

Universidad Nacional de Educación a Distancia



**Máster Universitario en Investigación en Ingeniería  
De Software Y Sistemas Informáticos**

Código 31105151 — Trabajo Fin de Máster

**PREDICCIÓN DE PRECIOS DE CRIPTOMONEDAS  
MEDIANTE UN MODELO DE INFERENCIA DIFUSO**

Memoria presentada por:

GUSTAVO PLAZA ROMA

Bajo la dirección de:

PEDRO JAVIER HERRERA CARO

Curso Académico 2020/2021

Septiembre 2021



Universidad Nacional de Educación a Distancia



**Máster Universitario en Investigación en Ingeniería  
De Software Y Sistemas Informáticos**

Código 31105151 — Trabajo Fin de Máster

**PREDICCIÓN DE PRECIOS DE CRIPTOMONEDAS  
MEDIANTE UN MODELO DE INFERENCIA DIFUSO**

Trabajo Tipo B

Memoria presentada por:

GUSTAVO PLAZA ROMA

Bajo la dirección de:

PEDRO JAVIER HERRERA CARO



# DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA LA DEFENSA DEL TRABAJO FIN DE MASTER

Fecha:10/09/2021

Quién suscribe:

Autor(a): Gustavo Plaza Roma  
D.N.I./N.I.E./Pasaporte.: 05717685T

Hace constar que es la autor(a) del trabajo:

Título completo del trabajo.  
PREDICCIÓN DE PRECIOS DE CRIPTOMONEDAS MEDIANTE UN MODELO DE INFERENCIA DIFUSO

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

## DECLARACIÓN:

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

Fdo.







**Impreso TFDM05\_AutorPbl. Autorización de publicación  
y difusión del TFM para fines académicos**

## Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es





# Resumen

En este trabajo se propone un sistema automático que opera el mercado de las criptomonedas. En este sistema se generan señales de compra y venta a partir de los datos del precio de las mismas. El sistema propuesto genera las señales de compra y venta usando indicadores técnicos clásicos, los usados en el trabajo son: RSI estocástico, Oscilador Estocástico y Nube de Ichimoku. Estos indicadores se transforman en tres variables de entrada difusas y una variable de salida que indica si existe una señal de compra o venta. El sistema de inferencia que se ha usado es del tipo Mamdani y utiliza un determinado número de reglas aplicadas en distintas estrategias. La implementación propuesta permite realizar operaciones en tiempo real y también probar estrategias aplicadas a periodos anteriores (*backtesting*). El software desarrollado consiste en un sistema distribuido compuesto por distintos nodos, lo que hace que sea simple y configurable. Unos nodos se encargan de ejecutar una estrategia difusa de *trading* para un par de criptomonedas, otro nodo se encarga de interactuar con el *exchange* donde se realizan las operaciones y otro nodo se encarga de presentar los resultados al usuario. El sistema difuso es aplicado a distintas criptomonedas (Bitcoin, Ethereum, Cardano y Binance Coin) en distintos intervalos temporales y los resultados obtenidos se comparan con los resultados que se obtendrían con un sistema basado en reglas clásicas.

**Keywords**— Lógica difusa, trading automático, criptomonedas, indicadores técnicos

# Abstract

In this work, an automatic system that operates the cryptocurrency market is proposed. In this system, buy and sell signals are generated using data price. The proposed system generates buy and sell signals using classic technical indicators, the ones used in the work are: Stochastic RSI, Stochastic Oscillator and Ichimoku Cloud. These indicators are transformed into three fuzzy input variables and an output variable that indicates if there is a buy or sell signal. The inference system is Mamdani type and uses a certain number of rules applied in different strategies. The proposed implementation allows to perform operations in real time and also to test strategies applied in previous periods (*backtesting*). The software developed consists of a distributed system made up of different nodes, which makes it simple and configurable. Some nodes are in charge of executing a fuzzy trading strategy for a pair of cryptocurrencies, another node is in charge of interacting with the exchange where the operations are carried out and another node is in charge of presenting the results to the user. The fuzzy system is applied to different cryptocurrencies (Bitcoin, Ethereum, Cardano and Binance Coin) in different time intervals and the results obtained are compared with the results that they would obtain with a system based on classical rules.

**Keywords**— Fuzzy logic, automatic trading, cryptocurrency, thechnical indicators



# Índice general

Índice general	I
Índice de tablas	III
Índice de figuras	V
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación	2
1.2. Objetivos	2
1.3. <i>Trading</i>	2
1.3.1. Tipos de <i>trading</i>	2
1.3.2. Ventajas del <i>trading</i> algorítmico	3
1.4. Sistemas de <i>trading</i> automáticos para criptomonedas	4
1.5. Descargo de responsabilidad	4
1.6. Organización de la memoria	4
<b>2. Estado del arte: lógica difusa</b>	<b>7</b>
2.1. Introducción a la lógica difusa	7
2.2. Sistemas difusos	8
2.3. Aplicaciones de la lógica difusa	10
2.4. Sistemas difusos y <i>trading</i>	11
<b>3. Solución propuesta</b>	<b>13</b>
3.1. Arquitectura del sistema difuso	13
3.2. Indicadores técnicos	14
3.3. Fuzzificación	17
3.4. Reglas difusas	20
3.5. Defuzzificación	20
3.6. Estrategias	20
3.6.1. Estrategia 1	21
3.6.2. Estrategia 2	23
3.6.3. Estrategia clásica	24
<b>4. Desarrollo del Sistema</b>	<b>27</b>
4.1. Requisitos de diseño	27
4.2. Estructura	28
4.2.1. Comunicación entre nodos	28

4.2.2. Configuración . . . . .	29
4.2.3. Nodos principales . . . . .	30
4.2.4. Obtención de datos . . . . .	32
4.2.5. Abrir posición . . . . .	33
4.3. Cerrar posición . . . . .	33
4.3.1. Intervalo temporal . . . . .	34
4.4. Notificaciones . . . . .	34
4.5. Instalación . . . . .	34
4.6. Ejecución . . . . .	35
4.6.1. Funcionamiento . . . . .	36
4.6.2. Logging . . . . .	36
4.6.3. Comparación del sistema frente a otros . . . . .	37
4.7. Limitaciones . . . . .	38
<b>5. Validación del sistema</b>	<b>39</b>
5.1. Intervalos temporales . . . . .	39
5.2. Resultados . . . . .	40
5.3. Discusión . . . . .	44
<b>6. Conclusiones y Trabajo Futuro</b>	<b>45</b>
6.1. Conclusiones . . . . .	45
6.2. Trabajos futuros . . . . .	45
<b>Referencias</b>	<b>47</b>
<b>Lista de acrónimos</b>	<b>50</b>
<b>A. Conceptos</b>	<b>53</b>

# Índice de tablas

3.1. Reglas difusas de la Estrategia 1 . . . . .	22
3.2. Reglas difusas de la Estrategia 2 . . . . .	24
5.1. Resultados del sistema entre el 1 de enero y 11 de febrero de 2021 . . . . .	42
5.2. Resultados del sistema entre el 1 de mayo y 31 de junio de 2021 . . . . .	43



# Índice de figuras

2.1. Definición de lógica difusa . . . . .	8
2.2. Función de pertenencia . . . . .	9
2.3. defuzzificación usando centroide . . . . .	9
2.4. Métodos de defuzzificación . . . . .	10
3.1. Arquitectura difusa del sistema . . . . .	13
3.2. Nube de Ichimoku . . . . .	18
3.3. Precio de BTC, nube de Ichimoku y RSI . . . . .	18
3.4. Función de pertenencia de las variables de entrada . . . . .	19
3.5. Función de pertenencia de la variable de salida (Posición) . . . . .	20
3.6. Precio de BTC entre 1 enero y 11 de febrero de 2021 . . . . .	21
3.7. Operaciones de compra y venta . . . . .	22
3.8. Funciones de pertenencia . . . . .	23
3.9. Operaciones de compra y venta . . . . .	23
3.10. Operaciones usando la estrategia clásica . . . . .	25
4.1. Arquitectura del <i>bot</i> de <i>trading</i> . . . . .	28
4.2. Uso de gRPC . . . . .	29
4.3. Flujo de ejecución de los nodos de estrategia . . . . .	31
4.4. Binance API . . . . .	32
4.5. Visualización de un mensaje enviado el sistema . . . . .	35
4.6. Intercambio de mensajes para obtener la hora y fecha . . . . .	36
4.7. Mensaje usado para abrir una operación . . . . .	37
5.1. Operaciones en <i>long</i> y <i>short</i> en un intervalo temporal de “1h” . . . . .	40
5.2. Operaciones en <i>long</i> y <i>short</i> en un intervalo temporal de “5m” . . . . .	40
5.3. Estrategia clásica. Operaciones en <i>long</i> y <i>short</i> en un intervalo temporal de “1h” . . . . .	41
A.1. Vela japonesa . . . . .	54





# Capítulo 1

## Introducción

Hoy en día los mercados financieros se han vuelto muy populares debido al alto beneficio que se puede obtener con ellos, sin embargo también son muy complejos y competitivos ya que los inversores necesitan tomar la decisión correcta en el momento oportuno. Los inversores intentan obtener un beneficio del mercado y para ello se usan unas expresiones matemáticas, cuyo objetivo es predecir la tendencia del mercado con antelación, que se llaman indicadores técnicos.

En los próximos capítulos se va a diseñar y desarrollar un sistema que permite realizar operaciones de compra y venta de criptomonedas de forma automática basado en reglas difusas. Estas reglas son definidas usando variables difusas obtenidas a partir de indicadores técnicos.

Este tipo de programas informáticos donde se opera en mercados financieros de forma autónoma, sin intervención humana, se denomina **Trading algorítmico**. Estos programas siguen unas reglas definidas y ejecutan órdenes de compra y venta de los activos. La principal ventaja del *trading* algorítmico es que podría generar beneficios con más rapidez que un humano.

La mayoría de los programas informáticos que operan en mercados financieros utilizan principalmente el mercado Forex debido a su gran volumen de operación diario, que en abril de 2016 fue de 5.09 billones de dólares [1]. En este trabajo en lugar de usar este mercado se va usar el mercado de las criptomonedas debido a los siguientes motivos:

- Opera 24 horas al día. Es un mercado que está siempre disponible incluido los fines de semana.
- Gran volumen de operación, en noviembre de 2019 el volumen comercializado sólo de Bitcoin fue de 22 billones de dolares [2].

La primera criptomoneda se creó en el año 2008 cuando se publicó el *whitepaper* de Bitcoin por “Satoshi Nakamoto”. Desde la creación de Bitcoin hasta hoy el número de criptomonedas se ha multiplicado exponencialmente y en mayo de 2021 se contabilizaban más de 10115 criptomonedas diferentes [3]. Aunque las criptomonedas son muy diferentes entre sí tanto en precio como en aplicación, todas comparten los mismos principios y la tecnología *blockchain*. Estas tecnologías no solo tienen su uso en las criptomonedas sino que *blockchain* implementa un sistema distribuido de transacciones mantenido por una red de nodos descentralizada.

Todas las figuras usadas en el trabajo son de construcción propia, han sido generadas usando diagrams.net [4] o el propio software desarrollado en este trabajo salvo la Figura 4.4 que es una captura de pantalla obtenida durante el desarrollo del trabajo.

## 1.1. Motivación

Nos encontramos en un momento en el que el acceso a sistemas basados en tecnologías de *blockchain* está al alcance de la mayoría de las personas. Las criptomonedas se consideran la principal aplicación de la tecnología *blockchain*, aunque tienen distintas aplicaciones, uno de sus usos principales es la especulación con su precio, lo que ha creado un mercado donde los inversores pueden participar con el fin de obtener un beneficio.

Los modelos basados en lógica difusa se adaptan de forma precisa a entornos donde existe una alta incertidumbre, por lo que analizar los mercados financieros usando sistemas difusos es una buena aproximación. Debido a esto se presenta la oportunidad de analizar, diseñar y desarrollar un sistema que permita realizar *trading* usando una lógica difusa para localizar las señales de entrada y salida del mercado.

## 1.2. Objetivos

Tal y como se ha mencionado en el apartado anterior, el objetivo de este TFM es, en primera instancia, realizar un análisis de los sistemas de *trading* en criptomonedas usando lógica difusa para después poder diseñar e implementar un software que automatice esta tarea.

El objetivo principal es diseñar un sistema autónomo basado en lógica difusa que pueda operar en el mercado de las criptomonedas. Como objetivos secundarios se definen los siguientes:

- Revisión bibliográfica de la lógica difusa aplicada en mercados financieros.
- Desarrollo de un software que implemente el sistema difuso.
- Verificación de la viabilidad del sistema implementado.

Puesto que a lo largo del trabajo se van a usar términos específicos del ámbito del *trading*, se ha creado el Anexo A donde se puede encontrar la definición de los conceptos usados.

## 1.3. *Trading*

El *trading* consiste en comprar y vender activos financieros como acciones, divisas, materias primas o criptomonedas a través de un *broker* con la intención de obtener una rentabilidad económica a corto plazo. Las personas que realizan esta acción se denominan *traders* y sus operaciones se basan en realizar operaciones en largo y operaciones en corto para obtener un beneficio.

### 1.3.1. Tipos de *trading*

Los modalidades de *trading* se pueden clasificar según el tiempo en que las operaciones estén abiertas, basado en esto existen los siguientes tipos:

1. *Day Trading*: las operaciones son abiertas y cerradas dentro del mismo día. Es una inversión a corto plazo ya que no se suele dejar abierta una operación para el día siguiente, este tipo se suele usar sobre todo en la bolsa ya que el “*premarket* del mercado” del día siguiente puede afectar a la operación.
2. *Scalping*: son operaciones a muy corto plazo, el *trader* realiza operaciones que pueden durar segundos.

3. *Swing Trading*: las operaciones pueden dejarse abiertas de un día al siguiente y suelen durar unos tres o cuatro días.
4. *Trading* tendencial o direccional: este tipo de *trading* no tiene un límite de tiempo para las operaciones sino que consiste en abrir posiciones en favor de la tendencia.

El *trading* algorítmico ha ganado mucha popularidad en los últimos años. Aunque no es sencillo saber el número de operaciones realizadas por programas informáticos, se estima que en 2019 en el mercado Forex el 92 % del volumen de *trading* es operado de forma automática [5] y en 2018, el 80 % en las bolsas de valores de los Estados Unidos [6].

La mayoría de los trabajos de investigación publicados sobre el uso de *trading* algorítmico usan inteligencia artificial para implementar el algoritmo como el artículo [7]. Hay muchos mercados diferentes adecuados para aplicar estrategias de *trading* algorítmico como el mercado Forex o los mercados de acciones, pero el aumento de popularidad de las criptomonedas, como Bitcoin, también ofrece nuevas posibilidades para aplicar esos algoritmos. Idealmente, los algoritmos desarrollados en este trabajo de investigación deberían ser aplicables a múltiples mercados, sin embargo, la atención se centrará en los mercados de criptomonedas por ahora.

### 1.3.2. Ventajas del *trading* algorítmico

La tecnología ha permitido automatizar gran cantidad de acciones y el *trading* ha sido una de ellas. Las principales razones que han provocado el desarrollo y crecimiento del *trading* automático son:

- Totalmente automatizado: la selección del activo a operar, identificación de la entrada, identificación de la salida de la posición y la ejecución de órdenes están totalmente automatizadas y no se requiere la participación de un usuario.
- Rentabilidad aproximada conocida: antes de aplicar estos sistemas en el mercado real, son probados aplicando la estrategia implementada con datos históricos del mercado (*backtesting*). Esto permite obtener una rentabilidad aproximada de la estrategia implementada.
- Opera continuamente: los sistemas automáticos pueden estar operando continuamente siempre que esté el mercado abierto. En el caso de las criptomonedas se puede operar durante las 24 horas del día, cualquier día, ya que es un mercado que siempre está abierto.
- Rapidez de ejecución: las operaciones basadas en algoritmos son más rápidas ya que pueden analizar el mercado y tomar decisiones en segundos. Esto permite encontrar entradas más precisas.
- Reducción de errores: puesto que la intervención humana no existe, se reduce la posibilidad de cometer errores humanos, como puede ser abrir una posición con un tamaño demasiado grande o abrir una posición cuando los indicadores técnicos no proporcionan una señal de entrada.
- Menor influencia de las emociones humanas: las emociones como el miedo o la codicia durante el *trading* producen que una posición que debería cerrarse con beneficios, se transforme en una con pérdidas. Hay gran multitud de artículos donde se profundiza en la influencia de las emociones en el *trading* como [8], [9] y [10].
- Diversificación: los sistemas automáticos permiten trabajar en distintos intervalos temporales y aplicar diferentes estrategias al mismo tiempo, lo que permite diversificar de una forma más eficiente la cartera del *trader*.

- Consistencia: cuando se realiza *trading* de forma convencional, seguir un plan de manera efectiva aunque la estrategia esté bien diseñada, es complejo debido a las fluctuaciones del mercado. En un sistema automático, las operaciones siempre se ejecutan siguiendo la estrategia puesto que están automatizadas.

## 1.4. Sistemas de *trading* automáticos para criptomonedas

Hoy en día se pueden encontrar sistemas de *trading* automático tanto gratuitos como de pago, entre ellos algunos permiten operar con criptomonedas. Hay que tener en cuenta que todos ellos requieren de cierto conocimiento para su uso y que antes de usarlos es necesario entender el análisis técnico y utilizarlos como una herramienta de *trading* y no como una caja negra donde introducir dinero y obtener un dinero mayor.

Una de las principales limitaciones que tienen estos sistemas de *trading* públicos ya sean de software libre o de pago es que no prestan mucha atención al *backtesting*. Que esta característica sea rápida y precisa es esencial para diseñar una estrategia y medir su rentabilidad.

Puesto que existe un gran número de opciones de librerías y aplicaciones de software libre para diseñar sistemas de *trading* automático a continuación se muestran las características principales de las más conocidas:

- Jesse [11]: es un aplicación muy recomendada para *backtesting* aunque no permite *trading* real en la versión actual, está previsto implementarlo en versiones futuras.
- FreqTrade [12]: es el sistema de *trading* automático de código abierto en Python más conocido. Es bastante antiguo y tiene gran cantidad de opciones lo que hace que sea complejo de usar.
- Gekko [13]: se encuentra escrito en Javascript. Su librería es compleja y es necesario estar familiarizado con este lenguaje para su uso. La principal ventaja de este sistema es que tiene interfaz gráfica.
- ZenBot [14]: también se encuentra escrito en Javascript. Es un proyecto que está descontinuado desde 2020 aunque su comunidad aún sigue activa.

## 1.5. Descargo de responsabilidad

Invertir en los mercados y criptomonedas es de alto riesgo dado a la volatilidad de los mercados y puede llevar a pérdidas de dinero. El propósito de este trabajo es educativo y nada de lo mostrado en los capítulos posteriores es una recomendación o consejo de inversión de ningún tipo.

## 1.6. Organización de la memoria

La memoria está estructurada como se indica a continuación:

- En el primer capítulo se encuentra una introducción al proyecto donde se habla de la motivación y los objetivos. También se realiza una introducción al *trading*.
- En el capítulo dos se realiza un estudio del estado del arte sobre la lógica difusa centrándose en la aplicación de esta a los sistemas de *trading*.
- En el tercer capítulo se detalla el modelo difuso propuesto. En este capítulo se explican los indicadores técnicos usados y las reglas difusas usadas en el sistema.

- En el cuarto capítulo se describe el funcionamiento del software donde se implementa el modelo propuesto.
- En el quinto capítulo se muestran los resultados obtenidos y se realiza una comparación de los mismos con los resultados obtenidos en un sistema que usa reglas clásicas.
- Por último, en el sexto capítulo se muestran las conclusiones obtenidas y se proponen posibles trabajos futuros.



## Capítulo 2

# Estado del arte: lógica difusa para el análisis técnico de criptomonedas

En este capítulo se va a realizar una revisión literaria de la lógica difusa y su aplicación a los sistemas de *trading*. En primer lugar se muestra una introducción a la definición, modelado y uso de sistemas difusos para después centrarse en la aplicación de estos en el *trading*.

### 2.1. Introducción a la lógica difusa

La lógica difusa se basa en la teoría de conjuntos difusos propuestos por Zadeh en 1965 [15]. La teoría de conjuntos difusos es una generalización de la teoría de conjuntos clásica, es decir, la teoría de conjuntos clásica es un subconjunto de la teoría de conjuntos difusos, como se muestra en la Figura 2.1a.

La Figura 2.1b muestra la función de pertenencia usada para modelar el subconjunto de “buena” calidad para una variable de entrada.

Los conjuntos difusos permiten modelar los eventos u objetos cuyas descripciones sean vagas o exista incertidumbre. Un ejemplo de esto podría ser que al describir que “el agua a 20 grados está templada”, en este caso el término lingüístico “templada” no se refiere a un valor que se pueda considerar verdadero o falso ya que no está claramente definido. Los conjuntos difusos permiten definir la temperatura del agua, haciendo que “templada” sea un valor difuso donde el valor de verdad de “20 grados” está representado por un valor entre 0 y 1.

La nomenclatura usada para definir el grado de pertenencia de un valor  $x$  en una variable lingüística es  $\mu_A(x)$ . Los límites entre las distintas variables lingüísticas no están bien delimitados, por ejemplo no existe una diferenciación clara entre “el agua está caliente” y “el agua está templada” ya que existe cierta superposición dentro de la escala de temperatura. Aunque está claro que el agua a 100 grados no se considera templada sino caliente.

Para transformar las variables clásicas en variables difusas se usa la función de pertenencia  $\mu_A(x)$  y el resultado de esta función es un valor entre 0 y 1 que representa la asociación entre el valor clásico y la variable lingüística. En la Figura 2.2 se muestra la función de pertenencia para la temperatura del agua usado anteriormente.

La forma de las funciones de pertenencia es variable dependiendo de la aplicación, normalmente las formas usadas para construirlas son: triangular, trapezoidal, sigmoide o gaussiana.

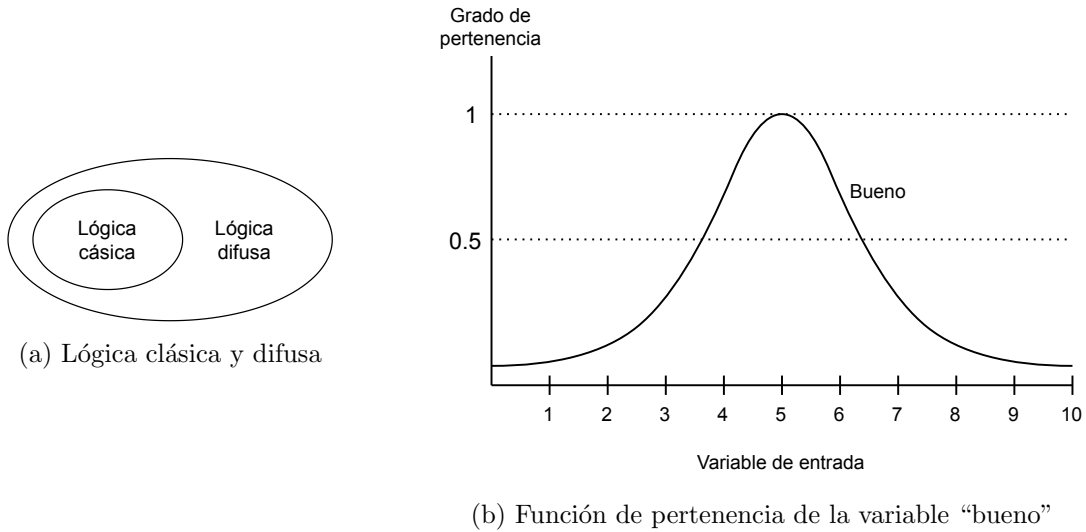


Figura 2.1: Definición de lógica difusa

## 2.2. Sistemas difusos

Los sistemas difusos almacenan el conocimiento utilizando reglas difusas que permiten tratar la información de entrada que es imprecisa. Según se evalúan todas las reglas del sistema, cada una de estas contribuye al valor de la variable de salida. Los conjuntos difusos resultantes de la evaluación de las reglas se combinan para obtener un valor clásico.

Los sistemas de inferencia difusa constan de tres etapas. La primera etapa de fuzzificación (*fuzzification*) donde se convierten los valores clásicos en variables lingüísticas difusas. La segunda etapa consiste en la aplicación de las reglas de la forma “Si ... entonces” donde se procesan las variables difusas y cuya salida es una variable difusa. La última etapa es la llamada defuzzificación (*defuzzification*) que consiste en convertir la variable difusa resultante en una variable clásica.

Las reglas difusas están formadas por dos partes: el antecedente (es la parte del “Si”) que se forma usando la entradas del sistema y los operadores *AND* y *OR* y el consecuente de la regla (es la parte del “entonces”) que se relaciona con la salida de las reglas [16]. Todas las reglas del sistema afectan en cierto grado a la conclusión y normalmente todas las reglas se combinan para producir un único valor clásico de salida.

Para transformar la salida del sistema difuso en una variable clásica se usa la etapa de defuzzificación, de la cual existen varios métodos. Uno de los métodos más usados es el “Centro de gravedad o centroide” donde se busca el punto de equilibrio a partir de la media de los valores de salida de cada una de las reglas. Este es el método que se va a usar en el modelo propuesto en capítulos posteriores. La ecuación con la que se calcula el centroide es 2.1. En la Figura 2.3 se muestra el valor obtenido al aplicar al método “centroide” como defuzzificación.

$$z = \frac{\int \mu(z) \cdot z dz}{\int \mu(z) dz} \quad (2.1)$$

Otros métodos de *defuzzificación* existentes son:

- Máximo Central donde la salida es la media de los valores de salida de aquellos que tienen un



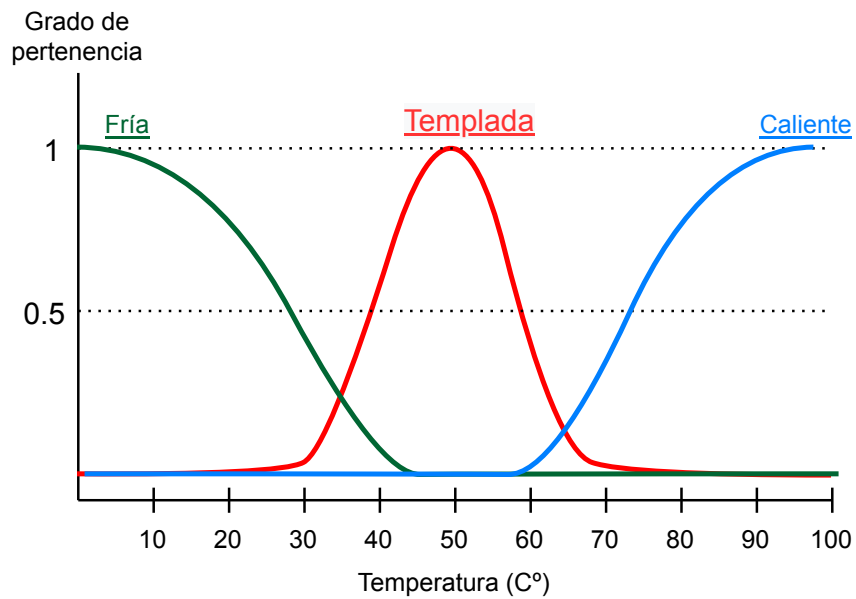


Figura 2.2: Función de pertenencia

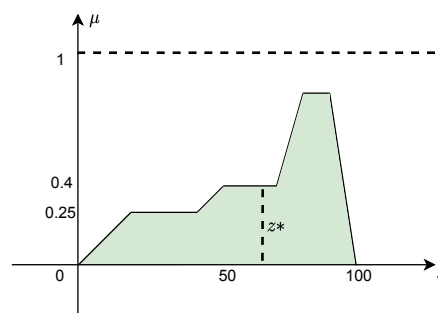


Figura 2.3: defuzzificación usando centroide

valor más alto de función de pertenencia.

- Máximo más pequeño donde la salida es el mínimo de los valores de salida de aquellos que tienen un valor más alto de función de pertenencia.
- Máximo más grande donde la salida es el máximo de los valores de salida de aquellos que tienen un valor más alto de función de pertenencia.

Se puede encontrar una lista detallada de los métodos de defuzzificación en el artículo de investigación [17]. En la Figura 2.4 se muestra la visualización del valor clásico obtenido usando distintos métodos de defuzzificación.

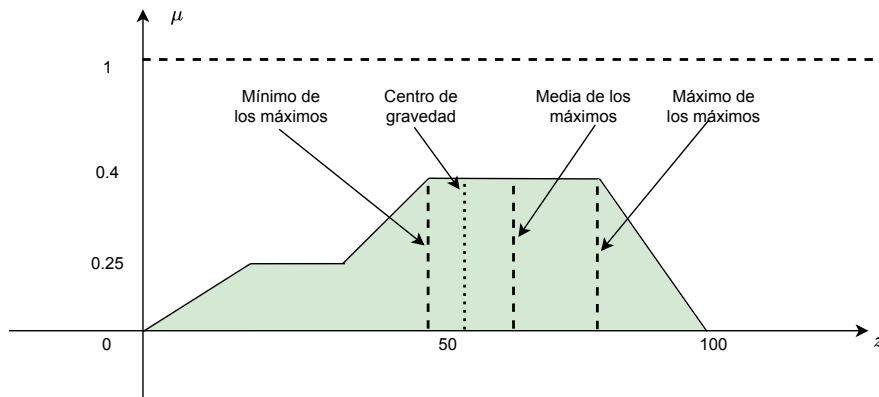


Figura 2.4: Métodos de defuzzificación

### 2.3. Aplicaciones de la lógica difusa

Aunque en este trabajo se ha usado la lógica difusa para predecir el precio de distintas criptomonedas a partir de indicadores técnicos difusos, esta se ha aplicado de forma satisfactoria en distintos campos durante los últimos años. Algunos ejemplos de aplicación se muestran a continuación:

En [18] se utiliza un modelo de inferencia difusa para determinar los lugares óptimos donde instalar parques eólicos y solares. Para ello se usan variables difusas que representan la velocidad del viento, energía solar o elevación del terreno, y con estas se deduce si el lugar es óptimo para la instalación del parque.

En [19] se estudia la madurez del tomate en base a un conjunto de reglas difusas. Para la construcción de las reglas difusas se usan parámetros característicos de los tomates como el tamaño, forma o color.

En [20] se propone una red neuronal difusa para ajustar los parámetros del lavado de una lavadora. El objetivo es reducir el consumo de agua y energía durante el lavado, para ello se usa un control difuso que ajusta la temperatura a la que calentar el agua y la velocidad del motor, usando como entrada la temperatura actual del agua y la cantidad de suciedad disuelta en el agua.

En [21] se usa un sistema de control difuso para calcular la corriente aplicada en una tubería enterrada para protegerla de la corrosión y otros agentes externos. El objetivo del sistema propuesto es proteger la tubería a la vez que se minimiza la energía empleada. Se ha usado un sistema difuso ya que las condiciones del suelo a lo largo de una tubería son muy complejas de modelar. En total el sistema cuenta con más de 100 reglas.

En [22] donde se presenta un sistema de apoyo a la toma de decisiones usando métodos difusos para automatizar la aplicación de guías de práctica clínicas. Estas guías usadas son unos escenarios que se le pueden presentar a un médico. La dificultad reside en la descripción lingüística ya que muchas pruebas médicas muestran la presencia de la enfermedad de forma probabilística y las acciones que deben tomar los médicos pueden ser vagas y contradictorias. Estas acciones se adaptan bien para usar métodos difusos. En el trabajo se usa un modelo difuso donde las entradas son los resultados de las pruebas médicas y la salida es la presencia de la enfermedad.

Otros campos de aplicación son la robótica [23], conducción autónoma [24], reconocimiento de patrones en imágenes [25] y las bases de datos [26] a través del lenguaje FQL [27].

## 2.4. Sistemas difusos y *trading*

En la literatura de los últimos años existen distintos autores que han propuesto sistemas difusos aplicados al *trading*. La mayoría de estos artículos se pueden dividir en tres enfoques distintos que se detallan a continuación.

La primera propuesta y más popular consiste en un sistema de *trading* que puede ser autónomo o un sistema de apoyo para un experto y consiste en seleccionar una serie de indicadores técnicos clásicos donde los más comunes son las medias móviles, MACD, RSI y SO. Después estos indicadores clásicos se convierten en variables difusas usando funciones de pertenencia. Una vez que se dispone de las variables difusas, se definen unas reglas difusas cuya salida es comprar, vender o mantener. Por último la salida de estas reglas se convierten en variables clásicas y se opera en el mercado. Este tipo de sistema se puede ver en [28] [29] y [30]. En la mayoría de estos sistemas se usa un sistema de inferencia difuso del tipo Mamdani que fue propuesto en [31] en el año 1977.

Otro enfoque también usado por distintos autores consiste en realizar un modelado difuso de las velas japonesas. Las velas japonesas están definidas usando cuatro valores obtenidos del precio (apertura, cierre, máximo y mínimo). Usando estos valores se obtienen unas variables difusas a través de funciones de pertenencia. Después al igual que en los artículos mencionados anteriormente hay una serie de reglas difusas cuya salida permite operar en el mercado. Artículos donde se muestran estos modelos son [32] y [33]. Al igual que los artículos anteriores en estos trabajos se usan un sistema de inferencia difuso de Mamdani.

Por último en la literatura existe otro enfoque también usado en los últimos años donde se proponen redes neuronales difusas. En estos artículos los modelos aprenden las reglas usando datos de entrenamiento, en vez de ser estas reglas de entrada definidas por el autor. La mayoría de estas redes neuronales constan de tres capas, en la primera se realiza el cambio de variables clásicas a difusas, en la segunda se aplican las reglas y en la tercera y última se realiza de nuevo el cambio de variables difusas a variables clásicas. Esta propuesta se muestra en [34] y [35]. Al contrario que en los otros dos enfoques mencionados anteriormente en estos trabajos se usa un sistema de inferencia difuso de Sugeno que fue propuesto en 1985 en [36].

Tras la revisión bibliográfica realizada se observa que existen diferentes perspectivas a la hora de aplicar la lógica difusa a la toma de decisiones en mercados financieros. La mayor parte de los artículos publicados en los últimos años se centran en tres esquemas distintos. El primero consiste en seleccionar un conjunto de indicadores técnicos y evaluar las compras y ventas usando reglas. El segundo consiste en usar redes neuronales difusas en conjunto con algoritmos genéticos y la tercera consiste en usar un análisis difuso de las velas japonesas.

Entre los indicadores técnicos usados se observa que la mayoría de artículos utilizan indicadores bastantes simples como las medias móviles y pocos de ellos usan indicadores complejos y modernos.

Todos los artículos estudiados aplican el modelo propuesto a índices, acciones y Forex pero ninguno de ellos es aplicado a criptomonedas como se va a realizar en este trabajo.



## Capítulo 3

# Solución propuesta

En este capítulo se muestra el sistema de inferencia difuso propuesto. La finalidad es exponer de forma detallada como el sistema es capaz de realizar la predicción del precio usando un conjunto de reglas difusas, para después en capítulos posteriores comparar los resultados obtenidos con un sistema basado en reglas clásicas. En primer lugar se muestra la estructura del sistema difuso, a continuación los indicadores técnicos usados en la estrategia, después se explica su conversión a variables de entrada difusas y por último se muestran las reglas y la etapa de defuzzificación.

### 3.1. Arquitectura del sistema difuso

El sistema implementado en este trabajo toma como entrada datos del precio de las criptomonedas, ya sean datos históricos o datos en tiempo real, con estos datos se calculan los valores de tres indicadores técnicos, estos se exponen con más detalles en el apartado 3.2. Los valores clásicos se transforman en variables difusas que son tratadas en un sistema de inferencia difuso de tipo Mamdani basado en reglas. Por último la salida de este sistema se transforma en órdenes de compra y venta. En la Figura 3.1 se muestra un esquema del sistema propuesto.

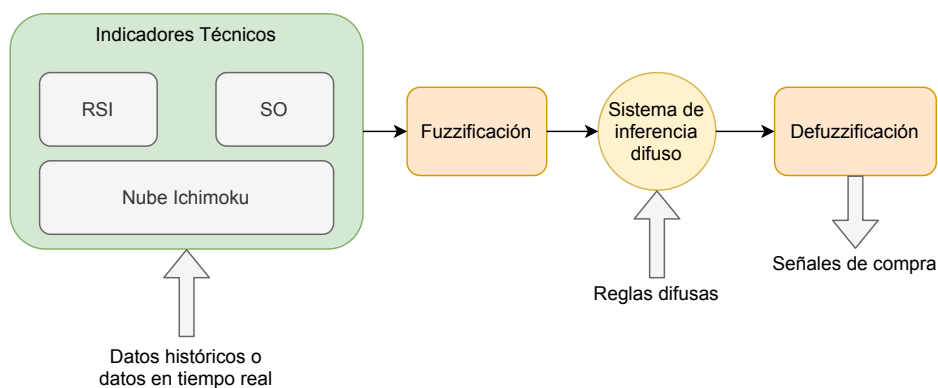


Figura 3.1: Arquitectura difusa del sistema

## 3.2. Indicadores técnicos

Los indicadores técnicos son expresiones matemáticas que a partir de datos históricos de los mercados intentan predecir el precio futuro de un activo. Existen gran cantidad de indicadores, algunos simples como una media móvil para medir la tendencia del mercado o algunos mucho más complejos que predicen zonas de soporte y resistencia del precio. En [37] se muestran algunos de los indicadores técnicos más usados hoy en día en los mercados financieros.

Los indicadores usados en este trabajo para construir una estrategia son: el oscilador estocástico (*stochastic oscillator*), el RSI estocástico (*Stoch RSI*) y la nube de Ichimoku (*Ichimoku cloud*).

La decisión de seleccionar estos tres indicadores se debe a su correcto funcionamiento en diferentes momentos del mercado. Con estos indicadores se cubren mercados con una fuerte tendencia alcista o bajista donde la nube de Ichimoku proporciona buenas señales de compra y venta y mercados laterales donde el oscilador estocástico funciona correctamente. El RSI estocástico es usado para reforzar las decisiones de compra y venta.

### Oscilador Estocástico (SO)

Los osciladores son un tipo de indicador cuyo valor varía entre cero y cien y son usados para detectar cuando un activo está sobrecomprado o sobrevendido. El oscilador estocástico añade una componente aleatoria a los osciladores, fue definido por Gorge Lane en 1970. La definición de este indicador según [38] y que es la que usaremos a partir de ahora es la siguiente:

$$\%K = \frac{Close - Lo}{Hi - Lo} \cdot 100 \quad (3.1)$$

Donde:

- *Close* es el precio de cierre del último periodo.
- *Hi* es el precio más alto del periodo para el que se está calculando el indicador.
- *Lo* es el precio más bajo del periodo para el que se está calculando el indicador.

Las características de este indicador son:

- El indicador consiste en dos líneas *%K* y *%D*. El cálculo de *%K* se muestra en la ecuación 3.1 y *%D* normalmente se calcula tres puntos sobre la media de *%K*.

$$\%D = 3 - \text{Media de \%K en el periodo} \quad (3.2)$$

- Compara el último precio de cierre del activo con el rango en el que se ha operado dicho activo en un periodo de tiempo específico.
- Este indicador puede dar falsas señales cuando un activo se encuentra en una fuerte tendencia alcista o bajista pero proporciona buenas señales de compra y venta cuando el activo se encuentra en un rango.
- Una característica muy importante de este indicador es que las líneas de *%K* y *%D* se pueden cruzar según varía el precio del activo. Estos cruces producen señales alcistas o bajistas. En apartados posteriores se usan estos cruces para producir señales de compra y venta.

### Índice de fuerza relativa (*Relative strength index*)

Este indicador no es usado directamente en el sistema propuesto pero es necesario definirlo para poder obtener otro indicador que si es usado. El índice de fuerza relativa (RSI) es un indicador técnico ampliamente usado propuesto por Wilder en 1978 en [39]. Este indicador es un oscilador de momento que mide la velocidad y cambio de los movimientos del precio. Su valor oscila entre 0 y 100. El cálculo del RSI en un periodo específico de tiempo mostrado en [40] y que es el utilizado en este trabajo se define de la siguiente manera:

$$RSI = 100 - \frac{100}{1 + RS} \quad (3.3)$$

donde  $RS$  es la fuerza relativa de los últimos  $n$  periodos y se define como:

$$RS = \frac{\text{EMA de } U}{\text{EMA de } D} \quad (3.4)$$

Para cada uno de los  $n$  periodos se calcula  $U$  y  $D$  de la siguiente manera:

$$\begin{aligned} U_i &= \text{Close}_i - \text{Close}_{i-1} \\ D_{i-1} &= \text{Close}_i - \text{Close}_i \end{aligned} \quad (3.5)$$

Si  $U$  es positivo para un periodo, entonces  $D$  se reemplaza con cero para ese periodo y viceversa, si  $D$  es positivo para un periodo, entonces  $U$  se reemplaza con cero para el periodo respectivo. Para calcular el RSI, se determina una media móvil exponencial (EMA) para  $U$  y para  $D$ .

Las características de este indicador son:

- Normalmente se calcula el RSI usando 14 periodos.
- Se considera que el activo está sobrevendido cuando los valores de RSI son menores de 30, aunque en mercados volátiles se puede usar 20.
- Se considera que el activo está sobrecomprado cuando los valores de RSI son mayores de 70, aunque en mercados volátiles se puede usar 80.
- Las señales de compra y venta pueden estar provocadas por divergencias entre la evolución del RSI y el precio que es el dato usado para calcular el indicador. Por ejemplo, cuando el valor del RSI alcanza un máximo local mayor al anterior y el precio alcanza un máximo local por debajo del anterior, el indicador proporciona una señal de compra.

### RSI estocástico

El RSI estocástico (*Stoch RSI*) es una extensión del RSI. Al contrario que el oscilador estocástico que es aplicado al precio, el *Stoch RSI* aplica la fórmula estocástica al índice de fuerza relativa. Desarrollado por Chande y Kroll en 1994, el *Stoch RSI* tiene como objetivo mitigar los problemas relacionados con el RSI.

La definición del *Stoch RSI* descrita en [41] y usada en este trabajo es la siguiente:

$$\text{Stoch RSI} = \frac{RSI - RSI_{LL}}{RSI_{HH} - RSI_{LL}}, \quad 0 < \text{Stoch RSI} < 1 \quad (3.6)$$

Donde:

- $RSI$  es el valor actual del RSI.

- $RSI_{LL}$  es el valor de RSI más bajo de los últimos  $n$  periodos.
- $RSI_{HH}$  es el valor de RSI más alto de los últimos  $n$  periodos.

Los límites del *Stoch RSI* se muestran en 3.7 y 3.8.

$$\text{Sobrecompra si Stoch RSI} > 0,7 \quad (3.7)$$

$$\text{Sobreventa si Stoch RSI} < 0,3 \quad (3.8)$$

Las características de este indicador son:

- Normalmente se calcula el *Stoch RSI* usando 14 periodos.
- El *Stoch RSI* se considera sobrevendido cuando el valor cae por debajo de 0.30.
- El *Stoch RSI* se considera sobrecomprado cuando el valor sube por encima de 0.70.

### Nube de Ichimoku (*Ichimoku Cloud*)

El indicador nube de Ichimoku fue propuesto por Goichi Hosada en la década de los sesenta aunque no ha sido hasta 2008 cuando se ha convertido en un indicador popular.

Este indicador está compuesto por cinco líneas: Tenkan Sen, Kijun Sen, Chikou Span, Senkou Span A y Senkou Span B. Para el cálculo de estas líneas se ha usado el modelo propuesto en [42] y se detalla a continuación.

- Tenkan Sen muestra un movimiento de precios a corto plazo, está calculado usando el precio en los últimos 9 periodos. Es el punto medio de los precios más altos y más bajos de un activo durante los últimos 9 períodos.

$$\text{Tenkan Sen} = \frac{\text{Máximo de 9 periodos} + \text{Mínimo de 9 periodos}}{2} \quad (3.9)$$

- Kijun Sen es el precio medio de los últimos 26 períodos. Por lo general, se usa junto con Tenkan Sen para identificar oportunidades de compra y venta. Cuando el precio está por encima del Kijun Sen, el impulso del precio aumenta y cuando el precio está por debajo del Kijun Sen, el impulso del precio disminuye.

$$\text{Kijun Sen} = \frac{\text{Máximo de 26 periodos} + \text{Mínimo de 26 periodos}}{2} \quad (3.10)$$

- Chinkou Span es el precio actual desplazado hacia atrás 26 períodos. Una tendencia tiende a ser al alza cuando Chinkou Span está por encima del precio y a la baja cuando el Chinkou Span está por debajo del precio.

$$\text{Chinkou Span} = \text{Precio actual retardado 26 periodos} \quad (3.11)$$

- Por último, Senkou Span A y Senkou Span B son dos líneas que forman la nube Kumo. Cuando Senkou Span A está por encima de Senkou B, es una señal alcista y cuando Senkou Span B está por encima de Senkou Span A, muestra una señal bajista. La formulación de Senkou Span A y Senkou Span B se muestra en las ecuaciones 3.12 y 3.13 respectivamente.



$$\text{Senkou Span A} = \frac{\text{Tenkan Sen} + \text{Kijun Sen}}{2} \quad (3.12)$$

Proyectado 26 periodos hacia adelante

$$\text{Senkou Span B} = \frac{\text{Máximo de 52 periodos} + \text{Mínimo de 52 periodos}}{2} \quad (3.13)$$

Proyectado 26 periodos hacia adelante

- Nube de Ichimoku (Kumo) es la diferencia entre Senkou Span A y Senkou Span B.

$$\text{Kumo} = \text{Senkou Span A} - \text{Senkou Span B} \quad (3.14)$$

- La nube de Ichimoku se puede utilizar tanto al alza como a la baja. Este indicador no proporciona señales correctas cuando el mercado no tiene una tendencia clara.
- Normalmente se usan 9, 26 y 52 periodos para el cálculo del indicador.
- Cuando el precio se encuentra sobre la nube de Ichimoku, indica que la tendencia es alcista.
- Cuando el precio se encuentra por debajo de la nube de Ichimoku, indica que la tendencia es bajista.
- Chikou Span ofrece información sobre la fuerza de la señal. Ante una posible señal de compra, Chikou Span debería situarse por encima de la nube y de Kijun.

En la Figura 3.2 se muestra el precio del Bitcoin y las líneas del indicador Ichimoku. En color azul se muestra la línea de Tenkan obtenida usando la ecuación 3.9, en amarillo se muestra la línea de Kijun obtenida usando la ecuación 3.10, en color verde o rojo se muestra la nube de Ichimoku obtenida mediante la ecuación 3.14. Por último en color negro se muestra el precio de Bitcoin.

En la Figura 3.3 se muestran los indicadores técnicos usados, se observa tanto la nube de Ichimoku, el oscilador estocástico se muestra en la línea de color azul, la línea amarilla representa %D (ecuación 3.2). Estas dos líneas son usadas para calcular el cruce con el oscilador estocástico.

Tras obtener los valores de los indicadores técnicos se lleva a cabo la fuzzificación de estos en variables difusas que serán la entrada del sistema de inferencia.

### 3.3. Fuzzificación

Para realizar la fuzzificación se van a usar funciones de pertenencia triangulares para las variables de entrada. En la Figura 3.4 se muestra la función de pertenencia del indicador de la nube de Ichimoku para ello se han usado tres variables lingüísticas (“bajo”, “medio” y “alto”). Para el resto de indicadores usados, *Stoch RSI* y cruce del oscilador estocástico, también se usan funciones de pertenencia triangulares.

Para obtener la variable difusa asociada a la nube de Ichimoku en primer lugar es necesario conocer si el precio se encuentra por encima o por debajo de esta. Para comprobar si el precio se encuentra por encima, por debajo o dentro de la nube de Ichimoku habría que comprobar que se cumplen las ecuaciones 3.15, 3.16 y 3.17 respectivamente.

$$\text{Debajo de la nube si } \text{Cierre} < \text{Senkou A} \text{ y } \text{Cierre} < \text{Senkou B} \quad (3.15)$$

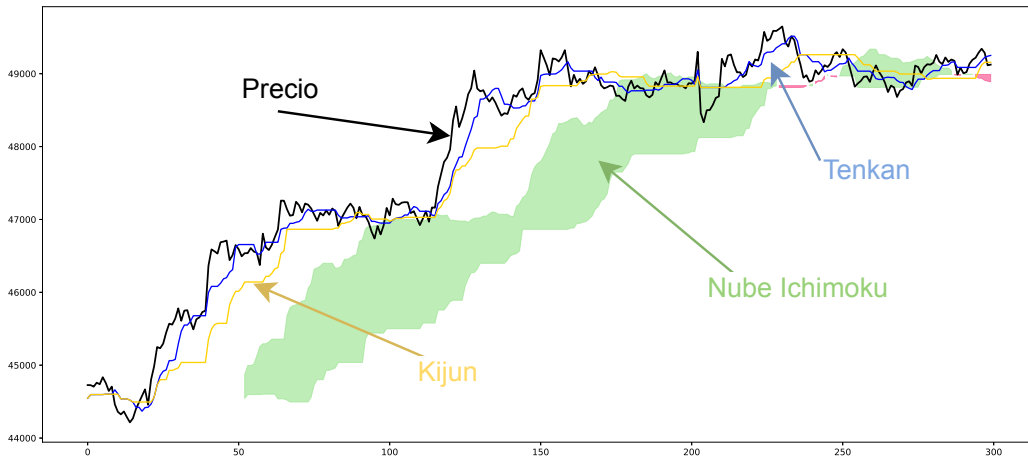


Figura 3.2: Nube de Ichimoku

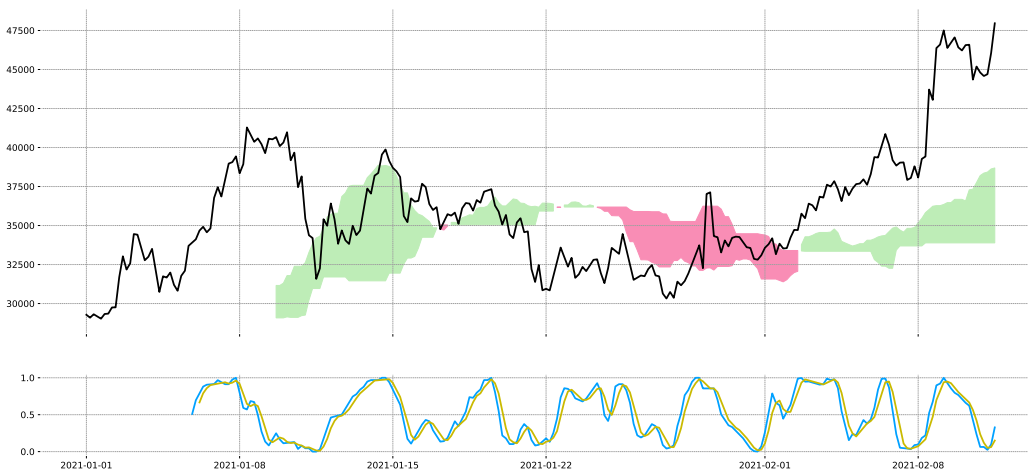


Figura 3.3: Precio de BTC, nube de Ichimoku y RSI

Encima de la nube si Cierre > Senkou A y Cierre > Senkou B (3.16)

Dentro de la nube si Cierre < Senkou A y Cierre > Senkou B (3.17)

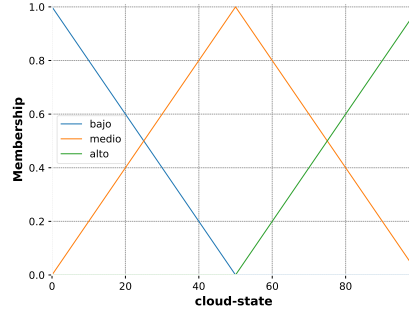


Figura 3.4: Función de pertenencia de las variables de entrada

La variable lingüística asociada a un determinado valor se obtiene al aplicar la función de pertenencia al resultado obtenido en la ecuación 3.18.

$$Nube_t = \begin{cases} 100 - (\text{Senkou A} - \text{Cierre}) & \text{Si Encima de la Nube} \\ 50 + \frac{\text{SenkouA} - \text{SenkouB}}{2} & \text{Si dentro de la Nube} \\ 0 + (\text{Cierre} - \text{Senkou B}) & \text{Si Debajo de la Nube} \end{cases} \quad (3.18)$$

Usando las dos líneas del oscilador estocástico (%K y %D), se puede definir un parámetro que indica el cruce entre ellas. Estos cruces pueden ser:

- Cruce bajista se produce cuando se cumple 3.19 indicando una señal de venta.
- Cruce alcista cuando se cumple 3.20 indicando una señal de compra.

$$\text{Cruce bajista en } t \text{ si } \%K_t < \%D_t \text{ y } \%K_{t-1} > \%D_{t-1} \quad (3.19)$$

$$\text{Cruce alcista en } t \text{ si } \%K_t > \%D_t \text{ y } \%K_{t-1} < \%D_{t-1} \quad (3.20)$$

Para calcular el valor del cruce bajista o alcista usamos las ecuaciones 3.21 y 3.22

$$\text{Valor Cruce}_t = \begin{cases} -|\%K_t - \%D_t| \cdot \%K_t & \text{Si Cruce bajista} \\ | \%K_t - \%D_t| \cdot \%K_t & \text{Si Cruce alcista} \end{cases} \quad (3.21)$$

La variable lingüística asociado a un determinado valor de cruce se obtiene al aplicar la función de pertenencia al resultado obtenido en la ecuación 3.22.

$$\text{Cruce}_t = \begin{cases} (\text{Valor Cruce})_t \cdot 100 \cdot (0,4 \cdot \frac{1}{0,4}) & \text{Si } (\text{Valor Cruce})_t > 0,2 \\ 0 & \text{Si } (\text{Valor Cruce})_t < -0,2 \\ (\text{Valor Cruce})_t \cdot 100 \cdot \frac{1}{0,4} & \text{En otro caso} \end{cases} \quad (3.22)$$

La variable lingüística asociado a un determinado valor del indicador *Stoch RSI* se obtiene al aplicar la función de pertenencia al resultado obtenido en la ecuación 3.23.

$$\text{Stoch RSI} = \text{Stoch RSI} \cdot 100 \quad (3.23)$$

### 3.4. Reglas difusas

Tras aplicar las ecuaciones mostradas en el apartado anterior, los indicadores técnicos ya se encuentran transformados en variables lingüísticas. A continuación se definen una serie de reglas que indican si es necesario comprar, vender o no realizar ninguna de ellas. Estas reglas son del tipo condicional “Si ... entonces ...” y compuestas de dos partes: en la primera se definen las condiciones de activación usando las variables de entrada (“Si *RSI es bajo* y *Cruce RSI es alto*”) y en la segunda parte se encuentra la conclusión que se relaciona con la salida del sistema (“entonces *Comprar*”). Un ejemplo de regla usada en el sistema sería:

SI ‘Nube es bajo’ y ‘RSI es bajo’ y ‘Cruce RSI es alto’ ENTONCES ‘Comprar’

Puesto que las salidas de las reglas son variables lingüísticas es necesario realizar la etapa de defuzzificación.

### 3.5. Defuzzificación

La variable de salida del sistema como ya se ha mencionado anteriormente indica si hay que comprar o vender una determinada criptomoneda. Esta variable se ha denominado “Posición”.

La función de transferencia de la variable de salida que indica la compra o venta del activo se muestra en la Figura 3.5, también es una función triangular y está formada por tres variables lingüísticas (“vender”, “indiferente” y “comprar”).

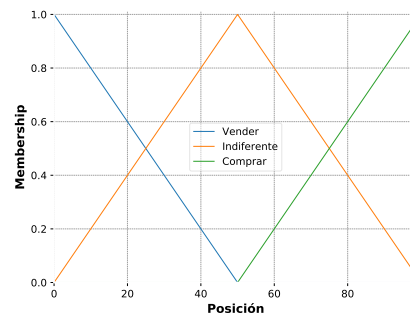


Figura 3.5: Función de pertenencia de la variable de salida (Posición)

Para llevar a cabo el proceso de defuzzificación de la variable de salida se va usar el método centroide (centro de masas) definido en el apartado 2.2. Tras la defuzzificación se obtiene un valor clásico de la variable de salida entre 0 y 100. Con este valor y usando una estrategia se decide si entrar o salir de una operación.

### 3.6. Estrategias

En este apartado se proponen dos estrategias distintas usando los indicadores técnicos y el sistema difuso propuestos en los apartados anteriores. Las dos estrategias se denominarán a partir de ahora “Estrategia 1” y “Estrategia 2” y a continuación se detallan cada una de ellas.

Para diseñar y ajustar estas estrategias se ha usado el rango temporal desde el 1 de enero de 2021 a las 00:00:00 (el *timestamp* en milisegundos es 1609459200000) hasta el 11 de febrero de 2021 a las 16:00:00 (el *timestamp* en milisegundos es 1609459200000).

### 3.6.1. Estrategia 1

En la Figura 3.6 se muestra la evolución del precio entre las fechas mencionadas anteriormente. Se ha usado un intervalo de 8 horas en las gráficas posteriores del capítulo para que sea más fácil de visualizar aunque el intervalo de 8 horas no se usará para aplicar las estrategias.



Figura 3.6: Precio de BTC entre 1 enero y 11 de febrero de 2021

En esta estrategia se van a usar las funciones de pertenencia mostradas en el apartado 3.3 por lo que los indicadores técnicos se dividen en tres variables lingüísticas que son: bajo, medio y alto.

La estrategia usa un total de nueve reglas para obtener la salida. Las reglas usadas son las mostradas en la Tabla 3.1. La variable de salida (denominada “Posición”) se muestra en el apartado 3.5 y puede tomar tres valores: comprar, vender o indiferente.

Usando los datos históricos del precio anteriormente mencionados se aplica el sistema difuso y tras realizar la etapa de defuzzificación se imponen las siguientes condiciones:

$$\text{Comprar si Posición} > 76 \quad (3.24)$$

$$\text{Vender si Posición} < 24 \quad (3.25)$$

Una vez que se ha abierto una posición la estrategia marca su cierre usando las siguientes condiciones:

- Si el precio toca un *stop loss* del 5%.
- Si se ha realizado una compra (*long*) no se cierra la operación hasta que la variable de salida no deja de indicar señal de compra usando la ecuación 3.24.

Tabla 3.1: Reglas difusas de la Estrategia 1

Condición 1	Condición 2	Condición 3	Salida
RSI es bajo	Cruce RSI es alto	-	Comprar
RSI es alto	Cruce RSI es bajo	-	Vender
Nube es alto	Cruce RSI es bajo	-	Vender
Nube es bajo	Cruce RSI es alto	-	Comprar
Nube es bajo	RSI es bajo	Cruce RSI es alto	Comprar
Nube es alto	RSI es alto	Cruce RSI es bajo	Vender
Nube es medio	RSI es medio	Cruce RSI es medio	Indiferente
Nube es alto	RSI es alto	Cruce RSI es alto	Indiferente
Nube es bajo	RSI es bajo	Cruce RSI es bajo	Indiferente

- Si se ha realizado una venta (*short*) no se cierra hasta que la variable de salida no deja de indicar señal de venta usando la ecuación 3.25.

En la Figura 3.7 se muestra las operaciones de compra y venta que se han realizado por esta estrategia en el periodo temporal estudiado. Las líneas verticales en color rojo indican que son operaciones en *short* y las líneas verticales en color verde indican las posiciones abiertas en *long*. La línea negra muestra el precio de Bitcoin y las franjas horizontales rellenas de color verde y rojo es la nube de Ichimoku.

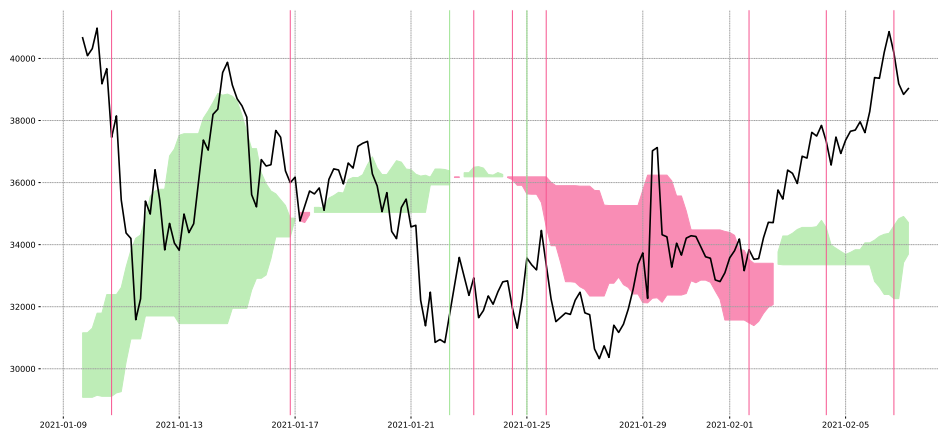


Figura 3.7: Operaciones de compra y venta

En este apartado únicamente se propone la estrategia y las condiciones para abrir y cerrar una posición, en apartados posteriores se analizarán y compararán los resultados.

### 3.6.2. Estrategia 2

En esta segunda estrategia se usan los mismos indicadores que la “Estrategia 1” mostrada en el apartado interior. En esta estrategia se van a cambiar las variables lingüísticas y las reglas para después comparar los resultados.

Las variables de entrada y la salida constan de cinco términos lingüísticos, en lugar de tres como en la estrategia anterior y los valores posibles se muestran en la Figura 3.8.

Para las entradas se han usado las siguientes variables lingüísticas: “muy bajo”, “bajo”, “medio”, “alto” y “muy alto”. Para la salida (“Posición”) se han usado las siguiente variables lingüísticas: “venta”, “venta moderada”, “indiferente”, “compra moderada” y “compra”.

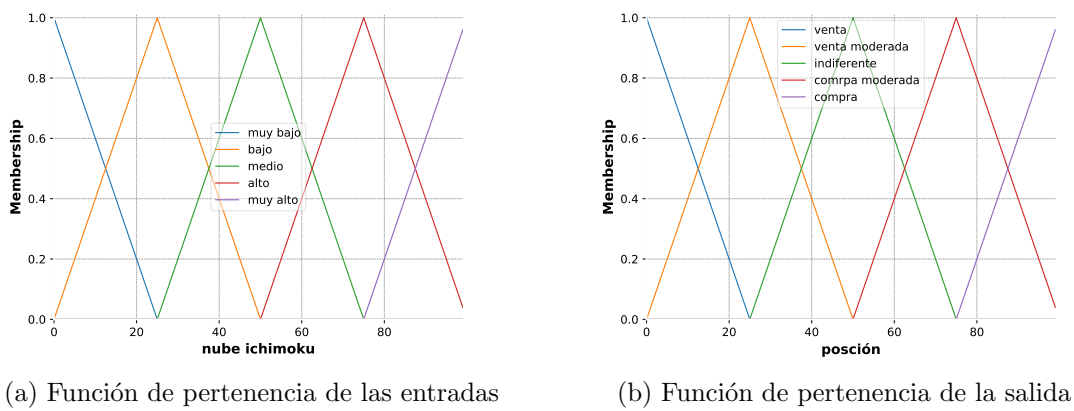


Figura 3.8: Funciones de pertenencia

En esta estrategia se han usado un total de dieciséis reglas difusas que se muestran en la Tabla 3.2. En la Figura 3.9 se muestra las operaciones de compra y venta que se han realizado en corto y en largo.

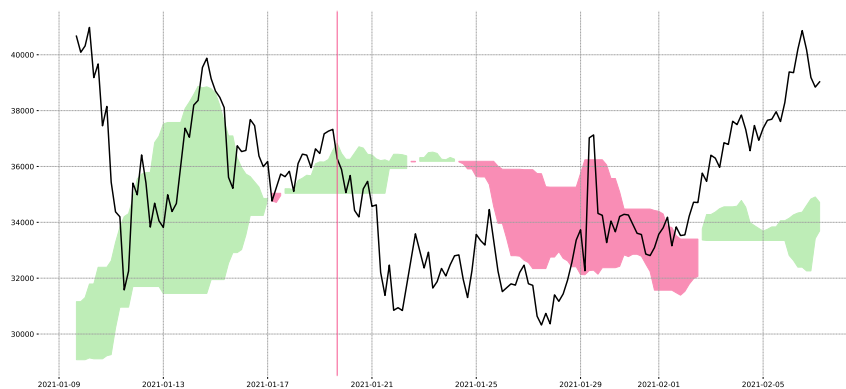


Figura 3.9: Operaciones de compra y venta

Tabla 3.2: Reglas difusas de la Estrategia 2

Condición 1	Condición 2	Condición 3	Salida
Nube es bajo o muy bajo	RSI es muy bajo	Cruce RSI es muy alto	Compra
Nube es bajo o muy bajo	RSI es medio	Cruce RSI es muy alto	Compra moderada
Nube es bajo o muy bajo	RSI es muy bajo	Cruce RSI es medio	Compra moderada
Nube es alto o muy alto	RSI es muy alto	Cruce RSI es muy bajo	Venta
Nube es alto o muy alto	RSI es alto	Cruce RSI es muy bajo	Venta moderada
Nube es alto o muy alto	RSI es muy alto	Cruce RSI es alto	Venta moderada
Nube es medio	RSI es medio	Cruce RSI es medio	Indiferente
Nube es alto o muy alto	RSI es muy alto	Cruce RSI es medio	Indiferente
Nube es bajo o muy bajo	RSI es medio	Cruce RSI es bajo	Indiferente
Nube es alto o muy alto	RSI es medio	Cruce RSI es alto	Indiferente
Nube es bajo o muy bajo	RSI es muy bajo	Cruce RSI es medio	Indiferente
Nube es medio	RSI es muy alto	Cruce RSI es muy bajo	Venta moderada
Nube es medio	RSI es muy bajo	Cruce RSI es muy alto	Compra moderada
Nube es bajo o muy bajo	RSI es muy alto	Cruce RSI es muy bajo	Venta moderada
Nube es alto o muy alto	RSI es muy bajo	Cruce RSI es muy alto	Compra moderada

### 3.6.3. Estrategia clásica

En este apartado se propone una estrategia clásica con la que comparar las estrategias difusas previamente descritas. Para poder hacer la comparación se han usado los mismos indicadores técnicos.

Las condiciones que tienen que cumplir estos indicadores para generar una señal de compra son:

- El precio se encuentra por encima de la nube de Ichimoku.
- El precio está sobrevendido ( $\text{Stoch RSI} < 0,3$ ).
- Se produce un cruce alcista (se cumple la ecuación 3.20).

Por otro lado se generará una señal de venta cuando se cumplen las siguientes condiciones:

- El precio se encuentra por debajo de la nube de Ichimoku.
- El precio está sobrecomprado ( $\text{Stoch RSI} > 0,7$ ).
- Se produce un cruce bajista (se cumple la ecuación 3.19)

En esta estrategia al igual que en las estrategias difusas, para cerrar una posición han de cumplirse las siguiente condiciones:



- Si el precio toca un *stop loss* del 5%.
- Si se ha realizado una compra (*long*) no se cierra la operación hasta que la variable de salida no deja de indicar señal de compra.
- Si se ha realizado una venta (*short*) no se compra hasta que la variable de salida no deja de indicar señal de venta.

Usando estas reglas, en la Figura 3.10 se muestran las compras (líneas verticales en color verde) y ventas (líneas verticales en color rojo) que se realizarían usando esta estrategia. En apartados posteriores se realizará una comparación entre el rendimiento que obtiene esta estrategia para distintas criptomonedas y distintos intervalos con el rendimiento obtenido con las estrategias difusas.

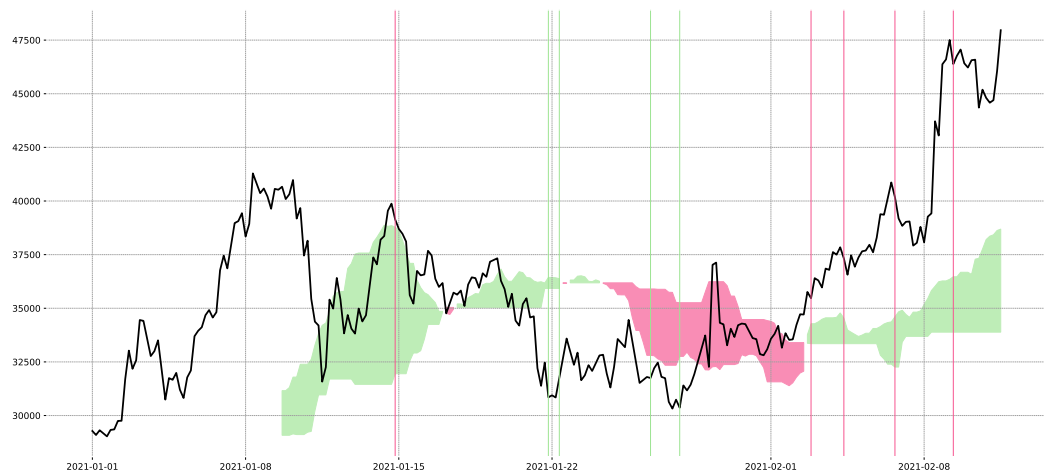


Figura 3.10: Operaciones usando la estrategia clásica



## Capítulo 4

# Desarrollo del Sistema

La solución propuesta en el trabajo es un bot de *trading* automático llamado “GCRYPTO BOT”. Todo el software se encuentra escrito en Python 3, concretamente se ha usado Python 3.7. Las condiciones que han llevado a la elección de este lenguaje son las siguientes:

- Librería de análisis técnico soportadas.
- Velocidad de prototipado, Python al ser un lenguaje interpretado reduce los tiempos de desarrollo.
- Facilidad de interoperabilidad entre sistemas operativos. Todos los desarrollos se han realizado bajo un sistema operativo Ubuntu 16.04 pero el sistema es portable a otros sistemas operativos como Windows.

El software propuesto se basa en un sistema distribuido ya que consta de diferentes nodos donde cada uno de ellos opera de forma independiente. Un nodo es el encargado de abrir una posición, gestionarla y cerrar dicha posición. Otro nodo es el encargado del *backtesting* para poder analizar el rendimiento de las estrategias, otro es el encargado de las notificaciones al usuario y cálculo de los resultados de la estrategia. En este capítulo se describe de forma más detallada la estructura y funcionamiento del sistema.

La librería seleccionada que implementa las operaciones difusas es Scikit-Fuzzy [43] ya que soporta Python 3. Su última versión liberada es de 2019 y proporciona una API para trabajar con conjuntos difusos. Esta librería no soporta el sistema de inferencia de Sugeno y solo permite usar el sistema de inferencia de Mamdani, puesto que el sistema propuesto que se utiliza es del tipo Mamdani esta librería es válida, en caso de necesitar modelar un sistema Sugeno se habría utilizado la librería Simpful [44].

Se ha decidido usar Scikit-Fuzzy ya que está construida sobre Scipy [45] y esto permite que sea más sencillo integrarla en el modelo y otras librerías usadas.

### 4.1. Requisitos de diseño

Antes de implementar el sistema se han propuesto una serie de requisitos que debe cumplir el sistema. Esto permite desarrollar la aplicación de una forma más sencilla y saber por donde empezar. Los requisitos mínimos que debe cumplir el sistema son:

- Ser capaz de realizar *trading* en tiempo real en un *exchange*.
- Ser capaz de realizar *backtesting* para obtener medidas de rentabilidad de las estrategias.

- Permitir el uso de varias estrategias de forma simultanea.
- Configurable, debe ser sencillo añadir nuevas estrategias y modificar las existentes.
- Estar disponible el mayor tiempo posible, en caso de que exista algún error, el sistema debe recuperarse y estar disponible de nuevo en el menor tiempo posible.
- Notificar al usuario el estado del sistema. Puesto que el sistema no tiene interfaz gráfica, debe existir algún modo en el que el usuario pueda conocer la rentabilidad del sistema, estado de las operaciones, etc.

## 4.2. Estructura

En la Figura 4.1 se muestra un esquema general de la aplicación. El sistema está compuesto por una serie de nodos imprescindibles que se detallan en el apartado 4.2.3 y otra serie de nodos que almacenan las estrategias de *trading* y proponen la apertura o cierre de las operaciones.

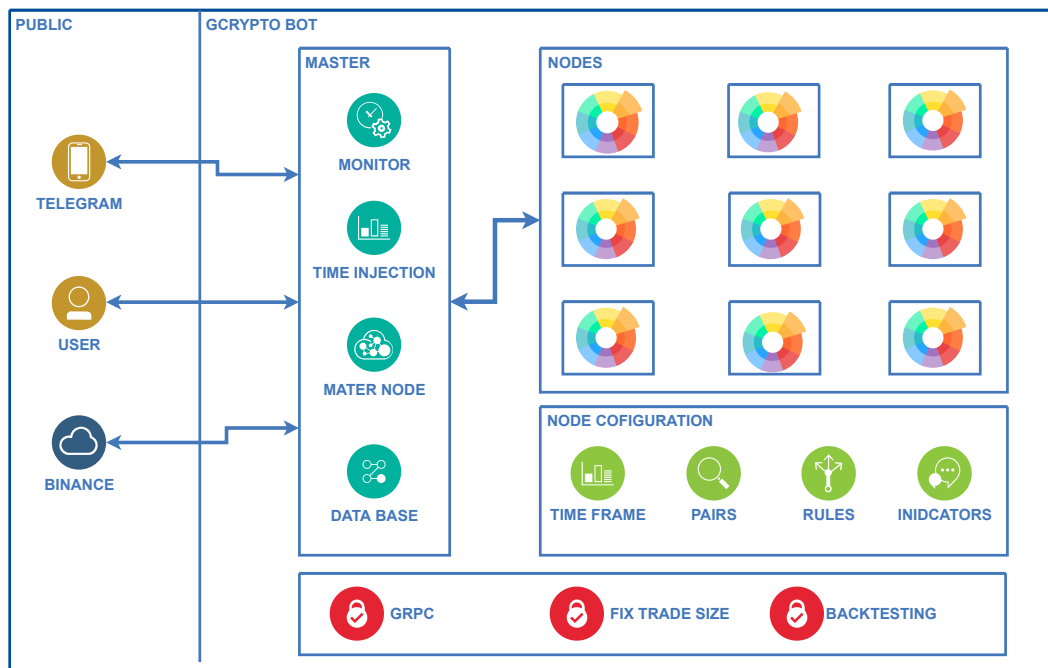


Figura 4.1: Arquitectura del *bot* de *trading*

En los siguientes apartados se describe de forma detallada la estructura del sistema.

### 4.2.1. Comunicación entre nodos

La comunicación entre los distintos nodos del sistema se realiza usando gRPC. Se ha seleccionado este mecanismo ya que es una de las formas más utilizadas en la actualidad en aplicaciones distribuidas [46]. Se podría decir que gRPC es una implementación moderna de los sistemas RPC (*Remote Procedure Callback*) donde se solventan sus limitaciones e inconvenientes. gRPC es más escalable, eficiente y seguro que sistemas basados en HTTP/REST para la comunicación entre aplicaciones [47].

La definición de los mensajes intercambiados entre los nodos se conoce como “lenguaje de definición de interfaz” (IDL) que en el caso de gRPC se usa *protocol buffers* (Protobuf). En la Figure 4.2 se muestra un ejemplo de la implementación de la comunicación entre dos aplicaciones usando gRPC.

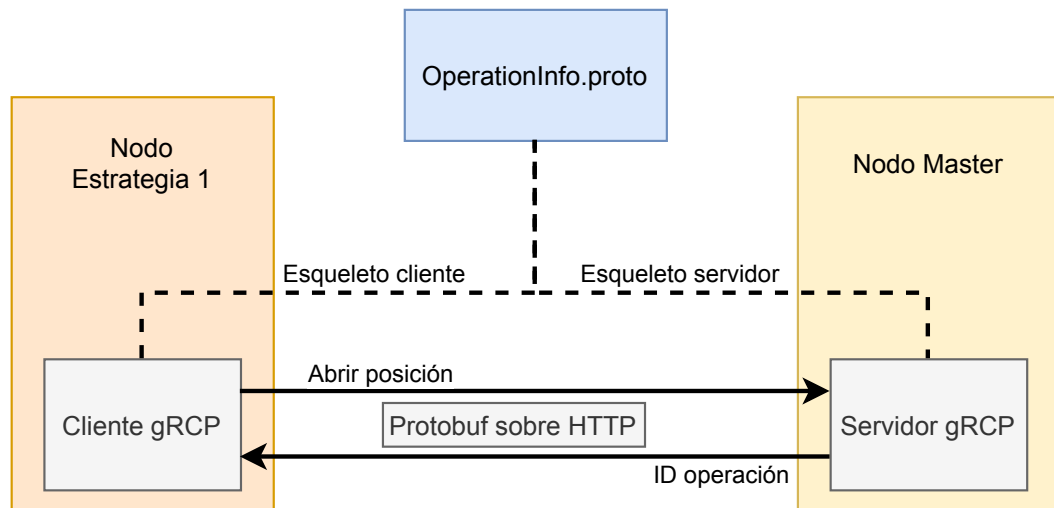


Figura 4.2: Uso de gRPC

### 4.2.2. Configuración

Toda la configuración del sistema está escrita usando el lenguaje TOML. TOML es un formato de ficheros que es legible y reutilizable [48] que permite anidar de forma sencilla estructuras de datos.

TOML es un lenguaje de configuración similar a YAML y es más avanzado que sus predecesores XML y JSON. A diferencia de sus predecesores, TOML admite tipos de datos que XML no admite y JSON los admite de forma incompleta [49].

Los archivos usados en el sistema en formato TOML no siguen una estructura estandarizada específica, excepto que utilizan el lenguaje TOML. A continuación se muestra un fragmento de un fichero de configuración del sistema.

```

1 strategy_name = "mystrategy"
2 timeframe = "15m"
3
4 takeProfit = 10.0
5 stopLoss = 20.0
6 enabled = true
7
8 [pairs]
9 UPairs = { "BTCUSDT" = { enable = "true" } }
```

### 4.2.3. Nodos principales

#### Nodo Máster

Este nodo es el que ejecuta el usuario y es el punto de entrada del programa. Las funciones que realiza el nodo son las siguientes:

- Abre y cierra las posiciones en el *exchange* basado en los mensajes recibidos de los diferentes nodos.
- Se comporta como servidor gRPC ya que solo recibe peticiones pero por si mismo no desencadena ninguna operación de compra y venta.
- Es el punto de entrada del sistema.
- Notifica al usuario la apertura y cierre de operaciones.
- Es el encargado de ejecutar los distintos nodos del sistema.

#### Nodo tiempo

Este es un nodo simple que se encarga de proveer datos temporales al resto del sistema. Se usa para poder probar la estrategia en momentos del pasado donde ya se sabe cuales son los siguientes datos del precio. En su fichero de configuración se configura la fecha y hora de inicio y este nodo lanza un servidor gRPC que devuelve la fecha y hora desde ese momento. La hora se proporciona usando UTC ya que es la usada en los *exchanges* para abrir y cerrar los intervalos temporales.

Cuando no se usa una fecha y hora pasadas sino que es un sistema de *trading* en tiempo real es necesaria la sincronización horaria del sistema donde se ejecuta el bot, para llevar a cabo esto se ha usado *chrony* como se propone en [50]. En el caso de este sistema no se requiere una sincronización menor a un segundo ya que los intervalos de tiempo usados son mayores al minuto.

#### Nodos estrategias

Estos nodos son los más importantes del sistema, implementan el sistema propuesto en el capítulo 3 y se encargan de generar las posibles operaciones a partir de los datos del precio obtenidos de un *exchange*. Estos nodos requieren un fichero de configuración donde se define la estrategia, el par de criptomonedas, intervalo temporal a operar, etc.

Las funcionalidades que realizan estos nodos son:

- Solicitan los datos de fecha y hora al nodo de tiempo.
- Solicitan los datos del precio al *exchange*.
- Aplican la estrategia definida a los datos y proporcionan al nodo master los datos necesarios para abrir una operación.

En la Figura 4.3 se muestra un diagrama simplificado del flujo que siguen estos nodos cuando se están ejecutando.

Estos nodos usan un bucle infinito donde se obtienen datos temporales del nodo temporal dependiendo del intervalo en el que se encuentra trabajando el nodo.

Cuando se obtienen los datos del *exchange*, el nodo los analiza y aplica la estrategia que tiene definida. Tras aplicar la estrategia se decide si es necesario abrir una posición, en caso de que exista una operación abierta se comprueba si es necesaria cerrarla.

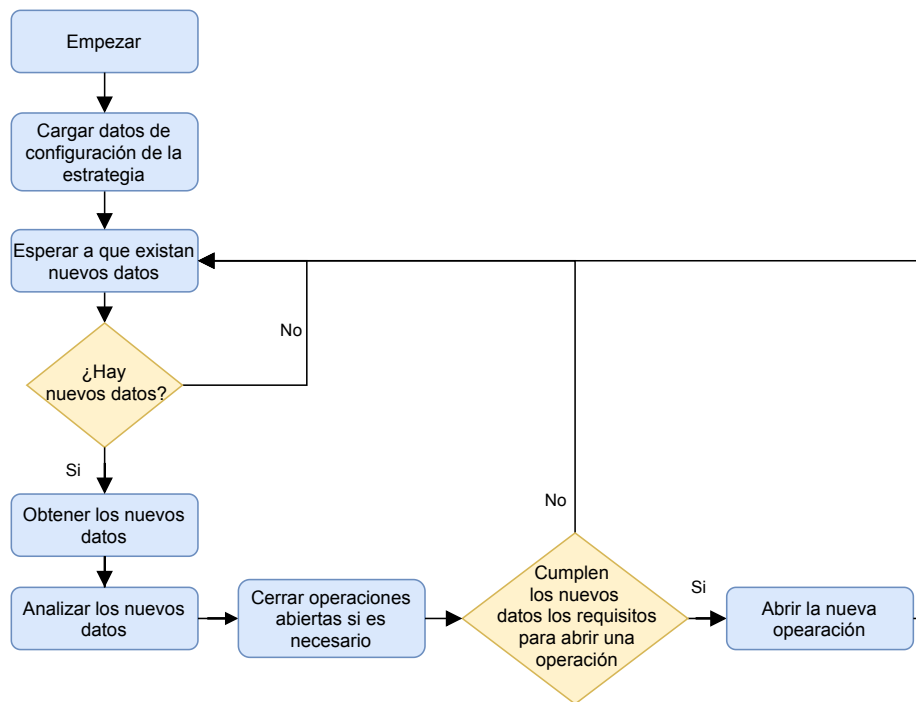


Figura 4.3: Flujo de ejecución de los nodos de estrategia

### Nodo base de datos

Este nodo se encarga de mantener una base de datos con el historial de las operaciones. Con esto se pueden ver las operaciones pasadas y poder obtener las estadísticas de las operaciones abiertas. Se pueden calcular los beneficios o pérdidas, rentabilidad de cada operación, etc.

Se ha usado `SQLite` como motor de base de datos [51]. Para acceder a los datos en vez de usar sentencias `sql` se ha usado un ORM *Object Relational Mappings*, concretamente se ha usado `peewee` [52].

Este nodo también es capaz de proponer el cierre de una operación al nodo master cuando una operación no ha alcanzado el punto de toma de beneficios y ha expirado el máximo tiempo que esa operación puede estar abierta.

### Nodo monitor

Este nodo es el encargado de comprobar el estado del resto de nodos y que ninguno de ellos esté caído. En caso de ser necesario este nodo es capaz de relanzar otros nodos. Otras funcionalidades que proporciona son:

- Calcular las estadísticas del sistema usando los datos aportados por el nodo de base de datos.
- Realizar las notificaciones al usuario. Se describe detalladamente en el apartado 4.4.
- Comprueba el estado del resto de nodos. En caso de un nodo no esté en funcionamiento, se encarga de ejecutarlo.
- Calcula la rentabilidad del sistema.

#### 4.2.4. Obtención de datos

Los datos que usa el bot son obtenidos del *exchange* de Binance. Este *exchange* provee una API REST [53] para que otras aplicaciones puedan operar en la plataforma. Se ha elegido Binance porque es el mayor *exchange* en la actualidad y su API es sencilla para obtener distintos datos de las criptomonedas, abrir y cerrar posiciones, etc.

Binance proporciona una librería en Python que simplifica el uso de su API. Para usar la API es necesario tener una cuenta en la plataforma y generar una clave pública y una privada para poder abrir y cerrar posiciones.

Para crear las claves es necesario acceder a la plataforma y crear un juego de claves que permitan realizar operaciones de *spot* y futuros como se muestra en la Figura 4.4.

Figura 4.4: Binance API

Tras realizar esto, desde el Python se puede abrir una conexión:

```
1 import binance
2 client = binance.Client(config.api_key, config.api_secret)
3 await client.load()
```

Podemos obtener los datos de un determinado par de criptomonedas entre dos puntos en el tiempo en la zona horaria UTC hay que realizar la siguiente llamada:

```
1 date_string_start = '2021-02-11 17:08:58'
2 date_string_end = '2021-03-11 17:08:58'
3 candle = await client.fetch_klines(symbol="BTCUSDT", start_time=str(
    ↪ timestamp_init), end_time=str(timestamp_end), interval="15m")
```

Obteniendo los siguientes datos que son interpretados según se indica en la API de Binance.

```
1 [[1613060100000, 47820.31, 48050.0, 47713.25, 47790.64, 944.67,
    ↪ 1613060999999, 45255916.92, 32650, 500.72, 23990167.63, 0],
2 [1613061000000, 47789.86, 47864.65, 47500.00, 47568.64, 1150.83,
    ↪ 1613061899999, 54778291.38, 35388, 518.48, 24677507.95, 0],
```



```

3     [1613061900000, 47568.64, 47716.70, 47337.18, 47590.53, 942.21,
      ↪ 1613062799999, 44825765.75, 28160, 462.82, 22023699.83, 0],
4     ...

```

#### 4.2.5. Abrir posición

Una funcionalidad clave para un sistema de *trading* automático es abrir una posición. Tras establecer la conexión usando la API de Binance y obtener los datos es posible abrir una posición.

Dentro de las posiciones que permite ejecutar Binance, las operaciones de compra y venta pueden ser de distintos tipos:

- “Valor de mercado” donde la operación se ejecuta instantáneamente al precio de mercado.
- “Precio límite” donde la operación se ejecuta cuando el precio alcanza un determinado precio.

Para abrir una operación usando la API de Binance se puede usar el siguiente código:

```

1     order = await client.create_order(
2         symbol="TRXBTC",
3         side=binance.Side.BUY,
4         order_type=binance.OrderType.LIMIT,
5         time_in_force=binance.TimeInForce.GTC,
6         quantity=Decimal("2000"), #Cantidad de TRX
7         price="0.00000150", # Valor de TRX/BTC
8         test=False,
9     )

```

Los datos necesarios para abrir una posición son:

- Símbolo: par de criptomonedas que se van a operar.
- Compra/Venta.
- Tipo de orden: “valor de mercado” o “precio límite”.
- Cantidad: tamaño de la operación.
- Precio: en caso de usar una operación de tipo “precio límite” es necesario especificar a qué precio se va a ejecutar la operación.

### 4.3. Cerrar posición

La API de Binance permite obtener la operaciones abiertas usando la siguiente llamada:

```

1     open_orders = await client.fetch_open_orders(symbol="TRXBTC")

```

Una vez obtenidas las operaciones en curso, es posible cerrar cualquiera de ellas usando el identificador de una operación:

```

1     close_order = await client.cancel_order(
2         symbol="TRXBTC",
3         order_id=order["orderId"]
4     )

```

### 4.3.1. Intervalo temporal

Seleccionar un intervalo temporal adecuado mientras se está realizando *trading* es algo muy importante ya que en una escala temporal la tendencia puede ser alcista pero en una temporalidad menor la tendencia puede ser bajista.

La API de Binance permite obtener los datos en distintos intervalos temporales, y el sistema puede trabajar en distintas temporalidades según la estrategia definida. En la plataforma de Binance todos los datos temporales están expresado en la zona horaria UTC por lo que el bot también trabaja en esta zona horaria.

Los intervalos temporales soportados en Binance y más usados en los sistemas de *trading* son:

- 1m - 1 minuto
- 5m - 5 minutos
- 15m - 15 minutos
- 30m - 30 minutos
- 1h - 1 hora
- 4h - 4 horas
- 1d - 1 día (diario)
- 1w - 1 semana (semanal)

## 4.4. Notificaciones

El sistema cuenta con un método de notificaciones al usuario que se realizan usando Telegram. Se ha elegido esta aplicación ya que crear un bot para esta aplicación que reciba mensajes desde el sistema de *trading* es sencillo.

Se ha usado el `BotFather` para crear el bot y desde Python usando la librería `requests` para enviar los mensajes. Un ejemplo de los mensajes recibidos en el bot de Telegram se muestra en la Figura 4.5.

Las notificaciones que se envían desde el sistema al usuario son:

- Apertura de una operación.
- Cierre de una operación y el resultado de esta.
- Inicio del sistema.
- Errores detectados.
- Cierres no controlado de algunos de los nodos.

## 4.5. Instalación

El proyecto está completamente escrito en Python por lo que se ha creado un fichero `requirements.txt` que permite instalar todas las dependencias del proyecto. La versión mínima del intérprete de Python necesaria es 3.5 aunque todas las pruebas se han realizado con la versión 3.7.

Se recomienda crear un entorno virtual donde instalar las dependencias y el proyecto. El entorno se puede crear usando el comando mostrado en Código 4.1.



Figura 4.5: Visualización de un mensaje enviado al sistema

## Código 4.1: Crear entorno virtual

```
1 python3 -m venv <name_of_virtualenv>
```

Para instalar las dependencias del proyecto es necesario activar el entorno virtual creado anteriormente e instalar las dependencias usando los comandos mostrados en Código 4.2.

## Código 4.2: Instalar dependencias

```
1 source env/bin/activate
2 python3 -m pip install -r requirements.txt
```

## 4.6. Ejecución

Tras la instalación correcta de las dependencias es necesario compilar los ficheros `.proto` en los que se definen los mensajes de comunicación entre los diferentes nodos. Esta compilación se realiza usando el módulo `grpc_tools` a través del comando mostrado en Código 4.3.

## Código 4.3: Compilación de los ficheros proto

```
1 python3.7 -m grpc_tools.protoc --proto_path=. *.proto --python_out=. --
  ↳ grpc_python_out=.
```

Después de llevar a cabo generación de los ficheros se puede ejecutar el nodo principal, este será el encargado de lanzar el resto de nodos para las distintas estrategias. En Código 4.4 se muestra el comando necesario para ejecutar el nodo principal.

## Código 4.4: Ejecución nodo principal

```
1 python3.7 master_node.py
```

### 4.6.1. Funcionamiento

A continuación se describe en detalle como se cargan los distintos nodos del sistema.<sup>9</sup>

1. Tras la ejecución del nodo principal, en primer lugar se carga el nodo de monitorización.
2. En segundo lugar se lanza el nodo de inyección de tiempo. Este nodo lee su fichero de configuración donde se configura la fecha de inicio a partir de la cual el resto de nodos obtienen la hora. En la Figura 4.5 se muestran los mensajes que recibe este nodo y la respuesta que proporciona.

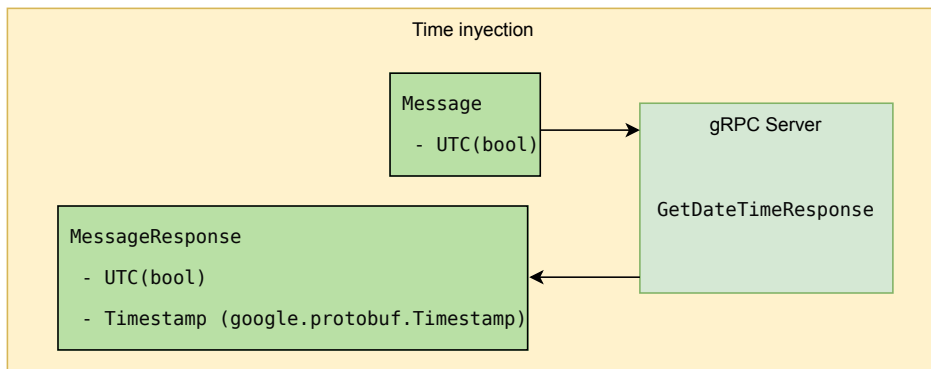


Figura 4.6: Intercambio de mensajes para obtener la hora y fecha

3. A continuación se lanza el nodo que maneja la base de datos. Cuando el nodo máster ejecuta una operación esta se añade a la base de datos usando este nodo. El mensaje enviado es el mismo que el mostrado en la Figura 4.7.
4. Tras esto, el nodo principal detecta las estrategias almacenadas en la carpeta `strategies` y ejecuta un nodo por cada estrategia. Los nodos de las estrategias comienzan a obtener valores del nodo de tiempo. Se analiza el precio del activo de esa estrategia y cuando se detecta un posible punto de entrada en una operación el nodo de estrategia se lo notifica al nodo máster usando los mensajes mostrados en la Figura 4.7.
5. Por último el nodo principal comienza a aceptar peticiones de cada uno de los nodos.

### 4.6.2. Logging

Se ha usado un sistema de logging que permite almacenar el funcionamiento del sistema, aquí se puede ver las reglas activas al abrir una operación, los valores de los indicadores técnicos clásico y difusos.

El formato elegido para los logs es el siguiente:

Código 4.5: Formato de logs

```
1 ' %(asctime)s [%(levelname)s] - [%(filename)s: %(lineno)d] %(name)s -
   ↳ %(message)s'
```

Donde:

- `asctime` es la fecha usando el formato ISO8601 [54].

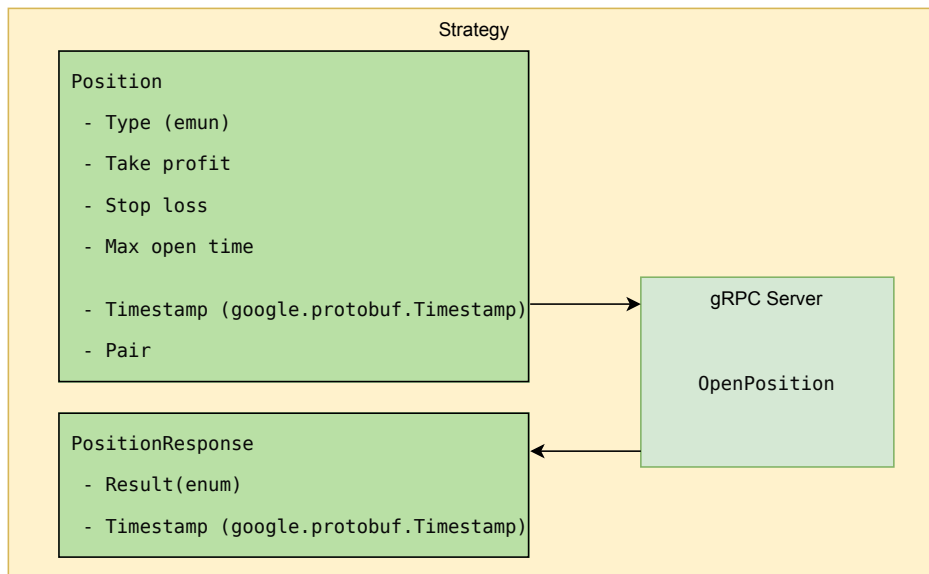


Figura 4.7: Mensaje usado para abrir una operación

- `textlevelname`: nivel del log.
- `filename`: nombre del fichero.
- `lineno`: línea del fichero.
- `name`: nombre del nodo.
- `message`: mensaje a escribir.

Los niveles del log usados son: *CRITICAL*, *ERROR*, *WARNING*, *INFO* y *DEBUG*.

### 4.6.3. Comparación del sistema frente a otros

A continuación se realiza un comparación más detallada del sistema propuesto con las principales características de las herramientas mostradas en la sección 1.4.

El sistema propuesto recopila las mejores características de las distintas herramientas:

- La principal ventaja del sistema propuesto frente al resto es que ninguna de las herramientas públicas permiten el uso de lógica difusa, todas ellas usan indicadores técnicos predefinidos para implementar las estrategias.
- Está desarrollado en Python al contrario que Gekko y ZenBot que usan JavaScript, esto permite un mejor rendimiento en *backtesting*.
- Jesse es una de las mejores herramientas ya que implementa gran cantidad de indicadores técnicos, tiene una gran comunidad y está altamente optimizado pero su principal desventaja es que no permite operaciones en tiempo real mientras que el sistema propuesto permite ambas: *backtesting* y operaciones en tiempo real.

- La simplicidad frente al resto de herramientas también es una gran ventaja de este sistema. Se pueden configurar las estrategias modificando un fichero `.toml` mientras que en el resto de herramientas para modificar las estrategias es necesario añadir código en Python o JavaScript haciendo que sea más complejo y se puedan añadir errores.
- Notificaciones: algunas herramientas constan de interfaz gráfica para notificar al usuario el estado del sistema, puesto que el sistema propuesto no tiene interfaz gráfica este notifica al usuario a través de una aplicación de mensajería.

## 4.7. Limitaciones

- La implementación no permite tener más de una operación abierta simultáneamente por estrategia. Cuando se produce una señal de compra o venta por una estrategia, se abre la operación con todo el capital disponible por lo que si la estrategia genera otra señal esta no se realizará. Se ha decidido tomar esta decisión ya que el sistema no proporciona un sistema de manejo del capital. En artículos como [55] y [56] se pueden encontrar algunos algoritmos que permiten subsanar esta limitación ajustando el tamaño de las posiciones a la salida del sistema difuso.
- Solo se permiten realizar operaciones en el *exchange* de Binance. Aunque hoy en día la mayoría de *exchanges* proporcionan una API para realizar operaciones, se ha elegido Binance ya que su API es implementada por distintas librerías haciendo su uso más sencillo.

## Capítulo 5

# Validación del sistema

En el capítulo 3 se ha detallado el funcionamiento del sistema difuso y en el capítulo 4 se ha descrito el funcionamiento de la implementación, en este capítulo se van a validar las estrategias definidas usando distintas criptomonedas e intervalos temporales. Para ello se va a realizar *backtesting* en distintos periodos temporales pasados. Se realizará una comparación entre las estrategias difusas propuestas en este trabajo y una estrategia clásica. Se ha decidido llevar a cabo esta comparación en lugar de realizar con una comparación con otros trabajos porque no se ha encontrado un trabajo de investigación que proponga un sistema que se pueda replicar de una forma sencilla a las criptomonedas. Esta aproximación de comparar las estrategias difusas con la estrategia clásica también es usado en distintos trabajos de investigación como [57] y [58].

### 5.1. Intervalos temporales

Para llevar a cabo la validación de las estrategias se van a usar dos periodos de tiempo, el primero es el mismo que se ha usado en el apartado 3.6 para diseñar la estrategia, este intervalo va desde el 1 de enero de 2021 a las 00:00:00 hasta el 11 de febrero de 2021 a las 16:00:00. El segundo periodo cubre desde el 1 de mayo a las 00:00:00 al 31 de junio de 2021 a las 16:00:00.

Las estrategias se van a aplicar a distintos intervalos temporales, los elegidos para validar la estrategia son: 5 minutos, 1 hora y 4 horas. Cuando se realiza el análisis en una escala temporal de 5 minutos se ha reducido el periodo en el que se aplica la estrategia debido a la gran cantidad de datos obtenida cuando se opera en un intervalo temporal pequeño, los datos usados van desde el 1 de enero de 2021 a las 00:00:00 hasta el 4 de enero de 2021 a las 16:00:00 y desde 1 de mayo a las 00:00:00 al 4 de enero de 2021 a las 16:00:00.

Para visualizar las compras y ventas realizadas por el sistema, a continuación se muestran las operaciones de las dos estrategias usando un intervalo temporal de una hora. La “Estrategia 1” ha realizado en total 26 operaciones y con beneficio de un 22.46%. La “Estrategia 2”, ha realizado en total 12 operaciones y ha tenido una pérdida del 13%.

En la Figura 5.1a se muestran las operaciones abiertas por la “Estrategia 1” en una escala temporal de 1 hora, las operaciones en *long* se muestran en franjas verticales de color verde, indicando el momento de apertura y cierre. En color rojo se muestran las operaciones en *short* realizadas por la misma estrategia. En color negro se muestra el precio del activo. En la Figura 5.1b se muestran los mismos datos para la “Estrategia 2”.

Si descendemos a la escala temporal de 5 minutos, en la “Estrategia 1”, se han realizado en total 25 operaciones y con un beneficio de 1.42%, usando la “Estrategia 2”, se han realizado en total 5

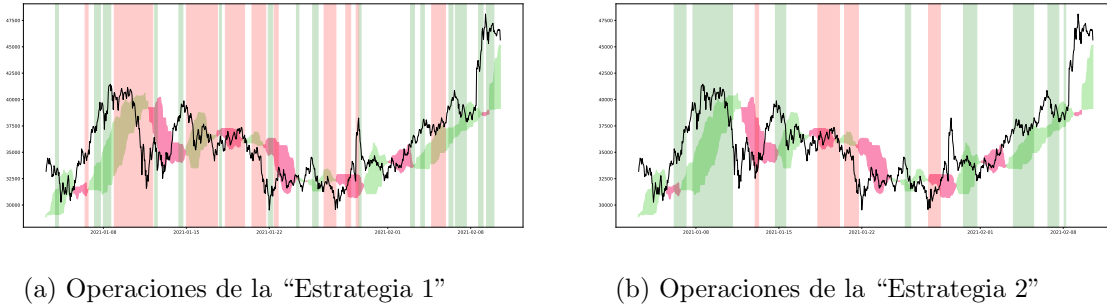


Figura 5.1: Operaciones en *long* y *short* en un intervalo temporal de “1h”

operaciones y ha tenido unas pérdidas del 12.03 %.

En la Figura 5.2a se muestran las operaciones abiertas por la “Estrategia 1” en una escala temporal de 5 minutos, las operaciones en *long* se muestran en franjas verticales de color verde, indicando el momento de apertura y cierre. En color rojo se muestran las operaciones en *short* realizadas por la misma estrategia. En color negro se muestra el precio del activo. En la Figura 5.2b se muestran los mismos datos para la “Estrategia 2”.

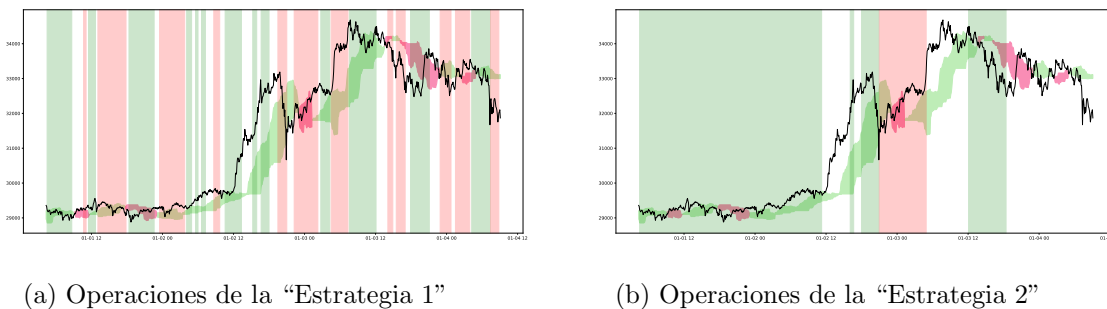


Figura 5.2: Operaciones en *long* y *short* en un intervalo temporal de “5m”

Para poder validar los resultados, también se aplica la estrategia clásica definida en el apartado 3.6.3 a los mismos periodos temporales a los que se han aplicado las estrategias difusas. En la Figura 5.3 se muestran las operaciones ejecutadas usando la estrategia clásica en *long* (bandas color verde) y en *short* (bandas en color rojo).

## 5.2. Resultados

En este apartado se van a aplicar las estrategias anteriores a distintas criptomonedas y comparar los resultados, las elegidas han sido las cuatro primeras con mayor capitalización de mercado (BTC, ETH, ADA y BNB).

En la Tabla 5.1 y Tabla 5.2 se muestran los resultados de las estrategias para las distintas criptomonedas y marcos temporales. En estas tablas se ha coloreado la columna de rendimiento, en color verde claro se encuentran las estrategias que han obtenido un rendimiento positivo y en rojo claro



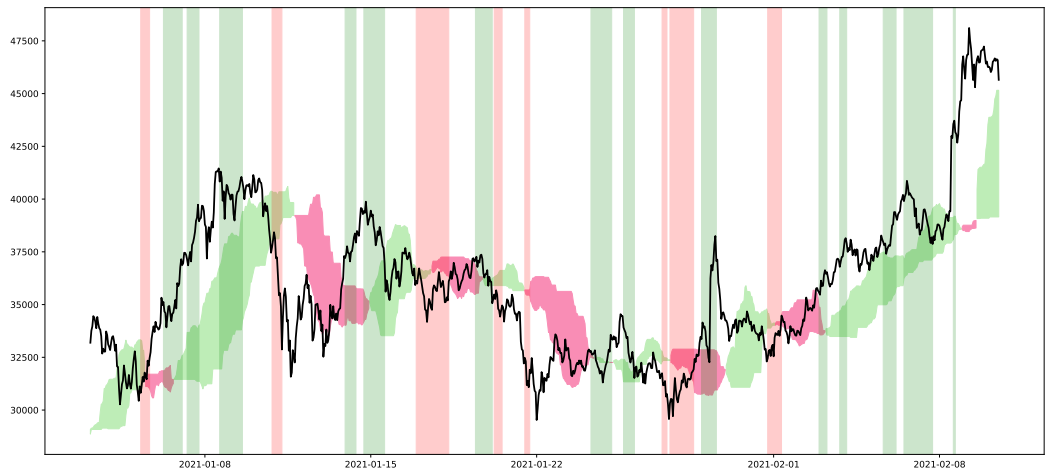


Figura 5.3: Estrategia clásica. Operaciones en *long* y *short* en un intervalo temporal de “1h”

las estrategias que han obtenido un rendimiento negativo. También se han resaltado en un color más brillante la estrategia que mejores y peores rendimientos ha obtenido en los dos periodos analizados.

Tabla 5.1: Resultados del sistema entre el 1 de enero y 11 de febrero de 2021

Intervalo	Par	Estrategia	Operaciones con beneficios	Operaciones con pérdidas	Operaciones totales	Media de beneficio por operación	Rentabilidad %
4h	BTC	Estrategia 1	2	4	6	-0.8	-5.1
		Estrategia 2	0	1	1	-5	-5
		Clásica	2	2	4	0.45	1.80
	ETH	Estrategia 1	3	4	7	1.3	9.7
		Estrategia 2	0	2	2	-5	10
		Clásica	4	1	5	1.06	5.30
	ADA	Estrategia 1	1	3	4	1.4	5.7
		Estrategia 2	0	1	1	-5	-5
		Clásica	2	0	2	1.9	3.8
	BNB	Estrategia 1	1	3	4	2.7	10.9
		Estrategia 2	1	0	1	0.5	0.5
		Clásica	2	2	4	2.29	9.1
1h	BTC	Estrategia 1	13	13	26	0.86	22.0
		Estrategia 2	5	7	12	1.1	-13.3
		Clásica	13	9	22	0.77	16.87
	ETH	Estrategia 1	15	13	28	1.1	33.1
		Estrategia 2	6	3	9	2.1	19.1
		Clásica	12	12	24	0.23	5.63
	ADA	Estrategia 1	14	16	30	0.8	25.37
		Estrategia 2	7	2	9	9.6	87.1
		Clásica	11	13	24	0.87	20.90
	BNB	Estrategia 1	16	10	26	2.9	76.4
		Estrategia 2	4	6	10	1.5	15
		Clásica	24	2	26	0.79	20.6
5m	BTC	Estrategia 1	14	11	25	0.05	1
		Estrategia 2	2	3	5	-2.4	-12.0
		Clásica	13	6	19	-0.13	-2.5
	ETH	Estrategia 1	18	8	26	0.48	12.7
		Estrategia 2	6	5	11	0.5	6.3
		Clásica	19	2	21	0.22	4.6
	ADA	Estrategia 1	13	12	25	-0.074	-1.8
		Estrategia 2	4	5	9	-1.58	-14.3
		Clásica	13	6	19	-0.36	-6.9
	BNB	Estrategia 1	6	9	15	-0.5	-8.7
		Estrategia 2	1	4	5	-1.1	-5.6
		Clásica	9	5	14	-1.11	-15.6

Tabla 5.2: Resultados del sistema entre el 1 de mayo y 31 de junio de 2021

Intervalo	Par	Estrategia	Operaciones con beneficios	Operaciones con pérdidas	Operaciones totales	Media de beneficio por operación	Rentabilidad %
4h	BTC	Estrategia 1	3	4	7	1.30	9.07
		Estrategia 2	0	4	4	-5.00	-20.00
		Clásica	3	5	8	1.46	11.69
	ETH	Estrategia 1	6	6	12	-0.72	-8.60
		Estrategia 2	2	4	6	-1.23	-7.41
		Clásica	4	9	13	-1.87	-24.33
	ADA	Estrategia 1	3	9	12	-1.80	-21.57
		Estrategia 2	1	4	5	-3.03	-15.13
		Clásica	5	3	8	-3.79	-30.36
	BNB	Estrategia 1	4	10	14	-1.54	-21.50
		Estrategia 2	0	5	5	-5.00	-25.00
		Clásica	6	6	12	-1.47	-17.7
1h	BTC	Estrategia 1	12	15	27	0.53	14.40
		Estrategia 2	5	4	9	-0.01	-0.11
		Clásica	18	6	24	0.64	15.35
	ETH	Estrategia 1	15	17	32	0.75	24.06
		Estrategia 2	5	6	11	2.65	29.20
		Clásica	20	8	28	0.89	24.91
	ADA	Estrategia 1	12	18	30	0.36	10.92
		Estrategia 2	5	8	13	1.59	20.66
		Clásica	11	14	25	-0.38	-9.41
	BNB	Estrategia 1	14	13	27	1.15	30.92
		Estrategia 2	4	6	10	-0.94	-9.36
		Clásica	14	6	20	1.59	31.87
5m	BTC	Estrategia 1	11	8	19	0.05	0.93
		Estrategia 2	2	1	3	0.05	0.14
		Clásica	7	4	11	-0.35	-3.89
	ETH	Estrategia 1	11	8	19	-1.05	-20.00
		Estrategia 2	1	4	5	-1.87	-9.33
		Clásica	9	4	13	-1.16	-15.13
	ADA	Estrategia 1	8	10	18	-0.16	-2.79
		Estrategia 2	10	6	16	0.03	0.44
		Clásica	13	6	19	-0.13	-2.5
	BNB	Estrategia 1	14	10	24	0.11	2.54
		Estrategia 2	5	3	8	0.44	3.51
		Clásica	5	7	12	-1.09	-13.07

### 5.3. Discusión

Tras presentar los datos obtenidos por las dos estrategias difusas y la estrategia clásica, se observan algunos datos relevantes.

La rentabilidad media de las tres estrategias en orden descendiente son:

- Estrategia 1 es 10,06 %.
- Estrategia 2 1,56 %.
- Estrategia Clásica es 1.29 %.

La escala temporal en la que mejor funcionan las tres estrategias es en “1 hora”, esto es se debe a la selección de indicadores. Todos los indicadores seleccionados son válidos para los intervalos en los que se ha aplicado pero idealmente tienen un rango de los intervalos en los que mejores señales producen. La nube de Ichimoku tiene su mejor comportamiento en intervalos desde “30m” hasta “4h”.

También cabe resaltar el número de operaciones, las estrategias menos complejas como la Estrategia Clásica y Estrategia 1 efectúan un número mucho mayor de operaciones. De media estas estrategias ejecutan un 150 % más de operaciones que la Estrategia 2.

El principal inconveniente de la estrategia clásica es que las operaciones son muy cortas como se puede apreciar en la Figura 5.3. Esto se debe a las condiciones de entrada y salida de la estrategia, al ser una estrategia clásica los indicadores cambian rápidamente haciendo que las operaciones se cierren antes de lo que deberían.

Uno de los inconvenientes de la Estrategia 2 es que realiza muy pocas operaciones, esto hace que si en un periodo de un mes solo se realizan dos operaciones y las dos se cierran por el *stop loss* tenga una pérdida del 10 % del capital.

También se observa que la Estrategia Clásica ha obtenido mejor rendimiento que las estrategias difusas en algunos escenarios aislados, esto se ha producido en intervalos donde ha habido variaciones del precio muy abruptas haciendo que esta estrategia abra posiciones que generen un gran beneficio, mientras que las estrategias difusas han tardado más en reaccionar a estos cambios tan abruptos y las operaciones han obtenido peores rendimientos.

Se puede concluir que las estrategias difusas obtienen mejor rendimiento que la estrategia clásica, también es más recomendable una estrategia que ejecute un mayor número de operaciones frente a una que realice pocas operaciones.

## Capítulo 6

# Conclusiones y Trabajo Futuro

En este capítulo se muestra un resumen de las conclusiones que se pueden obtener del trabajo mostrado en los apartados anteriores. Después se muestran los posibles trabajos futuros que se podrían desarrollar usando como base este trabajo.

### 6.1. Conclusiones

Como se ha mostrado, un sistema de *trading* automático puede ser una herramienta de ayuda para los *traders* con la que aumentar los beneficios obtenidos del mercado. Debido a la volatilidad de los mercados financieros de criptomonedas, la lógica difusa es una herramienta útil que permite lidiar con la incertidumbre que existe.

En el trabajo se han completado con éxito los objetivos propuestos al inicio, se ha analizado el estado del arte actual de los sistemas de *trading* automático así como la aplicación de la lógica difusa a este ámbito. Esta investigación ha permitido que se diseñe e implemente un sistema completo de *trading* automático basado en lógica difusa como se muestra en el capítulo 4, permite tanto probar estrategias en el pasado como operar en un *exchange* en tiempo real, esto es una gran ventaja sobre otros sistemas de *trading* automático de código abierto. Otras de sus grandes ventajas son su simplicidad de uso y fácil configuración. También tiene algunas limitaciones o mejoras que se proponen en este mismo capítulo más adelante.

El sistema ha sido puesto a prueba en periodos temporales del pasado reciente, usando distintos intervalos temporales y aplicado a distintas criptomonedas como se muestra en el capítulo 5. Los resultados obtenidos han resultado favorables ya que tienen una rentabilidad aceptable.

Las dos estrategias que se han probado tienen beneficios positivos aunque cabe señalar que una de ellas obtiene mejores resultados. El intervalo de una hora es en el que mejores resultados obtienen ambas estrategias.

El software usado durante el trabajo se encuentra disponible en el siguiente repositorio público de Bitbucket: [www.bitbucket.org/gustavoPlaza/cripto-bot](http://www.bitbucket.org/gustavoPlaza/cripto-bot).

### 6.2. Trabajos futuros

Aunque el sistema propuesto es completo y funcional, hay algunos aspectos del mismo que se podrían mejorar y que se detallan a continuación.

Aunque el sistema es altamente configurable, las reglas difusas que usa el sistema no lo son, por lo que un trabajo futuro consistiría en poder almacenar las reglas en los ficheros de configuración que usan los nodos de estrategias, existe el lenguaje FML (*Fuzzy markup language*) [59] que se podría usar para almacenar la reglas en los ficheros de configuración. La librería de lógica difusa usada en el trabajo (Scikit-Fuzzy) no permite usar este lenguaje por lo que sería necesario implementar la serialización y deserialización del mismo.

Otro posible trabajo sería la implementación de una interfaz gráfica para el sistema, esta podría ser una interfaz web que consuma los datos del sistema usando su base de datos. La interfaz gráfica también debería de permitir abrir y cerrar posiciones manualmente, cambiar configuración del sistema (pares de criptomonedas usados, intervalos temporales, ...) y visualizar el estado del sistema (número de operaciones abiertas, rentabilidad, ...)

El último trabajo futuro sería implementar un método para calcular el *stop loss* del sistema, actualmente este siempre está fijado a un 5% pero sería una gran mejora que este se calculase en base a los indicadores técnicos y estado del mercado.

# Bibliografía

- [1] Bank for International Settlements. Foreign exchange turnover in april 2016. <https://www.bis.org/publ/rpfx16.htm>, 2016. [Online; Accessed: 2021-08-13].
- [2] Lennart Ante. Bitcoin transactions, information asymmetry and trading volume. *Bitcoin Transactions, Information Asymmetry and Trading Volume (June 2, 2020)*. Lennart Ante. *Bitcoin transactions, information asymmetry and trading volume. Quantitative Finance and Economics*, 4(3):365–381, 2020.
- [3] Oliver Barsby. How many cryptocurrencies are there in 2021? <https://www.gfityesports.com/cryptocurrency/how-many-cryptocurrencies-are-there-2021-different-types-of-cryptocurrency/>, [Online; Accedido: 2021-02-25].
- [4] David Benson. Diagrams.net. <https://www.diagrams.net/>. [Online; Accessed: 2021-09-07].
- [5] Robert Kissell. *Algorithmic Trading Methods*. Academic Press, 2st edition, 2020. ISBN 9780128156308.
- [6] Silvia Amaro. Sell-offs could be down to machines that control 80 % of the us stock market, fund manager says, Dec 2018. <https://www.cnbc.com/2018/12/05/sell-offs-could-be-down-to-machines-that-control-80percent-of-us-stocks-fund-manager-says.html>, [Online; Accedido: 2021-02-25].
- [7] Thibaut Théate and Damien Ernst. An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications*, 173:114632, 2021.
- [8] Brett Steenbarger. *Enhancing trader performance: proven strategies from the cutting edge of trading psychology*, volume 276. John Wiley & Sons, 2012.
- [9] Steve Ward. *High Performance Trading: 35 Practical Strategies and Techniques to Enhance Your Trading Psychology and Performance*. Harriman House Limited, 2010.
- [10] Tom James. Trading psychology. In *Commodity Market Trading and Investment*, pages 199–206. Springer, 2016.
- [11] Jesse. The open-source python framework for trading cryptocurrencies. <https://jesse.trade/>. [Online; Accessed: 2021-08-13].
- [12] Freqtrade. Freqtrade. <https://www.freqtrade.io/en/stable/>. [Online; Accessed: 2021-08-13].

- [13] Mike van Rossum. Gekko. <https://gekko.wizb.it/>. [Online; Accessed: 2021-08-13].
- [14] Chase Sillevs. Zenbot. <https://github.com/DeviaVir/zenbot>. [Online; Accessed: 2021-08-13].
- [15] Lotfi Asker Zadeh. Fuzzy sets, information and control, 8: 338-353. *MathSciNet zbMATH*, 1965.
- [16] Oscar Cordón, Francisco Herrera, and Antonio Peregrín. Applicability of the fuzzy operators in the design of fuzzy logic controllers. *Fuzzy sets and systems*, 86(1):15–41, 1997.
- [17] Werner Van Leekwijck and Etienne E Kerre. Defuzzification: criteria and classification. *Fuzzy sets and systems*, 108(2):159–178, 1999.
- [18] Asma Zaynah Dhunny, Jay Rovisham Singh Doorga, Zaheer Allam, Roddy Lollchund, and Ravindra Boojhawon. Identification of optimal wind, solar and hybrid wind-solar farming sites using fuzzy logic modelling. *Energy*, 188:116056, 2019.
- [19] Joan Baez Ulic Dimatira, Elmer Dadios, Francisco Culibrina, Jo-Ann Magsumbol, John De-la Cruz, Kristine Sumage, Mary Tamar Tan, and Mike Gomez. Application of fuzzy logic in recognition of tomato fruit maturity in smart farming. In *2016 IEEE Region 10 Conference (TENCON)*, pages 2031–2035. IEEE, 2016.
- [20] Caro Lucas, Rasoul Mohammadi Milasi, and Babak Nadjar Araabi. Intelligent modeling and control of washing machine using locally linear neuro-fuzzy (llnf) modeling and modified brain emotional learning based intelligent controller (belbic). *Asian Journal of Control*, 8(4):393–400, 2006. doi: <https://doi.org/10.1111/j.1934-6093.2006.tb00290.x>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1934-6093.2006.tb00290.x>.
- [21] Mohamed Salem Almardy. *Design of fuzzy logic controller for the cathodic protection of underground pipelines*. PhD thesis, 1999.
- [22] James Warren, Gleb Beliakov, and Berend Van Der Zwaag. Fuzzy logic in clinical practice decision support systems. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, pages 10–pp. IEEE, 2000.
- [23] Rafael Kelly, Rodolfo Elias Haber-Guerra, and Fernando Reyes. Lyapunov stable control of robot manipulators: A fuzzy self-tuning procedure. *Intelligent Automation & Soft Computing*, 5(4):313–326, 1999. doi: 10.1080/10798587.1999.10750611.
- [24] Jose Eugenio Naranjo, Carlos Gonzalez, Ricardo Garcia, Teresa de Pedro, and Rodolfo Elias Haber. Power-steering control architecture for automatic driving. *IEEE Transactions on Intelligent Transportation Systems*, 6(4):406–415, 2005. doi: 10.1109/TITS.2005.858622.
- [25] Igor Anikin and Ellina S Anisimova. Handwritten signature recognition method based on fuzzy logic. In *2016 Dynamics of Systems, Mechanisms and Machines (Dynamics)*, pages 1–5. IEEE, 2016.
- [26] Janusz Kacprzyk, Ronald Robert Yager, and Sławomir Zadrozny. A fuzzy logic based approach to linguistic summaries of databases. 2000.
- [27] Weijian Zheng, Dali Wang, and Fengguang Song. Fql: An extensible feature query language and toolkit on searching software characteristics for hpc applications. *arXiv preprint arXiv:1905.09364*, 2019.



- [28] Yavuz Çapkan, Erdi Şenol, and Ulu Cenk. Fuzzy decision mechanism for stock market trading. *Avrupa Bilim ve Teknoloji Dergisi*, (26):6–11, 2021.
- [29] Ahmed Tealab, Hesham Hefny, and Amr Badr. Short-term stock market fuzzy trading system with fuzzy capital management. *International Journal of Intelligent Engineering and Systems*, 11(3), 2018.
- [30] Venkat Narayana Rao, Shayideep Sangam, et al. Application of fuzzy logic in financial markets for decision making. *International Journal of Advanced Research in Computer Science*, 8(3), 2017.
- [31] Ebrahim Mamdani. Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE transactions on computers*, 26(12):1182–1191, 1977.
- [32] Adam Marszałek and Tadeusz Burczyński. Modeling and forecasting financial time series with ordered fuzzy candlesticks. *Information sciences*, 273:144–155, 2014.
- [33] Partha Roy, Ramesh Kumar, and Sanjay Sharma. Fuzzy candlestick based stock market trading system using hammer pattern. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, 7(1):6–10, 2014.
- [34] Yoke Leng Yong, Yunli Lee, Xiaowei Gu, Plamen Angelov, David Chek Ling Ngo, and Elnaz Shafipour. Foreign currency exchange rate prediction using neuro-fuzzy systems. *Procedia computer science*, 144:232–238, 2018.
- [35] Pei-Chann Chang, Jheng-Long Wu, and Jyun-Jie Lin. A takagi–sugeno fuzzy model combined with a support vector regression for stock trading forecasting. *applied soft computing*, 38:831–842, 2016.
- [36] Tomohiro Takagi and Michio Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE transactions on systems, man, and cybernetics*, (1):116–132, 1985.
- [37] Tingwei Gao and Yueting Chai. Improving stock closing price prediction using recurrent neural network and technical indicators. *Neural computation*, 30(10):2833–2854, 2018.
- [38] Wayne Alexander Thorp. Id'ing when to buy and sell using the stochastic oscillator. *AAII Journal*, pages 24–34, 2000.
- [39] John Welles Wilder. *New concepts in technical trading systems*. Trend Research, 1978.
- [40] Adrian Țăran-Moroşan. The relative strength index revisited. *African Journal of Business Management*, 5(14):5855–5862, 2011.
- [41] Muhammad Asad Khan. *Impact of Technical Analysis on Karachi Stock Exchange of Pakistan*. PhD thesis, Qurtuba University of Science & Information Technology, Peshawar, 2016.
- [42] Matt Lutey and David Rayome. A primer on the ichimoku cloud indicator. *Journal of Marketing Development & Competitiveness*, 14(3), 2020.
- [43] SciKit-Fuzzy. Scikit-fuzzy. <https://pythonhosted.org/scikit-fuzzy/>. [Online; Accessed: 2021-08-13].

- [44] Simone Spolaor, Caro Fuchs, Paolo Cazzaniga, Uzay Kaymak, Daniela Besozzi, and Marco Nobile. *Simpful: A user-friendly python library for fuzzy logic*. *International Journal of Computational Intelligence Systems*, 13(1):1687–1698, 2020.
- [45] Eric Jones, Travis Oliphant, Pearu Peterson, et al. *Scipy: Open source scientific tools for python*. 2001.
- [46] Rajarshi Biswas, Xiaoyi Lu, and Dhabaleswar Panda. Designing a micro-benchmark suite to evaluate grpc for tensorflow: Early experiences. *arXiv preprint arXiv:1804.01138*, 2018.
- [47] Kasun Indrasiri and Danesh Kuruppu. *GRPC: Up and Running: Building Cloud Native Applications with Go and Java for Docker and Kubernetes*. O’Reilly Media, Inc., 2020.
- [48] Daniele De Martini, Gianluca Roveda, Alessandro Bertini, Agnese Marchini, and Tullio Facchinetti. A framework for automatic generation of fuzzy evaluation systems for embedded applications. *Nin*, 1:1, 2017.
- [49] Kim Möller. Developing a graphical user interface for creating chatbot configurations. 2018.
- [50] Peter Volgyesi, Abhishek Dubey, Timothy Krentz, Istvan Madari, Mary Metelko, and Gabor Karsai. Time synchronization services for low-cost fog computing applications. In *2017 International Symposium on Rapid System Prototyping (RSP)*, pages 57–63. IEEE, 2017.
- [51] SQLite. What is sqlite? <https://www.sqlite.org/index.html>. [Online; Accessed: 2021-08-13].
- [52] Peewee. Peewee: a simple and small orm. <http://docs.peewee-orm.com/en/latest/>. [Online; Accessed: 2021-08-13].
- [53] Binance. Binance api, Jun 2021. <https://binance-docs.github.io/apidocs/spot/en>, [Online; Accedido: 2021-05-25].
- [54] Gary Houston. *Iso 8601: 1988 date/time representations*, 1993.
- [55] Wee Mien Cheung and Uzay Kaymak. A fuzzy logic based trading system. In *Proceedings of the Third European Symposium on Nature-inspired Smart Information Systems*, volume 59, pages 1–60. Citeseer, 2007.
- [56] Amaury Hernandez-Aguila, Mario Garcia-Valdez, and Oscar Castillo. Money management for a foreign exchange trading strategy using a fuzzy inference system. In *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, pages 275–286. Springer International Publishing, 2018. doi: 10.1007/978-3-319-71008-2\_21. URL [https://doi.org/10.1007%2F978-3-319-71008-2\\_21](https://doi.org/10.1007%2F978-3-319-71008-2_21).
- [57] Yavuz Çapkan, Erdi Şenol, and Cenk Ulu. Fuzzy decision mechanism for stock market trading. *European Journal of Science and Technology*, jun 2021. doi: 10.31590/ejosat.951586. URL <https://doi.org/10.31590%2Fejosat.951586>.
- [58] Przemysław Juszczuk and Lech Kruś. Soft multicriteria computing supporting decisions on the forex market. *Applied Soft Computing*, 96:106654, 2020.
- [59] Giovanni Acampora. Fuzzy markup language: A xml based language for enabling full interoperability in fuzzy systems design. In *On the power of fuzzy markup language*, pages 17–31. Springer, 2013.

## Lista de acrónimos

**RSI** Índice de fuerza relativa

**SO** Oscilador estocástico

**TFM** Trabajo fin de máster

**RPC** *Remote Procedure Callback*

**IDL** *Intermediate definition language*



# Apéndice A

## Conceptos

### **Análisis técnico**

Es el estudio del mercado y del precio de activos usando patrones que se encuentran en el precio y en el volumen.

### ***Backtesting***

Es el proceso en el que una estrategia de *trading* es probada antes de ser empleada en un mercado en tiempo real. Permite al usuario conocer que si la estrategia funciona correctamente,

### ***Exchange***

Un exchange de criptomonedas es una plataforma en la que se realizan el intercambio entre unas criptomonedas y otras. También se puede convertir una criptomoneda a dinero fiduciario.

### **Futuros**

Es un tipo de producto financiero usado en el mundo de las inversiones que permite abrir operaciones en largo y en corto.

### ***Intraday***

Son los movimientos del precio de un activo que ocurren durante un día.

### **Liquidez**

Una buena liquidez se produce cuando un activo tiene un volumen diario alto, tener buena liquidez permite que el activo sea convertido a efectivo rápidamente. Un inmueble es una inversión sin liquidez.

### **Operación en corto (*short*)**

Es una operación en la que un activo financiero que no se posee (se toma prestado) se vende a un precio específico con la intención de comprarlo cuando este haya disminuido su precio para devolver el activo que ha sido prestado.

### Operación en largo (*long*)

Es una operación en la que un activo financiero se compra a bajo precio con la intención de venderlo cuando este haya aumentado su precio.

### Oscilador

Son un tipo de indicador técnico que permite identificar si un activo está sobrecomprado o sobrevendido.

### *Periodo o intervalo temporal*

Es la cantidad de tiempo más pequeña en la que se analiza el mercado en una determinada estrategia. Los intervalos temporales se definen por un comienzo, un cierre, el precio en el comienzo, el precio en el cierre, el precio máximo y mínimo en el periodo y el volumen operado en el periodo. Los intervalos temporales en el *trading* se pueden dividir en bajos que son aquellos menores a 1 hora, medios que son aquellos entre 1 hora y 4 horas y grandes que son aquellos mayores a 4 horas.

### *Premarket*

Es el periodo de actividad que ocurre en los mercados antes de que estos abran.

### Sobrecomprado

Ocurren en los mercados cuando un activo es comprado en exceso y sube a la parte superior del oscilador.

### Sobrevendido

Ocurren en los mercados cuando un activo es vendida en exceso y baja a la parte inferior del oscilador.

### *Stop loss*

Es el precio al que se cierra un operación de forma automática cuando esta está en pérdidas. La traducción en castellano sería “detener pérdidas”.

### Vela japonesa

Se usan las velas japonesas para representar el precio de los activos. Permiten representar los cuatro valores más importantes del precio durante un periodo: apertura, cierre, máximo y mínimo. En la Figura A.1 se muestra un ejemplo de dos velas japonesas.

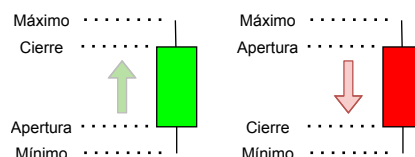


Figura A.1: Vela japonesa

**Volatilidad**

Mide los rangos de precios del mercado. Cuanto mayor sea el rango de precios desde el mínimo al máximo, mayor volatilidad tiene el mercado.

**Volumen**

Es el número de operaciones que son realizadas durante un periodo de tiempo.