

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

Departamento de Ingeniería de Software y Sistemas Informáticos



MÁSTER UNIVERSITARIO DE INVESTIGACIÓN EN INGENIERÍA DE
SOFTWARE Y SISTEMAS INFORMÁTICOS

31105151- Trabajo Fin de Máster

Codificación de Estrategias de Inversión en Mercados de Capitales: Aproximación Teórico Práctica mediante Asistente Automático

Autor

D.IGNACIO FUMANAL ANDRÉS

Director

D. Ismael Abad Cardiel

Septiembre 2022.

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

Fecha: 10/Septiembre/2022

Quién suscribe:

Autor(a): D. Ignacio Fumanal Andrés
D.N.I./N.I.E./Pasaporte.: 02858779V

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

Título completo del trabajo.

**“Codificación de Estrategias de Inversión en Mercados de Capitales:
Aproximación Teórico Práctica mediante Asistente Automático”**

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.



Resumen

La inversión de dinero con fines de crecimiento de capital es una actividad que ha ganado millones de adeptos desde la temprana existencia de los mercados de inversión.

Con la evolución y tecnologización del mercado de capitales la afluencia de dinero de particulares y empresas ha creado una masa dineraria flotante de tamaño enorme, que ha dado lugar a un interés sin precedentes por hacerse con una pequeña parte de esa riqueza.

Mediante técnicas de análisis de mercado se han logrado grandes avances en el seguimiento de la evolución del precio de los diferentes activos, hasta alcanzar una capacidad muy alta de detectar tendencias, patrones, y mecanismos de predicción de avances y retrocesos del valor.

La formulación del momento en el cual entrar y salir del mercado y por cuanto tiempo permanecer en él en una posición de compra o de venta y con qué volumen y qué exposición al riesgo de pérdidas, es lo que llamamos una estrategia de inversión. Con el advenimiento de sistemas basados en lógica difusa, ha sido posible flexibilizar las estrategias de inversión para formularlas de manera más próxima a la lógica y verbalización humanas, más analógicas, y menos rígidas y digitales.

En este trabajo se desarrolla la formulación de estrategias de inversión tal como pensamos los seres humanos, y se propone una mejora de traducción de esas estrategias a su codificación automática en código software mediante la aplicación de ingeniería de software.

Lista de Palabras Clave. Estrategia, inversión, trading, lógica difusa, lógica borrosa, Metatrader4, MQL4

INDICE DE CONTENIDOS

Listado de Figuras	10
Listado de Tablas	12
1. Introducción	14
1.1. <i>Objetivo General del trabajo</i>	
1.2. <i>Objetivos Específicos</i>	
2. Problemática y Estado del Arte	19
2.1. <i>Factores Motivacionales</i>	
2.2. <i>Estado del Arte</i>	
2.3. <i>Limitaciones</i>	
2.4. <i>Tendencias</i>	
3. Fundamentos de Trading	40
3.1. <i>Qué es trading</i>	
3.2. <i>Las Ondas de Elliot</i>	
3.3. <i>Criterios de Análisis de mercado</i>	
3.3.1 <i>Análisis Fundamental</i>	
3.3.2 <i>Análisis Técnico</i>	
1 <i>Ejemplo de Indicador Técnico: RSI</i>	
3.4. <i>Programabilidad: Plataformas y Lenguajes</i>	
3.5. <i>Estrategia Técnica de Trading</i>	
3.5.1 <i>Estrategia con Matemática Clásica</i>	
3.5.2 <i>Fundamentos de Lógica Difusa</i>	
3.5.3 <i>Estrategia Técnica Difusa</i>	

4. Evaluación y Comparación de la Solución	83
4.1. <i>Evaluación de la Solución</i>	
4.2. <i>Comparación de la Solución</i>	
5. Conclusiones y Trabajos Futuros	100
6. Referencias y Bibliografía	107
7. Anexos	114
7.1. <i>Anexo I. Código fuente Java</i>	
7.2. <i>Anexo II. Código fuente MQL4, generador por AutoTrader</i>	

Listado de figuras

Figura 1. Dos representaciones de una sesión mediante velas japonesas

Figura 2. Fórmulas de definición de velas difusas

Figura 3. Patrón de cinco ondas de Elliot

Figura 4. Reglas clásicas para un patrón Kicking bullish

Figura 5. Pseudo código del algoritmo de trading.

Figura 6. Etapas de un sistema Experto Fuzzy. (Thakera, S., Nagorib, V. 2018)

Figura 7. Ondas de Elliott numeradas

Figura 8. Resultados de trading con y sin EWT como mecanismo predictor de mercado

Figura 9. The FAO Cereal Price Index evolution.

Figura 10. World Cereals Price Index (Trading Economics.com)

Figura 11. Indicador RSI gráfico con intervalo 0 (sobreventa) a 100 (sobrecompra) (Naranjo, R. 2018)

Figura 12. Gráfica del activo de divisas EUR/USD

Figura 13. Plataforma comercial Metatrader 4. Vista general

Figura 14. Función de pertenencia del valor RSI en matemática clásica

Figura 15. Función de pertenencia de la variable de entrada en matemática difusa

Figura 16. Función de pertenencia de la variable de salida en lógica difusa

Figura 17. Función de pertenencia del valor RSI en los conjuntos borrosos definidos en t-2

Figura 18. Función de pertenencia del valor RSI en los conjuntos borrosos definidos en t-1

Figura 19. Métodos del Proceso de defuzzificación. (Jarusek, Eva., Kotyrba, M. 2021)

Figura 20. Ubicación de los centroides de la variable de salida

Figura 21. Obtención del valor *defuzzificado* de la variable lingüística de salida

Figura 22. Conjuntos difusos múltiples para estrategia de trading

Figura 23. Detalle de la pantalla de Ayuda del editor Eclipse de Java

Figura 24. Pantalla de AutoTrader

Figura 25. Nivel de indicador RSI para posición de venta

Figura 26. Selección del activo de operativa en mercado

Figura 27. Selección del indicador técnico para operativa

Figura 28. Autorización para operar en compras, en ventas y posiciones mixtas

Figura 29. Selección de los contratos de compra o venta en operativa

Figura 30. Detalle del botón generador del fichero con código MQL4

Figura 31. Carpeta conteniendo código MQL4 generado desde AutoTrader: AutoTrader.mq4

Figura 32. Editor MetaEditor de la plataforma Metatrader4. Código fuente generado: AutoTrader.mq4

Figura 33. Código fuente del método OnInit() generado por AutoTrader

Figura 34. Código fuente del método check_RSI_level()

Figura 35. Código fuente del método goBuy()

Figura 36. Método Java que pasa a fuente MQL4 los parámetros seleccionados por el trader

Figura 37. Fabricantes de plataformas low-code

Figura 38. Cuadrante mágico Gartner de plataformas low-code

Listado de Tablas

Tabla 1. Recomendación según los tres criterios de Othman & Schneider (2010)

Tabla 2. Reglas difusas para el patrón Kicking bullish

Tabla 3. Tabla de Reglas de Inferencia. Sistema Mamdani

Introducción

- 1.1 Objetivo General
- 1.2 Objetivos Específicos

Introducción

Está en la naturaleza del ser humano el adquirir bienes y hacerlos crecer. Con los bienes obtenidos se busca la perpetuación de la especie mediante la alimentación y el abrigo.

La acumulación de bienes más allá de lo necesario para la subsistencia permite al ser humano subir peldaños en la pirámide de Maslow donde la parte inferior apunta a las necesidades básicas, y la parte superior apunta a la obtención y acumulación de bienes de confort no imprescindibles para la vida diaria. La pirámide de Maslow la define así el IEP (iep.edu.es/las-5-fases-de-la-piramide-de-maslow) *“La pirámide de Maslow es una teoría sobre la motivación y trata de explicar qué impulsa la conducta humana.”*

En la edad media el modelo de intercambio de bienes por otros bienes necesarios que tienen otros da lugar al trueque, por medio del cual dos propietarios intercambiaban para su beneficio particular dos bienes que eran mutuamente necesarios o deseados. En la visión más liberal y evolucionada de la economía Adam Smith define *“como esos bienes intercambiados se originan en el excedente no necesario que tiene cada individuo implicado en la transacción y como cada individuo parece estar especializado en la adquisición o incluso producción del bien excedente que vende ‘intercambia’ con otro individuo productor de otro bien”* (Smith, A. 1794). Y apunta como ese intercambio da lugar a la especialización en la producción.

La recurrencia de ese modelo social económico de ganancia reiterada por venta de producción va progresivamente llamando al nacimiento de nuevos mercados donde intercambiar bienes producidos, ahora ya para venta y enriquecimiento.

Llevado al mundo de las divisas se puede comprar y vender divisas, y de hecho es habitual hacerlo en mesas de cambio, en oficinas de cambio, donde se entrega la moneda que vendes -por ejemplo, euros- y a cambio recibes la divisa deseada al precio que en ese momento esté cotizando la tasa de cambio de ambas divisas, más la comisión comercial que obtiene el cambista.

En este trabajo se analiza toda la mecánica de compra y venta de bienes en su forma divisa, utilizando las llamadas estrategias de inversión que permiten entender cuando es mejor comprar o vender y a qué precio. Siempre visto desde una perspectiva puramente de comercio para el enriquecimiento.

1.1 Objetivo General

Este trabajo explicará y tratará de demostrar como el lenguaje humano que expresa -en forma de estrategia- las intenciones de invertir bajo ciertas condiciones de mercado puede ser mapeado a una codificación de lenguaje de programación de dominio (lenguajes diseñados y especializados en negociar en el mercado de divisas) mediante un sistema interactivo de menús desplegados que capte las intenciones de inversión del *trader*, plasmando esas intenciones en código fuente - lenguaje MQL4- de un sistema experto de la plataforma Metatrader4.

1.2 Objetivos Específicos

- 1.2.1 Se demostrará la posibilidad de automatización de la codificación de una estrategia de inversión formulada con todos los parámetros que incluye
- 1.2.2 Se aplicará una aproximación generativa mediante lenguaje Java para producir un Sistema Experto (Expert Advisor) en lenguaje propio de plataforma de trading
- 1.2.3 Se demostrará que el factor humano de mayor influencia en la generación de pérdidas en las negociaciones en bolsa -la psicología temerosa de perder y ambiciosa de ganar más- puede ser eficientemente minorizado mediante la utilización de herramientas asistentes de codificación automática. Estas herramientas reemplazan el factor humano negativo de duda con el factor tecnológico de pragmatismo.

Problemática y Estado del Arte

2.1 Factores Motivacionales

2.2 Estado del Arte

2.3 Tendencias

2.4 Carencias

Problemática y Estado del Arte

2.1 Factores Motivacionales

Ganar dinero prediciendo la evolución del precio de algún valor en bolsa; esa motivación concita gran atención, y hay ya numerosos estudios en torno al tema desde hace decenas de años, cada vez involucrando más a la tecnología.

La promesa de obtener ganancias dinerarias -cuanto más rápidas mejor- hizo de la negociación en mercado de valores un campo de estudio e investigación muy popular durante las últimas décadas.

Tradicionalmente la operativa en bolsa se había realizado por análisis fundamental de los eventos locales y globales o por análisis técnico. La mala interpretación del análisis fundamental de eventos o la sobreestimación de indicadores técnicos a la hora de abrir posiciones llevaban a cerrar las operaciones con pérdidas. En la búsqueda de herramientas que ayudaran a los analistas a tomar mejores decisiones de inversión, la introducción de la informática y sus herramientas representó un gran avance. La relevancia de estas se hizo evidente al comprobar los resultados de su aplicación. Por ejemplo, en el campo del comercio electrónico se pudo observar un incremento de eficiencia operativa, así como una experiencia de mayor cercanía del cliente a toda la oferta de productos en cualquier campo comercial.

Lógica difusa, redes neuronales, sistemas mixtos, todos serán mencionados varias veces en este documento como aplicación de la informática a la predicción de la evolución del valor de los activos de mercado.

Dada la atención que ha captado este tema en la ciudadanía, la banca y la industria, consideramos que tiene suficiente relevancia como para escribir este documento en una

evaluación del progreso alcanzado hasta ahora, y en llamar la atención sobre conceptos que aún tienen un largo recorrido para ser investigados.

A lo largo de este documento se debatirán diversos enfoques y mecanismos aplicados como parte del estado del arte de la aplicación de la lógica difusa en trading.

Como parte de este repaso conceptos como Sistemas de ayuda a la toma de decisiones, lógica difusa y reglas IF THEN, redes neuronales, y muchos más se traerán a la atención del lector como parte fundamental del debate sobre las teorías que han aportado contribuciones en todo este tema.

Repasaremos los principales contribuidores en este campo -razón fundamental de la elección de la bibliografía a final de documento- principalmente por los casos de uso presentados en sus trabajos, que incluyen experimentos de aplicación de sistemas difusos. Se comentan sus contribuciones a las pruebas completados como sistemas de apoyo en la toma de decisiones de operativa, y los resultados obtenidos.

Al final del trabajo se facilitan más de treinta referencias de artículos que aportan información muy relevante en este campo. Estos artículos han sido referenciados a lo largo de este trabajo allá donde es pertinente dado la abundante información que existe en torno a este tema, y respetando escrupulosamente la autoría original para no caer en plagio. Se han empleado estos artículos para investigar los diferentes puntos de vista de fuentes muy diversas; de hecho, la abundancia de material en este tema hace compleja la selección de la bibliografía, entre la cual no podían faltar por supuesto los trabajos del Dr. Lofti Zadeh.

El reto de juntar y seleccionar las contribuciones más relevantes en este campo es muy grande. Tal como ya hemos mencionado este tema se ha hecho tan popular que las contribuciones existentes son abundantes. Revisaremos algunas de estas contribuciones basándonos en criterios tales como obtención de resultados positivos en ganancias, evolución prometedora de más resultados, o incluso los activos -beneficios- obtenidos tras la investigación.

Como se mencionó más arriba fue la introducción de la informática lo que creó un hecho relevante en varios campos, pero específicamente en el análisis del mercado de valores y el

almacenamiento de todo ese conocimiento y experiencia, y su aplicación automatizada bajo el nombre de *algotrading* (*algorithmic trading*).

Una experiencia temprana en la introducción de la informática fueron los sistemas de apoyo a la toma de decisiones, DSS, solución propuesta por Shim et al (2002) cuyo objeto era la creación de un sistema inteligente de agendado de reuniones utilizando agentes.

Su propuesta incluía un MNA o agente negociador de la reunión, un FIA o agente de inferencia difusa y un GLA o agente de aprendizaje genético.

Los tres agentes colaborarían entre ellos tal que FIA y GLA ayudarían al organizador de la reunión en su reserva mientras que el MNA desarrolla las tareas de recepción de la información de todos los asistentes a la reunión para ser reenviada al FIA.

La utilización de ambos probó ser una gran colaboración entre DSS y tecnología de agentes para lograr cierto grado de automatización y ayuda a humanos a través de la introducción de herramientas informáticas. En un paso más allá la investigación comenzó a incluir la lógica difusa. La lógica difusa la introdujo el Dr. Lofti Zadeh (1965) durante su estancia en la universidad de Berkeley, como una forma alternativa de evaluar la verdad -más y mejor adaptada a cómo vemos el mundo los humanos- desde un punto de vista diferente donde los tonos grises tomaban relevancia frente a solamente el blanco y el negro de la verdad en el enjuiciamiento de las cosas. Tal como menciona Martin (1982) en su Teoría de los Conjuntos Borrosos y la Toma de Decisiones, la pertenencia puede ser simple, escalonada, lineal, o de otros tipos, dependiendo de la intensidad de pertenencia. Más adelante en este trabajo se mencionan las diferentes funciones de pertenencia existentes.

A modo de ejemplo tomemos el caso de las horas del día y de la noche. Empleando el principio de verdad tradicional evaluaríamos como día solamente las horas en que brilla el sol, y como noche solo las horas en que hay oscuridad.

Ese mismo evento desde la perspectiva de la lógica difusa podría verse como una función continua progresiva según la luz del día crece o disminuye, de manera que según aumenta la luz progresivamente en el alba la etiqueta "es de día" va ganando más peso. Inversamente la etiqueta "es de noche" ganaría más peso según va anocheciendo. Si creáramos dos conjuntos matemáticos, uno llamado *HorasdeDia* y otro llamado *HorasdeNoche* entonces

cada hora de las 24 horas del día podría -en lógica difusa- pertenecer a cada uno de estos dos conjuntos con un valor tal que expresara -entre 0 y 1- cuanto cumple esa hora del día con “ser de día” o “ser de noche” respectivamente. Escojamos la hora 10am a modo de ejemplo práctico. Esa hora podría perfectamente pertenecer al conjunto *HorasdeDia* con un valor de 0.99, ya que cumple totalmente con que las 10am es una hora con brillo máximo del sol, o casi máximo; mientras que la pertenencia de la hora 10am al conjunto *HorasdeNoche* podría tener un valor aproximado de 0,01 porque su pertenencia a este conjunto refleja muy bajamente el concepto que representa este conjunto.

En este rango continuo de valores desde 0 hasta 1 la membresía máxima en un conjunto estaría expresada por el valor 1, y la mínima por el 0. La lógica difusa representa mejor la realidad humana analógica que la lógica binaria de verdadero o falso, o de pertenecer o no pertenecer; si bien la lógica difusa podría dar lugar a vaguedad o imprecisión.

Por otra parte, las redes neuronales emergieron como un campo de la inteligencia artificial en un intento de modelar los datos en una representación del comportamiento humano y el modo de pensamiento humano. Como puede verse claramente aplicar redes neuronales a problemas habituales emplearía muchísimos recursos computacionales haciendo la solución no ideal excepto para ciertas aplicaciones concretas.

Si tomamos las mejores cosas de ambos -lógica difusa y redes neuronales- surge otro concepto: sistemas neuro difusos, que abarca las redes neuronales en donde las variables de entrada se transforman en conjuntos difusos. Esta arquitectura tiene una ventaja básica consistente en que el conocimiento se crea desde datos acumulados, y así los resultados se interpretan con el uso de variables verbales difusas.

Como resultado, aplicando todo esto a trading podemos decir que los progresos realizados en predicción de valores en bolsa son sólidos no solo desde la perspectiva de la investigación sino también desde la operativa misma en el mercado de valores.

Estaríamos no obstante descuidando nuestro análisis si no indicáramos áreas en las que es necesario aun investigar más y más profundamente con la idea de mejorar la calidad de las predicciones de valores bursátiles. Esto lo haremos en la última parte de este trabajo.

En el plano más organizativo de la información investigada se ha encontrado tres capítulos principales de información, a saber: Lógica Difusa en torno al Trading, Redes Neuronales y Sistemas aplicados al Trading, Sistemas de Apoyo a la toma de decisiones. Es tal la cantidad de información encontrada que hemos centrado este trabajo fundamentalmente en los aspectos concretos de diseño de sistemas difusos de recomendación de operativa de trading -ayuda a la toma de decisión de comprar, vender o no operar-, si bien algunos trabajos enfatizan más en aumentar los beneficios y otros en disminuir el riesgo de pérdidas.

Con objeto de dar una perspectiva visual directa hemos aportado tablas -y figuras- que reflejan aspectos concretos de las técnicas que proponen los autores (ej. velas japonesas difusas, reglas difusas de entrada, ondas de Elliot, etc.).

2.2. Estado del Arte

Si miramos atrás, entre los trabajos de investigación realizados en el pasado en lógica difusa -de ahora en adelante en este trabajo anotado como FL- nos encontramos con el trabajo de Simutis (2000), en su investigación acerca de aplicaciones basadas en sistemas expertos que aprenden habilidades estrictamente humanas. Este incluía tanto datos almacenados históricos sobre negociación en mercado -de ahora en adelante en este trabajo anotado como Trading- como conocimiento experto aplicado.

Ambos fueron entonces formulados como sentencias FL del tipo IF THEN, sentencias lógicas condicionales de comportamiento.

La simulación del pensamiento humano es en sí mismo un objetivo claro de investigación. Doura et al (2002) trataron de aproximarse a este concepto en su trabajo *“Invertir empleando análisis técnico y Lógica difusa”*. El resultado de la investigación probó ser muy exitosa cuando varias compañías adoptaron sus teorías con la idea de aplicarlas formalmente, y así lo hicieron con éxito. FL fue nuclear en el enfoque que dieron y por lo tanto altamente reconocida en su aplicación práctica en campo.

Hasta su inclusión en el proceso de análisis, la decisión frente a una oportunidad de trading sin FL era clásicamente sólo comprar o vender, en caso de tomar una acción -siempre quedaba la opción de no operar ante la incertidumbre o escasez de indicadores

fundamentales o técnicos que invitaran a hacerlo-. La contribución de Othman & Schneider (2010) difuminó este enfoque aplicando FL de tal manera que empleando tres criterios de juicio sobre la tendencia de un activo, por ejemplo, la opinión de un experto -o sistema experto-, la ganancia por acción y la ratio precio/ganancias se podría formular una gama de recomendaciones no binarias -o sea en el conjunto de los sistemas difusos- sobre la acción a tomar, tal como COMPRAR FUERTE, COMPRAR, MANTENER, VENDER, VENDER FUERTE, cumpliéndose así la implantación formal de una lógica borrosa donde tradicionalmente todo se analizaba desde una perspectiva que tan solo ofrecía o una acción u otra, COMPRAR/VENDER o NO OPERAR-. El primer criterio, EO -opinión del experto- era binaria, es decir, si el precio es alto entonces la recomendación era vender y si el precio es bajo entonces la recomendación era comprar, por lo que el valor de EO era 0 o 1, mientras que los otros dos criterios sí que evidenciaban el tratamiento como un comportamiento difuso en un rango de valor real entre 0 y 1. Juntando el rango de valores de los tres criterios los autores crearon tablas de inferencia -con condicionales tipo IF THEN- para llegar a una recomendación integrada sobre compra o venta.

Tabla 1. Recomendación según los tres criterios de Othman & Schneider (2010)

TABLE I. EVALUATION RESULT FOR COMPANY A AND B.

Company	E/O	EPS	P/E	Recommendation (R)
A	1	0.909	0.88	0.75
B	0	0.667	0.15	0.454

En la tabla 1 podemos ver que el sistema de inferencia aplicado saca como resultado una recomendación -R- con valor entre 0 y 1 para comprar o vender activos -a lo largo de este trabajo hablamos de activos refiriéndonos a cualquier elemento financiero negociable en bolsa, si bien mencionamos en algún caso cual fue el activo sujeto de las pruebas de eficacia que probaron los autores de los trabajos estudiados-. Este sistema tiene la ventaja de ser difuso y de ser multicriterio, de ahí los buenos resultados obtenidos en dos aplicaciones en el NASDAQ y el IBEX cuando son comparados con sistemas previos no difusos.

No obstante, una conclusión importante es que esta recomendación no releva al operador de su responsabilidad última de ejecutar la compra o la venta con sus consecuencias. Esta es

la parte emocional que la inteligencia artificial no puede resolver frente al ser humano, aún.

La contribución de Simutis (2000) define la FL aplicada a informática -soft computing- como la acomodación a la imprecisión real del mundo. Efectivamente tal como vamos viendo en distintas contribuciones la mayor contribución de la FL es analizar datos con la amplia gama de tonalidades grises que compone cualquier decisión en una lógica humana no binaria. Este trabajo de Simutis (2000) fue una de las bases para el trabajo de Othman & Schneider (2010), siendo este trabajo una aplicación concreta a un sistema experto multicriterio y difuso.

En Trading una visualización posible de la evolución de precios de un activo es mediante las llamadas velas japonesas -otros sistemas de visualización incluyen la gráfica lineal, las barras de apertura, cierre, máximo y mínimo, etc.-, que consiste en un método gráfico muy sencillo que en cada sesión marca cuatro valores por cada “vela” o línea vertical coloreada que queda dibujada en la gráfica y que refleja para esa sesión los respectivos valores de apertura, de cierre, el más alto de la sesión, el más bajo de la sesión. Habitualmente las velas que representan en la sesión una ganancia de valor para el activo se colorean en verde mientras que las que representan pérdida se colorean en rojo. Este criterio es totalmente arbitrario y es el operador quien decide qué atributos de visualización aplica en su pantalla de operativa de mercado.

En su trabajo Naranjo y Santos (2019) emplearon la versión difusa de las velas japonesas para profundizar en una herramienta de ayuda a la decisión de tendencia del precio.

En su búsqueda de factores añadidos que influyen en la reducción de incertidumbre de la evolución del precio encontraron en su investigación la importancia que tiene la devaluación de la moneda con el paso del tiempo, e incluyeron esa variable en su consideración extendida de predicción de la tendencia del valor.

Más abajo podemos ver en la Figura 1 una representación de una vela japonesa, que se caracteriza por enseñarnos cuatro valores en la sesión temporal que representa -ya sea una sesión de un minuto o de un día o una semana-, a saber: precio de apertura y cierre que son los límites superior e inferior del cuerpo sólido de la vela, y precio máximo y mínimo de la sesión que se corresponden con los límites superior e inferior respectivamente de las

llamadas mechas de la vela, que son las líneas verticales que salen desde el cuerpo de la vela hacia arriba y hacia abajo. En la Figura podemos ver junto a cada vela -una de tendencia alcista o ganancia y otra de tendencia bajista o pérdida- una representación de gráfica lineal equivalente a la evolución del precio de la vela respectiva.

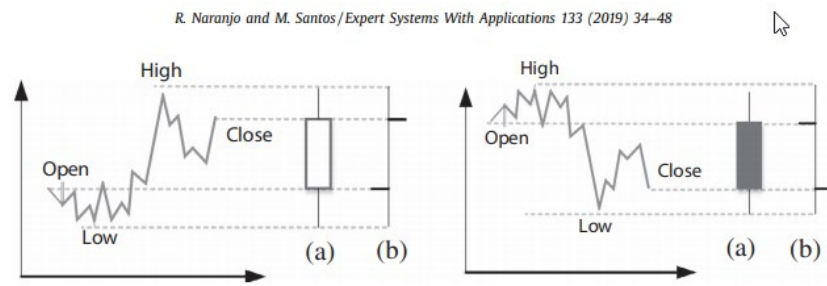


Figura 1. Dos representaciones de una sesión mediante velas japonesas.

El método seguido por los autores para trasladar el valor de una vela a unos parámetros de entrada de un sistema y con recorrido discreto fue aplicar como entradas los cuatro valores de la sesión y de ellos extraer sólo tres parámetros de entrada -estos ya difusos- dentro del rango 0 a 100: Lupper o longitud de la mecha superior, Lbody o longitud del cuerpo de la vela, y Llower o longitud de la mecha inferior de la vela. A continuación, parametrizaron con la siguiente fórmula estos tres valores, según vemos en figura 2.

$$L_{upper}(t) = 100 \cdot \frac{high(t) - \max(open(t), close(t))}{open(t)}$$

$$L_{lower}(t) = 100 \cdot \frac{\min(open(t), close(t)) - low(t)}{open(t)}$$

$$L_{body}(t) = 100 \cdot \frac{close(t) - open(t)}{close(t)}$$

Figura 2. Fórmulas de definición de velas difusas

Y a continuación crearon cuatro relaciones trapezoidales de membresía, a saber: NULA, CORTO, MEDIA, LARGO, que eran asignadas en cada sesión a cada una de las velas Llower, Lupper. Para el caso del parámetro Lbody proponían definir siete membresías triangulares y

trapezoidales con los siguientes tags: LARGO NEGRO, MEDIO NEGRO, CORTO NEGRO, NULO, CORTO BLANCO, MEDIO BLANCO, LARGO BLANCO. Las salidas difusas que obtienen en su planteamiento son una relación del tamaño del cuerpo de la vela con respecto al tamaño total de la vela, que lo llamaron RSize, más la posición dentro de la vela del cuerpo, a la que llamaron RPos. Para RPos definieron cinco funciones de membresía triangulares, y nueve a RSize. Mediante un sistema de inferencia de tipo Mamdani crearon reglas IF THEN hasta un total de 224 reglas difusas. Y de todas estas reglas se obtenía un criterio de predicción del precio basado en método k-NN -vecino más próximo-. Complementando este sistema con un criterio de entrada al mercado, una cantidad de capital y la estrategia de salida -cuando salir si entro al mercado en corto o largo- completaron su sistema, del cual concluyeron que tuvo una enorme importancia el haber considerado en trading tanto la devaluación de la moneda como la inflación como factores de influencia en su algoritmo. También incorporaron a su sistema un método regulado de gestión de capital. Su éxito se probó en el Nasdaq y en el índice Ibex, en los cuales obtuvieron beneficios aplicando su sistema difuso de predicción de la evolución del precio con velas japonesas. De hecho, su principal conclusión en el trabajo realizado es que en la predicción de evolución del precio tiene una enorme importancia considerar la evolución histórica de la devaluación monetaria, cosa que representa un hito de innovación en sistemas expertos de trading pues combina criterios técnicos con este criterio de análisis fundamental.

En un salto evolutivo se produjo hace varias décadas un planteamiento que consiste en que el mercado repite constantemente su comportamiento, a modo fractal. Esto lo estudió y emitió una hipótesis fundamental para el futuro el propio Elliot (1939) en su teoría de ondas.

Con los trabajos de Elliot (1939) en torno a la modelización -patronaje- de la evolución del precio Matviychuk (2006) desarrolla un artículo en torno a la bondad del modelo de Elliot -tres impulsos de subida de precio seguidos de dos impulsos de corrección a la baja- y la utilidad de un modelo de predicción que emplea esta teoría dada su bondad. Tal como hemos mencionado ya anteriormente en los sistemas de predicción de evolución del precio se recurre a múltiples factores como entrada a un sistema de inferencia de tipo Mamdani (1974) donde las consecuencias de las reglas son conjuntos difusos en el espacio de salida -versus el enfoque de Sugeno & Kanh (1988) donde las salidas son funciones nítidas de las

entradas-. La cuestión -Matviychuk (2006) - es cuales de todos los factores disponibles deben ser elegidos ya que la elección podría ser subjetiva y conducir a errores en caso de una elección no contrastada. En este punto es interesante hacer una parada para explicar que en el mundo del trading día existen no menos de cien indicadores técnicos a disposición de cualquier operador de bolsa. Adicionalmente existen indicadores fundamentales que también pueden ser incluidos simultáneamente en el algoritmo de predicción de evolución del precio del activo. Más abajo podemos ver en la Figura 3 una representación de evolución del precio cuando se mapea a un patrón de ondas de Elliott.

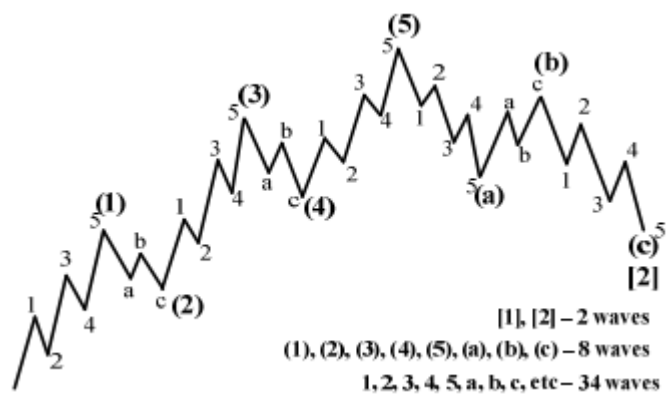


Figura 3. Patrón de cinco ondas de Elliot.

Como variables de entrada en su sistema propuesto y testado emplea una valoración de menor a mayor velocidad del precio -según avanza formando un patrón de Elliot- con términos como Slump ,Growth, Moderate Growth, Considerable Slump,etc., y así poder extraer de manera difusa -Moderate, Considerable- en qué fase del desarrollo del patrón Elliot está el precio. Para extraer una variable Y -que refleja la tendencia del precio- obtiene un promedio por pesos de varias entradas X. Si nos fijamos en la Figura 3 es esencial tal como menciona el autor no equivocarse en la extracción de datos ni en la aplicación de reglas IF THEN en qué momento está la gráfica respecto a la teoría de ondas de Elliot. Y con la aplicación de algoritmos de sistemas neuro difusos deja el sistema preparado para una batería de prueba de optimización de su sistema. El autor enfatiza la importancia de elegir bien las relaciones de membresía, que en el caso presente es de tipo trapezoidal ya que permite fijar límites a las variables.

En la etapa final de su trabajo Matviychuk (2006) realiza el análisis de eficiencia del sistema desarrollado. Para ello compra o vende activos cada vez que la variable Y saca el valor CS - considerable slump- o CG -considerable growth-, respectivamente. Los resultados obtenidos revelan una buena eficiencia general del sistema llegando hasta un 13.05% de rentabilidad, logrando 26 operaciones exitosas de un total de 42. Depositando un monto de 100.000\$ en la cuenta de trading el autor obtuvo en un año de negociación un balance final de 113.048\$. Su conclusión es que claramente la aplicación de FL a las ondas de Elliot hace del sistema un éxito, especialmente en la no necesidad de la opinión del experto ya que todo el sistema está basado en series temporales de datos financieros.

En un enfoque bastante radical respecto a otros trabajos previos hemos leído el trabajo de Morta (2015) con un enfoque muy esperable mirando en retrospectiva y sin embargo no tan frecuente en la bibliografía consultada. Se trata de construir un sistema de reglas, pertenencias, y entradas y salidas basadas en indicadores técnicos de trading. Aquí es importante hacer notar que el análisis técnico es algo que ha obtenido una atención muy alta recientemente, sin embargo, es el análisis fundamental el que tradicionalmente ha captado una atención casi completa. De manera concreta los autores emplean los indicadores MACD (Moving Average Convergence Divergence), ROC (Rate of Change) y Stochastic Oscillator. Con los datos facilitados por estos indicadores técnicos crearon en MATLAB la lógica de su sistema difuso de operativa que les generaba recomendaciones de COMPRA, VENTA, MANTENER. La prueba del sistema aplicada a una serie temporal registrada y guardada permitió conocer el comportamiento de su propuesta, de tal forma que en un período de 2.235 días de registro de resultados el sistema aplicado arrojó una recomendación de compra del 2%, mantener 90% y vender 8%. El intervalo difuso que determinaron como recomendación para compra fue cuando el valor de salida de la lógica era menor del 0.4, siendo los resultados mayores a 0.7 los que emitían una señal de recomendación de vender. El resto de los valores -90%- recomendaron mantener la posición, es decir no operar.

El trabajo de Naranjo (2019) vuelve de nuevo a captar la importancia de los patrones en el mundo del trading. Tal como hemos mencionado anteriormente toda la teoría de Elliot se basa en patrones. En este caso los autores se apoyan en el diseño de un sistema detector de

patrones que en la teoría de trading son anunciadores de buen momento para compra o venta de activos. Su sistema de patrón emplea los conocidos patrones de velas japonesas por nombre Hammer, Kicking y Piercing Line. Una vez reconocido un patrón se fusifica las velas y se normalizan las variables empleadas, para esto emplearon el trabajo de Leet et al. (2006). Como anteriormente, los métodos estudiados en este trabajo crearon una variable de entrada ya conocidas para nosotros llamadas Lupper, Lbody y Llower, pero además nuevas variables difusas fueron incluidas a saber Lgap (relación entre un gap y el valor de apertura o cierre de la sesión tras el gap), Trend(t) que es la tendencia de las dos últimas sesiones empleando la diferencia entre sus valores de cierre, y Ldifopen definido como la diferencia entre el mínimo valor de la sesión anterior y la apertura de la actual. Finalmente introducen una entrada llamada Ldifcentral que corresponde a la diferencia entre el valor central de la última sesión y el cierre de la actual sesión. Como variables de salida emplean un rango de 0 a 100 que indica la intensidad de la señal de venta ya que los patrones de reconocimiento detectan mercado en alza, de ahí la detección de velas bullish -velas que indican que la tendencia del precio es de caída- que indican la rotura de tendencia alcista. La variable x de salida es intercalada en intervalos en los valores 0,10,30,20-50, 40-70,60-90,80 y mayor, correspondiendo a cada uno una etiqueta de recomendación -sobre intensidad- de entrada: MANTENER, BAJO, MEDIO BAJO, MEDIO, MEDIO ALTO, ALTO. A continuación, en la Figura 4 podemos ver como se describen las reglas difusas de actuación para el patrón de vela Kicking. Y en la Tabla 1 podemos ver las reglas difusas definidas, también en este caso para una de las velas, la Kicking bullish.

Kicking Bullish:

```

IF High(t - 1) = Open(t - 1) AND
   Low(t - 1) = Close(t - 1) AND
   High(t) = Close(t) AND
   Low(t) = Open(t) AND
   Low(t) > High(t - 1) THEN Bullish = TRUE

```

Figura 4. Reglas clásicas para un patrón Kicking bullish

Tabla 2. Reglas difusas para el patrón Kicking bullish

Fuzzy rules for Kicking Bullish pattern.

Lupper (t-1)	Llower (t-1)	Lupper (t)	Llower (t)	Lgap	Lbody (t-1)	Lbody (t)	Bullish
NULL	NULL	NULL	NULL	LONG	BLACK	WHITE	HIGH
NULL	NULL	NULL	NULL	MIDDLE	BLACK	WHITE	MEDIUM_HIGH
NULL	NULL	NULL	NULL	SHORT	BLACK	WHITE	MEDIUM
in other case	HOLD						

Tras la construcción del sistema este es entrenado durante un período de dos años en el índice Nasdaq-100. Los resultados que arroja el entrenamiento son de una ganancia de 6.29% siendo considerado un éxito. En la fase de validación los resultados son del 6.97% de ganancia, aun mejores.

Como conclusiones los autores encuentran que es posible crear un sistema de operativa -de recomendación de operativa- que sea estable basándose en FL, y además el sistema no es una caja negra por cuanto puede ser reconfigurado en sus parámetros fácilmente basándose en experiencia y de esta manera realimentar con más histórico al algoritmo. Este sistema se diferencia de otros en que no recomienda comprar o vender estrictamente sino además lo hace indicando cuando es el mejor momento para operar y qué cantidad es recomendable operar basado en la intensidad de la señal. Esto último lo hace mediante el llamado money management, que no deja de ser un sistema de cálculo de capital disponible para operativa que se calcula dinámicamente en función del capital libre y la máxima pérdida que el operador quiere permitir que el sistema pueda alcanzar.

Una de las ramas más interesantes de trading es aquella en la que las máquinas toman el control del mercado en el llamado trading de alta frecuencia (HFT). Este se caracteriza por una duración breve de la vida de las operaciones, y la suma de las ganancias diarias o

mensuales se elevan hasta un 70% del volumen de operaciones para activos como el dólar, sin intervención humana en la operativa dada su velocidad, tan sólo participa el ser humano en el diseño y reglas. Las operaciones HFT -high frequency trading- necesitan de algoritmos que son ejecutados en tiempo real por las máquinas -servidores con aplicaciones de observación y operación de mercado- que ajusten especialmente bien el riesgo de cada operación. Tal cómo indicaba Zadeh (1975) aquellos sistemas que tienen una complejidad adicional requieren un enfoque significativamente diferente. En su trabajo Vella (2016) destaca cómo esta nueva realidad donde HFT opera a tal velocidad que los humanos no pueden participar de esa operativa hace necesario intercalar lógica difusa tipo dos para ajustar el nivel de riesgo de las operaciones automáticas. El incremento de HFT en mercado hizo que la UE -Unión Europea- regulara desde 2017 este entorno haciendo cambios y sugerencias para brokers y usuarios finales mediante la regulación MiFID2. Esta regula - intenta regular- el nivel de riesgo asociado a algoritmos y trading automático, entre otras cuestiones como seguridad. En su trabajo Vella (2016) precisamente aborda cómo mejorar el nivel de riesgo de HFT mediante la incorporación de ANN -Adaptive Neurofuzzy Networks- con FL tipo-1 combinadamente, es decir haciendo uso de ANFIS -Adaptive NeuroFuzzy Inference Systems-, Jang(1993). En su trabajo Vella (2016) precisamente aborda como la FL de orden superior -tipo 2- mediante sistemas FLS -fuzzy logic systems- es en primer lugar una innovación, y en segundo lugar aumenta el control de riesgo en las operaciones -hasta ahora casi siempre el énfasis en los trabajos, tal como hemos ido viendo, se hacía principalmente en obtener beneficios-. Los autores tienen la intencionalidad de hallar resultados mejorados mediante esta introducción de FL tipo 2, mediante la reducción de incertidumbre de acuerdo con los autores Heal & Millner (2014); Knight (1921); Nelson & Katzenstein (2014).

No obstante, según Wu y Mendel (2014) no hay evidencia matemática de que cambiando de un sistema FL tipo 1 a tipo 2 un controlador FL tipo 2 superará automáticamente a un controlador tipo 1.

Al respecto hay que mencionar que la implementación de ANFIS con el objeto de mejorar su aplicación en sistemas de trading ha sido abundantemente documentado Boyacioglu & Avci,

(2010); Chang, Wei, & Cheng, (2011); Chen, (2013); Gradojevic, (2007); Kablan & Ng, (2011); Tan, Quek, & Cheng, (2011); Vella & Ng, (2014b); Wei, Cheng, & Wu, (2014).

Según Rachenthin y Street (2013) en HFT hay un momento coincidente con el punto de ruptura del precio del activo, durante el cual se puede predecir la evolución del precio durante un minuto, y en menor extensión durante cinco minutos, disipándose su predictibilidad a los treinta minutos. Así Vella (2016) hace uso de temporalidad de dos minutos en su trabajo de investigación, pero alineando el sistema a la prioridad dictada por MiFID2 de controlar el riesgo de manera prioritaria en lugar de concentrar el resultado tan sólo en obtener pingües beneficios, de manera contraria a la tendencia apuntada por los sondeos realizados sobre IA, según apuntan Krollner et al. (2010) y Tsai & Wang (2009).

Es una idea común entre operadores de bolsa la creencia de la existencia de una correlación entre la volatilidad del mercado intradía y las rupturas (tratado por Choudhry, McGroarty, Peng & Wang, 2012; Gradojevic & Gençay, 2013; Holmberg et al., 2013; Schulmeister, 2009; Vella & Ng, 2014a).

Así, tratando la disponibilidad del histórico de datos intradía, Merton (1980) propuso la utilización de estos datos históricos para estimar la volatilidad mediante el cálculo de la suma al cuadrado de los beneficios diarios. Al respecto confirman Andersen y Bollerslev (1997) que la volatilidad calculada desde un sistema HFT intradía de acuerdo con los retornos intradía es efectivamente una medida de la volatilidad libre de error.

Así mismo la posibilidad de medir la intensidad de una señal -evolución del precio- de acuerdo con diferentes indicadores como podrían ser el volumen de negociación en cada minuto, por ejemplo, nos permite en trading seguir la evolución más o menos probable del valor del activo. Al respecto Vanstone & Finnie (2009) indican lo apropiado de comparar la señal actual respecto al período previo y detectar su incremento o decremento.

Entrando en la aplicación práctica de este sistema los autores Vella (2000) emplean el horizonte temporal de un minuto, emitiendo una predicción de la media de ingresos en los dos siguientes minutos y pasando esta predicción al algoritmo para que este emita una recomendación de entrada larga -comprar- o corta – vender- o esperar -mantener-. Para la simulación emplearon una cuenta con 250.000 Libras. El tamaño de la posición estándar

empleada fue de 10 Libras ya sea en compra o en venta. Una vez abierta una posición esta es cerrada de acuerdo con el parámetro TD que indica la duración exacta que permanecerá abierta la posición. Después de ese tiempo la posición será automáticamente cerrada. En caso de recibir una señal de operativa en la misma dirección mientras que ya está abierta una posición entonces los autores siguen la técnica de resetear TD y reiniciar su duración sin cerrar la operación y sin abrir una nueva a pesar de la señal a favor de hacerlo, en lugar de cerrar la posición en su momento previsto. Al respecto hay un trabajo que indica que este método está probado con éxito, Brabazon & O'Neill, (2006).

En la Figura 5 podemos ver de nuevo un algoritmo de lógica, si bien este algoritmo incluye nuevos parámetros hasta ahora no vistos, como la duración de vida de la operación en curso (tradeDuration).

```

Φ ← 0
if signal > RT and signal > prevsignal and balance-
tradesize > 0 then
    Φ ← long
end if
if signal < -1*RT and signal < prevsignal and balance-
tradesize > 0 then
    Φ ← short
end if
OPENTRADE(Φ)
for each open trade t do
    tradeDuration(t) ← tradeDuration(t)+1
    if tradeDirection(t) == Φ then
        tradeDuration(t) ← 0
    end if
    if tradeDuration(t) >= TD then
        CLOSETRADE(t)
    end if
end for

```

Figura 5. Pseudo código del algoritmo de trading.

Los autores hacen una consideración importante a la hora probar el sistema con su algoritmo en *back-testing* -pruebas previas a la puesta en producción comercial para comprobar que el sistema trabaja con corrección- consistente en probar el sistema en activos financieros que muestren tendencias diferentes y así evitar lo mencionado por Pardo (2011) en su aviso de que elegir activos con misma tendencia llevan a definir modelos que no pueden ser generalizables y que sólo funcionan -por lo tanto- en ciertos escenarios. Este proceder es raramente considerado, según tanto Krollner et al. (2010) como Tsai and Wang (2009).

Entre las conclusiones del trabajo Vella (2016) apunta tres: mejora de ANFIS mediante la introducción de conjuntos difusos tipo 2 y así reducir incertidumbre; aportación general que enriquece el conocimiento de trading HFT ya recogido en otros trabajos previos como Holmberg et al. (2013) o Schulmeister (2009).

Con este repaso y evaluación de conceptos de trading hemos tenido la ocasión de familiarizarnos con las pruebas más significativas realizadas para la mejora de la investigación de métodos que ayudan en la toma de decisiones de operativa en mercado en cualquiera de sus direcciones.

Recogiendo toda esta información encontramos una gran cantidad de trabajo muy concienciado y valioso, si bien por razones de concreción no se ha podido profundizar en exceso en los detalles concretos de cada uno de los experimentos realizados tras cada investigación abordada. No obstante, no era, efectivamente, el objetivo de este apartado incidir en los detalles últimos de cada caso sino más bien entender dónde estamos actualmente en tendencias de sistemas de lógica difusa de ayuda a la toma de decisiones de operativa bursátil. Antes de entrar a comentar tendencias es interesante mostrar la figura inferior ya que nos adentra en el concepto de sistema experto difuso, el cual es una forma avanzada, más humana por flexible, de un sistema de aprendizaje realimentado donde la base de conocimiento se realimenta con la experiencia de usuario. Esta forma de sistema experto es un avance en la vía de la lógica difusa, que será tratada más adelante.

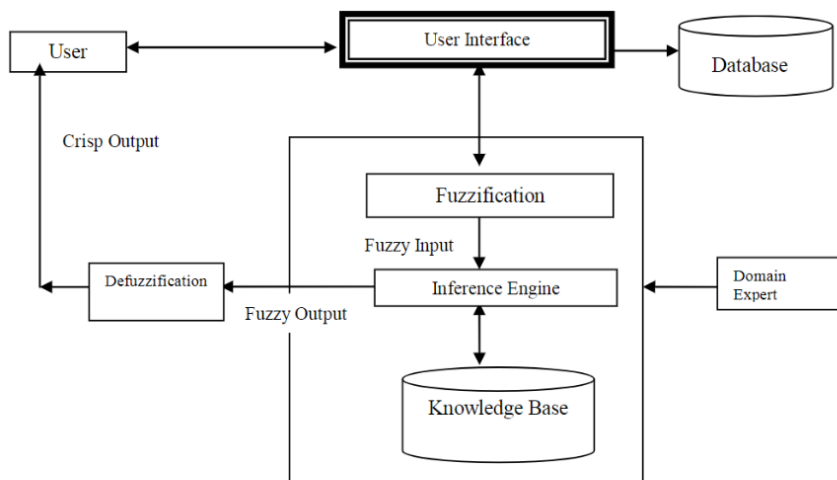


Fig. 1 Stages of Fuzzy Expert System

Figura 6. Etapas de un sistema Experto Fuzzy. (Thakera, S., Nagorib, V. 2018)

2.3 Tendencias

Una primera tendencia claramente identificable es una progresión positiva hacia la menor participación humana en la operativa. Este intento es el que han facilitado los fabricantes de plataformas software de operativa al incluir en los mismos los lenguajes de programación más o menos completos que facilitan al operador avanzado o los consultores de los operadores desarrollar herramienta de análisis tales como nuevos indicadores técnicos o indicadores técnicos difusos, así como sistemas expertos basados en lógica clásica o en lógica difusa. Esto es algo apuntado en su trabajo, Matviychuk (2006).

Una segunda tendencia es la hibridación de sistemas expertos mezclando el análisis fundamental con el análisis técnico, y ambos tratados con lógica difusa. Basarse en indicadores técnicos tiene la ventaja de emplear las herramientas más habitualmente utilizadas, pero además mejoradas con FL para suavizar recomendaciones muy rígidas que pueden con frecuencia llevar a posiciones de pérdidas.

La tercera tendencia que detectamos es un enorme interés por incluir en los sistemas expertos criterios de equilibrio que permitan no enfocar todo el éxito de los expertos difusos en crear solamente beneficios, sino además en alinearse a regulaciones más conservadoras, entre las cuales cabe incluir la gestión proteccionista del capital, como elemento de juicio en la entrada al mercado.

Finalmente, una tendencia que existe claramente en el mercado de valores -y tratada con abundantes trabajos en su conexión con la lógica difusa- es el trading de alta frecuencia, o HFT. Durante este trabajo lo hemos querido mencionar porque hay investigaciones al respecto muy bien documentados y con sistemas diseñados para contrarrestar el incremento del riesgo al estar expuesto constantemente a la volatilidad del mercado. Entre las razones principales que impulsan HFT automático está la bajada de costes para grandes instituciones al prescindir de perfiles humanos caros en operativa, a cambio de invertir en la

introducción de personal experto en campos como inteligencia artificial, ingeniería de software, científicos de datos, matemáticos de *big data*, desarrolladores de algoritmos, etc.

Estas capacidades junto a grandes estructuras de computación y almacenamiento han tornado en muy atractivo para grandes bancos de inversión y *brokers* la reducción de personal financiero a favor de personal analítico del dato y experto en algorítmica, que pueden diseñar e implantar sistemas ganadores en formato temporal 24x5 (el formato 24x7 no existe porque el calendario de mercado libera dos días a la semana de operativa).

2.4 Carencias

Tras la evaluación del estado del arte llama la atención la abundancia de investigación en torno a trading y en torno lógica difusa como método de mejorar las ganancias en trading, si bien también llama la atención de la escasez de investigación en torno a la investigación sobre interfaces facilitadoras de asistencia al usuario de trading, y en torno a la generación automática basado en interfaces de usuario.

En la temática del trading siempre y de manera incontestable se enseña que el mayor factor influyente en el aumento de ganancias y la reducción de pérdidas es la correcta gestión de la psicología humana; muy por delante de tener grandes conocimientos de economía, bolsa, sistemas e indicadores. Estas capacidades de adecuada gestión psicológica de la ambición y de temor a perder ganancias se llaman *psicotrading*.

Es la ansiedad humana por obtener mayores beneficios -y su influencia en la mala gestión de las emociones- la que produce con frecuencia una mala operativa, que termina con cierre de posiciones en pérdidas. Así mismo el miedo a obtener pérdidas también es otro desencadenante de reacciones emocionales -sobre todo en operadores con poca experiencia o formación- que llevan al cierre de operaciones con pérdidas o por no esperar al momento de la entrada en ganancias tras una pequeña corrección del precio o por no agotar la tendencia de ganancias hasta que el sistema nos indique con claridad -mediante velas progresivamente más acertadas y mediante velas doji- que es el momento de cerrar a

posición porque viene muy probablemente un giro de la tendencia del valor del activo (estos giros del valor del activo se tratan más adelante como la teoría de las Ondas de Eliot).

El *psicotrading* es claramente la perspectiva humana emocional en torno a la operativa en bolsa, que no puede ser ignorada ni mejorada por simples conocimientos técnicos. Todos los *brokers* dan formación en *psicotrading* a los clientes que invierten dinero en sus cuentas, para poder operar con más perspectiva y con menos ansiedad y ambición desmedida.

Pensamos que este tema y su conexión con la lógica difusa no ha sido tratado en toda la profundidad que requiere, y presenta un campo de oportunidad enorme para futuras mejoras en sistemas realmente automáticos de trading que soslayan el factor emocional humano.

Una derivada de la evitación del *psicotrading* para que ello no sea óbice para poder operar en bolsa es la hipótesis de este trabajo de investigación: generar y facilitar al *trader* usuario una interfaz de expresión de sus intenciones de inversión, la cual sea convertida en código fuente de un sistema experto de plataforma de trading. Se presenta en este trabajo una primera versión básica y sencilla de esa interfaz, por nombre **AutoTtrader©**.

Fundamentos de Trading

3.1 Qué es trading

3.2 Las Ondas de Elliot

3.3 Criterios de Análisis de mercado

3.3.1 Análisis Fundamental

3.3.2 Análisis Técnico

3.3.2.1 Ejemplo de Indicador Técnico: RSI

3.4 Programabilidad: Plataformas y Lenguajes

3.5 Estrategia Técnica de Trading

3.5.1 Estrategia Técnica Clásica

3.5.2 Fundamentos de lógica difusa

3.5.3 Estrategia Técnica Difusa

Fundamentos de Trading

3.1 Que es trading

Consiste en comerciar con el valor de activos de bolsa. Al igual que el comercio tradicional con cualquier objeto consiste en comprar o vender una cantidad definida de un objeto o mercancía a un precio acordado entre las partes. Actualmente el trading de bolsa se desarrolla sobre ordenadores de manera interactiva mediante plataformas software de negociación donde el usuario -privado o institucional- tiene una cuenta con fondos que es mantenida por un bróker.

En trading los activos no se llegan a tener físicamente, ni se llega a desprender de ellos el trader físicamente. Tan solo se especula con el valor al alza o a la baja del valor o mercancía tratada. Durante el tiempo que la operación está abierta en largo -compra- o en corto -venta- el trader acierta sobre su hipótesis de incremento o decremento del valor del activo y en ese caso gana cierta cantidad de dinero, o bien yerra y pierde la cantidad de dinero que aplique según haya utilizado mecanismos de protección de pérdidas llamados STOP LOSS. A mayor distancia del STOP LOSS desde el valor de compra o venta mayor pérdida, y a menor STOP LOSS menor pérdida. Igualmente, las ganancias pueden ser limitadas basadas en la hipótesis que haga el trader sobre la cantidad de valor que va a incrementar el activo de trading -en el caso de operación de compra- o según la hipótesis que haya hecho sobre la cantidad de valor que va a perder el objeto -en el caso de operación de venta-. La ganancia se hace realidad cuando la operación abierta sobre el valor alcanza el BUY STOP para operación de compra y SELL STOP para operación de venta. El trading trabaja sobre un marco temporal que es seleccionado por el trader en pantalla en tiempo real, y lo puede cambiar en cualquier momento para tener una vista temporal diferente. Para valores muy volátiles -cierta cantidad de divisas FOREX se encuentran en este caso- un marco temporal de 30 minutos o 15 minutos da cierta perspectiva, pero por debajo de esa frecuencia es difícil analizar la tendencia del valor de estas. Para otros activos como los cereales, ciertos metales, ciertas *utilities* el marco temporal diario o semanal da una vista de conjunto suficiente para entender la tendencia del activo.

3.2 Las Ondas de Elliott

Ralph Nelson Elliott publicó su descubrimiento sobre la evolución del precio del mercado para los valores de bolsa en su escrito *The Wave Principle*, bajo el acrónimo EWT (Elliott Wave Theory). Se publicó en el año 1938 y describía como el mercado se comporta conforme a una repetición periódica y predecible de impulsos de ascenso y descenso, alternativamente, del valor de los activos de bolsa. Estos ciclos de impulsos se denominan ondas de Elliott.

Definió cómo la tendencia del precio se proyecta en la tendencia actual -ascenso o descenso- en un movimiento primario de cinco olas, para a continuación proyectar una inversión del precio en la tendencia contraria en otras tres olas.

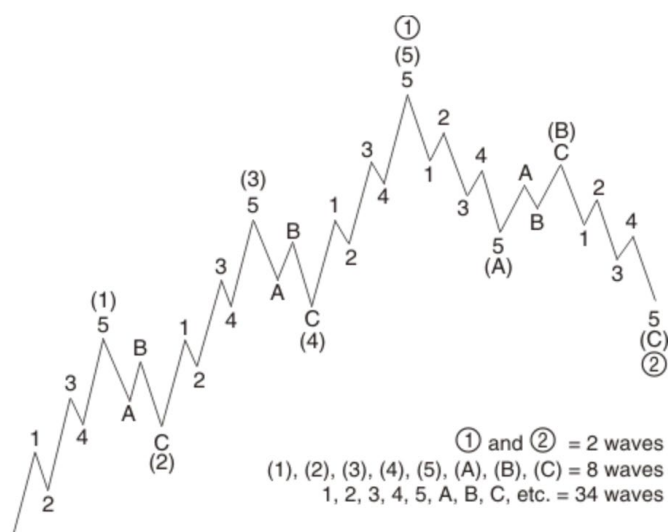


Figura 7. Ondas de Elliott numeradas

En la figura anterior podemos ver la evolución del precio de un valor de bolsa según las ondas de Elliott. Desde el comienzo de la gráfica hasta el punto (1) podemos ver claramente los cinco impulsos iniciales de tendencia de subida del valor -tres impulsos son de ascenso y dos son correctivos y menores-. Si continuamos analizando la gráfica entre los puntos (2) y (3) podemos observar la repetición de los cinco impulsos -tres de subida alternados con dos de bajada, correctivos-. Este

ritmo impulsos se repiten hasta cinco veces, para acabar con un impulso final de baja de tres impulsos -dos de subida alternados y menores. Esta repetición continua del patrón, así como el anidamiento del comportamiento de impulsos según hacemos zoom -es decir damos a la grafica un periodo mayor de análisis- nos permite detectar la fractalidad de los patrones de las ondas de Elliott; *“An important finding is that the theory founder discovered that those moves are fractals. What does it mean? One impulse wave contains another five waves in a smaller scale”* (Jarusek, Eva., Kotyrba, M. 2021).

Extraemos la definición de la dinámica de EWT (Jarusek, Eva., Kotyrba, M. 2021), *“The Elliott Wave Theory is based on the following principles:*

- *Each impulse is followed by a correction reaction*
- *Five waves (1–2–3–4–5) in the direction of the main trend are followed by three corrections (a–b–c) or a 5 : 3 move*
- *Move 5–3 terminates the cycle”*

¿Por qué es importante la teoría de ondas de Elliott? Esta teoría describe de manera sencilla y predecible el comportamiento del mercado para cualquier activo de bolsa, y es un *reflejo de la propia psicología de masas respecto al comercio.*

Podría parecer que al conocer de antemano -gracias a la EWT- la evolución del mercado seria casi trivial obtener ganancias ilimitadas para cualquier aficionado. Nada más lejos de la realidad, de hecho, más de un 70% de las personas que hacen trading pierden su dinero en el intento de ganar más. ¿Por qué es esto así?: la EWT tan solo nos da una explicación sobre el patrón de evolución del mercado, pero no nos dice la amplitud que tendrá cada impulso, lo cual es la clave para poder hacer ganancias entrando y saliendo del mercado en los momentos previos a la finalización de la amplitud del impulso.

Hay trabajos donde se han realizado pruebas estadísticamente significativas sobre la bondad de la predicción del mercado incluyendo la EWT como parte del método. A modo de ejemplo (Jarusek, R., Volna, E. Kotyrba. 2021), los autores estudian la predicción del mercado por varias vías y lanzan operaciones de trading. Lo que encuentran es que cuando incluyen la EWT como mecanismo

predicor la mejora de resultados se eleva desde 320K USD de ganancias hasta 410K USD de ganancias.

La figura inferior es una muestra clara de la influencia que tiene la utilización de fuentes de patrón como indicador de evolución, en este caso el patrón es del ondas de Elliot.

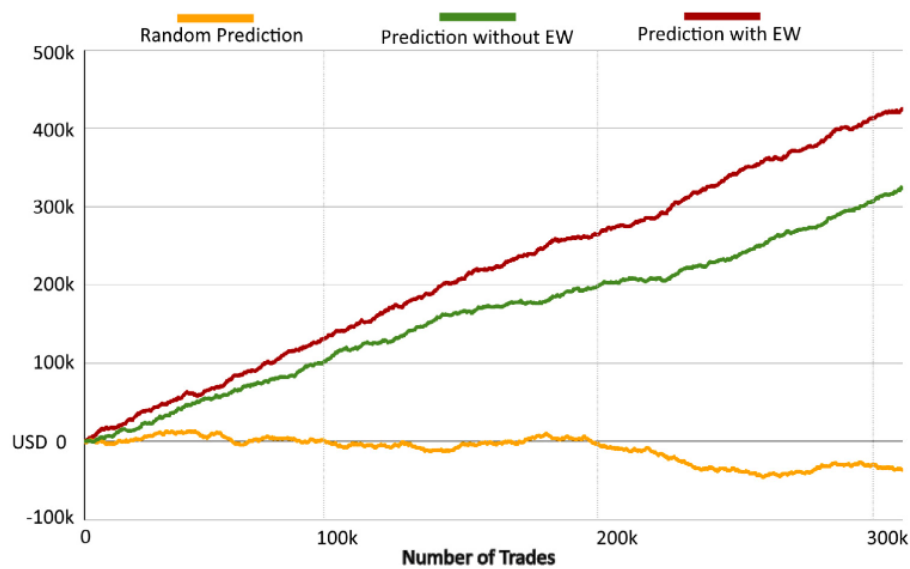


Figura 8. Resultados de trading con y sin EWT como mecanismo predictor de mercado

De esta manera, y sin tratar de considerar EWT como un mecanismo único e infalible sino más bien un patrón fiable- podemos afirmar que -someramente- ya sabemos cómo funciona el mercado para cualquier valor de bolsa, mediante los patrones de Elliott.

3.3 Criterios de Análisis de Mercado

La evolución de los precios de los activos bursátiles responde a múltiples estímulos, cuya reacción es el valor que marca en cada momento la bolsa.

Se dice que el valor que marca la bolsa en cada momento ya integra la totalidad de estímulos y cualesquiera factores influyentes en el valor del activo, es decir está descontado cualquier hecho que pueda influir en ese valor mostrado.

A la hora de tratar de predecir el valor al que evolucionará un valor de mercado se puede enfocar el análisis de dos formas, una es analizando el contexto financiero del valor bursátil en cuestión - sea una divisa, sea una empresa, sea cualquiera activo...- a través de, por ejemplo -no solamente- noticias financieras relacionadas; este sería un análisis fundamental. Por otra parte, existe el análisis llamado técnico donde la predicción de la evolución del precio se hace utilizando *graficacion* del valor pasado y herramientas desarrolladas para proyectar al futuro los valores pasados.

3.3.1 Análisis Fundamental

Como se ha mencionado anteriormente en este análisis del valor de un activo de observa y se integra cualquier noticia financiera, local o global, política, y de cualquier otro origen que pudiera tener influencia en la evolución del precio del activo. Un ejemplo sencillo de entender sería el caso de la evolución del valor del cereal mediante el análisis de la evolución del símbolo WHEAT que hace referencia el índice mundial del precio de este. Si tomamos este índice podemos ver que su precio tenía un ascenso progresivo muy pequeño o incluso se estancaba en torno al valor de 140 USD, pero desde abril de 2022 se dispara en bolsa el precio del WHEAT. ¿A qué se debe esto según el análisis fundamental?: la causa tiene su origen en la guerra de Ucrania tras la invasión de Rusia, porque en el mercado financiero donde se valora este activo hay miedo a que la guerra disminuya el cereal disponible que se origina en Ucrania y que por lo tanto haya escasez mundial -Ucrania es uno de los grandes graneros globales-. La escasez -incluso percibida, aunque no sea real- de un activo produce su incremento en precio -oferta/demanda-. Si miramos la gráfica de evolución de la figura 9 podemos ver claramente que el precio hasta febrero de 2022 se mantiene estable en torno a los 145 USD y ya en julio sufre una subida muy relevante debido a la inevitable guerra. Con este ejemplo se quiere ilustrar que tan solo se ha considerado el entorno político y macroeconómico para hacer un análisis rápido de la evolución del precio del índice del cereal; esto es análisis fundamental.

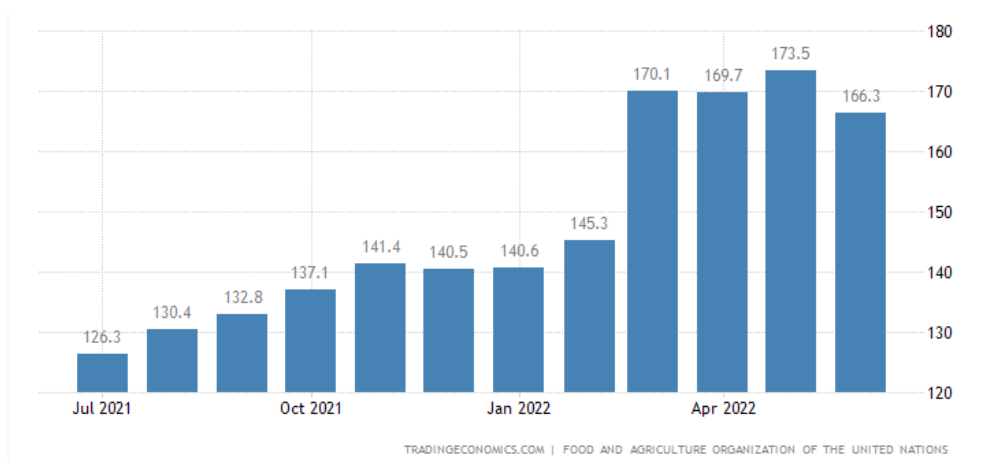


Figura 9. The FAO Cereal Price Index evolution.

A continuación se recoge un texto de **Trading Economics.com** que explica el análisis al respecto del valor: *“The Cereals Price Index is compiled using the International Grains Council (IGC) Wheat Price Index of 10 quotations, 1 US maize quotation and 16 rice quotations. The rice quotations are combined into groups consisting of higher quality Indica, lower quality Indica, Japonica and Aromatic rice. Within each group, a simple average of the relative prices of appropriate quotations is calculated; then the average relative prices of each of the three varieties are combined by weighting them with their trade shares. Indices for wheat, coarse grains and rice are combined using export value weights of the base period 2002–2004”.*

World Cereals Price Index								
The Cereals Price Index is compiled using the International Grains Council (IGC) Wheat Price Index of 10 quotations, 1 US maize quotation and 16 rice quotations. The rice quotations are combined into groups consisting of higher quality Indica, lower quality Indica, Japonica and Aromatic rice. Within each group, a simple average of the relative prices of appropriate quotations is calculated; then the average relative prices of each of the three varieties are combined by weighting them with their trade shares. Indices for wheat, coarse grains and rice are combined using export value weights of the base period 2002–2004.								
Actual	Previous	Highest	Lowest	Dates	Unit	Frequency		
166.30	173.50	173.50	48.60	1990 - 2022	Index Points	Monthly	2014-2016=100	

Figura 10. World Cereals Price Index (Trading Economics.com)

El origen del análisis fundamental está datado en 1934 y se atribuye a la contribución de los analistas financieros Dodd y Graham.

3.3.2 Análisis Técnico

Con este análisis la metodología empleada consiste fundamentalmente en graficar el histórico del precio pasado en la pantalla mediante líneas de tendencia y mediante las líneas de cortes de precio al alza -resistencias- y líneas de cortes de precio a la baja -soportes-. Las líneas de tendencias te indican en la dirección que fluye el precio del activo hasta que se termina el impulso actual -impulso de las ondas de Elliott-. Donde termina un impulso de subida suele coincidir que se encuentra una

línea de resistencia que no deja pasar el precio hacia arriba y se produce un rebote en el sentido contrario, iniciando así una corrección del precio a la baja hasta que se encuentra una línea de soporte, donde el precio no podrá bajar más.

A este somero cuadro hay que añadir la fractalidad, es decir la ocurrencia repetitiva a escala de este mismo patrón de impulsos de subida y bajada.

Adicionalmente en análisis técnico es común la búsqueda y detección de patrones de formas definidas -forma en W, forma en M, forma en V, etc.-. Cuando se detecta una forma patrón es muy frecuente que el comportamiento del precio a futuro sea el mismo que fue la última vez que apareció esa misma forma patrón anteriormente, con lo que se puede establecer una hipótesis de precio a futuro. Estos mecanismos mencionados pertenecen al análisis técnico basado en estadística histórica y evolución del precio. Además, es posible también utilizar alternativamente o incluso de manera complementaria otro mecanismo adicional llamado indicadores técnicos, los cuales son algoritmos pre-procesados tras años de estudio por parte de sus diseñadores, y que basados en diferentes observaciones y fórmulas nos permiten ver en pantalla una versión gráfica sencilla y legible visualmente de la evolución que tuvo el precio en el pasado y mediante números o códigos de colores u oscilaciones facilitan mucho la generación de hipótesis respecto al futuro del valor del activo.

Los indicadores técnicos más empleados son MACD, RSI, SMA, ADX, si bien hay una gran cantidad de indicadores disponibles en la mayoría de las plataformas, y adicionalmente el *trader* puede desarrollar sus propios indicadores basados en experiencia personal.

Dado que este trabajo se desarrolla en torno a estrategias de inversión basadas en análisis técnico a continuación se desarrolla uno de estos indicadores para poder entender cómo trabaja un *trader* con un indicador.

3.3.2.1 Ejemplo de Indicador Técnico: RSI

Se explica a continuación un indicador técnico muy popular, que permita al lector entender mejor la utilidad y la mecánica de uso de los indicadores técnicos. Se ha elegido el RSI por ser muy sencillo de leer y por ser muy popular. Este indicador es el que será utilizado en capítulos sucesivos para demostrar los objetivos de este trabajo al crear y probar estrategias de inversión rígidas y asistidas por lógica difusa.

Este indicador -extensamente empleado por *traders* por su alta fiabilidad- es un indicador del tipo llamado osciladores, y de manera continua va dando una lectura gráfica de valor en un rango entre 0 y 100 , respondiendo el valor 0 a un instante en el que hay una sobreventa del activo -posiblemente indicando fin de un impulso de bajada- , y respondiendo el valor 100 a un instante de sobrecompra -los *traders* activos en el mercado han abierto muchas operaciones de compra- por lo que es esperable que haya finalizado un impulso de subida -recordamos a qui los impulsos al alza y baja de las ondas de Elliott- y que por lo tanto el valor gire -oscile- hacia una tendencia de bajada del precio. Cuando hay una situación de sobreventa es esperable abrir una operación de trading de compra -posición larga-, e inversamente la operación más lógica cuando hay una situación de sobrecompra es de venta -posición corta-.

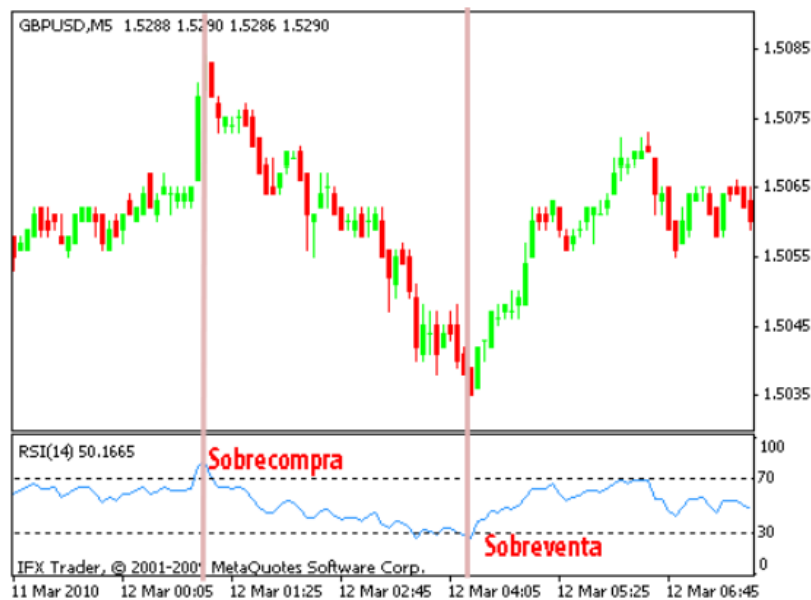


Figura 10. Ejemplo de zonas de sobrecompra y sobreventa proporcionadas por el indicador RSI.
(Forex, 2017)

Figura 11. Indicador RSI gráfico con intervalo 0 (sobreventa) a 100 (sobrecompra). (Naranjo, R. 2018)

La activación del indicador se hace de manera muy sencilla, se arrastra el indicador desde el menú de indicadores de la plataforma de trading hasta la gráfica del activo que está siendo analizado, en ese momento el RSI comenzará a dibujar su línea de evolución entre 0 y 100 según el valor del activo se va construyendo -vela roja o vela verde-.

Dentro de la pantalla del oscilador -zona inferior de la pantalla- se puede editar el valor de los límites, como por ejemplo 30 y 70 como vemos en la figura 11. Esos límites tan sólo son visuales e informativos, para que el trader tenga referencias visuales de la amplitud de oscilación del indicador; con esta información es con lo que el *trader* recoge información para componer el estado propicio o no para abrir una operación de compra o de venta.

Los indicadores deben ser leídos para ser interpretados. Además de la lectura de indicador de sobrecompra o sobreventa se puede hacer una interpretación llamada de divergencias del precio. Al respecto indica (Naranjo, 2018) *“Otra forma de interpretar el indicador RSI, como en otros tantos osciladores, es analizar las divergencias Precio/RSI, tanto positivas como negativas. Entre el RSI y el precio se pueden generar señales de compra y venta. - Divergencia alcista: ocurre cuando el precio hace un mínimo menor al mínimo anterior mientras que el RSI hace un mínimo mayor al mínimo anterior. La divergencia alcista es una señal de entrada en largo (compra). - Divergencia bajista: se da cuando en el precio se observa un máximo mayor al máximo anterior y en el RSI se observa un máximo menor al máximo anterior. La divergencia bajista es una señal de venta”*

Siempre que la plataforma permita programación -algunas plataformas incluyen un lenguaje de programación para automatizar el análisis e incluso el trading de operaciones- los valores del indicador -de cualquier indicador- pueden ser leídos y manipulados por el propio lenguaje de programación para tomar acciones (generar sonidos de alarma, generar señales visuales en pantalla, enviar un email de aviso, abrir una operación de compra o de venta, etc).

Como se puede concluir la potencia de los indicadores es máxima, y saber interpretarlos, y especialmente saber programar su información en la plataforma puede dar lugar a una automatización máxima con grandes rendimientos dinerarios sin requerir la presencia del *trader* frente a la pantalla.

Dos ideas muy relevantes para mencionar en relación con el análisis técnico:

- En análisis técnico se asume que el valor actual del activo ya integra todos los sucesos del mercado en torno a este
- El valor pasado de un activo no implica necesariamente el valor a futuro de este, aunque los valores tienen una tendencia repetitiva secuencial. Es por esto por lo que el análisis técnico tan solo consiste en un contexto de probabilidades y no puede de manera alguna anticipar un giro del precio en la dirección inesperada por causas incluso desconocidas

“Debido a la imposibilidad de obtener toda la información, ya que los mercados financieros se han ido convirtiendo, cada día más, en sistemas complejos influenciados por muchos factores económicos, políticos y psicológicos interrelacionados, y están caracterizados por no linealidades inherentes, fue surgiendo el análisis técnico, asentado en las teorías desarrollada por Charles H. Dow (Rhea, 1938) y Stansbury (1938). Los analistas técnicos se basan en no creer necesaria la información proporcionada por el análisis fundamental, ya que dicha información siempre queda reflejada en la evolución del precio de los mercados, y por tanto, es el propio mercado el objeto de sus análisis” (Naranjo, 2018).



Figura 12. Gráfica del activo de divisas EUR/USD

En la figura 10 se puede observar la pantalla de trabajo de trading que emplea cualquier empresa o particular que accede al mercado de capitales para observar o para operar compras y ventas de activos.

Descripción somera de la pantalla de la figura 10: la plataforma software de trading de la figura es ActivTrader de la casa de trading Activtrades. Esta es una plataforma web -solo web- de trading, escrita a medida para esta casa, si bien el cliente de la casa ActivTrades puede utilizar otras plataformas tal como Metatrader4 o Metatrader5, y muchas más. Lo que se ve en la captura es que cada barra recoge cinco minutos de actividad de mercado. Si la barra -conocida como vela japonesa- es de color verde entonces el precio subió en esos cinco minutos, y si la barra es de color rojo el precio bajó en esos minutos. El tamaño y el color de las velas japonesas nos indican la evolución del precio del activo para el horizonte temporal observado en pantalla; siendo este un parámetro configurable, así como los colores de las velas, la duración en minutos de cada vela, y muchas otras parametrizaciones que dan al *trader* la capacidad de adaptar la visualización de pantalla -de mercado- a su gusto. Las barras a pie de gráfica bajo cada vela japonesa indica el volumen de acciones -o contratos- que han sido negociados en e periodo temporal de esa vela japonesa.

3.4 Programabilidad: Plataformas y Lenguajes

Tal como se menciona anteriormente la plataforma software de observación de mercado existe en muchas marcas y desarrollos. A efectos de este trabajo se empleará la plataforma Metatrader4 ya que es una de las más comúnmente aceptadas en el mercado de trading, y tiene capacidades enormemente útiles para conocedores de desarrollo software.

Metatrader4 es una plataforma programable de trading, que lleva embebido un lenguaje de programación llamado MQL4 que permite realizar todas las acciones que realiza el *trader* de manera interactiva, pero codificando línea a línea esas acciones en un programa ejecutable desde dentro de la plataforma.

En el mercado de capitales existen cientos de activos con los que se puede negociar en bolsa, entre ellos las divisas. El cambio de divisas y su tendencia al alza o a la baja es uno de los activos bursátiles más populares que existe hoy en día. Para este activo -divisas- Metatrader es la plataforma comercial más popular que existe actualmente.

La plataforma base incluye el software de observación y trading de mercado. La mayoría de las entidades financieras de trading que facilitan la apertura de una cuenta de trading suelen ofrecer a sus clientes la plataforma Metatrader, si bien algunas no lo hacen. Esta plataforma facilita al *trader* todas las funcionalidades necesarias para observar y analizar el mercado para los activos que haya contratado en su cuenta de trading con su proveedor de acceso al mercado. Por ejemplo, un *trader* puede contratar en su cuenta de trading tan solo FOREX -divisas- y en ese caso podrá utilizar Metatrader -asumiendo que su proveedor de acceso al mercado bursátil lo ofrece como plataforma comercial- para ejecutar operaciones de compra y venta de lotes -o sus divisores más pequeños minilotes y microlotes- de los pares de divisas que desee. Un lote equivale a 100.000€ de valor, por lo que los *trader* más modestos normalmente operan con microlotes -equivalente a 10.000€, a veces con minilotes -equivalente a 1.000€-.

En este trabajo los ejemplos de estrategias, la programabilidad de plataforma, y los indicadores técnicos se demostrarán sobre plataforma Metatrader. Como activo financiero a lo largo de la parte más práctica de este trabajo se ha empleado FOREX.

Una opción que emplean *traders* que no quieren crear y monitorizar su propia estrategia es la llamada copia de *señales comerciales* donde el *trader* puede suscribirse a la cuenta de los muchos *traders* que comparte su estrategia de operativa en bolsa a cambio de un coste fijo o variable o fijo+variable. Esta es una buena opción para personas interesadas en hacer crecer sus ahorros, pero no quieren profundizar en la técnica y dedicación que requiere invertir en bolsa. La elección de la señal comercial copiada debe estar muy bien fundamentada para no tener pérdidas. Todos los *trader* que permiten copiar sus señales de trading tienen que publicar en Metatrader el histórico de ganancias y pérdidas que tienen en el tiempo, para que los posibles clientes sepan a qué atenerse si ponen su dinero a trabajar en mercado, con los riesgos que ello pueda implicar de pérdidas. Así mismo cualquier *trader* que considere que tiene una estrategia que genera ganancias consistentemente en el tiempo puede publicar su señal para que otros se suscriban a ella -siempre que cumpla con las condiciones mínimas exigidas por la entidad-.



Figura 13. Plataforma comercial Metatrader 4. Vista general.

Los proveedores de cuentas comerciales asumen que es más fácil perder dinero que ganarlo, haciendo trading. De hecho, es obligación legal de los proveedores de cuentas comerciales avisar a nuevos clientes que más del 70% de los clientes pierden dinero haciendo trading.

Por ello el nivel de preparación de los *traders* debe ser largo y muy exhaustivo antes de salir en real al mercado. Para ello existen las llamadas cuentas demo que son las cuentas que los proveedores de cuentas facilitan a sus clientes para probar estrategias nuevas, pero con dinero simulado, no con dinero real. En la figura 11 se puede ver la plataforma Metatrader4 mostrando cuatro activos FOREX -GBPUSD, EURGBP, USDJPY, XAUUSD- en una cuenta demo de cliente. Una vez que el *trader* tiene fuertemente afianzada su confianza sobre la estrategia de inversión generada es cuando la cuenta con dinero real se empieza a utilizar. El tiempo de preparación de un *traders* raramente será inferior a un año como mínimo antes de poder comenzar a utilizar un microlote de dinero real. En este tiempo aprenderá la teoría de mercado, la teoría del activo que quiere comercializar, la tecnología de la plataforma elegida, y la construcción de una primera estrategia sencilla que ejecutar.

Una funcionalidad ineludible de una plataforma de trading programable es el *backtesting*. Esta funcionalidad permite probar una estrategia programada, con efecto retroactivo al mercado que real de los días o meses o años pasados, para poder comprobar qué resultados habría dado esa estrategia si hubiera sido aplicada en ese intervalo temporal.

Antes de salir a mercado con una cuenta que contiene dinero real es muy recomendable realizar *backtesting* hasta que el resultado de la ejecución de la estrategia arroje ganancias consistentemente. Cualquier otro enfoque del mercado será muy probablemente conducente a pérdidas continuas de dinero.

“El simulador de estrategias comerciales MetaTrader 4 ha sido diseñado para testar y optimizar robots comerciales antes de comenzar a usarlos en el comercio real. El funcionamiento del simulador se basa en los datos históricos de las cotizaciones de las divisas. Durante el proceso de simulación, el robot comercial analiza las cotizaciones acumuladas, realizando operaciones comerciales virtuales de acuerdo con el algoritmo comercial implementado en él. Esto permite valorar cómo ha comerciado dicho asesor en el pasado, permitiendo así modelar su comportamiento en el trading real.

La [Función de optimización](#) incorporada en el simulador permite seleccionar los parámetros óptimos del programa comercial, obteniendo de esta forma el mejor resultado en el trading. Por ejemplo, es posible configurar los parámetros del robot comercial para obtener el máximo beneficio, para minimizar los riesgos, etcétera” (www.metatrader4.com).

La programabilidad de Metatrader4 es un activo fundamental de esta plataforma ya que permite automatizar el trading bajo los parámetros y condiciones de mercado que el *trader* programador haya diseñado. Obviamente no todo *trader* ha de ser un buen programador, es por ello por lo que hay una oferta abundante de desarrolladores de software en lenguaje MQL4 -lenguaje de la plataforma Metatrader4- que adaptan las estrategias que les explican los traders a un programa que automatiza esa estrategia.

El trading algorítmico (trading automático) es uno de los puntos fuertes de MetaTrader 4, que permite crear, poner a prueba y usar asesores expertos e indicadores técnicos de forma autónoma. Con su ayuda se elimina cualquier límite en las capacidades analíticas y comerciales de la plataforma. Directamente en la plataforma se ha incorporado el propio entorno de desarrollo MQL4

IDE (Integrated Development Environment), que permite crear expertos comerciales (robots comerciales, Expert Advisors) e indicadores técnicos de prácticamente cualquier complejidad. Su núcleo es MQL4, un lenguaje de desarrollo de estrategias comerciales orientado a objetos que se distingue por su alta productividad, flexibilidad y funcionalidad” (www.metatrader4.com).

El lenguaje de programación embebido en Metatrader4 es el lenguaje MQL4, muy similar a C++, orientado a objetos, y sobre todo con gran capacidad de manejar mucho cálculo y analizar largas series de datos almacenadas. El análisis de series temporales es una parte esencial del trading técnico ya que toda la proyección a futuro del valor del activo se realiza en base al análisis de los valores del pasado.

Los diferentes desarrollos soportados en MQL4 se dividen como sigue:

- **Asesor Experto:** en inglés Expert Advisor o EA, es un programa MQL4 que dota al *trader* de total autonomía, y no necesita realizar análisis de mercado ni seguimiento de indicadores ni del valor del activo, ya que el programa codificado realiza todas esas tareas; incluida la compra o la venta del activo en cuestión y su seguimiento hasta el cierre de la operación. Evidentemente este es el nivel superior de trading algorítmico automatizado, por lo que su desarrollo no es algo trivial y tiene muchas implicaciones técnicas y de consecuencias de pérdidas dinerarias si el EA trabaja incorrectamente en sus análisis de mercado. En este punto es bueno recordar el tema tratado en capítulos anteriores respecto a la ausencia de plasticidad de pensamiento en los programas de trading; estos no reaccionan frente a condiciones de mercado que no hayan sido programadas explícitamente. Es relevante en este punto mencionar que el EA puede no ser un elemento de riesgo si es programado en ejecución manual o semiautomática, es decir, el *trader* puede delegar el análisis de mercado al EA y cuando este encuentra condiciones favorables de compra o venta, entonces activa con alarmas visuales o sonoras o escritas por email al *trader* para que este entre en la plataforma y verifique personalmente si -efectivamente- las condiciones de mercado sugieren una entrada a mercado tomando una posición en largo o corto, y si así fuera este procederá a la compra o venta de manera manual.

- **Indicador Técnico a medida:** tal como se explicó anteriormente los indicadores son algoritmos que calculan la evolución del precio del activo y la dibujan en pantalla de manera muy visual. Adicionalmente a la librería de indicadores técnicos de la plataforma, también se pueden diseñar y escribir indicadores nuevos a la medida del *trader*. El riesgo de errores de desarrollo del indicador no es tan elevado como en el caso del EA ya que no incluye la ejecución de órdenes de compra o venta de activos sino únicamente el análisis del mercado y su graficación en pantalla según el algoritmo desarrollado.

También en este punto hay que hacer una llamada de atención: un indicador técnico mal escrito puede inducir a tomar decisiones equivocadas, por lo que el resultado que muestra el indicador en pantalla debe ser testado intensivamente -en una cuenta demo- antes de emplearlo como indicador técnico comercial con dinero real.

- **“Script”:** el script es la forma más básica de desarrollo MQL4 y consiste en una serie secuencial de instrucciones que son ejecutadas una vez de principio a fin. Puede incluir tareas básicas como limpiar la pantalla de indicadores técnicos, así como imprimir la fecha y hora en pantalla, o enviar un email al *trader* con alguna información relevante.
-

3.5 Estrategia Técnica de Trading

A continuación, se desarrolla el concepto y parámetros de una estrategia de inversión basada en análisis técnico. Este trabajo – a modo de recordatorio- no se centra en estrategias de inversión fundamentales sino en cómo generar el código fuente -en lenguaje MQL4, plataforma Metatrader4- que responde a una estrategia de inversión obtenida tras su formulación con algún indicador -o conjunto de indicadores- técnico, formulada por el trader.

Con el objeto de hacer más directo y más sencilla la comprensión de este trabajo, se desarrolla una estrategia con un solo indicador técnico para no centrar la atención del lector tanto en el mundo del trading como en el objeto fundamental que es la automatización del trading mediante programación software.

La formulación de la estrategia tiene varios parámetros que deben ser definidos según el tipo de trading que se elige realizar. Estos parámetros serán parte de este apartado en sus diferentes subapartados.

3.5.1 Estrategia Técnica Clásica

Como se ha comentado anteriormente los indicadores técnicos de mercado son fórmulas que dan luz a algoritmos que calculan el potencial siguiente valor del precio al que va a evolucionar el actual precio de un activo, basándose en uno o más parámetros del valor pasado de ese mismo activo.

De los múltiples indicadores de mercado se ha elegido -por su popularidad- el indicador RSI como indicador para la construcción y ejecución de la estrategia modelo que se enuncia, desarrolla y automatiza en este trabajo. El RSI visto desde la plataforma de programación MQL4 tiene el siguiente formato sintáctico:

```
double iRSI(  
    string      symbol,      // symbol  
    int         timeframe,   // timeframe  
    int         period,      // period  
    int         applied_price, // applied price  
    int         shift        // shift
```

Los parámetros de trabajo de RSI -definido como variable predefinida en la plataforma con nombre iRSI- se definen a continuación:

symbol nombre del activo de mercado para el que se mide el RSI

timeframe horizonte temporal de la medición del RSI

period media del periodo de cálculo del valor

applied_price este valor puede ser elegido de entre los preexistentes.

Puede ser el valor al cierre de la vela, a la apertura, el más bajo de la vela, el más alto, la media de los valores de la vela, valor típico, valor ponderado):

El cálculo del valor típico es el resultado de la fórmula $(ALTO + BAJO + CIERRE) / 3$, mientras que el cálculo del valor ponderado responde a la fórmula $(ALTO + BAJO + CIERRE + CIERRE) / 4$.

Si se presta un poco de atención a la fórmula empleada es sencillo darse cuenta de que se puede elegir un tipo de precio al indicador RSI que enfatice más un valor que otro de la fórmula, como es el caso del valor de CIERRE en el cálculo del tipo de valor ponderado, donde el valor al que cerró la vela en el horizonte temporal elegido se computa dos veces en la fórmula.

Shift este parámetro indica cuantos pasos de horizonte temporal en el pasado queremos medir para esta iteración del cálculo. Así por ejemplo el valor -1 indicaría que la fórmula del valor RSI se calcula para la vela anterior a la vela en curso de mercado, es decir al último segmento de apertura a cierre de mercado en el horizonte temporal con el que está trabajando el trader.

Empleando lenguaje MQL4 se podría calcular un RSI con la siguiente sentencia:

```
_indicador_RSI = iRSI (EURUSD, 0, 14, PRICE_CLOSE, -2)
```

El resultado sería depositar sobre la variable previamente definida `_indicador_RSI` el valor que se obtiene de la lectura de la vela anterior a la última cerrada -parámetro -2 – según el valor al que cerró esa vela -parámetro `PRICE_CLOSE`- para el activo de mercado de divisas EURUSD -precio de cambio del euro al dólar americano-.

Veamos otro ejemplo:

```
_indicador_RSI = iRSI (USDJPY, 30, 21, PRICE_WEIGHTED, -1)
```

El resultado sería depositar sobre la variable previamente definida `_indicador_RSI` el valor que se obtiene de la lectura de la vela última cerrada -parámetro -1 – según el valor ponderado al que cerró esa vela -parámetro `PRICE_WEIGHTED`- para el activo de mercado de divisas USDJPY -precio de cambio del dólar americano al yen japonés-.

Como se puede ver es sencillo invocar mediciones del valor del activo del precio del activo para un indicador técnico cualquiera. Con el valor obtenido se puede manipular mediante sentencias condicionales la acción deseada: ninguna, comprar el activo, vender el activo, etc.

Cuando el valor RSI es más alto que 70 se entiende que el activo se ha comprado en exceso y seguramente -siempre es sólo una hipótesis y nunca hay garantía- su valor bajará, por lo que una sentencia condicional para vender si el valor alcanzara 70 sería como sigue:

```
if (iRSI(NULL,0,14,PRICE_CLOSE,0)>70) ;
```

```
<accion de venta...>
```

```
<accion 2 de venta ...>
```

Inversamente cuando el valor del RSI baja de 30 se sobreentiende -sin garantías, solo una hipótesis- que un activo ha sido sobrevendido en el mercado y por lo tanto poco probable que su valor siga bajando, por lo que la acción pertinente sería probablemente comprar ese activo. Para ello la sentencia condicional aplicable sería:

```
if (iRSI(NULL,0,14,PRICE_CLOSE,0)<30) ;
```

```
<accion de compra...>
```

```
<accion 2 de compra ...>
```

¿Como quedaría representada la función gráfica de pertenencia de los valores del indicador RSI en matemática clásica? . En el siguiente diagrama se representa la rigidez de la pertenencia, 0 o 1, de los valores del RSI al conjunto de valores que pertenecen al conjunto llamado valor SOBREVENDIDO, y al conjunto SOBRECOPRADO. Es muy evidente la rigidez de la matemática clásica al evaluar el estado del indicador de cara a realizar una hipotética inversión de compra (valor SOBRECOPRADO) o de venta (valor SOBREVENDIDO).

EVOLUCIÓN DEL VALOR DEL INDICADOR RSI EN LOS CONJUNTOS DE MATEMÀTICA CLÀSICA

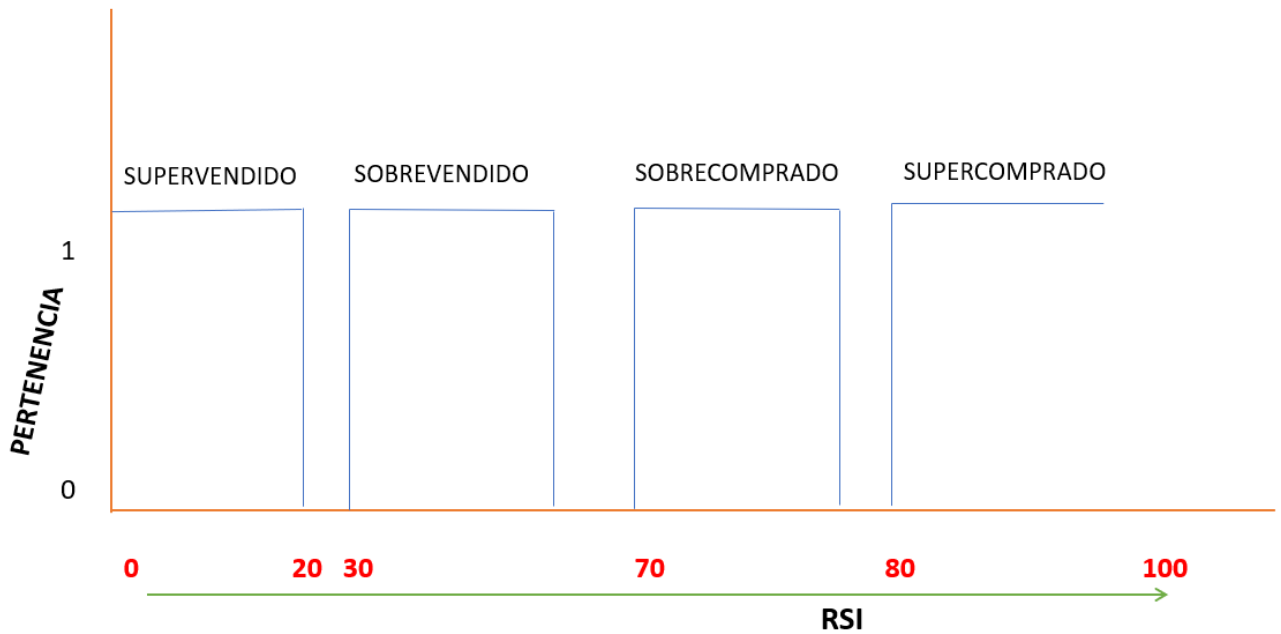


Figura 14. Función de pertenencia del valor RSI en matemática clásica

Seguendo el diagrama anterior la estrategia quedaría definida y enunciada de la siguiente manera: *tomar una posición larga de mercado (compra) ante una indicación de sobreventa según el indicador RSI e valor 30, y tomar solamente una posición (no comprar más de una vez) , y no considerar posiciones cortas (ventas del activo).*

La sentencia condicional de la estrategia sería la siguiente:

IF Activo = NOT.Comprado

 IF RSI ≤ 30 THEN COMPRAR

 END

END

La aproximación de este algoritmo responde a una entrada de lógica matemática más tradicional y rígida, tal que siempre que el valor de RSI sea igual o inferior a 30 se producirá una compra del activo. Este método de afrontar las inversiones es el más habitual y por estático o rígido que parezca es el que conoce y aplica la gran mayoría de traders; si bien el valor decisivo no tiene por qué ser exactamente 30, y podría ser cualquier valor inferior (normalmente nunca sería un valor mayor porque el mercado no suele estar sobrevendido por encima de 30).

A continuación, se ablanda la estrategia mencionada aplicando lógica difusa, para que el proceso de decisión de compra de activo sea la consecuencia de una evaluación más flexible en el recorrido del valor del indicador RSI según la teoría de conjuntos difusos.

3.5.2 Fundamentos de lógica difusa

En este apartado se aporta una introducción a la lógica difusa por una razón de evolución: un claro campo de desarrollo a futuro para trading automático es el trading difuso, donde las estrategias definidas por el trader se basan en definiciones muy flexibles del estado de los indicadores en su recorrido a lo largo de las sesiones de mercado.

El doctor Lofti Zadeh, padre de la teoría de conjuntos borrosos (*Fuzzy Sets*), ideó una matemática de lógica más flexible que la lógica de la matemática clásica, donde los elementos de un conjunto no tienen por qué pertenecer a ese conjunto al cien por cien de pertenencia sino en cualquier grado en el intervalo $>0\%$ y $\leq 100\%$.

Aunque de origen soviético, se mudó a EE. UU. por la fuerte atracción que dejaron en él los misioneros que conoció en Teherán en época temprana. Dice Zadeh en su entrevista con Betty Blair *“Para mí, representaban -los misioneros de EE. UU.- lo mejor que se podía encontrar en los Estados Unidos: gente del Medio Oeste con fuertes raíces. Eran realmente 'buenos samaritanos' dispuestos a entregarse por el beneficio de los demás. Así que este tipo de actitud me influyó profundamente. También me inculcaron un profundo deseo de vivir en los Estados Unidos”*

Ya en EE. UU. cursó un máster en ingeniería eléctrica en el MIT, doctorándose posteriormente en Columbia, y llegando a catedrático en esa misma universidad.

A partir de su posterior estancia en la universidad de Berkeley publica su trabajo sobre conjuntos difusos, en 1965.

En su informe para el *NASA Contractor Report* titulado *“Towards a theory of fuzzy systems”* (1969) el *abstract* es muy indicativo de su visión estratégica de la nueva teoría de conjuntos: *“The concept of a fuzzy set, that is, a class which admits of intermediate grades of membership in it, opens the possibility of analyzing such systems both qualitatively and quantitatively by allowing the input and/or the output and/or the state of the system to range over fuzzy”*.

Como se menciona en el capítulo anterior, aunque pueda parecer inverosímil las horas del día pueden -bajo la teoría de conjuntos difusos- ser elementos del conjunto “Horas de la noche”, ciertamente con un grado de pertenencia progresivamente menor según avanzan las agujas del reloj. Como menciona en el *abstract* de su trabajo Zadeh, la clave de la membresía es *“intermediate grades of membership in it”*, es decir entre 0 y 1 se establece el rango de valores de pertenencia al conjunto que incluye toda la gama granular de números reales contenidos en ese intervalo, para cada elemento del mismo, en función de su identificación con la naturaleza del conjunto. Un elemento que se identifica muy poco con el conjunto tendrá un grado de membresía próximo a 0, mientras que el caso contrario incrementaría hasta 1 su membresía en el conjunto. Esto aporta un matiz riquísimo de intervalos de valores no conocidos hasta entonces en la teoría clásica de conjuntos matemáticos donde las opciones para cualquier elemento son pertenecer o no pertenecer al conjunto.

Esta teoría de conjuntos difusos se convierte así en un buen apoyo de modelado de sistemas no lineales.

En su trabajo Zadeh (1969) define las propiedades elementales de los conjuntos difusos, donde define como un conjunto difuso A en el espacio X como un par de valores que indican el valor del elemento en el conjunto, y su grado de pertenencia a ese conjunto, tal como sigue,

$$A = \{(x, \mu_A(x))\}, \quad x \in X$$

$\mu_A(x)$ es el término que indica el grado de pertenencia -de identificación- del elemento x dentro del conjunto A . E indica el autor que intuitivamente el conjunto A es una clase sin límites definidos de manera exacta, es decir una clase donde un punto x puede tener un grado de membresía que es intermedio entre la no pertenencia y la pertenencia completa. Para Zadeh lo importante en este punto era denotar que ese conjunto A podría ser definido de manera precisa asociando cada elemento x del conjunto A con su grado de membresía asignado. El intervalo de pertenencia va desde la no membresía, es decir el valor 0, hasta la membresía completa al conjunto, es decir valor 1 o membresía al 100%.

Para explicar el concepto él empleó un conjunto A definido como los números reales muy superiores al número 1, y en torno a este desarrolla conceptos:

El conjunto $A = \{x \mid x \gg 1\}$ es el conjunto de los números reales superiores -muy superiores- al número uno. La definición del conjunto quedaría escrita como sigue:

$$\begin{aligned} \mu_A(x) &= 0 && \text{for } x \leq 1 \\ &= (1 + (x - 1)^{-2})^{-1} && \text{for } x > 1 \end{aligned}$$

A continuación, se establece un concepto de enorme valor, y en su momento totalmente innovador. Se trata del concepto nuevo de pertenencia, en contraste con el concepto de pertenencia clásica a un conjunto, con notación \in , donde un elemento era o no era miembro de ese conjunto. A partir del nacimiento de la teoría de conjuntos difusos se establece una pertenencia progresiva, matizada, de tal manera que el símbolo tradicional de pertenencia pasa a ser el símbolo \in .

El autor quería introducir desde ese momento de manera nítida la diferencia que existía desde ya, entre pertenecer a un conjunto -al modo clásico—o pertenecer asociadamente a un conjunto; el último hace referencia a la pertenencia de los elementos x en el conjunto A cuando los valores de x no se identifican ni con la pertenencia nula (valor 0) ni con la pertenencia total (valor 1).

Desde este hito Zadeh desarrolla todas las propiedades matemáticas clásicas conocidas (contenido, igualdad, complemento, unión, intersección, producto algebraico, suma algebraica, relación, composición de relaciones).

Fuzzificación

Llamamos *fuzzificación* -recurriendo a un anglicismo- al proceso por el que ablandamos -suavizamos- el criterio de pertenencia de un elemento a un conjunto definido, viniendo de un criterio más rígido de pertenencia a ese conjunto. Es decir, de alguna manera humanizamos criterios muy rígidos adaptándolos a la flexibilidad del pensamiento y toma de decisiones humanos, viniendo de criterios discretos computacionales. Hablamos aquí de tecnología de lenguaje y de lingüística, al definir conjuntos con palabras nativamente abstractas (poco, mucho, más, algo menos, sustancialmente menor, casi nulo, poco probable, casi igual, etc.).

Esto está muy documentado, *The Fuzzy logic uses similar pattern as of how humans think. Hence fuzzy logic is used to deal with information having imprecise terms, for example height is tall, age is Young* (Thakera, S., Nagorib, V. 2018).

En realidad, lo que ejecutamos al *fuzzificar* es mapear un rango de valores numéricos a adjetivos calificativos o adverbios.

Hence fuzzy logic is used to provide a sort of interface between human brains and computer. It can simultaneously handle exact and vague data. The fuzzy logic uses fuzzy sets having a value between 0 and 1. (Thakera, S., Nagorib, V. 2018)

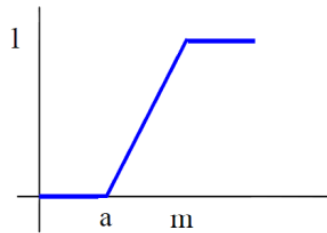
Para el caso de este trabajo la *fuzzificación* tiene un valor muy relevante pues como veremos más adelante tenemos que establecer un mecanismo por el que convertimos reglas de estrategias de inversión rígidas a reglas más suaves en el enunciado de sus intenciones, que hagan el sistema automático, pero más parecido a un método nativo *humano de calificar a las cosas*.

En la *fuzzificación* se emplea la medida de pertenencia de un elemento a uno o más conjuntos difusos en un porcentaje determinado de membresía; estos conjuntos han sido previamente

nombrados con reglas lingüísticas humanas. Esa pertenencia está tabulada en unas funciones de pertenencia bien conocidas, de las que vemos a continuación a mencionar las principales:

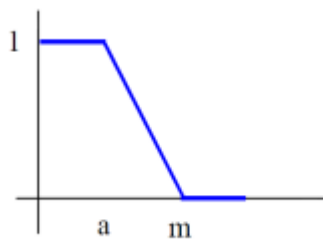
Gamma

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{m-a} & \text{si } a < x < m \\ 1 & \text{si } x \geq m \end{cases}$$



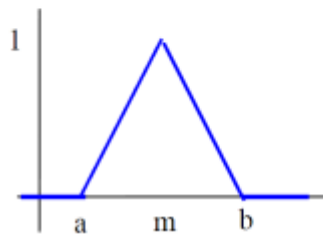
Gamma Inversa

$$\mu(x) = \begin{cases} 1 & \text{si } x \leq a \\ \frac{m-x}{m-a} & \text{si } x \in (a, m) \\ 0 & \text{si } x \geq m \end{cases}$$



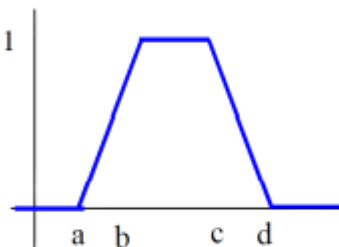
Triangular (Lambda)

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{m-a} & \text{si } a < x \leq m \\ \frac{b-x}{b-m} & \text{si } m < x \leq b \\ 0 & \text{si } x \geq b \end{cases}$$



Trapezoidal (Pi)

$$\mu(x) = \begin{cases} 0 & \text{si } x \leq a \\ \frac{x-a}{b-a} & \text{si } a < x \leq b \\ 1 & \text{si } b < x \leq c \\ \frac{d-x}{b-c} & \text{si } c < x \leq d \\ 0 & \text{si } x > d \end{cases}$$



Al igual que en matemática clásica los conjuntos difusos soportan operaciones matemáticas tales como igualdad, inclusión unión, intersección, complementario, etc.

En el proceso de *fuzzificación* que veremos más adelante en un ejemplo concreto de estrategia de trading, el primer paso será la definición inteligente, bien asesorada, de los conjuntos difusos que sustentarán el proceso lógico difuso.

Defuzzificación

Tras dotar al proceso de una lógica difusa con conjuntos difusos definidos mediante *fuzzificación*, y tras aplicar reglas de inferencia, llegaos a un último paso que se verá mas adelante en un ejemplo concreto, llamado *defuzzificación*, por el cual las conclusiones del proceso deben ser cuantificadas en números para poder aplicarlas a un sistema que trata con variables numéricas. Es decir, la *defuzzificación* es el viaje de regreso desde el inicio del sistema *fuzzificado*, hacia la cuantificación final numérica no borrosa. Los métodos de defuzzificación son variados, desde el cálculo por máximo, por media de centros, por centro de área. En el ejemplo desarrollado más adelante se emplea el centro del área o centroide.

3.5.3 Estrategia Técnica Difusa

El espacio de aplicación de la lógica difusa del doctor Zadeh se aplicaría fácilmente mapeando los valores de entrada del indicador RSI -para ser más precisos los intervalos de valor del activo- a etiquetas lingüísticas que definan estados más o menos pertenecientes, del valor. Estas etiquetas lingüísticas serian la definición borrosa -humana- de las entradas. A continuación, se realiza paso a paso la aplicación de lógica difusa:

Comenzamos por definir una variable lingüística llamada ESTADO DEL ACTIVO, que define la situación actual del activo de mercado en función de si ha sido coprado o vendido en cantidades que llevan e indicador al límite inferior -muy vendido- o superior -muy comprado-.

En matemáticas clásicas no existen nada más que las variables numéricas, sin embargo con la matemática que sustenta la lógica difusa aparece el concepto de variable lingüística que tiene por

objeto manejar apreciaciones imprecisas de los valores de un activo en términos de valoración humana, tales como MUY, MUCHO, BASTANTE, DE VEZ EN CUANDO , etc., porque esa es la manera en la que el ser humano cuantifica las cosas salvo cuando aplica matemáticas clásicas para obtener un valor numérico en una operación.

ETIQUETAS SINTACTICAS DEL VALOR DE ENTRADA SEGÚN INDICADOR RSI

SUPERVENDIDO

SOBREVENDIDO

SOBRECOPRADO

SUPERCOMPRADO

Estas etiquetas son arbitrarias, pero definen bien la percepción real de la mente humana respecto al concepto que esta evalúa.

En lógica difusa los valores definidos anteriormente tendrían una expresión matemática visual como sigue:

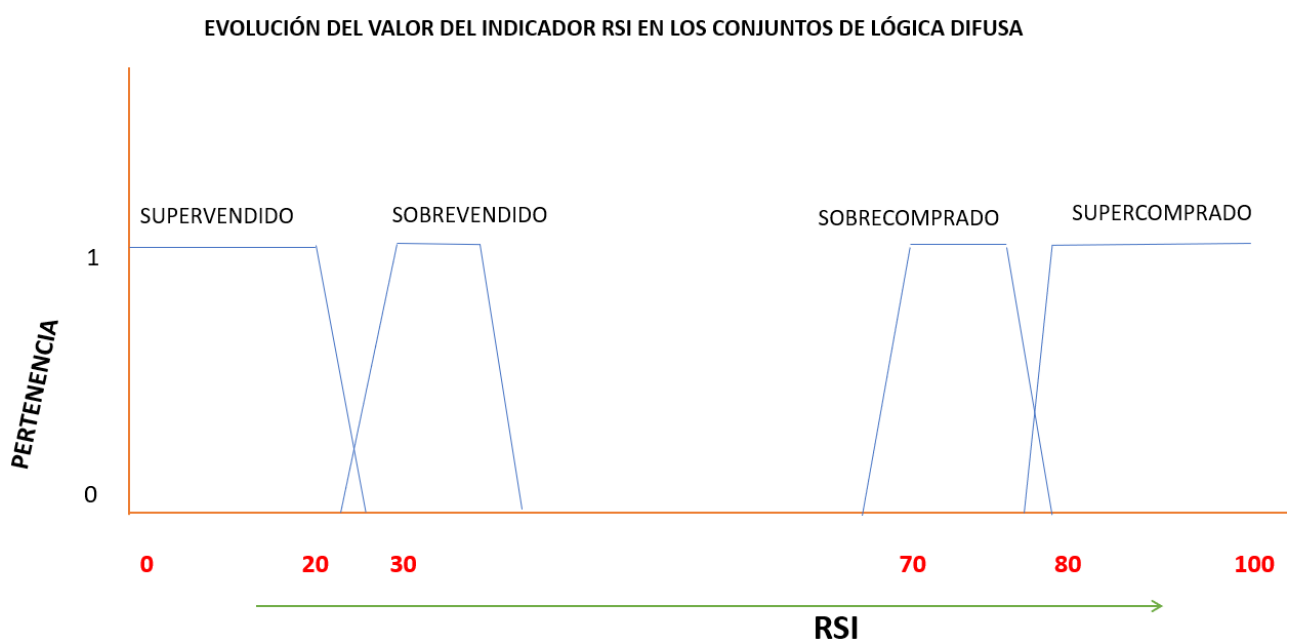


Figura 15. Función de pertenencia de la variable de entrada en matemática difusa

La interpretación de este diagrama es la siguiente: un valor cualquiera del indicador RSI -en el eje de abscisas- tendrá un grado de pertenencia diferente para uno o más conjuntos del intervalo de valores que puede adoptar el indicador y definidos por conjuntos con etiquetas lingüísticas. Por ejemplo, si el valor del indicador RSI tomara el valor 25, al proyectar ese valor sobre el eje de ordenadas se obtendrá el valor de pertenencia al conjunto llamado SOBREVENDIDO y que será de 0,2, y se obtendrá el valor de pertenencia al conjunto llamado SUPERVENDIDO y que será de 0,2 también. Otra verbalización sería que el valor $RSI=25$ es al 20% un valor SOBREVENDIDO y al 20% un valor SUPERVENDIDO.

Para el caso de nuestra estrategia con lógica difusa emplearemos dos valores de RSI, el del periodo -2 y el del periodo -1, es decir si nuestra gráfica muestra una vela japonesa de valores cada cinco minutos, nuestra estrategia tendría en cuenta para la variable de salida -comprar o no comprar el activo- el valor del indicador RSI en el periodo de los últimos cinco minutos y el del periodo de los anteriores cinco minutos al anterior, es decir $t-2$ y $t-1$, siendo la vela japonesa actual la que en tiempo real estará representando los valores de los actuales cinco minutos, o t_0 .

La variable de salida de nuestro sistema será también lingüística y representa la actuación a realizar en función del valor de las variables lingüísticas de entrada.

En el siguiente paso definimos la variable de salida empleando etiquetas de variable sintácticas, como sigue:

ETIQUETAS SINTÁCTICAS DEL VALOR DE SALIDA

POCO

MUCHO

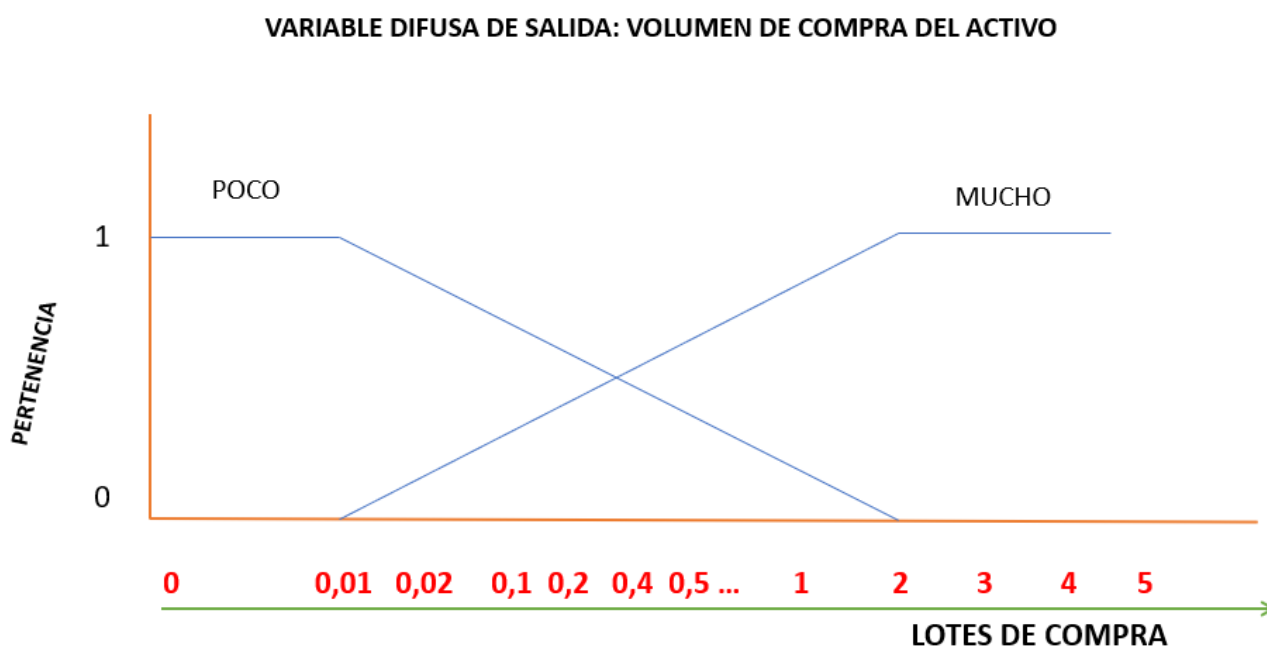


Figura 16. Función de pertenencia de la variable de salida en lógica difusa

La variable de salida tomará el valor resultante de la operación de unión o conjunción de las variables de entrada aplicado sobre la variable de salida, la cual nos indicará por inferencia gráfica cuantos lotes del activo se procederá a comprar, desde 0 hasta 5. Estos valores son arbitrarios y elegidos por un experto en trading. Esto apunta un hecho importante de este trabajo: si bien la hipótesis a lo largo del trabajo de investigación es que se puede automatizar la codificación de una estrategia de trading, también es imprescindible que los valores arbitrarios elegidos para las variables sintácticas de entrada y salida del sistema difuso sean asignados por un experto en la materia de trading. Si no fuera esto así, el resultado sería que el sistema automático generaría código fuente automatizadamente, pero fundamentado en un error operativo inversionista.

Más adelante en el trabajo se ofrece el código fuente en lenguaje Java que facilita la interfaz de operación del usuario con su generador del código fuente en lenguaje MQL4 de la plataforma Metatrader4.

A continuación, se desarrolla un ejemplo concreto que es referencia de la generalización aplicable cuando se ejecuta el código software de automatización, de manera que en el ejemplo podemos ver cuál es ese proceso de computación programado que realiza el código desde la perspectiva de análisis de las variables lingüísticas de entrada y salida para valores concretos del indicador RSI.

La problemática que trataremos de resolver en este ejemplo es obtener la respuesta a la pregunta concreta y práctica del día a día de un trader: ¿Cuántos contratos de mi activo debo comprar o vender si el valor de RSI en t-2 es 71 y el valor de RSI en t-1 es 83?

Paso 1: Definición de las reglas de inversión que empleará nuestro sistema automático difuso:

Pueden ser definidas múltiples reglas de actuación, pero a efectos de obtener una explicación e implementación muy sencilla se reduce a cuatro reglas el modelo del ejemplo:

Regla #1

Si el activo en (t-2) está SOBRECOPRADO y el activo en (t-1) está SUPERCOPRADO entonces entrar al mercado y VENDER POCO

`SOBRECOPRADO(t-2) .AND. SUPERCOPRADO(t-1) -----> VENDER POCO`

Regla #2

Si el activo en (t-2) está SOBREVENDIDO y el activo en (t-1) está SUPERVENDIDO entonces entrar al mercado y comprar POCO

`SOBREVENDIDO(t-2) .AND. SUPERVENDIDO(t-1) -----> COMPRAR POCO`

Regla #3

Si el activo en (t-2) está SUPERCOPRADO y el activo en (t-1) está SUPERCOPRADO entonces entrar al mercado y VENDER MUCHO

`SUPERCOPRADO(t-2) .AND. SUPERCOPRADO(t-1) -----> VENDER MUCHO`

Regla #4

Si el activo en (t-2) está SUPERVENDIDO y el activo en (t-1) está SUPERVENDIDO entonces entrar al mercado y comprar MUCHO

SUPERVENDIDO(t-2) .AND. SUPERVENDIDO(t-1) -----> COMPRAR MUCHO

Siguiendo estas reglas sencillas se concluye que la compra o venta del activo será el resultado de la aplicación y combinación de las reglas anteriores. Por tratar de ganar simplicidad se han incluido solo cuatro reglas, si bien en procesos habituales la densidad de reglas puede ser mayor bajo la necesidad de acotar bien los casos aplicables.

La interpretación de las reglas enunciadas se conoce como inferencia borrosa, y nos devolverá como resultado las conclusiones en forma de variables lingüísticas de salida a partir de las entradas. Como parte del proceso de inferencia los métodos más conocidos son el de Mamdani (más aplicable a casos menos complejos) y el TSK (acrónimo de sus autores, Takagi-Sugeno-Kan) que está más orientado a casos complejos y donde la salida no es un conjunto difuso, sino que son funciones lineales. En nuestro ejemplo y desarrollo optamos por el método Mamdani por una razón clara: tan solo empleamos en nuestro sistema difuso dos variables -o mejor dicho una variable, pero expresada en dos marcos temporales diferentes que se concatenan necesariamente para obtener la variable de salida- y buscamos obtener una salida en unos conjuntos difusos definidos; cuya salida será eventualmente cuantificada mediante un proceso de *defuzzificación* (o desborrosificación).

Tabla 3. Reglas de Inferencia. Sistema Mamdani.

REGLAS DE LÓGICA DIFUSA EN EL SISTEMA DE MAMDANI

(t-2) \ (t-1)	SUPERVENDIDO	SOBRE VENDIDO	SOBRECOMPRADO	SUPERCOMPRADO
SUPERVENDIDO	COMPRAR MUCHO			
SOBRE VENDIDO	COMPRAR POCO			
SOBRECOMPRADO				VENDER POCO
SUPERCOMPRADO				VENDER MUCHO

Paso 2: Fuzzificación (borrosificación) de los valores RSI de entrada mediante el cálculo del grado de pertenencia del valor a cada conjunto difuso -fuzzy set- definido del rango posible del valor

Valor de las entradas del indicador RSI en t-2:

hacemos lectura visual en la pantalla de la plataforma Metatrader y nos arroja un valor de 71. Este es un valor que en la definición de conjunto borrosos de la Figura 14 nos habla de un elemento del conjunto borroso SOBRECROMPRADO al que pertenece en un 100% -valor de 1- como puede apreciarse en la figura 16.

La expresión de los grados de pertenencia a los diferentes conjuntos quedaría expresada de la siguiente manera:

$$\mu_{\text{SUPERVENDIDO}}(71) = 0$$

$$\mu_{\text{SOBREVENDIDO}}(71) = 0$$

$$\mu_{\text{SOBRECOMPRADO}}(71) = 1$$

$$\mu_{\text{SUPERCOMPRADO}}(71) = 0$$

Empleando las uniones de los antecedentes de las reglas concluimos que el valor del activo en t-2 es SOBRECROMPRADO al 100% ya que 71 está dentro del rango de valores de esta etiqueta lingüística.

A continuación, la representación gráfica:

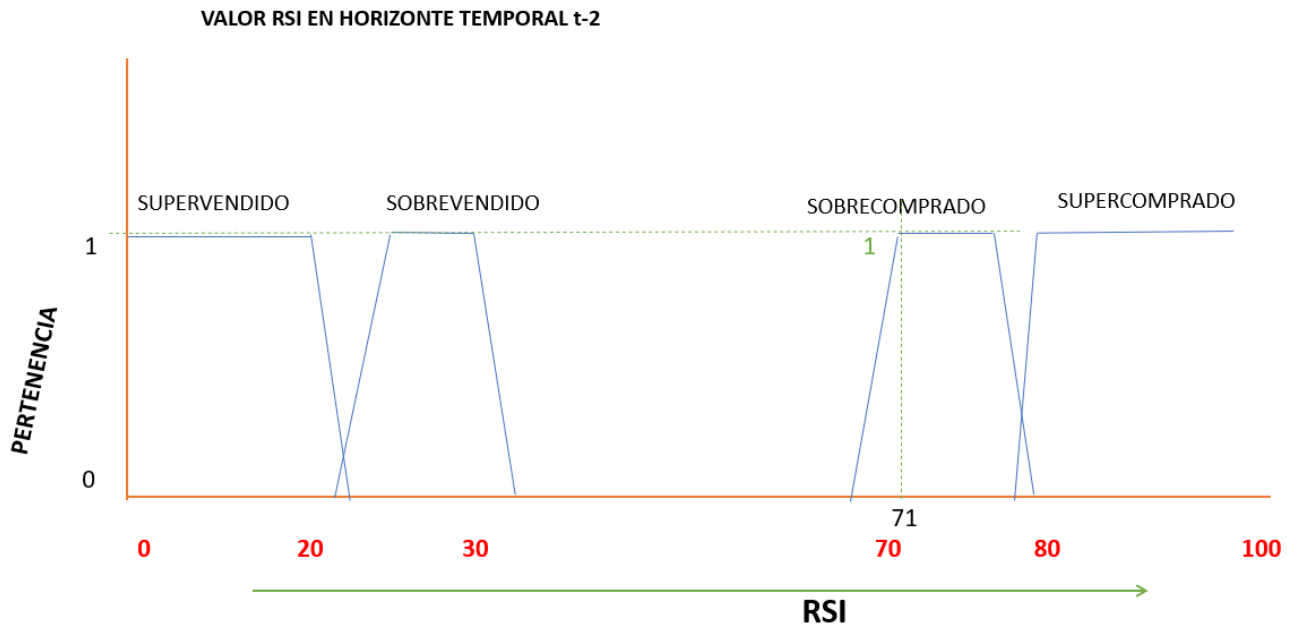


Figura 17. Función de pertenencia del valor RSI en los conjuntos borrosos definidos en t-2

Valor de las entradas del indicador RSI en t-1:

hacemos lectura visual en la pantalla de la plataforma Metatrader y nos arroja un valor de 83.

Este es un valor que en la definición de conjunto borrosos de la Figura 14 nos habla de un elemento del conjunto borroso SUPERCOMPRADO al que pertenece en un 100% -valor de 1- como puede apreciarse en la figura 17.

La expresión de los grados de pertenencia a los diferentes conjuntos quedaría expresada de la siguiente manera:

$$\mu_{\text{SOBREVENDIDO}}(71) = 0$$

$$\mu_{\text{INDEFINIDO}}(71) = 0$$

$$\mu_{\text{SOBRECOPRADO}}(71) = 0$$

$$\mu_{\text{SUPERCOMPRADO}}(71) = 1$$

Empleando las uniones de los antecedentes de las reglas concluimos que el valor del activo en t-1 es el de SUPERCOMPRADO.

Visto gráficamente:

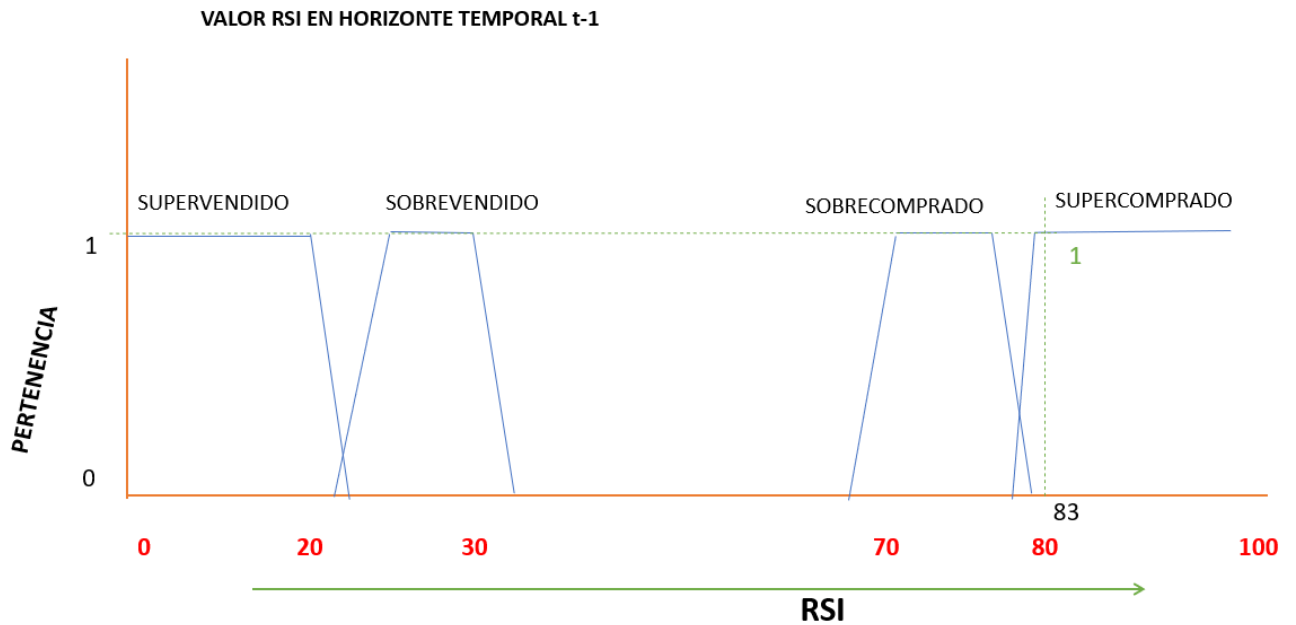


Figura 18. Función de pertenencia del valor RSI en los conjuntos borrosos definidos en t-1

La obtención de los valores de pertenencia se extrae mediante la aplicación de las ecuaciones de pertenencia mostradas en el capítulo de Introducción a la Lógica Difusa. Por razones de simplificación en nuestro ejemplo se ha inferido el valor por graficación directa.

Paso 3: Fijación del valor de la salida. Defuzzificación (desborrosificación). En este paso del proceso se obtendrán los valores numéricos a partir de los valores de las lecturas de las entradas

La lógica difusa nos ofrece varios métodos de cálculo del valor de salida, entre los que contamos con el método centroide, método MOM, método de bisectriz, método SOM, método LOM.

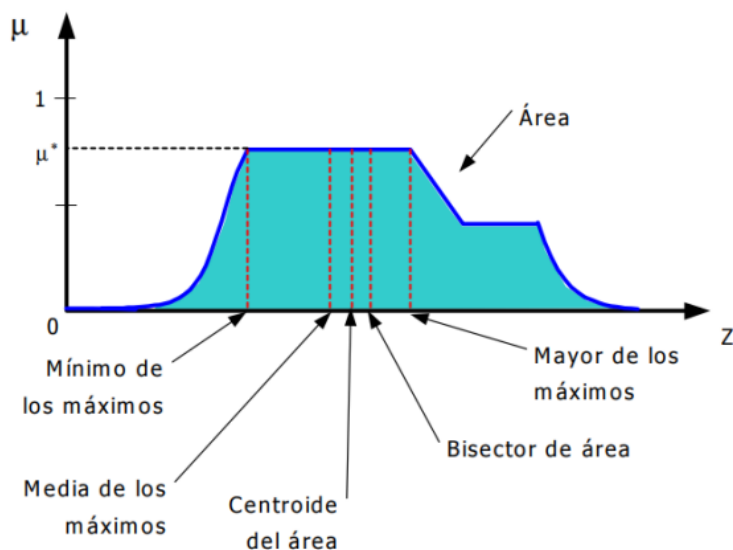


Figura 3.23: Diferentes métodos de defuzzificación.

Figura 19. Métodos del Proceso de Defuzzificación. Sanahuja, S (2017).

Por centrarnos en uno, el método del centroide, podemos definir el centroide como el centro de masas del intervalo del valor de la variable de salida. Su formulación teórica es la siguiente:

$$c = \frac{\int_S x\mu(x)dx}{\int_S \mu(x)dx}$$

Para su cálculo práctico proyectamos el valor del conjunto que tiene el máximo grado de pertenencia sobre el eje de abscisas y obtenemos el valor centroide de la salida. Para el conjunto de valores que pertenecen al conjunto POCO -hace referencia a la salida comprar poco- el centroide se sitúa en 0,01 (esto hace referencia a comprar 0,01 lotes, o contratos, del activo). Para el conjunto MUCHO el centroide se sitúa en el valor 2 (compra de dos lotes o contratos del activo).

El valor de los centroides es muy relevante en tanto en cuanto servirá como multiplicador de los valores de pertenencia obtenidos en las variables de entrada, y así obtener el valor final a aplicar en la salida mediante ponderación.

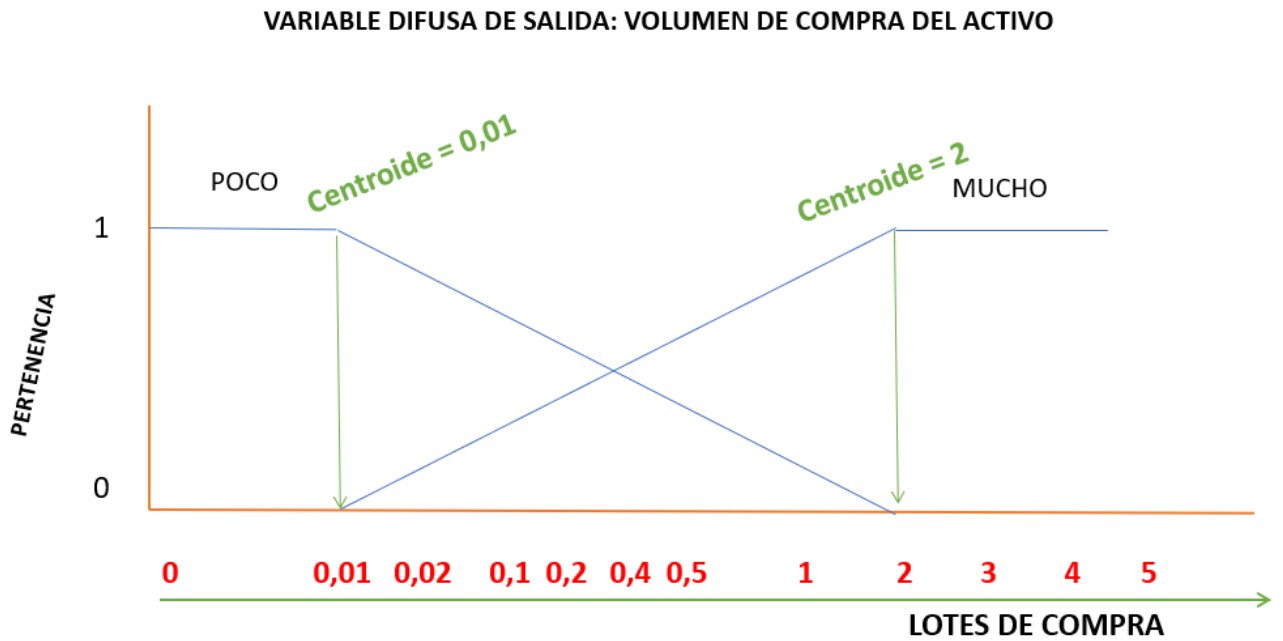


Figura 20. Ubicación de los centroides de la variable de salida

Atendiendo a nuestro sistema de reglas se identifica el cumplimiento de la regla Regla #1, ya que en el marco temporal t-2 el activo está SOBRECOPRADO y en el marco temporal t-1 el activo está SUPERCOMPRADO.

Aunque las variables de salida se enuncian como variables no numéricas sino lingüísticas, el resultado obtenido obviamente debe ser numérico, y concretamente un número real. Calculamos desde los valores de los consecuentes de las variables de entrada, y desde los centroides, el valor numérico de la variable de salida.

Análisis de cumplimiento de reglas:

Regla #1

Si el activo en (t-2) está SOBRECOPRADO y el activo en (t-1) está SUPERCOMPRADO entonces entrar al mercado y VENDER POCO

$$\begin{array}{ccccccc}
 \text{SOBRECOPRADO}(t-2) & \text{.AND.} & \text{SUPERCOMPRADO}(t-1) & \text{---->} & \text{VENDER POCO} \\
 1 & & 1 & &
 \end{array}$$

Esta regla se cumple al 100% ya que ambos consecuentes encajan los condicionales.

Regla #2

Si el activo en (t-2) está SOBREVENDIDO y el activo en (t-1) está SUPERVENDIDO entonces entrar al mercado y comprar POCO

SOBREVENDIDO(t-2) .AND. SUPERVENDIDO(t-1) -----> COMPRAR POCO

0

0

Esta regla no se cumple

Regla #3

Si el activo en (t-2) está SUPERCOMPRADO y el activo en (t-1) está SUPERCOMPRADO entonces entrar al mercado y VENDER MUCHO

SUPERCOMPRADO(t-2) .AND. SUPERCOMPRADO(t-1) -----> VENDER MUCHO

0

0

Esta regla no se cumple

Regla #4

Si el activo en (t-2) está SUPERVENDIDO y el activo en (t-1) está SUPERVENDIDO entonces entrar al mercado y comprar MUCHO

SUPERVENDIDO(t-2) .AND. SUPERVENDIDO(t-1) -----> COMPRAR MUCHO

0

0

Esta regla no se cumple

La regla #1 cumple al 100% para (t-2) y al 100% para (t-1), por lo que la salida se infiere directamente en la gráfica en la marca 0,01 de salida; es decir la recomendación del sistema difuso dado el

cumplimiento de las reglas de inferencia es comprar 0,01 contratos del activo, dada la evolución del valor del indicador técnico RSI en los dos últimos intervalos de tiempo t-2 y t-1.

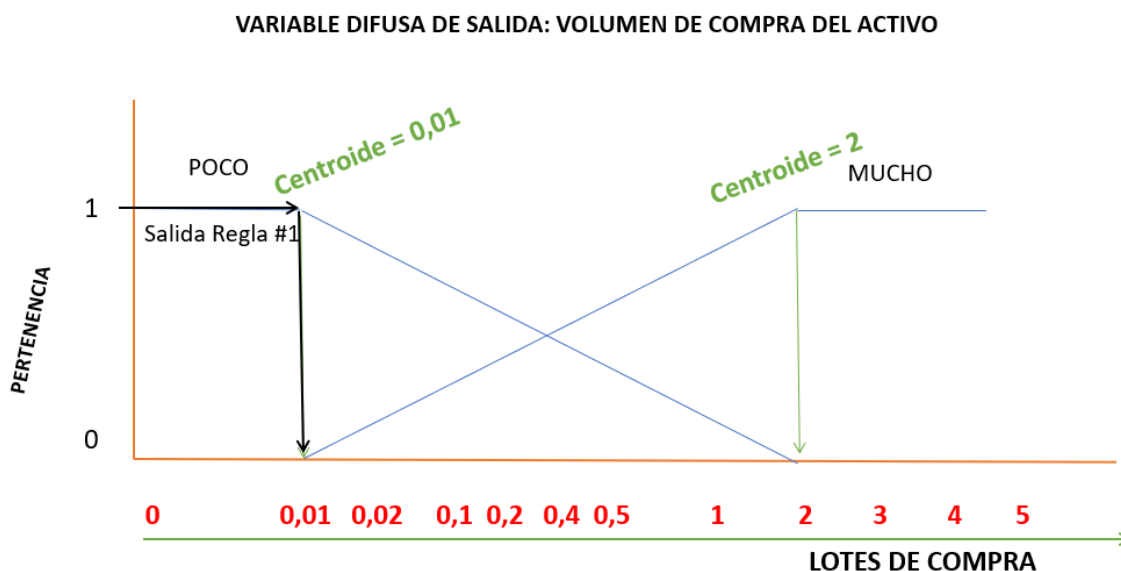


Figura 21. Obtención del valor *defuzzificado* de la variable lingüística de salida.

Por razones de sencillez de entendimiento se han empleado a propósito valores que permiten obtener inferencias muy claras y directas sin necesidad de realizar cálculos de ponderación. El valor de este trabajo reside principalmente en la transferencia del conocimiento del proceso, y para ello la elección de un ejemplo sencillo de aplicar tiene una importancia alta.

El ejemplo de reglas de inferencia aplicadas es muy básico. Es un caso real posiblemente se pueden aplicar hasta 16 o 20 reglas de inferencia para que cada estadio de evolución del indicador RSI tenga etapas difusas de corto recorrido en su evolución.

En ese caso los conjuntos difusos definidos podrían ser los siguiente:

ETIQUETAS SINTACTICAS DEL VALOR DE ENTRADA SEGÚN INDICADOR RSI

SUPERVENDIDO

SOBREVENDIDO

VENDIDO

NEUTRO
COMPRADO
SOBRECOMPRADO
SUPERCOMPRADO

En ese ejemplo la gráfica de conjuntos difusos sería la siguiente:

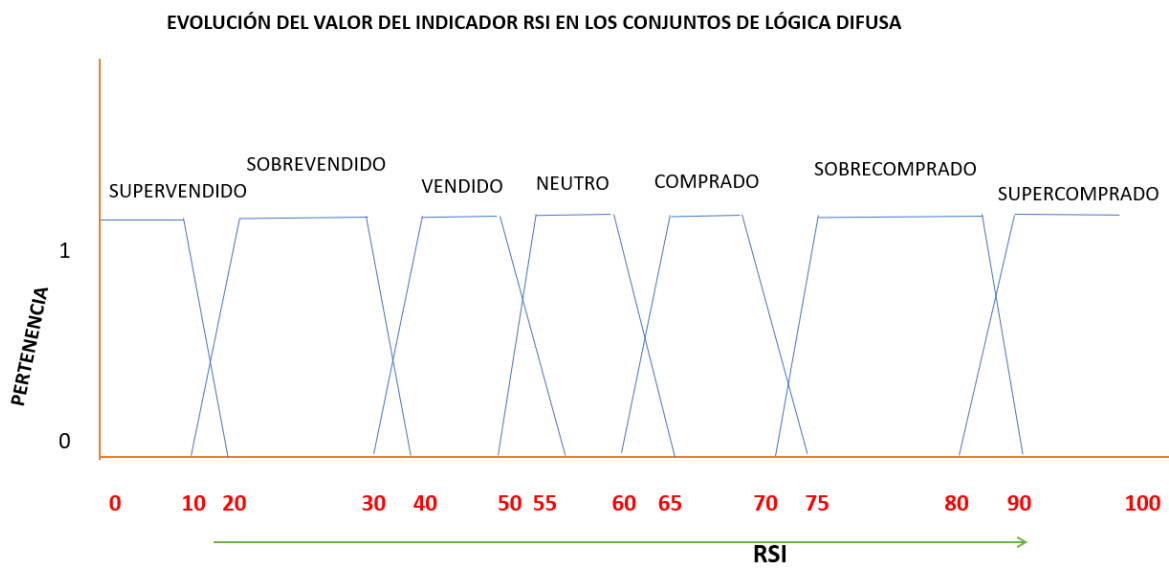


Figura 22. Conjuntos difusos múltiples para estrategia de trading

Como se mencionó anteriormente estas definiciones son arbitrarias, al igual que las definiciones de salidas difusas. La consecuencia de esa arbitrariedad es que la definición tiene que estar bien asesorada por un experto en el proceso que va a ser *fuzzificado*, y haber entrenado la lógica del proceso mediante iteración de pruebas en entorno de laboratorio o de simulación hasta corroborar científicamente que el planteamiento responde a la realidad con un grado elevado -nunca garantizado- de certidumbre.

Evaluación y Comparación de la Solución

Evaluación De la Solución

La propuesta de este trabajo como valor diferenciador es principalmente la automatización de la generación del código MQL4 que ejecuta la operativa deseada por el *trader* -estrategia de trading-; la captación de las intenciones del *trader* en su estrategia de inversión mediante la selección múltiple de diferentes opciones de menús desplegables que le ofrece la aplicación desarrollada a tal efecto, llamada **AutoTrader©**, un desarrollo realizado sobre entorno de desarrollo Eclipse en lenguaje de programación Java.

A continuación, profundizamos en los detalles de la solución elaborada con lenguaje de programación Java sobre IDE Eclipse y librerías SWING gráficas.

Recursos de programación utilizados Para desarrollar este generador de bases de datos no-code sobre SQL hemos utilizado un entorno de desarrollo consistente en: Lenguaje de programación Java. Entorno de desarrollo Eclipse, como IDE. Librerías gráficas SWING basadas en AWT.

La plataforma de ejecución es un ordenador con procesador Intel i7 con 16GB de memoria ram de estado sólido y disco duro de 500Gbyte también de estado sólido. El sistema operativo utilizado es Windows 10, versión de 64 bits. Como editor de código fuente se ha empleado Eclipse IDE.

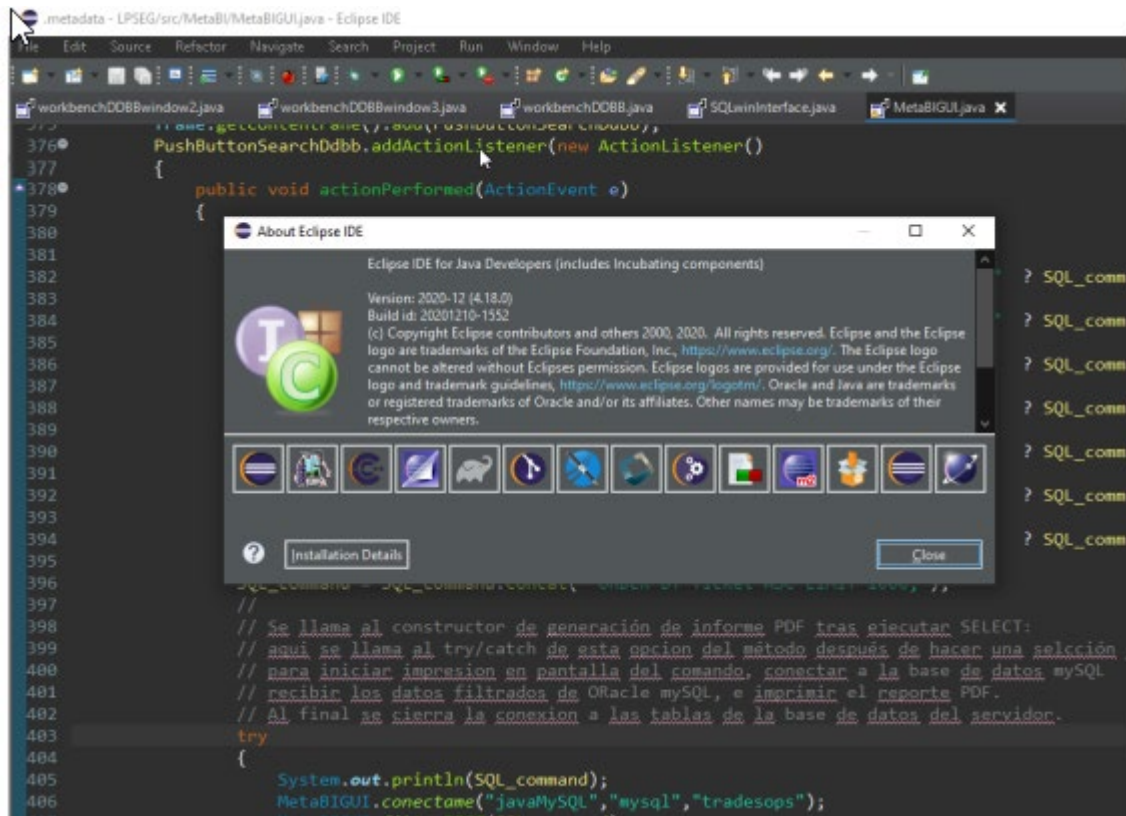


Figura 23. Detalle de la pantalla de Ayuda del editor Eclipse de Java

La solución presentada en este trabajo de investigación es un desarrollo acotado de un código software no-code que resuelve el paradigma intencional de atención al trader que no conoce la programación o no es experto en ella, pero que sin embargo necesita la potencia y la flexibilidad de un entorno no code donde sus intenciones de trading -a través de estrategias definidas- sean programadas automáticamente.

El objeto de investigación de este trabajo es profundizar en la posibilidad real de crear un entorno asistente totalmente parametrizable en el contexto específico de un trader. Como condición base se incluyó que la plataforma sobre la que se realizara la investigación y el desarrollo de prueba fuera una plataforma comercial muy conocida y valorada por los traders. Por esa razón se eligió Metatrader4.

El desarrollo fundamental del paradigma consiste en una pantalla interactiva donde el usuario final -trader- define su estrategia seleccionando opciones en menús desplegables. La interacción que se ha buscado es para nivel de conocimiento bajo o medio en informática, si bien sería deseable impartir una sesión formativa de 30 minutos al trader para enseñar a utilizar la herramienta y a adaptarla un poco a la medida de su necesidad. La tecnología subyacente es programación Java sobre Eclipse; los menús han sido generador con

librería SWING y la captación de las intenciones del trader quedan almacenadas en las variables enteras, de coma flotante, fecha/hora o string que reciben las selecciones de menús realizadas.

Con el desarrollo diseñado se ha hecho opaco al usuario toda la complejidad asociada. Este módulo permite al usuario final -trader- seleccionar los parámetros de su estrategia de trading sin modificar ninguna de sus intenciones.

En una sola pantalla de trabajo el trader es interrogado por cada uno de los parámetros de entrada a una estrategia de compra o venta de mercado de capitales, tales como el activo sobre el que quiere operar, a partir de qué valor del indicador RSI considera que está el valor sobrevendido o sobrecomprado, qué horario de operativa querrá utilizar, cuantos puntos básicos de ganancia o de pérdida debe permitir el Expert Advisor cuando esté en ejecución antes de cerrar la operación abierta, etc.

AutoTrader (c)2020/22

Entradas a la Estrategia

SOBREVENDIDO 20

SOBRECOPRADO 70

Activo EURUSD

Indicador Técnico SMA

Periodo De Indicador t-1

Control de Ganancias

Limite Ingreso Diario 100 €

Pips de Ganancia 10

Pips de Pérdida 5

Mercado

Hora Apertura de Operativa 9 am

Hora Cierre de Operativa 4 pm

Posiciones Permitidas LARGO

Volumen de Compra 0.01

Volumen de Venta 0.01

MagicNumber 1234

Generar Expert Advisor MT4;

Figura 24. Pantalla de AutoTrader

En la figura anterior vemos la pantalla única de interacción del trader con el sistema. Una vez que el trader haya realizado la elección de los campos de estrategia, esta será generada en un robot -Expert Advisor- en lenguaje de plataforma Metatrader4, de nombre MQL4.

A continuación, se pueden ver algunas capturas de las opciones de menús que se despliegan para que el trader haga su elección de parámetros para la estrategia de inversión:

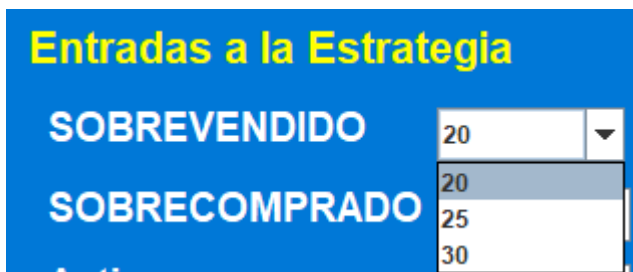


Figura 25. Nivel de indicador RSI para posición de venta

El menú anterior permite seleccionar el nivel de indicador RSI que disparará la compra -activo sobrevendido- del número de lotes seleccionados del activo.

En la figura siguiente vemos el menú de selección del activo que puede seleccionar el trader para operar.



Figura 26. Selección del activo de operativa en mercado

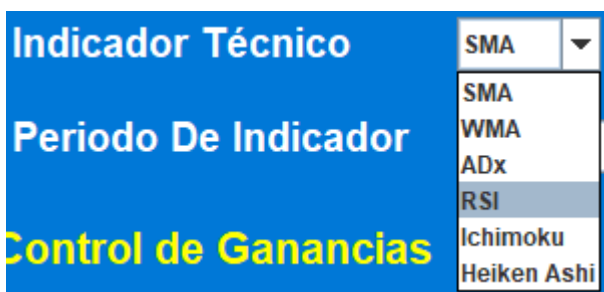


Figura 27. Selección del indicador técnico para operativa

En este trabajo se ha centrado la atención sobre el indicador técnico RSI, pero existen muchos otros indicadores técnicos, los cuales son ofrecidos al trader en e menú de la figura anterior-

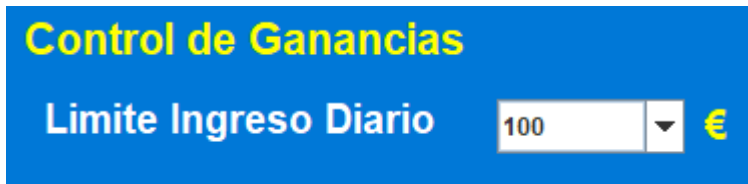


Figura Selección de tope de ganancias para parada de operativa

Es frecuente entre traders determinar una cantidad de ingresos diaria que se quiere obtener antes de parar la operativa, así como la cantidad ganada que indicará que es el momento de cerrar la operativa por el día. En la figura anterior vemos el menú de selección de la cantidad máxima diaria topada.

La figura inferior es la que muestra la opción de menú que permite que el Expert Advisor generado en MQL4 abra automáticamente posiciones de compra -largo- o de venta -corto- o mixtas simultáneas haciendo hedging.

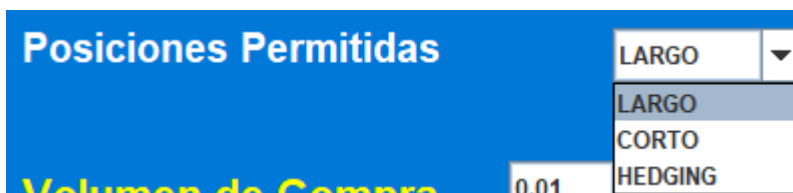


Figura 28. Autorización para operar en compras, en ventas y posiciones mixtas

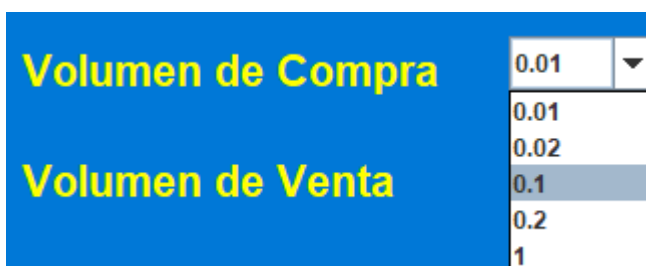


Figura 29. Selección de los contratos de compra o venta en operativa

La figura anterior muestra la opción de menú que permite seleccionar cuantos lotes o minilotes o microlotes del activo será puesto en venta o serán comprados por el Expert Advisor cuando se dispare la operativa.

La operativa completa consiste en los siguientes pasos:

- Lanzamiento de la aplicación y selección de campos por el trader
- Pulsación del botón Generar Expert Advisor MT4
- Importar en Metatrader el fichero fuente MQL4 generado desde AutoTrader

Como se puede apreciar la operativa es enormemente sencilla, pero como se ha mencionado anteriormente será necesario que el desarrollador o comercializador de la aplicación AutoTrader imparta un pequeño curso de los tres pasos de la operativa anterior.

Del párrafo anterior se deduce que AutoTrader genera un fichero con el código software escrito en lenguaje MQL4 que deberá ser importado en la plataforma Metatrader. Veamos algunas capturas que enseñan la operativa y los resultados:

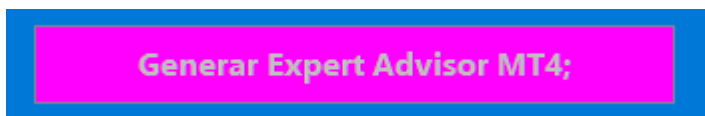


Figura 30. Detalle del botón generador del fichero con código MQL4

La pulsación del botón de la figura anterior genera dos acciones: muestra en la consola de Eclipse el fuente MQL4 generado, para comodidad del operador, y además genera el fichero con extensión .mq4 para ser importado y compilado en la plataforma Metatrader4.

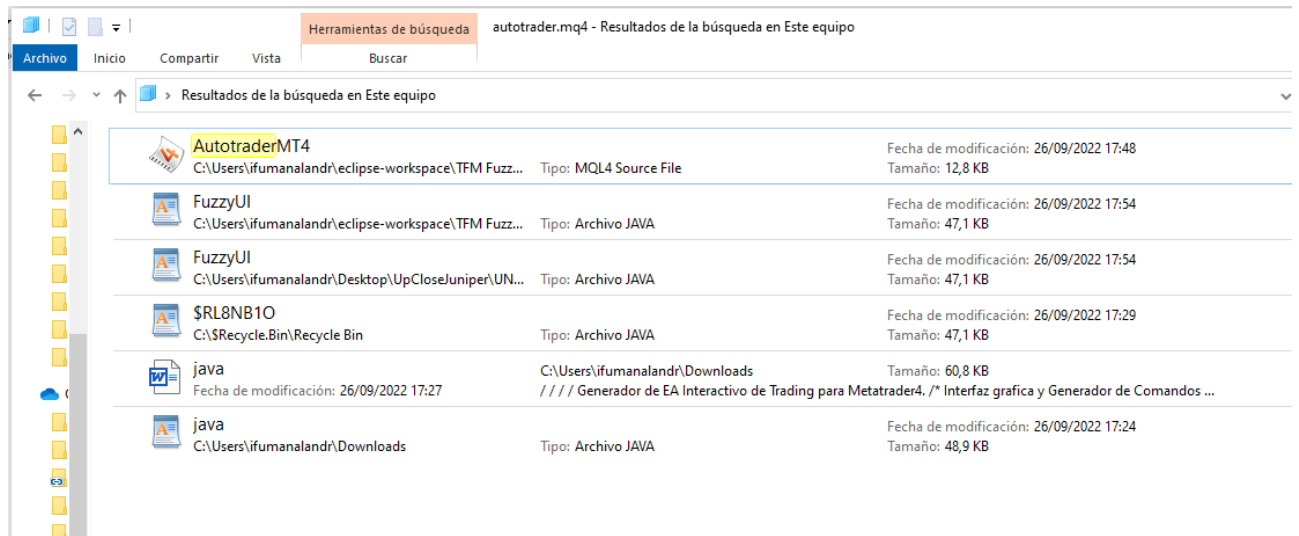


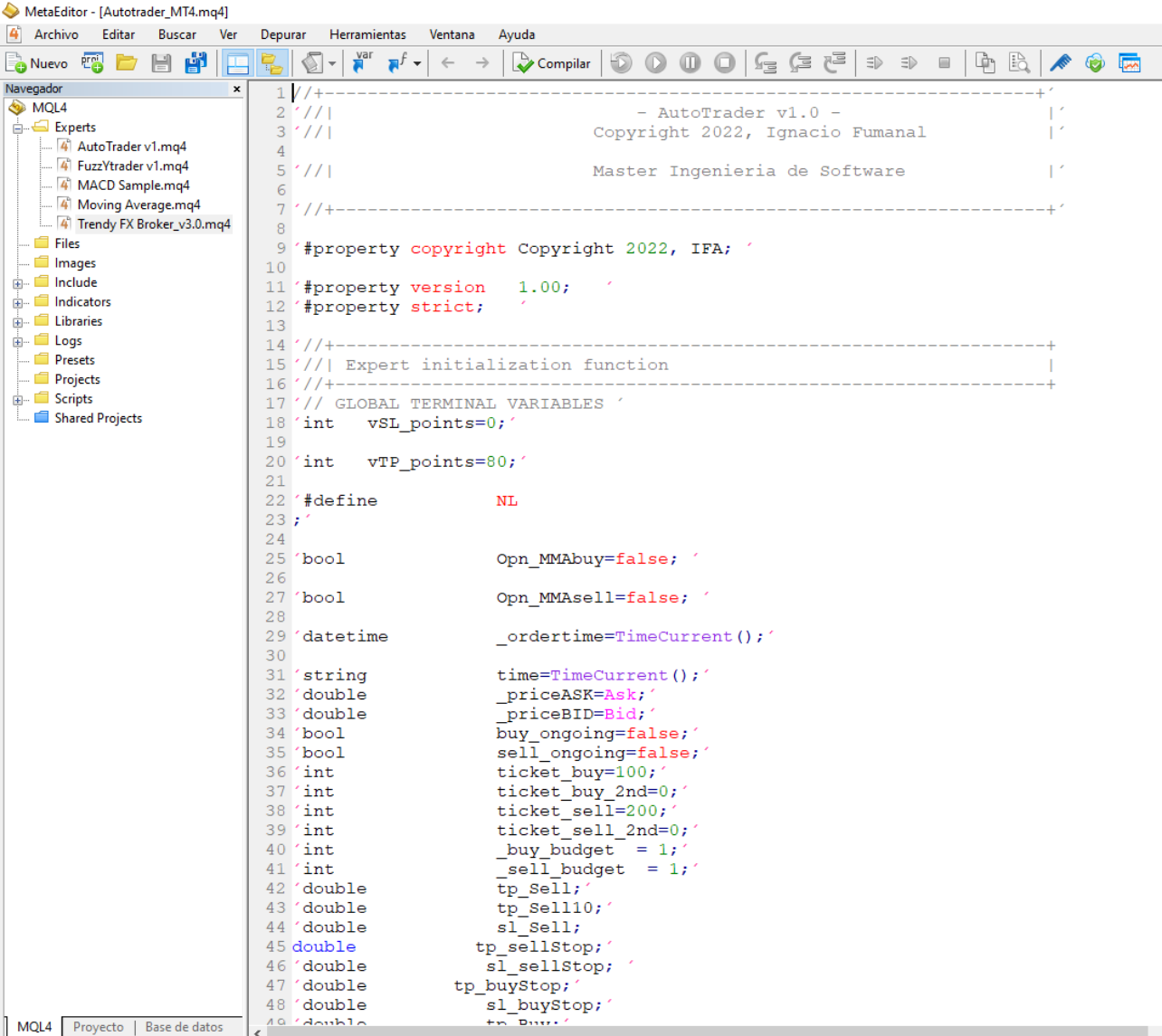
Figura 31. Carpeta conteniendo el fuente MQL4 generado desde AutoTrader: AutoTrader.mq4

EL fichero generado por AutoTrader© se deposita en la carpeta seleccionada para su depósito.

Al clicar sobre el fichero generado -teniendo abierta la plataforma MetaTrader4- se abre una nueva pantalla de Expert Advisor en el editor que trae embebido la plataforma, por nombre MetaEditor.

En la figura anterior se puede ver el detalle de la carpeta donde el código fuente Java de la aplicación AutoTrader ha depositado el fichero MQL4 generado.

A continuación, las capturas correspondientes:



```
1 //+-----+
2 ///           - AutoTrader v1.0 -
3 ///           Copyright 2022, Ignacio Fumanal
4
5 ///           Master Ingenieria de Software
6
7 //+-----+
8
9 #property copyright Copyright 2022, IFA;
10
11 #property version 1.00;
12 #property strict;
13
14 //+-----+
15 /// Expert initialization function
16 //+-----+
17 /// GLOBAL TERMINAL VARIABLES
18 int vSL_points=0;
19
20 int vTP_points=80;
21
22 #define NL
23 ;
24
25 bool Opn_MMAbuy=false;
26
27 bool Opn_MMAsell=false;
28
29 datetime _ordertime=TimeCurrent();
30
31 string time=TimeCurrent();
32 double _priceASK=Ask;
33 double _priceBID=Bid;
34 bool buy_ongoing=false;
35 bool sell_ongoing=false;
36 int ticket_buy=100;
37 int ticket_buy_2nd=0;
38 int ticket_sell=200;
39 int ticket_sell_2nd=0;
40 int _buy_budget = 1;
41 int _sell_budget = 1;
42 double tp_Sell;
43 double tp_Sell10;
44 double sl_Sell;
45 double tp_sellStop;
46 double sl_sellStop;
47 double tp_buyStop;
48 double sl_buyStop;
49 double tp_Buy;
```

Figura 32. Editor MetaEditor de la plataforma Metatrader4. Código fuente generado: AutoTrader.mq4

La figura anterior es la vista del editor que trae embebido la plataforma Metatrader4 para que los traders con habilidades de desarrollo puedan codificar en lenguaje MQL4 sus propios indicadores técnicos, scripts, o incluso Expert Advisors como es el caso de AutoTrader.

La estructura del código fuente generado desde Java mantiene la estructura siguiente:

- Cabecera
- Sentencias property
- Definición de variables globales
- Definición de variables AutoTrader
- Métodos MQL4 de operativa bursátil

A continuación, podemos ver algunas capturas de los métodos principales generador por Autotrader en MQL4:

```
int OnInit();
{
    Sleep(20);
    buy_ongoing = false;
    sell_ongoing = false;
    ticketclosed = false;
    _StopbyError = false;
    _buy_budget=1;
    _sell_budget=1;
    explore_order = true;
    _today = datetime(); daily_revenue = 0;
    working_hours=false;
    Opn_MMAbuy = true; Opn_MMAsell = true;
    return(0);
};
```

Figura 33. Código fuente del método OnInit() generado por AutoTrader

```
void OnTick();
{
    check_RSI_level(RSIlevel_to_Buy, RSIlevel_to_Sell);
    check_opening_hours(); // Allows trading just between 08:00 and 18:00;
    if ((buy_ongoing == false) && (sell_ongoing == false) && (working_hours == true) && (daily_revenue < 200));
    {
        goBuy(Opn_MMAbuy);
        Comment("Open_bid: ", open_bid, " Open SL: ", open_SL, " Open TP: ",open_TP, " Operation: ",name + NL);
        if (_trail_buy == true);
        {
            TrailBuy(ticket_buy) ;
            if (OrderProfit() < 0);
            {
                TrailSell(ticket_sellStop);
            }
        }
    };
    if ((buy_ongoing == false) && (sell_ongoing == false) && (working_hours == true) && (daily_revenue < 200));
    {
        goSell(Opn_MMAsell);
        Comment("Open_bid: ", open_bid, " Open SL: ", open_SL, " Open TP: ",open_TP, " Operation: ",name + NL);
        if (_trail_sell == true);
        {
            TrailSell(ticket_sell) ;
            if (OrderProfit() < 1);
            {
                TrailBuy(ticket_buyStop);
            }
        }
    };
};
```

Figura Código fuente del método OnInit() generado por AutoTrader

```
'void check_RSI_level(int RSIlevel_to_Buy, int RSIlevel_to_Sell);'  
{;  
    if (RSIlevel_to_Buy <= iRSI(NULL, 15,3,PRICE_CLOSE,1));'  
        {;  
            Opn_MMABuy == true;    ;'  
                };'  
        if (RSIlevel_to_Sell >= iRSI(NULL, 15,3,PRICE_CLOSE,1));'  
            {;  
                Opn_MMAsell == true;'  
                    };'  
            };'  
};'
```

Figura 34. Código fuente del método check_RSI_level()

El método anterior es el que ecanea continuamente el valor del indicador RSI para disparar -según los valores seleccionados por el trader- el proceso de compra o venta de activos.

```
'void goBuy(bool Opn_MMABuy);'  
{;  
    if ((Opn_MMABuy == true) && ( _buy_budget == 1 ));'  
        {;  
            if (IsTradeAllowed() == true);'  
                {;  
                    RefreshRates();'  
                    {;  
                        open_ask = Ask; open_bid = Bid;'  
                        tp_Buy = open_ask + Point * _tp; tp_Buy = NormalizeDouble(tp_Buy , Digits);'  
                        tp_Buy10 = open_ask + Point * _tp10; tp_Buy10 = NormalizeDouble(tp_Buy10, Digits);'  
                        sl_Buy = open_ask - Point * _sl; sl_Buy = NormalizeDouble(sl_Buy, Digits);'  
                    };'  
                    ticket_buy = OrderSend(Symbol(),OP_BUY,LotSize_buy, open_ask ,3, 0 , 0 , _ordercomment,MagicNumber,0,clrGreen);'  
                    Sleep(500);'  
                    {;  
                        write_Screenshot(ticket_buy);'  
                        OrderSelect(ticket_buy,MODE_TRADES);'  
                        if (ticket_buy <0);'  
                            {;  
                                _StopbyError = true;'  
                                return;'  
                            };'  
                        if (ticket_buy > 1);'  
                            {;  
                                open_bid = sl_Buy;'  
                                tp_sellStop = open_bid - _tp * Point; tp_sellStop = NormalizeDouble(tp_sellStop,Digits);'  
                                sl_sellStop = open_ask + _sl * Point; sl_sellStop = NormalizeDouble(sl_sellStop,Digits);'  
                                ticket_sellStop = OrderSend(Symbol(),OP_SELLSTOP,LotSize*2, open_bid ,3, sl_sellStop,tp_sellStop , _orderc  
                                Sleep(500);'  
                                write_Screenshot(ticket_sellStop);'  
                                _trail_buy = true;'  
                                buy_ongoing = true;'  
                                _buy_budget = 0;'  
                                _sell_budget = 1 ;'  
                                OrderAnalyticOpen(ticket_buy);'  
                                highest_high = Ask; lowest_low = Bid;'  
                            };'  
                        };'  
                    };'  
                };'  
            };'  
        };'  
};'
```

Figura 35. Código fuente del método goBuy()

El método goBuy() es el método llamado cuando el nivel de indicador RSI seleccionado por el trader alcanza un valor que dispara la compra del activo seleccionado por el trader en la pantalla de parametrización de AutoTrader.

Existe un método equivalente para disparar la venta, llamado go Sell().

Estos métodos de compra y venta de activos utilizan los parámetros que el trader ha seleccionado por menús, tales como el volumen de compra o venta, límite de perdidas -stop loss-, límite de ganancias -take profit-, etc.

```
//
// Captacion de las opciones de Trading que el trader ha seleccionado
//
int RSI_level_to_Sell = Integer.valueOf((String) buttonSobrevendido.getSelectedItem());
int RSI_level_to_Buy = Integer.valueOf((String) buttonSobrecomprado.getSelectedItem());
float LotSize_buy = Float.valueOf((String) buttonVolumencompra.getSelectedItem());
String Symbol = (String) buttonActivo.getSelectedItem();
String vela = (String) buttonPeriodoindicador.getSelectedItem();
int ingresoDaily = Integer.valueOf((String) buttonIngresodiario.getSelectedItem());
String tipoPosicion = (String) buttonPosicionespermitidas.getSelectedItem();
int tradeFrom = Integer.valueOf((String) buttonHoraapertura.getSelectedItem());
int tradeTill = Integer.valueOf((String) buttonHoraapertura.getSelectedItem());
int vSL_ = Integer.valueOf((String) vSL.getSelectedItem());
int vTP_ = Integer.valueOf((String) vTP.getSelectedItem());

//
MQL4_source = MQL4_source.concat("//////////////////////////////////// \r\n");
MQL4_source = MQL4_source.concat("//////// Variables setting from JAVA screen to MQL4 ////////// \r\n");
MQL4_source = MQL4_source.concat("// \r\n");
MQL4_source = MQL4_source.concat("int RSILEvel_to_Buy= " + RSI_level_to_Buy + "; \r\n");
MQL4_source = MQL4_source.concat("int RSILEvel_to_Sell= " + RSI_level_to_Sell + "; \r\n");
MQL4_source = MQL4_source.concat("int LotSize_buy= " + LotSize_buy + "; \r\n");
MQL4_source = MQL4_source.concat("int LotSize_sell= " + LotSize_sell + "; \r\n");
MQL4_source = MQL4_source.concat("double vSL=" + vSL_ + "; \r\n");
MQL4_source = MQL4_source.concat("double vTP=" + vTP_ + "; \r\n");
MQL4_source = MQL4_source.concat("double MagicNumber=" + MagicNumber + "; \r\n");
```

Figura 36. Método Java que pasa a fuente MQL4 los parámetros seleccionados por el trader

```
////////////////////////////////////
//////// Variables setting from JAVA screen to MQL4 //////////
//
int RSILEvel_to_Buy= 20;
int RSILEvel_to_Sell= 70;
int LotSize_buy= 0.01;
int LotSize_sell= 0.01;
double vSL=5;
double vTP=10;
```

Figura 37. Código MQL4 generado con AutoTrader conteniendo parámetros definidos por el trader

Comparación de la Solución

Las principales aportaciones en el universo del desarrollo de software para trading incluyen:

- Detectores de patrones repetitivos
- Generadores de macros y rutinas
- Desarrollos a medida de algoritmos de trading según especificaciones
- Desarrollo a medida de Expert Advisors
- Desarrollo a medida de Scripts e Indicadores Técnicos propios
- Intérpretes desde varios lenguajes a MQL4
- Interfaces entre lenguajes y MQL4
- Sistemas de menús nativos en MQL4 sobre terminal Metatrader, aunque de interfaz e interacción muy árida

Precisamente por la fuerte programabilidad de la plataforma Metatrader abundan excelentes desarrolladores sobre su lenguaje MQL, en los foros de las dos versiones de la plataforma, Metatrader4 y 5, se intercambia de manera muy fluida todo tipo de dudas, consejos, y ofertas de desarrollo a medida.

La oferta interactiva con interfaz gráfico moderno, amigable y experto no es una oferta disponible de manera regular, probablemente porque el mundo del trading acoge muy bien a traders expertos y desarrolladores expertos, pero la casi totalidad de desarrollos ofrecidos son a medida mediante generación de expert advisors o scripts en MQL4 nativo, cuyo ejecutable es transferido al comprador para instalación y ejecución en la propia plataforma.

La aplicación que se aporta en este trabajo es una contribución de investigación, es decir si bien su operativa es totalmente apta para el consumo, está más pensada para inspirar un camino de evolución de desarrollo del *algotrading* pasivo y asistido con código LOWCODE / NOCODE, donde el *trader* apenas necesita interaccionar con su plataforma Metatrader, para poder enfocar toda su atención y tiempo al análisis del mercado y sus tendencias.

La aplicación AutoTrader es un generador automático basado en NOCODE, es decir el usuario obtiene una aplicación que le guía por sistemas de menús, y que según las selecciones realizadas se produce un código fuente de manera totalmente opaca al usuario. Esta aplicación generada está realizada a la medida del trader, ya que está parametrizada por él mismo. Veamos el contexto actual de lenguajes de alto rendimiento low-code para entender su posicionamiento y crecimiento progresivo:

Si cogemos un listado de proveedores de LCAP (low code application platform) impresiona ver la abundancia de soluciones disponibles comercialmente o de manera abierta:

Listado de fabricantes de plataformas de desarrollo low-code

ACTICO	FileMaker	N-LAB Software	Skuid
AgilePoint	Fliplet	Novacura	snappii
Alpha Software	GeneXus	Novulo	Software AG
AppGyver	Graphite GTC	Olympe	Thinkwise
Appian	Grexx	OpenText	TIBCO Software
AppSheet	GW Apps	Oracle	TrackVia
Arrayworks	Haulmont	OrangeScape	Twenty57
Axway	i-exceed	OutSystems	UGROUND
Barium	iTiZZIMO	ParaSQL	Velneo
Betty Blocks	Joget	Pegasystems	WaveMaker
bpm'online	Kintone	Plains Mobile	WebRatio
Bubble	Kony	Plasma	WEMAIK
Bubblz	LANSA	PowWow Mobile	Workday
Caspio	MatsSoft	ProntoForms	Xiaoshouyi
Cherwell Software	Mendix	Quick Base	Xojo
CitizenDeveloper	michaels, ross & cole (mrc)	Robocoder	XOne
Covius Technology Solutions	MicroPact	Salesforce	
Dell Boomi	Microsoft	SAP	
EachScape	MicroStrategy	Scopeland Technology	
Enstar	Qalcwise	ServiceNow	
Faveod	Neptune Software	Servoy	
FHOSTER	Ninox Software	Simplicité Software	

Figura 37. Fabricantes de plataformas low-code

Una parte importante de este trabajo de selección de las mejores plataformas está socialmente encargado a los grandes analistas tales como Forrester Research, o Gartner. Si cogemos el cuadrante mágico de Gartner para plataformas low-code nos encontramos una densa oferta de plataformas de desarrollo, y casi todas proceden de casas de software de alta calidad tales como ServiceNow, Microsoft, Caspio, Oracle, etc.



Figura 38. Cuadrante mágico Gartner de plataformas low-code

La competencia es ya enorme en el campo de reducción de costes de desarrollo trasladando los desarrollos sin desarrollo (low code o no code) a los propios departamentos del negocio, mediante un paquete adaptable sin complejidad, y con algo de formación simplemente.

Gartner predice que "para el año 2023, más del 50 % de las empresas medianas a grandes adoptarán una plataforma de aplicaciones low-code como una de sus plataformas estratégicas de aplicaciones"

Por experiencia personal en desarrollo en varios campos muy distintos tales como aplicativos para finanzas del sector de fabricación del automóvil, sistemas de tráfico aéreo, sistemas de cálculo estándar para empresa PYME, y más, una conclusión es clara: la mejor medición de la bondad de introducción de paquetes software low-code con paradigma intencional es la productividad del desarrollo. Es decir, tomando un ejemplo concreto, real, vivido en primera persona en los tiempos tempranos de paquetes low-code en los años 90, mediante el paquete PRO-IV (de la compañía Mc Donnell Douglas Information Systems) ya entonces era posible desarrollar una aplicación de gestión de complejidad media para empresa, para casi cualquier departamento, incluyendo entrada de datos, almacenamiento de datos, conectores de flujos entre procesos, transferencia de datos entre etapas sucesivas del negocio (producción, logística, almacén, fábrica, etc) consiguiendo tiempos de desarrollo de unas 20 a 30 horas para obtener un producto de una calidad 6 a 7/10 y con total autonomía para el usuario.

Muy posiblemente el mejor mecanismo de medición de la posibilidad de triunfar por el camino de estos productos es un sencillo algoritmo donde calculamos el coste del desarrollo a medida de una aplicación más su coste de soporte a cinco años (TCO 5Y) y lo comparamos con el coste de desarrollar y mantener la aplicación equivalente en no-code más su soporte. Claramente no hay posible punto de comparación entre ambos costes. Eso es lo que justifica que en dos o tres años al menos la mitad de las empresas estarán implantando de manera habitual sus aplicaciones low o no code.

AutoTrader© puede considerarse un compilador automático MQL4 desde Java. La implantación de la lógica difusa acerca la lingüística humana a la matemática clásica mediante la aplicación de reglas de inferencia que mapean conceptos abstractos humanos a números reales. La síntesis de entradas, salidas, y reglas de inferencia se obtiene sin conocimiento alguno de programación ya que es AutoTrader© el único encargado de interactuar con el *trader* en lenguaje de menús cerrados, para trasladar sus elecciones a la codificación de la estrategia de inversión.

En el camino de desarrollo a futuro del *algotrading* pasivo asistido, la aplicación aportada puede ser una semilla desde la que crezcan iniciativas de mejora y perfeccionamiento de la captación de intenciones del individuo que se dedica a hacer inversiones de manera desatendida, abundando más y más en la mejor interpretación de sus intenciones; esto se puede lograr mediante tecnología del lenguaje interpretando opciones elegidas por el usuario y adelantándose a sus pretensiones. Un ejemplo sería el usuario que en el menú de selección de los niveles de entrada del indicador RSI en la aplicación AutoTrader© elije desde la primera entrada valores muy conservadores del mismo, lo

cual ya determina una manera de interpretar el trading, de la cual se puede inferir casi automáticamente los volúmenes probables de inversión que el usuario querrá seleccionar, y la aplicación puede ofrecer automáticamente valores de las salidas difusas por inferencia de las entradas, dando siempre la opción a ser editadas y alteradas por el usuario. En conclusión, la evolución debería probablemente ir por el camino de la pasividad casi total del usuario al interactuar con la plataforma, y del aprendizaje de la plataforma en cada nueva interacción con el usuario. Estos términos nos llevan al camino de la inteligencia artificial sintética, generativa, donde la generación de código automático es una respuesta inmediata y muy próxima al pensamiento e intenciones del usuario.

Conclusiones y Trabajos Futuros

Conclusiones y Trabajos Futuros

El campo del desarrollo software y el campo de la ingeniería de software estar acaparando una atención sin precedentes en el mundo en general debido a las mejoras que el software a medida puede traer casi a cualquier campo operativo.

Hasta este momento el mayor logro obtenido ha sido muy probablemente la generalización del uso de software para mejorar la llegada de muchos beneficios tecnológicos a la sociedad, a periferias más lejanas. La socialización del software a través de aplicaciones, Apps, Widgets, desarrollos a medida, etc. Es un proceso que ha sido largo y nunca de duración inferior a 50 años. Hoy día han sido los teléfonos móviles y su portabilidad de Apps una de las principales plataformas de socialización del software y sus beneficiosos para el día a día de casi cualquier ciudadano. Ejemplos de aplicaciones de alto beneficio para la sociedad incluyen:

- . Interacción digital con la administración de salud
- . Interacción digital con la banca
- . Interacción digital con los colegios y universidades
- . Interacción digital con las cosas (electrodomésticos, dispositivos del hogar)

En el contexto de este trabajo tiene especial relevancia el progreso alcanzado en trading móvil; la realización de trading desde la pantalla de teléfono móvil. Esta variedad de training permite la continua negociación de activos en bolsa desde cualquier lugar donde se encuentre el operador de bolsa, siempre que tenga conexión de datos. No es el teléfono

móvil una plataforma cómoda y completa para desarrollo de un entorno de trading como lo es de manera muy ventajosa un ordenador personal fijo o portátil, pero desde un punto de vista estrictamente funcional la aplicación de trading adaptada al teléfono es exactamente la misma que la desarrollada originalmente para ordenador. Esto implica que el desarrollo de un robot o sistema experto tal como ha sido explicado en este trabajo sigue estando vigente y operativo en la versión móvil de la plataforma de trading, de manera que la aplicación AutoTrader© ejecutada en el ordenador sigue siendo operativa en el teléfono móvil ya que esta instancia es tan solo una ejecución de la cuenta de usuario de Metatrader4 pero lanzada desde un sistema operativo distinto, en este caso el teléfono móvil. Por tanto, todo lo desarrollado en este trabajo es total aplicación en portabilidad allá donde se encuentre el usuario de algo trading difuso, con su teléfono móvil y su sistema experto generado desde AutoTrader©.

El desarrollo de este generador automático de código fuente entre lenguajes Java y MQL4 en sus formatos léxicos, semánticos y sintácticos, y su fase previa de análisis de arquitectura han dado lugar a adentrarse en una perspectiva creativa nueva, sobre la enorme potencialidad de utilización de la generación automática de código fuente que pueda ser empleado de manera recurrente, aunque el código fuente vaya cambiando. Con tan solo asegurarse de que el sistema de menús ofrecido por la aplicación está haciendo las preguntas correctas al usuario de la aplicación es suficiente para garantizar el adecuado trabajo del programa.

Obviamente esta versión 1.0 de AutoTrader© no es un intérprete completo y profesional si bien es totalmente operativo para día a día de trading difuso, pero es interesante apreciar que el nivel de progreso logrado es superior a lo esperado al principio del análisis. La mayor parte de opcionales de estrategias de trading están incluidas en este compilador automático presentado en este trabajo. Es de especial interés mencionar la sencillez de desarrollo software que ofrecen ambos ambientes: Java mediante IDE Eclipse, así como MQL4. Los dos lenguajes de programación son bastante amigables, potentes, están bien documentados, y son flexibles en el desarrollo tanto de aplicaciones orientadas a objetos como aplicaciones menos sofisticadas orientadas a arquitecturas de desarrollo más bien secuenciales sencillas.

Por la forma en que ha sido escrito este intérprete sería muy sencillo añadir más capacidad interpretativa de las intenciones del trader, o cualquier tipo de especificidades propias de una operativa basada en estrategias complejas, ya que el lenguaje Java mediante sus librerías de cálculo, de gráficos y menús hace el trabajo del desarrollador francamente amigable y sencillo. Es decir, el nivel de reaprovechamiento del código que se ofrece en esta primera versión es completo de cara a una versión evolucionada del intérprete de estrategias difusas. El código está diseñado modularmente y con una recursividad muy flexible.

Pensando en una evolución hacia una versión superior, añadir nuevas opciones sería tan sencillo como añadir métodos individuales por cada nueva opción de estrategia o añadir nuevas bifurcaciones condicionales sintácticas -del tipo IF THEN -.

Este intérprete de intenciones de estrategias de inversión es un buen producto para generar código MQL4 en gran cantidad desde el sistema de interrogación al *trader* por medio de menús cerrados desplegables.

Prestando atención y dedicación al mercado de aplicaciones desarrolladas a medida sobre MQL4 se capta rápidamente el nivel de complejidad tan elevado al que se ha llegado en el sector del trading, como respuesta a la llamada generalizada mundial que hay en el afán por enriquecerse de manera rápida comprando y vendiendo activos de bolsa desde una pantalla de ordenador. Aunque ya se ha mencionado anteriormente en este trabajo, merece la pena volver sobre el hecho de que más de un 70% de los usuarios de trading pierden el dinero que depositaron en sus cuentas de *brokers* tras ejecutar trading, lo cual invitó a la Union Europea a la imposición de las medidas MIFID (Markets in Financial Instruments Directive; MiFID II, Directiva 2014/65, que se basa en la mejora de las reglas ya adoptadas por MiFID y su entrada en vigor es el 3 de enero de 2018.), con el objeto de aumentar el control y transparencia del inversionismo, con los siguientes objetivos (Banco Cooperativo Español. Website):

- Reforzar la protección y transparencia al inversor y las normas de conducta, así como las condiciones de competencia en la negociación y liquidación de instrumentos financieros.

- Introducir un marco regulatorio común que unifique los servicios financieros en los países de la Unión Europea
- Aumentar la transparencia y la supervisión de los mercados financieros y velar por el correcto funcionamiento de los mercados en la formación de precios.
- Regular el comportamiento de las entidades financieras, incrementando la competencia entre las mismas.

Al respecto la misma fuente informa de cómo se adapta el producto de inversión al usuario mediante la aplicación de MIFID:

Test de idoneidad

Evalúa los conocimientos y experiencia del cliente, sus objetivos de inversión y su capacidad financiera. Este test se realizará, en todo caso, cuando exista asesoramiento en materia de inversión (mediante contrato).

No obstante lo anterior para determinadas contrataciones de instrumentos financieros y de manera puntual, Caja Rural también evaluará los conocimientos y experiencia del cliente, sus objetivos de inversión y su capacidad financiera.

Test de conveniencia

Evalúa los conocimientos y experiencia del cliente sobre el instrumento que se dispone a contratar. Aplica cuando se trata de un producto complejo y la iniciativa de contratación reside en el cliente o cuando, en el caso de los productos no complejos, el cliente indica que la iniciativa de la operación es de Caja Rural.

Este riesgo tan elevado de pérdidas del dinero depositado en una cuenta de trading es probablemente el hecho más significativo que hace que el mercado de desarrollo de *expert advisors* se haya acelerado tanto, poniendo en manos de informáticos la interpretación de la evolución del precio de los activos mediante complejas herramientas software, en un intento de los *traders* de ser asistidos a la hora de tomar la decisión de entrar a operar.

Precisamente este hecho es el que animó la idea que dio lugar a este trabajo, aportando - como trabajo de investigación de este máster en ingeniería de software- una pequeña mejora y un posible camino de futuras investigaciones.

En relación con el posible futuro de los generadores de código difuso para estrategias de inversión tenemos una fuerte opinión positiva a favor de una evolución por el camino de desarrollos que faciliten las siguientes evoluciones:

- . Generadores automáticos por interpretación más y más avanzada de las intenciones del *trader*, sin desarrollo por usuarios sino tan solo por ingenieros informáticos, salvo excepciones de *traders* muy maduros en herramientas informáticas

- . Convergencia del desarrollo con los lenguajes de desarrollo NOCODE; esto propicia el desarrollo software paradigmático a medida de cada aplicación de mercado sin *inteligencia informática* del usuario

- . Desarrollo progresivamente más implicado en inteligencia artificial, donde la aplicación generada automáticamente aprenda en tiempo real del estado del mercado, y pueda realizar recomendaciones al *trader* para cambiar patrones de su estrategia o parametrizaciones de esta

- . Avances en tecnología de lenguaje que permitan una interacción del *trader* con la aplicación de desarrollo automático por la vía oral, haciendo a la aplicación progresivamente *más humana* en su interpretación del lenguaje de *trader*, activándose mas y mas en la captación de sus intenciones, incluso a través de la lectura del tono y timbre de su voz.

Como objetivo concreto a futuro parece altamente interesante y prometedor enfocar la atención en la investigación y el desarrollo enfocados en dos frentes:

. Aprendizaje progresivo por parte del generador automático, y autocorrección de su propio código fuente, en base a los errores y desviaciones captados entre las intenciones y las realidades obtenidas, medido en ganancias o pérdidas

. Aprendizaje progresivo de las intenciones del usuario mediante iteraciones sucesivas de análisis de sus requerimientos expresados en versiones anteriores del generador.

Estos campos de investigación y desarrollo aparecen como una propuesta muy atractiva para afrontar un programa de doctorado en ingeniería de software e inteligencia artificial.

Referencias y Bibliografía

Referencias y Bibliografía

Andersen, T., Bollerslev, T. 1997. Intraday periodicity and volatility persistence in financial markets.

Blair, B. Interview with Lotfi Zadeh (December 1999) in "[Famous People: Then and Now](#) [Lotfi Zadeh, Creator of Fuzzy Logic \(1921–\)](#)" *Azerbaijan International* (7.4) (Winter 1999)

Boyacioglu, M., Derya, A. 2010. An Adaptive Network-Based Fuzzy Inference System (ANFIS) for the prediction of stock market return: The case of the Istanbul Stock Exchange.

Brabazon, A., O'Neill, M. 2006. *Biologically Inspired Algorithms for Financial Modelling*. Springer.

Chang, J.R., Wei, L.Y. Cheng, C-H. 2011. A Hybrid ANFIS model based on ar and volatility for TAIEX forecasting. *Applied soft computing*, 11(1).

Chen, M.Y. 2013. A hybrid ANFIS model for business failure prediction utilizing particle swarm optimization and subtractive clustering. *Information Sciences*.

Choudhry, T. , McGroarty, F. , Ke Peng , Wang, S. 2012. High frequency exchange-rate prediction with an artificial neural network.

Dourra, H., Siy, P. 2002. Investment using technical analysis and fuzzy logic. *Fuzzy sets and systems* vol127.

Elliott, R. 1939. The Wave Principle. *Financial World* magazine.

Instituto Europeo de Posgrado. Conócenos Noticias". <https://www.iep.edu.es/las-5-fases-de-la-piramide-de-maslow/>

Gradojevic, N. 2007. Non-linear, hybrid exchange rate modeling and trading profitability in the foreign exchange market..

Gradojevic, N., Gençay, R. Fuzzy logic, trading uncertainty and technical trading. 2013.

Heal, G., Millner, A. 2014. Reflections uncertainty and decision making in climate change economics.

Holmberg, U., Lönnbark, C., Lundström, C. 2013. Assessing the profitability of intraday opening range breakout. Strategies.

Jarusek, R., Volna, E. Kotyrba, M. 2021. FOREX rate prediction improved by Elliott waves patterns based on neural networks. University of Ostrava, Department of Informatics and Computers, 30. dubna 22, 70103, Ostrava, Czech Republic.

Kablan, A., Wing Lon Ng. 2011. 1921. Intraday high-frequency FX trading with adaptive neuro-fuzzy inference systems.

Knight, F.H. Risk, uncertainty, and Profit.

Krollner, B., Vanstone, B., Finnie, G. 2010. Financial time series forecasting with machine learning techniques: a survey.

Lee, C.H., Jo, G.S. 1999. Expert system for predicting stock market timing using a candlestick chart.

Mamdani, E.H. 1974. Applications of Fuzzy Algorithms for Simple Dynamic Plant.

Martin, E. 1982. La teoría de los conjuntos borrosos y la toma de decisiones. Revista Española de financiación y contabilidad.

Mateos-Aparicio, P. & Rodríguez-Moreno, A. 2019. The Impact of Studying Brain Plasticity, Front. Cell. Neurosci. 13:66. doi:[10.3389/fncel.2019.00066](https://doi.org/10.3389/fncel.2019.00066)

Matviychuk, A. 2006. Fuzzy logic Approach to Identification and Forecasting of Financial Time Series Using Elliot Wave Theory. Vol. XI, No. 2, No, p. 51-68 Fuzzy Economic Review.

Merton, R. 2002. On estimating the expected return on the market: An exploratory investigation. MIT.

Morta, R., Morta, S., Dadios, E. 2015. Modeling Market Data and Technical Indicators with Fuzzy Logic to develop a Trading Algorithm for an Individual Stock in the Philippines. 8th IEEE International Conference Humanoid, Nanotechnology, Information Technology. The Institute of Electrical and Electronics Engineers Inc. (IEEE). 9-12 December 2015. Philippines

Naranjo, R. 2018. Tesis doctoral, Modelado de sistemas de inversión mediante lógica borrosa como soporte a la toma de decisiones en mercados bursátiles. Universidad Complutense de Madrid

Naranjo, R., Arroyo, J., Santos, M. 2019. Fuzzy modeling of stock trading with fuzzy candlesticks. * Faculty of Computer Sciences, University Complutense of Madrid. February 2017. Naranjo., Santos, M. A fuzzy decision system for money investment in stock markets based on fuzzy candlesticks pattern recognition. May 2019.

Nelson, S.C., Katzenstein, P.J. 2014. Uncertainty, risk and the financial crisis of 2008.

Othman, S., Schneider, E., 2010. Decision Making Using Fuzzy Logic for Stock Trading Computer and Information Sciences. 978-1-4244-6716-711. IEEE.

Pardo, R. 2008. The Evaluation and Optimization of Trading Strategies. Vol 314. New Jersey: John Wiley and Sons.

Rechenthin, M., NickStreet, W. 2013. Using conditional probability to identify trends in intra-day high-frequency equity pricing.

Sanahuja, S. 2017. Sistemas de Control con Lógica Difusa: Métodos de Mamdani y de Takagi-Sugeno-Kang (TSK). Estancia en Prácticas y Proyecto Final de Grado.

Schulmeister, S. 2009. Profitability of technical stock trading: Has it moved from daily to intraday data.

Smith, A. 1794. [«Capítulo II»](#). *Investigacion de la naturaleza y causa de la riqueza de las naciones*. Tomo 1.

Shim, J.P., Warkentin, M., Coutney, F., Power D., Sharda, R. Carlson, C. 2002. Past, Present and Future of Decision support technology". *Decision support system*, vol.33.

Simutis, R. 2000. Fuzzy Logic Based Stock Trading System. Department of Informatics, Vilnius University.

Stankevicius, G., Simutis,R. 2000. Realization of Human Skills Based Break-Out trading strategies using Fuzzy Logic methods. Kaunas University of Technology. Aachen, Germany.

Sugeno, M. and Kang, G.T. 1988. Structure Identification of Fuzzy Model. *Fuzzy Sets and Systems*.

Tan, Z., Quek, C., & Cheng, P. Y. 2011. Stock trading with cycles: a financial application of ANFIS and reinforcement learning. *Expert Systems with Applications*.

Thakera, S., Nagorib, V. 2018. Analysis of Fuzzification Process in Fuzzy Expert System. FCAIT, GLS University, Ahmedabad 380006, India. GLS University, Ahmedabad 380006, India.

Troianoa, L., P.Kriplani, P. 2011. Supporting trading strategies by inverse fuzzy transform.

Tsai, C., Wang, S. 2009. Stock price forecasting by hybrid machine learning techniques.

Vanstone, B., Finnie, G. 2010. Enhancing stock market trading performance with ANNs.

Vella, V., Wing LonNg. 2014. Enhancing risk-adjusted performance of stock market intraday trading with Neuro-Fuzzy systems.

Vella, V., Ng Lon, Ng. 2014. A Dynamic Fuzzy Money Management Approach for Controlling the Intraday Risk-Adjusted Performance of AI Trading Algorithms.

Vella, V., Lon Ng, W. 2016. Improving risk-adjusted performance in high frequency trading using interval type-2 fuzzy logic. Centre for Computational Finance and Economic Agents (CCFEA). 0957-4174/© 2016 Elsevier Ltd

Wei, L-Y., Cheng, C.-H., Wu, H.-H. 2014. A hybrid ANFIS on n-period moving average model to forecast TAIEX stock. Applied soft computing.

Wu, D., Mendel, J.M. 2014. Designing practical Interval type-2 fuzzy logic systems made simple.

Zadeh, A. 1975. The concept of a linguistic variable and its application to approximate reasoning. Computer Sciences Division, Department of Electrical Engineering and Computer Sciences, and the Electronics Research Laboratory, University of California, Berkely, California 94720 USA

Zadeh, A. 1969. TOWARD A THEORY OF FUZZY SYSTEMS. Prepared by UNIVERSITY OF CALIFORNIA Berkeley, Calif. for NATIONAL AERONAUTICS AND SPACE ADMINISTRATION WASHINGTON, D. C.

Code Generation in Action. Jack Herrington. MANNING.

Mastering Regular Expressions. Jeffrey E.F. Friedl. O'REILLY.

Ejercicios de Programación en Java. F.M.Perez. Eduinnova.

Manual Básico de Java. ESIME-CULHUACAN.

Java Swing. M.Loy. O'REILLY. Swing, A Beginner's Guide. H.Schildt. McGrawHill.

Anexos

Anexo I: Código Fuente

```
////////////////////////////////////  
//  
//  
// Ignacio Fumanal Andres. Master de Investigacion en Ingenieria de Software.  
//  
// Generador de EA Interactivo de Trading para Metatrader4.  
//* Interfaz grafica y Generador de Comandos MQL4  
//*  
//* Generacion Atuomatica de Codigo para Trading Automatico con Metatrader  
//* TFM. ETSI Informatica. UNED  
//*  
//* (c)2022  
//*  
//  
//  
//Relacion de importaciones de librerias necesarias para la compilacion de todos  
//los fuentes de este desarrollo  
//  
import java.awt.Color;  
import java.awt.SystemColor;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.Font;  
//  
import java.io.FileOutputStream;  
import java.io.FileWriter;  
import java.io.IOException;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import javax.print.DocFlavor.STRING;  
import javax.swing.DefaultComboBoxModel;  
import javax.swing.JComboBox;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JRadioButton;  
import javax.swing.JButton;  
import javax.swing.JTextPane;  
//  
// Esta es la clase pública que encierra el código fuente completo.  
// Inicializamos variables que utilizaremos  
//  
public class FuzzyUI  
    {  
        String MQL4_source = "";  
private JFrame frame;
```

```

//
//
//Aqui iniciamos el principal metodo de llamada a todos los fuentes.
//
// Metodo MAIN()
//
//
public static void main(String[] args)
{
    try
        {
            FuzzyUI window = new FuzzyUI();
            window.frame.setVisible(true);
        }

    catch (Exception e)
        {
            e.printStackTrace();
        }

    finally
    {
        //
    }
}

//
//
//Metodo de llamada a la inicializacion
//
//
public FuzzyUI() throws Exception
{
    initialize();
}

//
//En este metodo se realiza la inicializacion de la capa grafica.
// Desde esta se llama a los demas metodos de insercion de datos
//La carga principal de desarrollo de este metodo es grafica a traves
//de la libreria SWING
//
//
//Metodo INITIALIZE()
//
public void initialize()
{
    // Construimos la pantalla de presentacion
    //
    frame = new JFrame();
    frame.getContentPane().setBackground(SystemColor.textHighlight);
    frame.setBounds(100, 100, 904, 625);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().setLayout(null);
    //
    // Construimos los titulos de cajas y captacion de Opciones Trading
    JLabel lbltitle = new JLabel("AutoTrader");
    lbltitle.setForeground(Color.CYAN);
    lbltitle.setFont(new Font("Tahoma", Font.BOLD, 39));
    lbltitle.setBounds(242, 12, 244, 32);
    frame.getContentPane().add(lbltitle);
}

```

```

JLabel lblSobrevendido = new JLabel("SOBRE VENDIDO");
lblSobrevendido.setForeground(Color.WHITE);
lblSobrevendido.setFont(new Font("Dialog", Font.BOLD, 20));
lblSobrevendido.setBounds(33, 161, 171, 27);
frame.getContentPane().add(lblSobrevendido);

JLabel lblSobrecomprado = new JLabel("SOBRE COMPRADO");
lblSobrecomprado.setForeground(Color.WHITE);
lblSobrecomprado.setFont(new Font("Dialog", Font.BOLD, 20));
lblSobrecomprado.setBounds(33, 201, 187, 27);
frame.getContentPane().add(lblSobrecomprado);

JComboBox buttonSobrecomprado = new JComboBox();
buttonSobrecomprado.setModel(new DefaultComboBoxModel(new String[] {"70", "75",
"80"}));
buttonSobrecomprado.setSelectedIndex(0);
buttonSobrecomprado.setToolTipText("Indicator index;");
buttonSobrecomprado.setMaximumRowCount(3);
buttonSobrecomprado.setBackground(Color.WHITE);
buttonSobrecomprado.setBounds(230, 205, 94, 27);
frame.getContentPane().add(buttonSobrecomprado);
int RSI_level_to_Buy = Integer.valueOf((String)
buttonSobrecomprado.getSelectedItem());

JComboBox buttonSobrevendido = new JComboBox();
buttonSobrevendido.setModel(new DefaultComboBoxModel(new String[] {"20", "25",
"30"}));
buttonSobrevendido.setSelectedIndex(0);
buttonSobrevendido.setToolTipText("BUY, SELL, All;");
buttonSobrevendido.setMaximumRowCount(3);
buttonSobrevendido.setBackground(Color.WHITE);
buttonSobrevendido.setBounds(227, 165, 94, 27);
frame.getContentPane().add(buttonSobrevendido);
int RSI_level_to_Sell = Integer.valueOf((String)
buttonSobrevendido.getSelectedItem());

JLabel lblActivo = new JLabel("Activo");
lblActivo.setForeground(Color.WHITE);
lblActivo.setFont(new Font("Dialog", Font.BOLD, 20));
lblActivo.setBounds(33, 239, 146, 27);
frame.getContentPane().add(lblActivo);

JComboBox buttonActivo = new JComboBox();
buttonActivo.setToolTipText("Whether generated commissions;");
buttonActivo.setModel(new DefaultComboBoxModel(new String[] {"EURUSD", "USDJPY",
"IBEX35", "DAX"}));
buttonActivo.setSelectedIndex(0);
buttonActivo.setMaximumRowCount(3);
buttonActivo.setBackground(Color.WHITE);
buttonActivo.setBounds(230, 243, 94, 27);
frame.getContentPane().add(buttonActivo);
String Symbol = (String) buttonActivo.getSelectedItem();

JLabel lblcopyright = new JLabel("(c)2020/22");
lblcopyright.setForeground(Color.CYAN);
lblcopyright.setFont(new Font("Dialog", Font.BOLD, 15));
lblcopyright.setBounds(482, 24, 135, 27);
frame.getContentPane().add(lblcopyright);

```

```

JButton PushButtonGenerateMQL4 = new JButton("Generar Expert Advisor MT4;");
PushButtonGenerateMQL4.setForeground(new Color(192, 192, 192));
PushButtonGenerateMQL4.setBackground(new Color(255, 0, 255));
PushButtonGenerateMQL4.setFont(new Font("Ebrima", Font.BOLD, 16));
PushButtonGenerateMQL4.setBounds(499, 525, 320, 39);
frame.getContentPane().add(PushButtonGenerateMQL4);

JLabel lblEntradas = new JLabel("Entradas a la Estrategia");
lblEntradas.setForeground(Color.YELLOW);
lblEntradas.setFont(new Font("Dialog", Font.BOLD, 22));
lblEntradas.setBounds(23, 118, 298, 32);
frame.getContentPane().add(lblEntradas);

JLabel lblTechnicalIndicator = new JLabel("Indicador Técnico");
lblTechnicalIndicator.setForeground(Color.WHITE);
lblTechnicalIndicator.setFont(new Font("Dialog", Font.BOLD, 20));
lblTechnicalIndicator.setBounds(33, 277, 230, 32);
frame.getContentPane().add(lblTechnicalIndicator);

JComboBox buttonTechIndicator = new JComboBox();
buttonTechIndicator.setToolTipText("Indicador de Trading;");
buttonTechIndicator.setModel(new DefaultComboBoxModel(new String[] {"SMA", "WMA",
"ADX", "RSI", "Ichimoku", "Heiken Ashi"}));
buttonTechIndicator.setSelectedIndex(0);
buttonTechIndicator.setMaximumRowCount(6);
buttonTechIndicator.setBackground(Color.WHITE);
buttonTechIndicator.setBounds(255, 281, 72, 27);
frame.getContentPane().add(buttonTechIndicator);
String TecInd = (String) buttonTechIndicator.getSelectedItem();

JLabel lblPeriodoIndicador = new JLabel("Periodo De Indicador");
lblPeriodoIndicador.setForeground(Color.WHITE);
lblPeriodoIndicador.setFont(new Font("Dialog", Font.BOLD, 20));
lblPeriodoIndicador.setBounds(33, 324, 250, 32);
frame.getContentPane().add(lblPeriodoIndicador);

JComboBox buttonPeriodoindicador = new JComboBox();
buttonPeriodoindicador.setToolTipText("Vela Japonesa de toma de valor;");
buttonPeriodoindicador.setModel(new DefaultComboBoxModel(new String[] {"t-1", "t-
2", "t-3", "t-4"}));
buttonPeriodoindicador.setSelectedIndex(0);
buttonPeriodoindicador.setMaximumRowCount(4);
buttonPeriodoindicador.setBackground(Color.WHITE);
buttonPeriodoindicador.setBounds(298, 332, 63, 27);
frame.getContentPane().add(buttonPeriodoindicador);
String vela = (String) buttonPeriodoindicador.getSelectedItem();

JLabel lbMercado = new JLabel("Mercado\r\n");
lbMercado.setForeground(Color.YELLOW);
lbMercado.setFont(new Font("Dialog", Font.BOLD, 24));
lbMercado.setBounds(472, 118, 298, 32);
frame.getContentPane().add(lbMercado);

JLabel lblHoraApertura = new JLabel("Hora Apertura de Operativa");
lblHoraApertura.setForeground(Color.WHITE);
lblHoraApertura.setFont(new Font("Dialog", Font.BOLD, 20));
lblHoraApertura.setBounds(472, 161, 285, 27);

```

```

frame.getContentPane().add(lblHoraApertura);

JLabel lblHoraCierre = new JLabel("Hora Cierre de Operativa");
lblHoraCierre.setForeground(Color.WHYTE);
lblHoraCierre.setFont(new Font("Dialog", Font.BOLD, 20));
lblHoraCierre.setBounds(472, 203, 260, 27);
frame.getContentPane().add(lblHoraCierre);

JLabel lblControlganancias = new JLabel("Control de Ganancias");
lblControlganancias.setForeground(Color.YELLOW);
lblControlganancias.setFont(new Font("Dialog", Font.BOLD, 22));
lblControlganancias.setBounds(23, 379, 298, 32);
frame.getContentPane().add(lblControlganancias);

JLabel lblLimiteIngresoDiario = new JLabel("Limite Ingreso Diario");
lblLimiteIngresoDiario.setForeground(Color.WHYTE);
lblLimiteIngresoDiario.setFont(new Font("Dialog", Font.BOLD, 20));
lblLimiteIngresoDiario.setBounds(33, 422, 213, 27);
frame.getContentPane().add(lblLimiteIngresoDiario);

JComboBox buttonIngresodiario = new JComboBox();
buttonIngresodiario.setToolTipText("BUY, SELL, All;");
buttonIngresodiario.setModel(new DefaultComboBoxModel(new String[] {"100", "200",
"300", "400", "500", "1000"}));
buttonIngresodiario.setSelectedIndex(0);
buttonIngresodiario.setMaximumRowCount(6);
buttonIngresodiario.setBackground(Color.WHYTE);
buttonIngresodiario.setBounds(259, 426, 94, 27);
frame.getContentPane().add(buttonIngresodiario);

JLabel lblPosicionesPermitidas = new JLabel("Posiciones Permitidas");
lblPosicionesPermitidas.setForeground(Color.WHYTE);
lblPosicionesPermitidas.setFont(new Font("Dialog", Font.BOLD, 20));
lblPosicionesPermitidas.setBounds(472, 254, 260, 27);
frame.getContentPane().add(lblPosicionesPermitidas);

JComboBox buttonPosicionespermitidas = new JComboBox();
buttonPosicionespermitidas.setToolTipText("BUY, SELL, All;");
buttonPosicionespermitidas.setModel(new DefaultComboBoxModel(new String[]
{"LARGO", "CORTO", "HEDGING"}));
buttonPosicionespermitidas.setSelectedIndex(0);
buttonPosicionespermitidas.setMaximumRowCount(3);
buttonPosicionespermitidas.setBackground(Color.WHYTE);
buttonPosicionespermitidas.setBounds(767, 258, 94, 27);
frame.getContentPane().add(buttonPosicionespermitidas);
String tipoPosicion = (String) buttonPosicionespermitidas.getSelectedItem();

JComboBox buttonHoraapertura = new JComboBox();
buttonHoraapertura.setToolTipText("BUY, SELL, All");
buttonHoraapertura.setModel(new DefaultComboBoxModel(new String[] {"9", "10"}));
buttonHoraapertura.setSelectedIndex(0);
buttonHoraapertura.setMaximumRowCount(2);
buttonHoraapertura.setBackground(Color.WHYTE);
buttonHoraapertura.setBounds(767, 161, 42, 27);
frame.getContentPane().add(buttonHoraapertura);
int tradeFrom = Integer.valueOf((String) buttonHoraapertura.getSelectedItem());

```

```

JComboBox buttonHoracierre = new JComboBox();
buttonHoracierre.setToolTipText("BUY, SELL, All");
buttonHoracierre.setModel(new DefaultComboBoxModel(new String[] {"4", "5"}));
buttonHoracierre.setSelectedIndex(0);
buttonHoracierre.setMaximumRowCount(2);
buttonHoracierre.setBackground(Color.WHITE);
buttonHoracierre.setBounds(767, 205, 42, 27);
frame.getContentPane().add(buttonHoracierre);
int tradeTill = Integer.valueOf((String) buttonHoraapertura.getSelectedItem());

JLabel lblEURO = new JLabel("€");
lblEURO.setForeground(Color.YELLOW);
lblEURO.setFont(new Font("Dialog", Font.BOLD, 20));
lblEURO.setBounds(363, 422, 29, 32);
frame.getContentPane().add(lblEURO);

JLabel lblVolumenDeCompra = new JLabel("Volumen de Compra");
lblVolumenDeCompra.setForeground(Color.YELLOW);
lblVolumenDeCompra.setFont(new Font("Dialog", Font.BOLD, 22));
lblVolumenDeCompra.setBounds(472, 324, 244, 32);
frame.getContentPane().add(lblVolumenDeCompra);

JLabel lblVolumenDeVenta = new JLabel("Volumen de Venta");
lblVolumenDeVenta.setForeground(Color.YELLOW);
lblVolumenDeVenta.setFont(new Font("Dialog", Font.BOLD, 22));
lblVolumenDeVenta.setBounds(472, 379, 244, 32);
frame.getContentPane().add(lblVolumenDeVenta);

JComboBox buttonVolumencompra = new JComboBox();
buttonVolumencompra.setToolTipText("");
buttonVolumencompra.setModel(new DefaultComboBoxModel(new String[] {"0.01",
"0.02", "0.1", "0.2", "1"}));
buttonVolumencompra.setSelectedIndex(0);
buttonVolumencompra.setMaximumRowCount(5);
buttonVolumencompra.setBackground(Color.WHITE);
buttonVolumencompra.setBounds(715, 324, 72, 27);
frame.getContentPane().add(buttonVolumencompra);
float LotSize_buy = Float.valueOf((String)
buttonVolumencompra.getSelectedItem());

JComboBox buttonVolumenventa = new JComboBox();
buttonVolumenventa.setToolTipText("");
buttonVolumenventa.setModel(new DefaultComboBoxModel(new String[] {"0.01",
"0.02", "0.1", "0.2", "1"}));
buttonVolumenventa.setSelectedIndex(0);
buttonVolumenventa.setMaximumRowCount(5);
buttonVolumenventa.setBackground(Color.WHITE);
buttonVolumenventa.setBounds(715, 379, 72, 27);
frame.getContentPane().add(buttonVolumenventa);
float LotSize_sell = Float.valueOf((String)
buttonVolumenventa.getSelectedItem());

JLabel lblAm = new JLabel("am");
lblAm.setForeground(Color.YELLOW);
lblAm.setFont(new Font("Dialog", Font.BOLD, 16));
lblAm.setBounds(816, 161, 72, 32);
frame.getContentPane().add(lblAm);

```

```

JLabel lblPm = new JLabel("pm");
lblPm.setForeground(Color.YELLOW);
lblPm.setFont(new Font("Dialog", Font.BOLD, 16));
lblPm.setBounds(816, 201, 72, 32);
frame.getContentPane().add(lblPm);

JLabel lblMagicNumber = new JLabel("MagicNumber");
lblMagicNumber.setForeground(Color.YELLOW);
lblMagicNumber.setFont(new Font("Dialog", Font.BOLD, 22));
lblMagicNumber.setBounds(472, 432, 244, 32);
frame.getContentPane().add(lblMagicNumber);

JComboBox MagicNumber = new JComboBox();
MagicNumber.setToolTipText("");
MagicNumber.setModel(new DefaultComboBoxModel(new String[] {"1234", "5648",
"6208"}));
MagicNumber.setSelectedIndex(0);
MagicNumber.setMaximumRowCount(3);
MagicNumber.setBackground(Color.WHITE);
MagicNumber.setBounds(726, 441, 72, 27);
frame.getContentPane().add(MagicNumber);

JLabel lblPipsDeGanancia = new JLabel("Pips de Ganancia");
lblPipsDeGanancia.setForeground(Color.YELLOW);
lblPipsDeGanancia.setFont(new Font("Dialog", Font.BOLD, 22));
lblPipsDeGanancia.setBounds(26, 469, 213, 32);
frame.getContentPane().add(lblPipsDeGanancia);

JLabel lblPipsDePrdida = new JLabel("Pips de Pérdida");
lblPipsDePrdida.setForeground(Color.YELLOW);
lblPipsDePrdida.setFont(new Font("Dialog", Font.BOLD, 22));
lblPipsDePrdida.setBounds(26, 512, 197, 32);
frame.getContentPane().add(lblPipsDePrdida);

JComboBox vTP = new JComboBox();
vTP.setToolTipText("BUY, SELL, All;");
vTP.setModel(new DefaultComboBoxModel(new String[] {"10", "15", "20"}));
vTP.setSelectedIndex(0);
vTP.setMaximumRowCount(3);
vTP.setBackground(Color.WHITE);
vTP.setBounds(255, 474, 42, 27);
frame.getContentPane().add(vTP);

JComboBox vSL = new JComboBox();
vSL.setToolTipText("BUY, SELL, All;");
vSL.setModel(new DefaultComboBoxModel(new String[] {"5", "10", "15"}));
vSL.setSelectedIndex(0);
vSL.setMaximumRowCount(3);
vSL.setBackground(Color.WHITE);
vSL.setBounds(259, 517, 42, 27);
frame.getContentPane().add(vSL);
//
// Generacion de la estructura fija de codigo fuente MQL4 para trading automatico
//
PushBtonGenerateMQL4.addActionListener(new ActionListener()
{
    public void actionPerformed(ActionEvent e)
    {

```



```

MQL4_source = MQL4_source.concat("int
ticket_sell_2nd=0;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           _buy_budget =
1;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           _sell_budget =
1;`\\r\\n`");
MQL4_source = MQL4_source.concat("double
tp_Sell;`\\r\\n`");
MQL4_source = MQL4_source.concat("double
tp_Sell10;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           sl_Sell;" +
"\\r\\n");
MQL4_source = MQL4_source.concat("double
tp_sellStop;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           sl_sellStop;
`\\r\\n`");
MQL4_source = MQL4_source.concat("double
tp_buyStop;`\\r\\n`");
MQL4_source = MQL4_source.concat("double
sl_buyStop;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           tp_Buy;`\\r\\n`");
MQL4_source = MQL4_source.concat("double
tp_Buy10;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           sl_Buy;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool
played_order_sound=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool           Ans1=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool           Ans2=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool           Ans3=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool           Ans4=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("datetime
now_time=Time[0];`\\r\\n`");
MQL4_source = MQL4_source.concat("string           name =
`BOTON`;`\\r\\n`");
MQL4_source = MQL4_source.concat("int
_typeArrow=181;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           _priceArrow =
0;`\\r\\n`");
MQL4_source = MQL4_source.concat("double           _yAxis=Bid;`\\r\\n`");
MQL4_source = MQL4_source.concat("string           _arrowColor =
clrAliceBlue;`\\r\\n`");
MQL4_source = MQL4_source.concat("int
_filehandle=0;`\\r\\n`");
MQL4_source = MQL4_source.concat("string
_subfolder=`Analytics`;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           counter=0;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           ticket = 0;`\\r\\n`");
MQL4_source = MQL4_source.concat("string
_ordercomment=StringConcatenate(`ROBOT Order `,Symbol());`\\r\\n`");
MQL4_source = MQL4_source.concat("bool
_trail_buy=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool
_trail_sell=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("bool
_StopbyError=false;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           magic =
1909;`\\r\\n`");
MQL4_source = MQL4_source.concat("int           period;`\\r\\n`");

```

```

        MQL4_source = MQL4_source.concat("bool
ticketclosed=false;`r\n");
        MQL4_source = MQL4_source.concat("double
ticket_buyStop;`r\n");
        MQL4_source = MQL4_source.concat("double
ticket_sellStop;`r\n");
        MQL4_source = MQL4_source.concat("double
open_ask;`r\n");
        MQL4_source = MQL4_source.concat("double
open_bid;`r\n");
        MQL4_source = MQL4_source.concat("double
Pips=0;`r\n");
        MQL4_source = MQL4_source.concat("double
STP=0;`r\n");
        MQL4_source = MQL4_source.concat("double
_profit=0;`r\n");
        MQL4_source = MQL4_source.concat("double
open=0;`r\n");
        MQL4_source = MQL4_source.concat("double
stop=0;`r\n");
        MQL4_source = MQL4_source.concat("long
Volume_log;`r\n");
        MQL4_source = MQL4_source.concat("string
Comment_log;`r\n");
        MQL4_source = MQL4_source.concat("string
result_log;`r\n");
        MQL4_source = MQL4_source.concat("string
optype_log;`r\n");
        MQL4_source = MQL4_source.concat("double
lowest_low;`r\n");
        MQL4_source = MQL4_source.concat("double
highest_high;`r\n");
        MQL4_source = MQL4_source.concat("bool
explore_order=true;`r\n");
        MQL4_source = MQL4_source.concat("double
Spread=NormalizeDouble((Ask-Bid)/Point(),1);`r\n");
        MQL4_source = MQL4_source.concat("double
retrace_up;`r\n");
        MQL4_source = MQL4_source.concat("double
retrace_down;`r\n");
        MQL4_source = MQL4_source.concat("double
bandwidth;`r\n");
        MQL4_source = MQL4_source.concat("double
daily_revenue=0;`r\n");
        MQL4_source = MQL4_source.concat("datetime
_today;`r\n");
        MQL4_source = MQL4_source.concat("datetime
date_op;`r\n");
        // date of closing operation (BUY or SELL)
        MQL4_source = MQL4_source.concat("datetime
today_ops;`r\n");
        // date of today
        MQL4_source = MQL4_source.concat("bool
working_hours=false;`r\n");
        MQL4_source = MQL4_source.concat("bool
revenue_recorded_flag=true;`r\n");
        MQL4_source = MQL4_source.concat("string
_strategy;`r\n");
        MQL4_source = MQL4_source.concat("double
_drawdown_pct;`r\n");
        MQL4_source = MQL4_source.concat("double
open_SL;`r\n");
        MQL4_source = MQL4_source.concat("double
open_TP;`r\n");
        MQL4_source = MQL4_source.concat("int
n;`r\n");

        // button vars
        MQL4_source = MQL4_source.concat("input int
_tp=20;`r\n");
        MQL4_source = MQL4_source.concat("input int
_tp10=10;`r\n");
        //
        //
        // Captacion de las opciones de Trading que el trader ha
seleccionado
        //
        int RSI_level_to_Sell = Integer.valueOf((String)
buttonSobrevendido.getSelectedItemAt());
        int RSI_level_to_Buy = Integer.valueOf((String)
buttonSobrecomprado.getSelectedItemAt());

```

```

        float LotSize_buy = Float.valueOf((String)
buttonVolumencompra.getSelectedItemAt());
        String Symbol = (String) buttonActivo.getSelectedItemAt();
        String vela = (String) buttonPeriodoindicador.getSelectedItemAt();
        int ingresoDaily = Integer.valueOf((String)
buttonIngresodiario.getSelectedItemAt());
        String tipoPosicion = (String)
buttonPosicionespermitidas.getSelectedItemAt();
        int tradeFrom = Integer.valueOf((String)
buttonHoraapertura.getSelectedItemAt());
        int tradeTill = Integer.valueOf((String)
buttonHoraapertura.getSelectedItemAt());
        int vSL_ = Integer.valueOf((String) vSL.getSelectedItemAt());
        int vTP_ = Integer.valueOf((String) vTP.getSelectedItemAt());

        //
        MQL4_source =
MQL4_source.concat("////////////////////////////////////
\r\n");
        MQL4_source = MQL4_source.concat("//////// Variables setting from
JAVA screen to MQL4 ////////// \r\n");
        MQL4_source = MQL4_source.concat("// \r\n");
        MQL4_source = MQL4_source.concat("int          RSILEvel_to_Buy= " +
RSI_level_to_Sell + "; \r\n");
        MQL4_source = MQL4_source.concat("int          RSILEvel_to_Sell= " +
RSI_level_to_Buy + "; \r\n");
        MQL4_source = MQL4_source.concat("int          LotSize_buy= "+
LotSize_buy + "; \r\n");
        MQL4_source = MQL4_source.concat("int          LotSize_sell= "+
LotSize_sell + "; \r\n");
        MQL4_source = MQL4_source.concat("double        vSL=" + vSL_ +
"; \r\n");
        MQL4_source = MQL4_source.concat("double        vTP=" + vTP_ +
"; \r\n");
        MQL4_source = MQL4_source.concat("double        MagicNumber=" +
MagicNumber + "; \r\n");
        MQL4_source = MQL4_source.concat("// \r\n");
        MQL4_source = MQL4_source.concat("// \r\n");
        MQL4_source = MQL4_source.concat("// \r\n");
        //
        //
        //
        //////////////////////////////////////
//          button f(x)
        MQL4_source = MQL4_source.concat("int OnInit(); \r\n");
        MQL4_source = MQL4_source.concat("  {; \r\n");
        //ObjectsDeleteAll();
        MQL4_source = MQL4_source.concat("          Sleep(20); \r\n");
        MQL4_source = MQL4_source.concat("          buy_ongoing =
false; \r\n");
        MQL4_source = MQL4_source.concat("          sell_ongoing =
false; \r\n");
        MQL4_source = MQL4_source.concat("          ticketclosed =
false; \r\n");
        MQL4_source = MQL4_source.concat("          _StopbyError =
false; \r\n");
        MQL4_source = MQL4_source.concat("          _buy_budget=1; \r\n");

```

```

        MQL4_source = MQL4_source.concat("        _sell_budget=1;`r\n`");
        MQL4_source = MQL4_source.concat("        explore_order =
true;`r\n`");
        MQL4_source = MQL4_source.concat("        _today = datetime());
daily_revenue = 0;`r\n`");
        MQL4_source = MQL4_source.concat("
        working_hours=false;`r\n`");
        MQL4_source = MQL4_source.concat("        Opn_MMABuy = true;
Opn_MMAsell = true;`r\n`");
        // VARIABLES AFFECTING button init()
        MQL4_source = MQL4_source.concat("        return(0);`r\n`");
        MQL4_source = MQL4_source.concat("
};`r\n`");
        MQL4_source = MQL4_source.concat("void OnDeinit(const int
reason);`r\n`");
        MQL4_source = MQL4_source.concat("        };`r\n`");
        //
        MQL4_source = MQL4_source.concat("
};`r\n`");
        //
        MQL4_source = MQL4_source.concat("void OnTick();`r\n`");
        MQL4_source = MQL4_source.concat("        };`r\n`");
        MQL4_source = MQL4_source.concat("
        check_RSI_level(RSIlevel_to_Buy, RSIlevel_to_Sell);`r\n`");
        MQL4_source = MQL4_source.concat("        check_opening_hours());
// Allows trading just between 08:00 and 18:00;`r\n`");
        MQL4_source = MQL4_source.concat("        if ((buy_ongoing ==
false) && (sell_ongoing == false) && (working_hours == true) && (daily_revenue <
200));`r\n`");
        MQL4_source = MQL4_source.concat("        };`r\n`");
        MQL4_source = MQL4_source.concat("
        goBuy(Opn_MMABuy);`r\n`");
        MQL4_source = MQL4_source.concat("
        Comment(`Open_bid: `, open_bid, ` Open SL: `, open_SL, ` Open TP: `,open_TP, `
Operation: `,name + NL);`r\n`");
        MQL4_source = MQL4_source.concat("        if
(_trail_buy == true);`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        TrailBuy(ticket_buy) ;`r\n`");
        MQL4_source = MQL4_source.concat("
        if (OrderProfit() < 0);`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        TrailSell(ticket_sellStop);`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        };`r\n`");
        MQL4_source = MQL4_source.concat("
        if ((buy_ongoing ==
false) && (sell_ongoing == false) && (working_hours == true) && (daily_revenue <
200));`r\n`");
        MQL4_source = MQL4_source.concat("        };`r\n`");

```

```

        MQL4_source = MQL4_source.concat("
goSell(Opn_MMAsell);`r\n");
        MQL4_source = MQL4_source.concat("
Comment(`Open_bid: `, open_bid, ` Open SL: `, open_SL, ` Open TP: `,open_TP, `
Operation: `,name + NL);`r\n");
        MQL4_source = MQL4_source.concat("                                if
(_trail_sell == true);`r\n");
        MQL4_source = MQL4_source.concat("                                {
;`r\n");
        MQL4_source = MQL4_source.concat("
TrailSell(ticket_sell) ;`r\n");
        MQL4_source = MQL4_source.concat("
if (OrderProfit() < 1);`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");
        MQL4_source = MQL4_source.concat("
        TrailBuy(ticket_buyStop);`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
        //
        MQL4_source = MQL4_source.concat("void check_RSI_level(int
RSIlevel_to_Buy, int RSIlevel_to_Sell);`r\n");
        MQL4_source = MQL4_source.concat("        {;`r\n");
        MQL4_source = MQL4_source.concat("        if (RSIlevel_to_Buy <=
iRSI(NULL, 15,3,PRICE_CLOSE,1));`r\n");
        MQL4_source = MQL4_source.concat("        };`r\n");
        MQL4_source = MQL4_source.concat("        Opn_MMAbuy
== true; ;`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
        MQL4_source = MQL4_source.concat("        if
(RSIlevel_to_Sell >= iRSI(NULL, 15,3,PRICE_CLOSE,1));`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");
        MQL4_source = MQL4_source.concat("
        Opn_MMAsell == true;`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
        MQL4_source = MQL4_source.concat("
};`r\n");
//
        MQL4_source = MQL4_source.concat("void goBuy(bool
Opn_MMAbuy);`r\n");
        MQL4_source = MQL4_source.concat("        { ;`r\n");
        MQL4_source = MQL4_source.concat("        if ((Opn_MMAbuy ==
true) && ( _buy_budget == 1 ));`r\n");
        MQL4_source = MQL4_source.concat("        };`r\n");
        MQL4_source = MQL4_source.concat("        if
(IsTradeAllowed() == true);`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");

```

```

        MQL4_source = MQL4_source.concat("
RefreshRates();\r\n");
        MQL4_source = MQL4_source.concat("
{;\r\n");
        MQL4_source = MQL4_source.concat("
open_ask = Ask; open_bid = Bid;\r\n");
        MQL4_source = MQL4_source.concat("
tp_Buy = open_ask + Point * _tp; tp_Buy = NormalizeDouble(tp_Buy ,
Digits);\r\n");
        MQL4_source = MQL4_source.concat("
tp_Buy10 = open_ask + Point * _tp10; tp_Buy10 = NormalizeDouble(tp_Buy10,
Digits);\r\n");
        MQL4_source = MQL4_source.concat("
sl_Buy = open_ask - Point * _sl; sl_Buy = NormalizeDouble(sl_Buy,
Digits);\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
ticket_buy = OrderSend(Symbol(),OP_BUY,LotSize_buy, open_ask ,3, 0 , 0 ,
_ordercomment,MagicNumber,0,clrGreen);\r\n");
        MQL4_source = MQL4_source.concat("
Sleep(500);\r\n");
        //MQL4_source = MQL4_source.concat("
Comment(`
BUY Ticket: `,ticket_buy,`Ask: `, open_ask,` SL: `,sl_Buy,"
TP: `,tp_Buy,` sl: `,_sl,` tp: `,_tp );\r\n");
        MQL4_source = MQL4_source.concat("
{;\r\n");
        //MQL4_source = MQL4_source.concat("draw_vSL("vSL", sl_Buy);
draw_vSL("TvSL", sl_Buy);draw_vTP( "vTP", tp_Buy );
draw_vTP10("vTP10",tp_Buy10);\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
write_Screenshot(ticket_buy);\r\n");
        MQL4_source = MQL4_source.concat("
OrderSelect(ticket_buy,MODE_TRADES);\r\n");
        MQL4_source = MQL4_source.concat("
if (ticket_buy <0);\r\n");
        MQL4_source = MQL4_source.concat("
{;\r\n");
        //MQL4_source = MQL4_source.concat("Alert("STOP EA. Fail Opening BUY
ORDER ",Symbol(), " Error: ", GetLastError());\r\n");
        MQL4_source = MQL4_source.concat("
_stopbyError = true;\r\n");
        MQL4_source = MQL4_source.concat("
return;\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
if (ticket_buy > 1);\r\n");
        MQL4_source = MQL4_source.concat("
{;\r\n");
        MQL4_source = MQL4_source.concat("
open_bid = sl_Buy;\r\n");
        MQL4_source = MQL4_source.concat("
tp_sellStop = open_bid - _tp * Point; tp_sellStop =
NormalizeDouble(tp_sellStop,Digits);\r\n");

```

```

        MQL4_source = MQL4_source.concat("
        sl_sellStop = open_ask + _sl * Point; sl_sellStop =
NormalizeDouble(sl_sellStop,Digits);`r\n");
        MQL4_source = MQL4_source.concat("
        ticket_sellStop = OrderSend(Symbol(),OP_SELLSTOP,LotSize*2, open_bid
,3, sl_sellStop,tp_sellStop , _ordercomment,MagicNumber,0,clrBlue);`r\n");
        MQL4_source = MQL4_source.concat("
        Sleep(500);`r\n");

        // {;`r\n");

draw_vSL("vSL", sl_sellStop); draw_vTP( "vTP", tp_sellStop );

                                                                 //
                                                                 // }

        MQL4_source = MQL4_source.concat("
write_Screenshot(ticket_sellStop);`r\n");
        MQL4_source = MQL4_source.concat("
        _trail_buy = true;`r\n");
        MQL4_source = MQL4_source.concat("
        buy_ongoing = true;`r\n");
        MQL4_source = MQL4_source.concat("
        _buy_budget = 0;`r\n");
        MQL4_source = MQL4_source.concat("
        _sell_budget = 1 ;`r\n");
        MQL4_source = MQL4_source.concat("
        OrderAnalyticOpen(ticket_buy);`r\n");
        MQL4_source = MQL4_source.concat("
        highest_high = Ask; lowest_low = Bid;`r\n");
        //MQL4_source = MQL4_source.concat("Alert("OrderBuy: ", ticket_buy,
" ",Symbol()," @",open_ask,"  SellStop:", ticket_sellStop, " @", open_bid
;`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
//
        MQL4_source = MQL4_source.concat("void goSell(bool
Opn_MMAsell);`r\n");
        MQL4_source = MQL4_source.concat("
        };`r\n");
        MQL4_source = MQL4_source.concat("
        if ((Opn_MMAsell
== true) && (_sell_budget == 1 ));`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");
        MQL4_source = MQL4_source.concat("
        if
(IsTradeAllowed() == true);`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");
        MQL4_source = MQL4_source.concat("
        RefreshRates();`r\n");
        MQL4_source = MQL4_source.concat("
        {;`r\n");
        MQL4_source = MQL4_source.concat("
        open_ask = Ask;open_bid = Bid;`r\n");

```

```

MQL4_source = MQL4_source.concat("
tp_Sell = open_bid - _tp*Point; tp_Sell = NormalizeDouble(tp_Sell,
Digits);`r\n`");
MQL4_source = MQL4_source.concat("
tp_Sell10 = open_bid - _tp10 * Point; tp_Sell10 =
NormalizeDouble(tp_Sell10, Digits);`r\n`");
MQL4_source = MQL4_source.concat("
sl_Sell = open_bid + _sl * Point; sl_Sell = NormalizeDouble(sl_Sell,
Digits);`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
MQL4_source = MQL4_source.concat("
ticket_sell = OrderSend(Symbol(),OP_SELL,LotSize_sell, open_bid
,3,0,0, _ordercomment,MagicNumber,0,clrGreen);`r\n`");
MQL4_source = MQL4_source.concat("
Sleep(500);`r\n`");
//MQL4_source = MQL4_source.concat("Comment("SELL Ticket:
",ticket_sell,"Bid: ", open_bid," SL: ",sl_Sell," TP: ",tp_Sell," sl: ",_sl," tp:
",_tp + NL );`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
//MQL4_source = MQL4_source.concat("draw_vSL("vSL",
sl_Sell);draw_vSL("TvSL", sl_Sell);draw_vTP( "vTP", tp_Sell );draw_vTP10("vTP10",
tp_Sell10);`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
MQL4_source = MQL4_source.concat("
OrderSelect(ticket_sell,MODE_TRADES);`r\n`");
MQL4_source = MQL4_source.concat("
if (ticket_sell <0);`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
//MQL4_source = MQL4_source.concat(" Alert("STOP EA. Fail
Opening SELL ORDER ", Symbol(), " Error: ", GetLastError());`r\n`");
MQL4_source = MQL4_source.concat("
_StopbyError=true;`r\n`");
MQL4_source = MQL4_source.concat("
return;`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
MQL4_source = MQL4_source.concat("
if (ticket_sell > 1);`r\n`");
MQL4_source = MQL4_source.concat("
");`r\n`");
MQL4_source = MQL4_source.concat("
open_ask = sl_Sell;`r\n`");
MQL4_source = MQL4_source.concat("
tp_buyStop = open_ask + Point * _tp;`r\n`");
MQL4_source = MQL4_source.concat("
sl_buyStop = open_ask - Point * _sl;`r\n`");
MQL4_source = MQL4_source.concat("
ticket_buyStop =
OrderSend(Symbol(),OP_BUYSTOP,LotSize*2,open_ask,3,sl_buyStop,tp_buyStop ,
_ordercomment,MagicNumber,0,clrBlue);`r\n`");
//MQL4_source = MQL4_source.concat(" Comment("BUYSTOP
Ticket: ",ticket_buyStop,"Ask: ", open_ask," SL: ",sl_buyStop," TP: ",tp_buyStop,"
sl: ",_sl," tp: ",_tp + NL );`r\n`");

```



```

        MQL4_source = MQL4_source.concat("
        Sleep(500);    ;\r\n");
    //{
    // draw_vSL("vSL", sl_buyStop); draw_vTP( "vTP", tp_buyStop );
    // }
    MQL4_source = MQL4_source.concat("
    sell_ongoing = true;\r\n");
    MQL4_source = MQL4_source.concat("
    _trail_sell = true;\r\n");
    MQL4_source = MQL4_source.concat("
    _sell_budget = 0;\r\n");
    MQL4_source = MQL4_source.concat("
    _buy_budget = 1;\r\n");
    MQL4_source = MQL4_source.concat("
    OrderAnalyticOpen(ticket_sell);\r\n");
    MQL4_source = MQL4_source.concat("
    highest_high = Ask; lowest_low = Bid;\r\n");
    //MQL4_source = MQL4_source.concat("                Alert("Sucess
OrderSell(): ", ticket_sell, " ",Symbol());\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    //
    //
    MQL4_source = MQL4_source.concat("                void
TrailBuy(double ticket);\r\n");
    MQL4_source = MQL4_source.concat("                {;\r\n");
    //
    MQL4_source = MQL4_source.concat("                while (
buy_ongoing == true ) // Trail while the position is open;\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");

    //.. of several operators
    //
    MQL4_source = MQL4_source.concat("                if
(Orderwasclosed(ticket_buy) == true) // Checks if Order was automatically closed by
system for TP or STL;\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    MQL4_source = MQL4_source.concat("
    closeticketbuy();\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    MQL4_source = MQL4_source.concat("
    if (OrderSelect(ticket_buy,SELECT_BY_TICKET) == false);\r\n");
    MQL4_source = MQL4_source.concat("
    );\r\n");
    //MQL4_source = MQL4_source.concat("
Alert("STOP EA @TrailBuy. Fail TicketBuy: ", ticket_buy, " ",Symbol(), " Error: ",
GetLastError());\r\n");
    MQL4_source = MQL4_source.concat("
    _StopbyError = true;\r\n");
    MQL4_source = MQL4_source.concat("
    buy_ongoing = false;\r\n");

```

```

return;`r\n`");
MQL4_source = MQL4_source.concat("
};`r\n`");
MQL4_source = MQL4_source.concat("
else;`r\n`");
MQL4_source = MQL4_source.concat("
{;`r\n`");
//MQL4_source = MQL4_source.concat("
Alert("Positive BUY TRAIL ORDERSELECT: ",ticket_buy," ", Symbol()),". Current Profit:
", (NormalizeDouble(_priceBID,Digits)-NormalizeDouble(OrderOpenPrice(),Digits))/Point,"
pip;`r\n`");`r\n`");
MQL4_source = MQL4_source.concat("
open=NormalizeDouble(OrderOpenPrice(),Digits);`r\n`");
MQL4_source = MQL4_source.concat("
//;`r\n`");
MQL4_source = MQL4_source.concat("
RefreshRates();`r\n`");
MQL4_source = MQL4_source.concat("
//;`r\n`");
MQL4_source = MQL4_source.concat("
_profit=OrderProfit();check_low_and_high();`r\n`");
MQL4_source = MQL4_source.concat("
if ( Bid > tp_Buy10 && Bid > sl_Buy );`r\n`");
MQL4_source = MQL4_source.concat("
{;`r\n`");
//MQL4_source = MQL4_source.concat("
sl_Buy = Bid ; draw_vSL ("TvSL",sl_Buy);`r\n`");
MQL4_source = MQL4_source.concat("
};`r\n`");
MQL4_source = MQL4_source.concat("
if ( Bid >= tp_Buy || Bid < sl_Buy );`r\n`");
MQL4_source = MQL4_source.concat("
{ ;`r\n`");
MQL4_source = MQL4_source.concat("
Ans2=OrderClose(ticket_buy,LotSize,Bid,3,clrBlue);`r\n`");
MQL4_source = MQL4_source.concat("
Sleep(500);`r\n`");
MQL4_source = MQL4_source.concat("
Ans1=OrderDelete(ticket_sellStop,clrGreen);`r\n`");
MQL4_source = MQL4_source.concat("
closeticketbuy();`r\n`");
MQL4_source = MQL4_source.concat("
update_daily_revenue();daily_revenue=daily_revenue+_profit;`r\n`");
// return;
MQL4_source = MQL4_source.concat("
};`r\n`");
MQL4_source = MQL4_source.concat("
};`r\n`");
MQL4_source = MQL4_source.concat("
};`r\n`");
MQL4_source = MQL4_source.concat("
} ;`r\n`");
//
MQL4_source = MQL4_source.concat("void TrailSell(double
ticket);`r\n`");
MQL4_source = MQL4_source.concat("{;`r\n`");
MQL4_source = MQL4_source.concat("while ( sell_ongoing=
true);`r\n`");

```

```

        MQL4_source = MQL4_source.concat("{\r\n");
        MQL4_source = MQL4_source.concat("if
(Orderwasclosed(ticket_sell)==true) // Checks if Order was automatically closed by
system for TP or STL ;\r\n");
        MQL4_source = MQL4_source.concat("{\r\n");
        MQL4_source = MQL4_source.concat("closeticketsell();\r\n");
        MQL4_source = MQL4_source.concat("return;\r\n");
        MQL4_source = MQL4_source.concat("};\r\n");
        MQL4_source = MQL4_source.concat("if (sell_ongoing==true);\r\n");
        MQL4_source = MQL4_source.concat("{\r\n");
        MQL4_source = MQL4_source.concat("if
(OrderSelect(ticket_sell,SELECT_BY_TICKET)==false);\r\n");
        MQL4_source = MQL4_source.concat("{\r\n");
        //MQL4_source = MQL4_source.concat("Alert("STOP EA @TrailSell. Fail
TicketSell: ", ticket_sell, " ",Symbol(), " Error: ", GetLastError());\r\n");
        MQL4_source = MQL4_source.concat("_StopbyError = true;\r\n");
        MQL4_source = MQL4_source.concat("sell_ongoing = false;\r\n");
        MQL4_source = MQL4_source.concat("return;\r\n");
        MQL4_source = MQL4_source.concat("};\r\n");
        MQL4_source = MQL4_source.concat("else ;\r\n");
        MQL4_source = MQL4_source.concat("{\r\n");
        //MQL4_source = MQL4_source.concat("Alert("Positive SELL TRAIL
ORDERSELECT: ",ticket_sell," ", Symbol(),". Current Profit: ",
(NormalizeDouble(_priceBID,Digits)-NormalizeDouble(OrderOpenPrice(),Digits))/Point,"
pip;\r\n");\r\n");
        //
        MQL4_source = MQL4_source.concat("RefreshRates();\r\n");
        //
        MQL4_source = MQL4_source.concat("if ( Ask < tp_Sell10 && Ask <
sl_Sell );\r\n");
        MQL4_source = MQL4_source.concat("{\r\n");
        //MQL4_source = MQL4_source.concat("
sl_Sell = Ask; draw_vSL ("TvSL",sl_Sell);\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
_profit=OrderProfit(); check_low_and_high();\r\n");
        MQL4_source = MQL4_source.concat("                                     if ( Ask
<= tp_Sell || Ask > sl_Sell );\r\n");
        MQL4_source = MQL4_source.concat("
{ ;\r\n");
        MQL4_source = MQL4_source.concat("
Ans2=OrderClose(ticket_sell,LotSize,Ask,3,clrRed);\r\n");
        MQL4_source = MQL4_source.concat("
Sleep(500);\r\n");
        MQL4_source = MQL4_source.concat("
Ans1=OrderDelete(ticket_buyStop,clrBlue);\r\n");
        MQL4_source = MQL4_source.concat("
closeticketsell();\r\n");
        MQL4_source = MQL4_source.concat("
update_daily_revenue();daily_revenue=daily_revenue+_profit;\r\n");
        //return;
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");
        MQL4_source = MQL4_source.concat("
};\r\n");

```


Anexo II: Código MQL4 generado con AutoTrader

```
//+-----+
///  
- AutoTrader v1.0 - |
///  
Copyright 2022, Ignacio Fumanal |
///  
Master Ingenieria de Software |
///  
+-----+
#property copyright Copyright 2022, IFA;
#property version 1.00;
#property strict;
///  
+-----+
///  
| Expert initialization function |
+-----+
// GLOBAL TERMINAL VARIABLES
int vSL_points=0;
int vTP_points=80;
#define NL
```

```

;
'bool      Opn_MMABuy=false; '
'bool      Opn_MMASell=false; '
'datetime  _ordertime=TimeCurrent();
'string    time=TimeCurrent();
'double    _priceASK=Ask;
'double    _priceBID=Bid;
'bool      buy_ongoing=false;
'bool      sell_ongoing=false;
'int       ticket_buy=100;
'int       ticket_buy_2nd=0;
'int       ticket_sell=200;
'int       ticket_sell_2nd=0;
'int       _buy_budget = 1;
'int       _sell_budget = 1;
'double    tp_Sell;
'double    tp_Sell10;
'double    sl_Sell;
double     tp_sellStop;
'double    sl_sellStop;
'double    tp_buyStop;
'double    sl_buyStop;
'double    tp_Buy;
'double    tp_Buy10;
'double    sl_Buy;

```

```
'bool    played_order_sound=false;'  
'bool    Ans1=false;'  
'bool    Ans2=false;'  
'bool    Ans3=false;'  
'bool    Ans4=false;'  
'datetime now_time=Time[0];'  
'string  name = 'BOTON';'  
'int     _typeArrow=181;'  
'double  _priceArrow = 0;'  
'double  _yAxis=Bid;'  
'string  _arrowColor = clrAliceBlue;'  
'int     _filehandle=0;'  
'string  _subfolder='Analytics' ;'  
'int     counter=0;'  
'int     ticket = 0;'  
'string  _ordercomment=StringConcatenate('ROBOT Order ',Symbol());'  
'bool    _trail_buy=false;'  
'bool    _trail_sell=false;'  
'bool    _StopbyError=false;'  
'int     magic = 1909;'  
'int     period;'  
'bool    ticketclosed=false;'  
'double  ticket_buyStop;'  
'double  ticket_sellStop;'  
'double  open_ask;'  
'double  open_bid;'  
'double  Pips=0;'  
'double  STP=0;'  
'double  _profit=0;
```

```

'double  open=0;'
'double  stop=0;'
'long    Volume_log;'
'string  Comment_log;'
'string  result_log;'
'string  optype_log;'
'double  lowest_low;'
'double  highest_high;'
'bool    explore_order=true;'
'double  Spread=NormalizeDouble((Ask-Bid)/Point(),1);'
'double  retrace_up;'
'double  retrace_down;'
'double  bandwidth;'
'double  daily_revenue=0;'
'datetime _today;'
'datetime date_op;'
'datetime today_ops;'
'bool    working_hours=false;'
'bool    revenue_recorded_flag=true;'
'string  _strategy;'
'double  _drawdown_pct;'
'double  open_SL;'
'double  open_TP;'
'int     n;'
'input int  _tp=20;'
'input int  _tp10=10;'
'////////////////////////////////////'
'///// Variables setting from JAVA screen to MQL4  /////'
'//

```



```

int    RSlevel_to_Buy= 20;
'int   RSlevel_to_Sell= 70;
'int   LotSize_buy= 0.01;
'int   LotSize_sell= 0.01;
'double vSL=5;
'double vTP=10;
'double
MagicNumber=javax.swing.JComboBox[,726,441,72x27,layout=javax.swing.plaf.metal.MetalComb
oBoxUI$MetalComboBoxLayoutManager,alignmentX=0.0,alignmentY=0.0,border=,flags=328,maxi
mumSize=,minimumSize=,preferredSize=,isEditable=false,lightWeightPopupEnabled=true,maximu
mRowCount=3,selectedItemReminder=1234];
'//
//
//
int OnInit();
' {';
'   Sleep(20);
'   buy_ongoing = false;
'   sell_ongoing = false;
'   ticketclosed = false;
'   _StopbyError = false;
'   _buy_budget=1;
'   _sell_budget=1;
'       explore_order = true;
'       _today = datetime(); daily_revenue = 0;
'       working_hours=false;
'       Opn_MMABuy = true; Opn_MMAsell = true;
'       return(0);
'           };
'void OnDeinit(const int reason);

```

```

'
'           {'
'
'           };
'
' void OnTick();
'
'           {'
'
'           check_RSI_level(RSIlevel_to_Buy, RSIlevel_to_Sell);
'
'           check_opening_hours();           // Allows trading just between 08:00 and 18:00;
'
'           if ((buy_ongoing == false) && (sell_ongoing == false) && (working_hours == true)
&& (daily_revenue < 200));
'
'           {'
'
'           goBuy(Opn_MMABuy);
'
'           Comment('Open_bid: ', open_bid, ' Open SL: ', open_SL, ' Open TP:
',open_TP, ' Operation: ',name + NL);
'
'           if (_trail_buy == true);
'
'           {'
'
'           TrailBuy(ticket_buy) ;
'
'           if (OrderProfit() < 0);
'
'           {'
'
'           TrailSell(ticket_sellStop);
'
'           };
'
'           };
'
'           if ((buy_ongoing == false) && (sell_ongoing == false) && (working_hours == true)
&& (daily_revenue < 200));
'
'           {'
'
'           goSell(Opn_MMASell);
'
'           Comment('Open_bid: ', open_bid, ' Open SL: ', open_SL, ' Open TP:
',open_TP, ' Operation: ',name + NL);
'
'           if (_trail_sell == true);
'
'           {'
'
'           TrailSell(ticket_sell) ;

```

```

        if (OrderProfit() < 1);
        {
            TrailBuy(ticket_buyStop);
        }
    };
};
};

void check_RSI_level(int RSIlevel_to_Buy, int RSIlevel_to_Sell);
{
    if (RSIlevel_to_Buy <= iRSI(NULL, 15,3,PRICE_CLOSE,1));
    {
        Opn_MMABuy == true; ;
    }
    if (RSIlevel_to_Sell >= iRSI(NULL, 15,3,PRICE_CLOSE,1));
    {
        Opn_MMAsell == true;
    }
};

void goBuy(bool Opn_MMABuy);
{ ;
    if ((Opn_MMABuy == true) && ( _buy_budget == 1 ));
    {
        if (IsTradeAllowed() == true);
        {
            RefreshRates();
        }
        open_ask = Ask; open_bid = Bid;
        tp_Buy = open_ask + Point * _tp; tp_Buy =
NormalizeDouble(tp_Buy , Digits);

```



```

'
'           if (Orderwasclosed(ticket_buy) == true) // Checks if Order was
automatically closed by system for TP or STL;'
'
'           {'
'
'               closeticketbuy();'
'
'           } ;'
'
'           if (OrderSelect(ticket_buy,SELECT_BY_TICKET) == false);'
'
'           {'
'
'               _StopbyError = true;'
'
'               buy_ongoing = false;'
'
'               return;'
'
'           };'
'
'           else;'
'
'           {'
'
'               open=NormalizeDouble(OrderOpenPrice(),Digits);'
'
'               //;'
'
'               RefreshRates();'
'
'               //;'
'
'               _profit=OrderProfit();check_low_and_high();'
'
'               if ( Bid > tp_Buy10 && Bid > sl_Buy );'
'
'           };'
'
'           };'
'
'           if ( Bid >= tp_Buy || Bid < sl_Buy );'
'
'           {' ;'
'
'
'               Ans2=OrderClose(ticket_buy,LotSize,Bid,3,clrBlue);'
'
'               Sleep(500);'
'
'               Ans1=OrderDelete(ticket_sellStop,clrGreen);'
'
'               closeticketbuy();'
'
'
'
'
'           update_daily_revenue();daily_revenue=daily_revenue+_profit;'
'
'           };'

```

```

        }';
    };
};

void TrailSell(double ticket);
{
while ( sell_ongoing= true);
{
if (Orderwasclosed(ticket_sell)==true) // Checks if Order was automatically closed by system for
TP or STL;
{
closeticketsell();
return;
};
if (sell_ongoing==true);
{
if (OrderSelect(ticket_sell,SELECT_BY_TICKET)==false);
{
_StopbyError = true;
sell_ongoing = false;
return;
};
else ;
{
RefreshRates();
if ( Ask < tp_Sell10 && Ask < sl_Sell );
{
};
};
profit=OrderProfit(); check_low_and_high();
if ( Ask <= tp_Sell || Ask > sl_Sell );

```



```
'      { ;'  
'      Ans2=OrderClose(ticket_sell, LotSize, Ask, 3, clrRed);'  
'      Sleep(500);'  
'      Ans1=OrderDelete(ticket_buyStop, clrBlue);'  
'      closeticketsell();'  
'      update_daily_revenue(); daily_revenue=daily_revenue+_profit;'  
'      };'  
'      };'  
'      };'  
'      };'  
'      };'
```

