading> (accessed Oct. 21, 2021).
- [27] "Drawdown y Rentabilidad: Una pareja peculiar - Rankia."  
<https://www.rankia.com/blog/trading-para-dummies/2017883-drawdown-rentabilidad-pareja-peculiar> (accessed Oct. 16, 2021).
- [28] "El impacto ambiental de la nube | Opciones | Consumo conciente."  
<https://opcions.org/es/articulo/el-impacto-ambiental-de-la-nube/> (accessed Oct. 15, 2021).

## Anexos

### **Acrónimos:**

API – Application Programming Interface

IDE – Integrated Development Environment

CSV – Comma Separated Values

SQL – Structured Query Language

IA – Inteligencia Artificial

CFD – Contract for Difference

SOR – Smart Order Router

HTTP – Hypertext Transfer Protocol