

Trabajo Fin de Grado



Desarrollo de un servicio de inversión en bolsa utilizando MQL4 y Java

Fecha:

Autor: **Juan Antonio Tora Cánovas**

Índice del contenido:

1. Introducción.....	4
2. Victoria privada: etapas que he pasado como trader el primer año de investigación y operativa real.	6
3. Conceptos fundamentales y bases de los mercados financieros.....	9
3.1- Diferentes productos financieros: los principales.....	9
3.2- Diferentes formas de analizar los mercados financieros	13
3.3- Cuándo NO operar en los mercados	14
3.4- Proceso que sigue todo trader hasta el éxito	15
3.5- Los 10 grandes errores que te hacen “un trader perdedor”	21
3.6- ¿Después de haber seguido todos los pasos anteriores sigues perdiendo dinero?	22
3.7- Bolsas mundiales más importantes y su interconexión	23
3.8- Introducción al Análisis Técnico de los Mercados Financieros	26
3.9- Ejemplos de Análisis Financiero	29
3.10- Gestión del riesgo.....	32
3.11- Tipos de comisiones presentes operando CFD's, Opciones Binarias y Acciones.	35
3.12- Comparativa de rentabilidad mensual: Opciones Binarias Vs CFD's.	38
4. Diseño general de la APP	44
4.1- Visión General.....	44
4.2- Flujo-Grama del Front-End (Java)	46
4.3- Flujo-Grama del Back-End (MetaTrader4)	47
4.4- Apariencia real de la aplicación Final	48
5. MetaTrader 4	50
5.1- Toma de datos en Tiempo Real con MetaTrader 4	50
5.2- Uso de la GUI de Meta Trader 4.....	53
5.3- Expert Advisors	62
5.4- Indicadores	63
5.5- Scripts	66
5.6- Plantillas	67
5.7- Estructura de archivos de Meta Trader 4	68
6. MQL4: servidor Jast.....	70
6.1- Mis primeros programas creados en MQL4	70
6.2- Paradigma de programando para operar automáticamente en los mercados desarrollando herramientas MQL4.....	77
6.3- Estrategia de Trading subyacente del Proyecto.....	80

6.4- Definimos nuestra Plantilla para cargar los indicadores creados	81
6.5- Indicadores propios implementados (utilizados en el proyecto)	82
6.6- Desarrollando el Expert Advisor que opera CFD's automáticamente al recibir la orden del programa Java	84
6.7- Simulador: probando los EAs	86
7. Cliente JAST: programa desarrollado en Java	88
7.1- Arquitectura programa Java y llamadas recursivas	88
7.2- Programas que captura la pantalla cada hora y cada 5 minutos	90
7.3- Programa controlador CFD's (explicado)	92
7.3- Lectura de noticias macro-económicas con ERROR 403	94
7.4- Eliminando ERROR 403, pero sin lograr HTML todavía	96
7.5- Código Python que devuelve el HTML deseado	97
7.6- Solución al problema de acceso Prohibido 403 (explicado)	98
8. Team Viewer	100
9. Conclusiones y trabajos futuros	103
10. Bibliografía	105

1. Introducción

Me presento como un alumno de la Universidad Politécnica de Cartagena al final de mi carrera universitaria de Ingeniería Telecomunicación, especializado en ingeniería telemática. Decidí desarrollar este proyecto porque, en mi opinión, al acabar la carrera de ingeniería de telecomunicaciones todo alumno graduado cuenta con conocimientos básicos para lograr desarrollarse como emprendedor tecnológico, además de un empleo en el ámbito de las TIC, y así ganar un sueldo extra. Desde que entré a estudiar esta carrera busqué desarrollar una idea que me permitiera obtener beneficios de forma automática utilizando la tecnología.

En este caso yo utilizo la programación Java y de bolsa para automatizar procesos, tales como: tomar decisiones lógicas en lugar de emocionales, gracias a la ejecución de un programa software, además de obtener beneficios constantes sin estar presente. Este objetivo personal se ha logrado, desarrollado y explicado a lo largo de este proyecto, es decir: lograr una solución tecnológica que me aportara un colchón económico vitalicio.

Por otra parte, el proyecto se ha dividido en 2 vías temáticas claramente diferenciadas:

1. **Primera parte: INVESTIGACIÓN.**

En esta parte podrá encontrar lo siguiente: descripción del proceso personal por el que he tenido que pasar como trader para llegar a tener relativo éxito en mi operativa financiera, estudio legal y regulatorio de organismos de mercado, estudio estadístico y muestral para cada tipo de estrategia y producto, qué cosas me fue mejor y con cuales peor, riesgos financieros y tipos de comisiones, presentación selectiva (recopilada de más de 1200 páginas de libros) a cerca de mercados financieros, descripción de los tipos de productos financieros más relevantes que existen en el mercado, trading emocional y como evitarlo, etc.

2. **Primera parte: PERSPECTIVA TÉCNICA DEL PROYECTO BASADA EN SOFTWARE.**

Se explicará los mecanismos y herramientas software desarrolladas para lograr una solución software robusta, ante la necesidad que se nos plantea cubrir, de un trader minorista con empleo y poco tiempo para operar, y explicando esta solución técnica de forma detallada y esquematizada a lo largo del proyecto.

NOTA: El software ha sido implementado en Java y MQL4.

Para lograr todos los objetivos recién descritos desde la perspectiva técnica, este proyecto describe un software que creará un sistema económicamente rentable. Los pasos técnicos que se han dado han sido contrarios en orden a cómo se muestra en el Proyecto, es decir, primeramente en desarrollo Java (aprovechando mi larga experiencia) y posteriormente formándome en lenguaje MQL4 para crear operativa automática en mercados financieros. Cuando se han terminado ambas partes por separado se ha desarrollado una comunicación entre la herramienta Java y el software automático MQL4. De esta manera el cliente podría iniciar y parar procesos automáticos complejos en un sistema Windows local. También se crea una selección de códigos MQL4 a modo de estrategias de trading automáticas, y posteriormente se han testeado con un simulador utilizando el programa MetaTrader4 (todos los resultados son mostrados).

Para terminar, se plantea el uso de TeamViewer para control remoto del sistema y algunas posibles continuaciones propuestas para este proyecto.

Destacar por último que la aplicación software resultante de este proyecto fue llamada "Jast": está formada por el "*cliente Jast*" hecho en Java y el "*servidor Jast*" hecho en MQL4 (lenguaje basado en C).

2. Victoria privada: etapas que he pasado como trader el primer año de investigación y operativa real.

Este proceso lo he separado en 20 fases de mi proceso personal:

1º) Entro en operaciones de opciones binarias y me desmiento la idea de “el trading es fácil”. Empiezo en cuentas demo y pronto me doy cuenta de la gran dificultad real existente.

2º) Empiezo a **discernir la información valiosa** de la engañosa. La formación valiosa es de pago, pero no toda la de pago es valiosa.

3º) Tomo documentación diaria y continuada en canales “sofisticados” de **Youtube** y lectura de los 3 grandes **libros de bolsa** por excelencia. Me permite una apertura mental hacia productos financieros extras a Opciones Binarias: Acciones, CFD's, etc.

4º) Empiezo a formarme en **análisis técnico, análisis chartista y análisis fundamental**.

5º) Recopilación de material y libros para aprender **programación MQL4** para implementar análisis técnico mediante software propio de MetaTrader4: programación de ExpertAdvisors, plantillas, indicadores técnicos y scripts.

6º) Decido sectorizar mi aprendizaje de finanzas, ya maduro, en **3 tipos de productos financieros**: Opciones Binarias, Acciones y CFD's.

7º) Abandono la operativa en cuenta DEMO y paso a operar en cuenta real en CDF, Opciones Binarias y Acciones.

8º) Me formo de manera profesional y con dinero real para operar con **Opciones Binarias**. Pongo a prueba mi templanza en trading de alta frecuencia. Resultó ser mi primer fracaso en el trading.

9º) Paso a operar **en CFD con cuenta real**. Abro cuentas en diferentes brokers, tanto nacionales como internacionales. Replico la misma operación en 3 brokers simultáneamente, manualmente. En este punto descubro que cada bróker cobra **una comisión diferente** y todo lo que esto afecta en la rentabilidad final.

10º) Aprendo **qué regulaciones de mercados financieros son los principales** para conocer que un bróker es honrado y justo en las transacciones dentro de los mercados financieros. Algunas regulaciones más importantes son estas: **GSC** (isla de Man, Gran Bretaña), **FCA** (organismo internacional encargado de garantizar comportamientos justos de Reino Unido), **CNMV** (Comisión Nacional del Mercado de Valores en España), **CNB** (Czech National Bank), etc.

11º) **Desarrollo mi primero software para opciones binarias con MQL4. Utiliza alarmas sonoras** en MQL4. Tener en cuenta que combiné conocimientos de programación MQL4 y análisis de mercados financieros, en esos momentos ambos recién adquiridos. Pese a automatizar el proceso al máximo. El miedo es la emoción más fuerte: este me lleva a quebrar una micro-cuenta REAL. Abandono las opciones binarias definitivamente.

12) **Me formo en trading emocional** para un trading real eficaz. Lectura obligatoria del libro VIVIR DEL TRADING: libro escrito por un psiquiatra ruso que explica los procesos mentales que el cerebro lleva a cabo realmente en el trading.

13º) Vuelvo al mercado con cuenta real, pero esta vez en CFD, y todo es más complejo a primera vista. Necesito una **base fuerte de gestión de capital**. Ahora en CFD el dinero que puedo perder o ganar es infinito y debo formarme bien en cómo gestionar eficazmente mi cartera de inversión.

14º) Entre el análisis técnico, chartista o de fundamentales **me termino identificando como analista técnico**. Huyo siempre de operar en momentos de salidas en noticias fundamentales, porque como analista técnico el mercado a veces no responde como preveo que responda técnicamente. El análisis técnico me permite usar funciones estadísticas y matemáticas.

15º) Profundizo sobre todos los tipos de comisiones existentes en los CFD's, llegando a haber distintos tipos de contrato y hasta un total de 5 comisiones diferentes que salvar antes de ver un solo céntimo.

16º) El CFD me estresa en gran medida, por lo que pruebo el mundo de las acciones. Para ello **compro BITCOIN y otros productos diversificados**. Esto no es minería de moneda digital, sino **ACCIONES sobre productos ya existentes**. Mis beneficios vendrán de la diferencia entre la compra y la venta de cada

“acción”, con pérdidas limitadas y ganancias ilimitadas: yo gano si vendo más caro que cuando compré.

17º) Las acciones van viento en popa, y mi estrés y tiempo de dedicación necesario disminuye. Es el momento de estudiar por cuenta propia la legalidad de los paraísos fiscales, dónde se sitúan y la razón de su existencia. También estudio los negocios de tipo piramidal y de tipo **HYiP** (“**high yield investment program**” o programa de inversión de alto rendimiento).

19º) Tras un diseño software inicial en MetaTrader4 de un sistema de trading con alarmas **empiezo a desarrollar mi TFG propiamente dicho**. Tanto la plataforma Java, que dará la interfaz de usuario final, como la inteligencia automatizada del sistema de trading subyacente en MetaTrader4. Empiezo también a realizar comparativas reales con uso de software capturador de resultados 24h y 7 días a la semana: se logra estimar la rentabilidad entre CFD y Opciones binarias despreciando a las emociones personales.

20º) Me aficiono a leer un poco cada día de libro de empresa y talento, tras leer recomendaciones de Warren Buffet (el padre de los mercados financieros americanos). Por esto busco referiblemente autores americanos.

Algunos libros que sugiero leer puede encontrarlos en la bibliografía, apartado “Libros de Coaching recomendados”.

3. Conceptos fundamentales y bases de los mercados financieros.

3.1- Diferentes productos financieros: los principales

Definición de “Producto Financiero”:

“Contrato, acción, bono u obligación de deuda que equivale a una cantidad monetaria, y que puede ser una inversión simple (sin derecho a propiedad) u otorgar derechos a la hora de invertir en mercados. Los productos financieros son comprados o vendidos gracias al acceso al mercado que nos proporcionará un intermediario (casas de bolsa, bancos o brokers), y deberemos pagar una comisión por acceder a dicho mercado”.

En adelante nosotros entenderemos por “**producto financiero**” lo siguiente: manera de operar en los mercados con reglas de comercio bien diferenciadas entre sí. Los productos financieros de los que se hablará en este proyecto son: acciones, opciones binarias y CFDs.

A continuación se hace una breve descripción de cada uno de estos productos:

Acciones: producto financiero de mercados más antiguo y simple que hay. Consiste en comprar una porción de una empresa (llamada “acción”) a un precio acordado, con perspectivas de vender esta participación a un precio superior en el futuro. Es un producto de muy largo plazo. El beneficio se obtiene al vender la acción cuando esté más cara del precio al que se compró. *No confundir ACCIONES con el reparto de dividendos dentro de la empresa, pues las acciones no te darán derecho al reparto de dividendos.*

Pros:

-El capital que puedo perder es finito, e igual al precio que valió mi acción multiplicado por el nº de acciones compradas.

-La esperanza matemática de sacar beneficios puede llegar a ser positiva si se hace un buen análisis financiero antes de invertir.

Contras:

-El tiempo transcurrido entre compra y venta la acción es a largo plazo (meses o años).

CFD: es el producto financiero más arriesgado que existe. Aquí se venden y se compran CONTRATOS, no acciones. El “contrato” que se compra exige obligaciones legales al propietario del contrato, no a la empresa propietaria, es decir, en caso de no acertar la dirección del mercado el trader deberá pagar las pérdidas al mercado. El término “CFD” significa de sus siglas del inglés “Contrat For Difference”. Los contratos que compre o venda usted operando en CFD presentarán pérdidas o ganancias inmediatas, y la cantidad de volumen de cada contrato va expresada en LOTES, MILIOTES o MICROOTES.

Pros:

-Los beneficios son ilimitados si usted acierta el sentido de la tendencia y sale de la operación antes de que se ponga el mercado en su contra.

-Puede ganar dinero tanto cuando el mercado sube como cuando baja. Solo necesita acertar el sentido en el cual irá.

-Puede colocar stops de protección o stops de beneficios para minimizar pérdidas y maximizar ganancias. También puede programar órdenes al mercado para que se ejecuten órdenes PUT o CALL a cierto nivel de precio estimado por usted.

-Existe gran multitud de software para programar las operaciones, stops loss, stop profit, etc. Además su precio no es elevado o incluso incluye los servicios en las comisiones por operación.

-Si usted es un buen analista de mercados financieros cada mínima variación del mercado usted ingresará dinero en su cuenta: desde 0,01€ por cada variación mínima del precio a mi favor.

Contras:

-No se necesita disponer de todo el capital para invertir en los mercados financieros, pues el bróker te hace un préstamo: se denomina “apalancamiento financiero”. Podrás invertir más dinero del que tienes, y endeudarte rápidamente.

-Pérdidas ilimitadas: Podemos llegar a perder más del capital que tenemos en nuestro “bank” o cuenta del bróker si combinamos los movimientos imprevistos del precio con un alto apalancamiento. La deuda podría ser vitalicia.

-Una buena gestión del riesgo no implica que obtengas beneficios: recomendado no invertir más del 2% del capital por operación y mercado elegido. Si la operación está en pérdidas del 2% considerando la inversión **nominal** (fija) y **marginal** (variable) salte de la operación YA!

-Aunque coloques **stop-loss** correctos podrías tener pérdidas mayores si existen saltos del precio bruscos en tu contra. A éstos se les denomina GAPS y si ocurren tu bróker te sacará. Cuando esto pase tú tendrás pérdidas mayores de las que estabas dispuesto a correr.

-Gran opacidad en la operativa interna de tus órdenes hacia el mercado:

a) Algunos brokers no ejecutan sus órdenes hasta transcurridos unos segundos y las retrasan.

b) La mayoría de brókers no envían al mercado todas las operaciones: el bróker solo enviará tus órdenes si eres un operador que registra beneficios para que te los pague el mercado, pero si registras pérdidas el bróker nunca enviará sus órdenes al mercado. De esta manera ellos se ahorran pagar la comisión por acceder al mercado. A este tipo de bróker se les denomina “dealers”.

-Cualquier operación que envíes al mercado siempre empezará en pérdidas de 30 pips, o puntos mínimos. A eso súmele comisiones de acceso y diferencias del precio entre el precio real y el ofrecido por su bróker.

Opciones Binarias: se basa en acertar la dirección que tomará el precio de un activo a la hora que usted indique como límite. Usted sabrá de antemano la hora de entrada (colocación de orden por usted) al mercado y la hora de salida. Estos tiempos serán fijos una vez la operación se inicie.

Pros:

-Operativa mínima de 60 segundos, y un máximo de varios días.

-Rápido acceso, rápidos beneficios.

-Producto financiero más moderno.

-Pérdidas limitadas al capital invertido, nunca superior a la cantidad de tu BANK.

Contras:

-Existe la falsa creencia de que “es muy fácil ganar dinero en opciones binarias”. Dominar este producto financiero correctamente suele llevar entre 6 meses y 2 años a tiempo completo, y no siempre se logra.

-La esperanza matemática de obtener beneficios es negativa: cuando se pierde una operación te quitan el 100% de tu inversión y cuando ganas solo se paga el 70% de tu inversión. Los mejor brokers rondan el 85% de retorno al ganar una operación, pero aun así hay probabilidad negativa.

-Este producto financiero genera unas emociones elevadas de avaricia, porque puedes ganar gran cantidad de dinero rápidamente. Este producto financiero potencia el “trading emocional” y dificulta el “trading racional”, y realizando trading emocional se pierde el trading objetivo en el mercado financiero. ¡Directamente a la quiebra!

3.2- Diferentes formas de analizar los mercados financieros

Se pueden analizar los mercados basándonos en noticias macroeconómicas, análisis técnico y análisis gráfico. Las 3 formas de análisis son altamente aceptadas por la comunidad mundial de traders, y se explican brevemente por separado a continuación:



Análisis de Noticias Fundamentales: las noticias hacen que el mercado fluctúe irracionalmente en décimas de segundo como consecuencia de que salga una noticia negativa inesperada. Este análisis es el más importante de los 3, pues da explicación a las reacciones más impulsivas del mercado. No necesariamente será también el más efectivo. Personalmente siempre recomiendo tener en cuenta las noticias fundamentales, aunque usted sea otro tipo de analista, y así evitará sustos innecesarios.

Análisis Técnico: este análisis pretende predecir la dirección del precio de un activo utilizando indicadores técnicos, basados en cálculos estadísticos, medias móviles y algoritmos matemáticos. Los indicadores técnicos de los más importantes son: MACD, RSI, medias móviles y niveles Fibonacci.



Análisis Gráfico: el analista gráfico solo utiliza una gráfica del precio para predecir el sentido y nivel objetivo que debe alcanzar el precio. No usa nada más que la gráfica para interpretar el sentido del mercado. Como mucho puede utilizar velas japonesas y pintar líneas sobre las gráficas en niveles de resistencia o de soporte del precio. Nunca utilizará ningún indicador matemático. Un analista gráfico busca soportes, resistencias y figuras de consolidación del precio. Un analista técnico es también analista gráfico, pero un analista gráfico jamás será un analista técnico.



3.3- Cuándo NO operar en los mercados

Jamás operar un mercado ni hacer especulación de ningún tipo si usted está emocionalmente sensible, alterado o cansado. Esto suele generar una MALA lectura del mercado y la operación que se coloque saldrá normalmente perdedora. De nada sirve saber leer patrones gráficos si se altera nuestra percepción de la realidad mediante estados emocionales inapropiados para su lectura. El mercado siempre se mueve por emociones, aunque tratemos de interpretarlo mediante indicadores técnicos.

El profesional que analiza el mercado debe tener una base de conocimientos sobre finanzas para poder asumir riesgos controlados, pero también formación emocional para controlar su miedo y su euforia.

EUFORIA



MIEDO



Un buen símil para explicar la relación que se establece entre el trader y el mercado sería este: “un trader que opera en los mercados financieros para cazar operaciones que le retornen ganancias es como aquel marinero que se lanza al océano para pescar pescado que pueda vender.” Un marinero no puede influir en el océano así como el océano no desea matar al marinero. El marinero debe saber cuándo puede y cuándo no navegar. Lo mismo ocurre entre un trader y el mercado: **el mercado no desea arruinar al trader, ni el trader puede afectar al mercado, sino que el trader debe saber leer las condiciones del mercado para entrar a éste, salir vivo y con ganancias.**



3.4- Proceso que sigue todo trader hasta el éxito

El trading es un proceso que no es corto, y requiere mucha voluntad para lograr finalizarlo con éxito. No hay ningún trader que no inicie este proceso desde las pérdidas. Un trader es cualquier persona que toma sus propias decisiones, libre y de forma lógica en los mercados financieros. Hacer "trading" no significa ahorrar o invertir ni ganar, sino que se debe complementar esto con la inversión y el ahorro. Un verdadero trader no busca ni beneficios ni dinero, sino cazar una buena oportunidad en el mercado a corto plazo y mejorar cada día. Después, al lograr esto, el dinero vendrá sólo: es el resultado del trading bien hecho.



Las fases de este proceso son las siguientes:

1ª Fase: ENTRAR AL MERCADO POR MOTIVOS EQUIVOCADOS

Muchas personas se inician en los mercados por los que representa el concepto de "ganar dinero" y "libertad financiera". Otros porque le gusta el juego. Esto es lo que conocemos como "avaricia", y es lo que te hará perder todo tu dinero. La avaricia crea traders impulsivos, descontrolados y, finalmente, arruinados.

2ª Fase: UTILIZAR MAL LAS CUENTAS DEMO

Comenzar con una cuenta DEMO es el primer paso de cualquier trader, pero tiene sus riesgos. Esto debe de hacerse para practicar una estrategia sin riesgo económico, pero es un arma de doble filo. El trader novato que empiezan a ganar dinero en cuenta DEMO a veces se vuelve arrogante: el comportamiento arrogante incrementa el sentimiento de avaricia, y la avaricia eleva la propensión al riesgo. Un trader con un riesgo elevado te acerca más rápido a la siguiente fase...

3ª Fase: TU PRIMERA GRAN PÉRDIDA

La primera gran pérdida en muchas ocasiones supera el 50% de tu cuenta. Suele ocurrir por el exceso de confianza que genera la sensación de “dominio sobre los mercados” al comenzar a entenderlos y ganar dinero. Tarde o temprano te pegas el “BOOM” y el mercado te recuerda por primera vez que de nada te sirve tener control sobre los mercados. Aprendes que la última palabra siempre la tiene el mercado. Ahora entiendes que el concepto de “riesgo” es muy real, y no es divertido.

4ª Fase: EL MIEDO A PERDER DEMASIADO

El sentimiento del miedo si instala definitivamente en tu mente tras la primera gran pérdida. La presencia cada vez más activa del “miedo a perder” puede bloquearte y frenar tus operaciones. Empiezas a desconfiar de los informes financieros, de los mercados, de tus decisiones, etc. Cuando las decisiones las empieza a tomar tu miedo a la larga tendrás más probabilidad de que vuelvas a perder todo tu capital.

Entonces, ¿Cómo enfrentas el miedo? Hay 2 maneras: dejar que el miedo te domine (que los mercados y tus emociones te dominen) o aprender a dominar tu miedo y tus propias emociones (realizar tus operaciones basadas solamente en la lógica).

En esta etapa estás luchando por lograr el control emocional, tienes deseos de operar cada día, llevas tus pérdidas al trabajo o tu familia, etc. Es decir, estás obsesionado con los mercados financieros. Apliquemos entonces la siguiente frase: “si tú quieres dominar el mundo, debes primero dominarte a ti mismo”-

Los mejores traders profesionales están en constante posición de “Pocker Face” durante su tiempo de operativa diaria, por lo que no se le mueve ni un solo músculo ni frente a las pérdidas ni frente a las ganancias. Tenga pérdidas o ganancias no las llevan a casa ni les afectarán. Creen en su estrategia ganadora.

5ª Fase: CUANDO TRAS EL MIEDO TOMA FUERZA LA EUFORIA

Lograste controlar el miedo y nada te aterra, pues acabas de salir de tu parálisis emocional creada por el miedo. En este momento te arruinas por segunda vez (volviendo a la fase 2ª y 3ª). Acaba de entrar en juego una segunda variable emocional: **la euforia**.

Ahora debes de aprender a seguir controlando el miedo, pero también la euforia. Ambas emociones deberás mantenerlas en equilibrio: son el Yin y el Yan emocional que todo trader debe controlar.

Si los mercados van en tu contra entra en juego la euforia (te sientes invencible y crees que tú siempre llevas la razón). Si los mercados van a tú favor empiezas a tener miedo porque sabes que el dinero que vas ganando no es tuyo hasta cerrar la operación (cierras la operación demasiado rápido y ganas menos de lo que podrías ganar por sentir miedo a perder lo que ahora puedes ganar).

El miedo siempre será más fuerte, y esto se aprecia cuando las masas se descontrolan en las bajadas de precios y el mercado entra en pánico. En términos de mercados financieros al miedo se le llama “**los osos**” y a la avaricia se le llama “**toros**”: *los osos son más fuerte que los toros, encargándose los osos de bajar el precio y los toros de subirlo*. Entonces los TOROS ganan en momentos de euforia y los OSOS en momentos de miedo.

Entonces, un trader en esta fase deberá **dominar el miedo** (para maximizar las ganancias) **y gestionar su euforia** (para no apalancarse en una operación perdedora hasta la muerte). Un buen trader se saldrá del mercado con la misma velocidad en que este se dé la vuelta en su contra: el más rápido es el que gana.

6ª Fase: HUYE DEL TRADING EMOCIONAL

Establecer una estrategia lógica que ponga las probabilidades de tu parte es básico, y desarrollar disciplina para cumplir tu estrategia al pie de la letra es lo que permitirá cumplir tu estrategia.

Hay un concepto fundamental que necesitas abrazar en esta profesión: **“sigue la tendencia”, “la tendencia es tu amiga”**. Tú no aprendes trading en cursos, webinars, etc, sino cuando empiezas a operar con dinero real, comienzas a observar los mercados y empiezas a perder dinero. Solo de esta manera aceptas que no tienes formas de predecir el futuro y empiezas a escuchar al mercado. Toda interpretación aceptas que estará sujeta al riesgo. En el fondo tú no sabes nada.

Comienza a plantearte trazar tu propio plan de trading y trata de mejorarlo cada día aprendiendo más y mejor a interpretar los mercados.

7ª Fase: CAMBIA TU ACTITUD Y SÉ HUMILDE

Aumentar tu humildad. La humildad no es necesaria para realizar mejor o peor trading, sino que sin ella no puedes empezar el proceso de aprendizaje sobre trading. Mientras que cuando estás en la arrogancia la culpa de tus pérdidas la tiene el mercado, cuando estás en la humildad aceptas que tú eres el único responsable de las decisiones que tomas.

En este punto solo estamos aprendiendo a construir cosas extraordinarias, dignas de la minoría que está dispuesta a ser humilde y dominar sus emociones ante el mercado.

8ª Fase: PERFECCIONANDO MI ANÁLISIS TÉCNICO

El trader que se forma en análisis técnico lo hace porque acepta que no sabe nada, y comienza a aprender a utilizar herramientas que le permiten leer los mercados financieros e interpretarlos. En este preciso momento es cuando comienzas a seguir a los mercados y a estudiarlos.

Los traders profesionales pasan la mayoría de su tiempo de trading estudiando hacia dónde se mueven los mercados y leyéndolos. Solo después de tener algo muy claro se mueve en el sentido del mercado y operan (menos del 40% del tiempo de trading diario lo gastan operando).

El análisis técnico debe de dominarse a la perfección. Sin embargo, hay muchos “traders” que utilizan un análisis técnico muy básico y aun así aseguran dominar los mercados financieros.

No puedes dominar una estrategia de análisis técnico sin con un control básico de los indicadores y la estrategia al completo: si no se entienden la estrategia al a perfección raro es que puedas obtener beneficios a largo plazo, pues ninguna estrategia es perfecta.

9ª Fase: CONSOLIDACIÓN DE UN PLAN DE TRADING

Ahora estamos por fin en posición de conocer qué hará muy probablemente el mercado, aceptamos que mi análisis está sujeto a riesgo y controlo mis emociones relativamente bien. Estamos cerca del final del camino hacia el éxito como trader.

Teníamos una estrategia más o menos buena ya definida. Ahora es el momento de perfeccionar la estrategia hacia un buen plan de trading integral. El plan de trading es el elemento crucial que vence a la emociones, y ya tienes todas las herramientas para definir tu propio plan de trading. Pero ¿qué debe reunir un plan de trading? Principalmente 5 cosas:

- 1-Metas financieras.
- 2-Metodología técnica.
- 3-Gestión del riesgo emocional
- 4-Gestión de capital.
- 5-Unificar todo lo anterior mediante el uso de un proceso mecánico y repetitivo.

Solo ganarás dinero si tu deseo es el de leer el mercado, entender el mercado, seguirlo. Jamás centres tu operativa en el deseo de ganar dinero. El negocio está en tú capacidad para vencer a las probabilidades.

10ª Fase: CONTROL ABSOLUTO DE TUS PÉRDIDAS

Cuando encuentras un adecuado plan de trading y lo empiezas a ejecutar con disciplina empiezas a tener pérdidas razonables, gestionables y compensables con las ganancias. No olvides leer libros y más libros de “Money Management” y poner esos conocimientos en práctica. Una gestión eficaz de tu *bank* es clave para asegurar tu supervivencia en la bolsa.

Tómate el tiempo necesario para perfeccionar el punto 9.

Fase Final: ENGORDA TU BOLA DE NIEVE

Esta fase es la que decide si te conviertes en un trader profesional, porque no debes descuidar la continua mejora de metodologías que apunten a construir riquezas en el tiempo de forma sistemática y más rápidamente. Tampoco olvides reinvertir tus ganancias en este mismo negocio.

Ahora estarás ganando algo de dinero de manera consistente en el tiempo, pero aun no te permite vivir de ello. Solo un 5% de la gente que inició este proceso hacia el éxito tiempo atrás ha podido llegar hasta aquí.

Aun peor, solo una quinta parte de ese glorioso 5% de afortunados podrá encontrar en el trading la libertad financiera que estaba buscando y ganarse la vida con ello. A nivel personal tampoco eres el mismo que al comienzo este proceso como trader, pues tú arrogancia a disminuido, tu humildad ha aumentado, estas capacitado para hacer una gestión de capital eficaz en cualquier negocio y posees un auto-control emocional muy superior al que tenías cuando empezaste esto. Ni si quiera tu perspectiva de enfocar la vida ahora es la misma que la de hace unos pocos meses.

Tu objetivo ahora es el de agrandar “la bola de nieve” que representa el cúmulo total de tu capital. Es decir, deberás agregar más capital sobre el capital que ya tienes invertido una vez obtengas beneficios. Cualquier libro de inversiones de éxito en los mercados financieros termina con la misma reflexión: **“el dinero se hace de manera sistemática a lo largo del tiempo, teniendo paciencia y capitalizando las ganancias a lo largo del tiempo”**. En el libro “Snow Ball” de Warren Bafer (padre de Wall Street) se comparte las razones que le llevan a ser uno de los hombres más ricos del mundo. También señala lo que a su parecer fueron las decisiones y los pensamientos más importantes como inversor.

Relájate, capitaliza y reinvierte. Después repite el proceso una y otra vez.

3.5- Los 10 grandes errores que te hacen “un trader perdedor”

1. Creer en el dinero fácil.



2. Creer que “perder dinero en los mercados financieros” es sinónimo de “fracasar en los mercados financieros”.



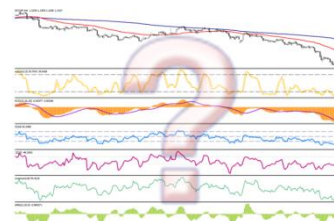
3. Improvisar las operaciones en los mercados financieros.



4. Ir contra la tendencia.



5. Abusar de los indicadores técnicos. Usa los menos posibles.



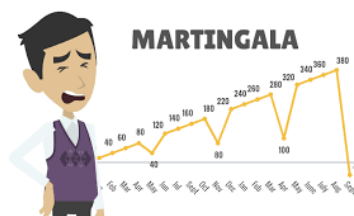
6. No colocar Stop Loss.



7. Improvisar la cantidad a invertir por operación.



8. Hacer Martín Gala.



9. Querer abordar mucho mercados y no centrar tu atención.

10. Dejar de estudiar los mercados: “cuando más se estudia de algo, menos se sabe de ello”.



3.6- ¿Después de haber seguido todos los pasos anteriores sigues perdiendo dinero?

No gana el que más indicadores usa, ni quien usa el sistema más complejo. Realmente, el que gana es aquel que focaliza, el estratega, el que concentra un gran método en algo fácil, el que entiende que con 3 líneas y 2 indicadores técnicos puede ser suficiente para generar beneficios a través del uso de un plan de trading.

La disciplina es lo que permitirá ejecutar el plan de trading sin que se mueva un solo músculo de la cara cuando se gana ni cuando pierde una operación o más operaciones seguidas.

¿Quién toma tus decisiones al operar en los mercados? ¿Tus miedos o tú lógica a partir de una seria y profunda reflexión? La disciplina es la variable clave que está presente en cualquier cosa que hagas para crear cosas extraordinarias: una dieta alimenticia, una carrera universitaria, el deporte de élite, etc. En el trading pasa lo mismo: necesitas disciplina.

¿Qué cosas en común tienen las personas que logran cosas extraordinarias? Veamos estos 6 puntos en común:

- 1-Son grandes estrategias.
- 2-Planifican cada detalle perfectamente.
- 3-Controlan sus emociones.
- 4-Gestionan sus miedos.
- 5-Ejecutan un plan predefinido.
- 6-Perseveran y tienen paciencia hasta lograr sus objetivos.

No puedes creer entonces jamás a nadie que te diga que ganar en los mercados financieros es fácil, ni que obtendrás ganancias en poco tiempo.

En el trading no solo hay 2 opciones (las de comprar o vender), sino 3 opciones: comprar, vender o abstenerse de operar. El trader desarrollará la paciencia de un franco tirador esperando la oportunidad perfecta para entrar y salir del mercado con el máximo beneficio posible, estando en el mercado cuanto menos tiempo mejor: “pocas operaciones de gran beneficio”.

“La diferencia entre el hombre rico y el hombre pobre está en cómo usa la persona el tiempo libre y su dinero” (frase sacada del libro **PADRE RICO PADRE POBRE**).

3.7- Bolsas mundiales más importantes y su interconexión

Las principales bolsa mundiales, colocadas por orden horario geográfico y de apertura, son las siguientes:

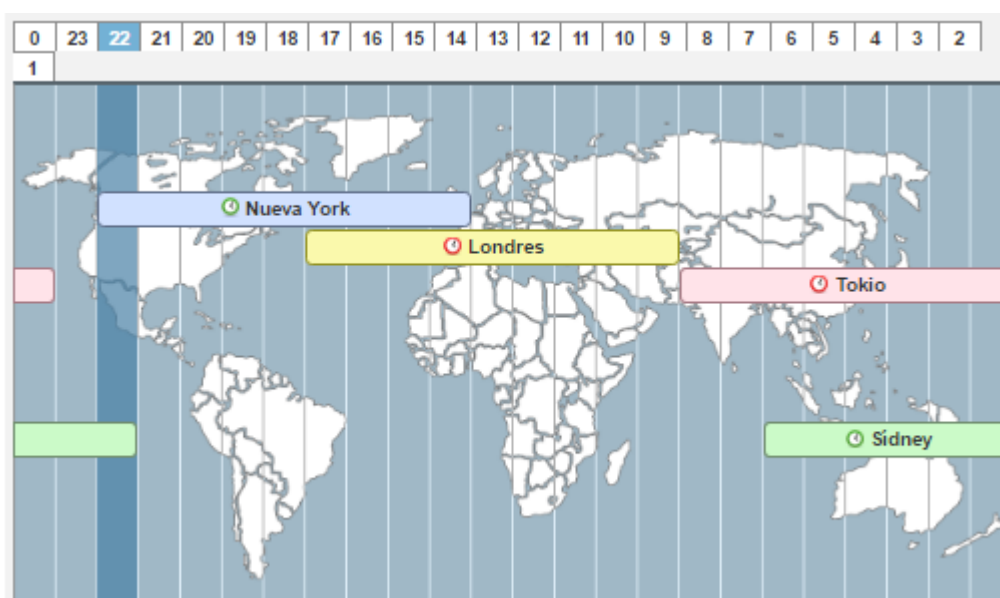
1º) Sídney

2º) Tokio

3º) Londres

4º) Nueva York

En la siguiente imagen podemos ver en detalle cuándo abre cada mercado y su franja horaria, considerando la hora de Madrid (España):



¿Se mueven todas las divisas a la misma hora de la misma manera? Por supuesto que NO. La bolsa de China es la que inicia los movimientos diarios de las divisas europeas y americanas (EUR, USD y JPY). Veamos qué pasa a lo largo del día frecuentemente según se abren los mercados mundiales:

➔ **Tokio** abre a las 00:00h y cierra a las 9:00h.

➔ **China** abre a las 3:00h.

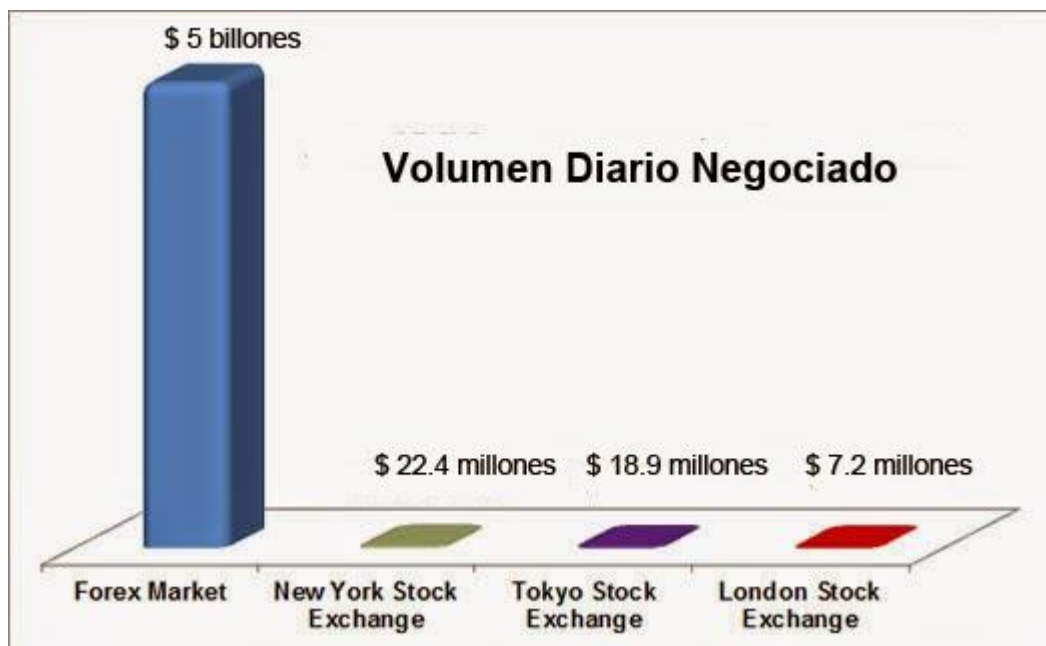
- Los mercados se inician cada día a moverse por los traders de Asia: Entre las 8:30 y 9:30h hay una lucha entre los mercados de Asia y Europa por cerrar unos y abrir otros lo mejor posible. Esta hora es llamada “La Ventana de Europa”.

→ **Londres** abre a las 9:00h y cierra a las 18:00h.

→ **Wall Street** abre a las 14:30h y cierra a las 22:00h.

¿Cuál es entonces la mejor hora para operar cada divisa? Puesto que mis 3 divisas preferidas mantienen al EURO en común, las 3 podrán operarse de 9 a 18h. No obstante, recomiendo operar de 9 a 14h en España, porque así se opera solo por traders de Europa, y todo es más simple. Si atendemos a las contra-monedas de las divisas sería: EUR/GBP mejor operarlo por la mañana, EUR/USD por la tarde, EUR/JPY por la noche. Dicho lo anterior, operar de 9 a 14h simplifica mucho el análisis financiero.

Comparemos ahora las bolsas mundiales fijándonos en el volumen de capital que mueve diariamente en cada una de ellas. Se ordenan de mayor a menor capital de izquierda a derecha. FOREX mueve más capital diariamente que la suma conjunta de capital que se mueve en el resto de mercados mundiales. La diferencia es abrumadora:



Dentro del mercado FOREX no todas las monedas se comercializan lo mismo. Los operadores tienen divisas preferidas. La moneda más comercializada es el Dólar Americano (USD), seguida por el Euro (EUR), el Yen japonés (JPY) y la Libra Esterlina (GBP) en ese orden. El Yuan chino no significa que no se comercie, sino que los operadores chinos prefieren operar el Yen japonés y aunar fuerza China y Japón en divisas ajenas al Yuan chino.

Veamos una gráfica que aclare visualmente cuál moneda se comercia más y cuál menos:



Por tanto, será prudente operar principalmente las divisas EUR/USD, EUR/GBP y EUR/JPY, por ejemplo, como ya he mencionado anteriormente. Estas divisas en concreto son las que uso yo para realizar mi operativa diaria. Cuando el Euro sube o baja en una de mis 3 divisas suelen ocurrir lo mismo en las otras 2, aunque no con la misma velocidad.

3.8- Introducción al Análisis Técnico de los Mercados Financieros

En esta sección se van a dar 9 principios o ideas básicas para realizar un buen análisis Técnico de mercados:

1. El mercado no está todo el tiempo haciendo lo mismo, es decir, el 70% del tiempo está en tendencia (alcista o bajista) y el 30% del tiempo está en lateralidad o retroceso.

2. El mercado no se mueve linealmente, sino que describe “ondas” no lineales. El estudio de dichas ondas recibe el nombre de “ondas de Eliot”. Estos estudios revelan que generalmente los mercados tienen avances a favor de la tendencia, retrocesos en contra y, a veces, interferencias impulsivas:

→La **tendencia** principal tiene **3 fases a favor** y **2 fases en contra** de la tendencia.

→El **retroceso** tiene **3 fases a favor** de dicho retroceso y **2 en contra** del retroceso.



3. La “Teoría de Dow” nos permite 1º detectar si estamos ante una tendencia y si es alcista o bajista. Después podremos determinar hasta dónde llegará un retroceso de la tendencia. Un buen uso de la Teoría de Dow nos dará la oportunidad de entrar justo a favor de la tendencia y salir antes de que esta se vuelva en nuestra contra.

4. Conocer el uso de indicadores y osciladores. Al menos debemos conocer uno que mida la fuerza de la tendencia (MACD) y otro que nos indiquen la probabilidad de cambio de tendencia (RSI).

5. Aprender a colocar líneas de soportes y resistencias relevantes del precio, realizando para ello únicamente un análisis visual de la gráfica. Esto permitirá establecer niveles objetivo del precio. Este tipo de trading lo llamo “trading mediante búsqueda de objetivos”. Los ratios de B/R (beneficio-riesgo) deben ser siempre >2 .

6. Utilización de 2 o más medias móviles que se cruzan. Tener claro que esta técnica no es útil en mercados que no se encuentren en tendencia. La otra dificultad reside en la correcta elección de los periodos de cada una de las medidas móviles. En mi opinión, las medias móviles son los peores indicadores porque se debe ser experto en esta técnica para no perder oportunidades por entrar tarde.

7. Las Bandas de Bollinger están definidas por 3 medias móviles y utilizan periodo de 20 periodos con desviación típica = 2. El indicador está dentro del grupo de “osciladores” técnicos, como cualquier media móvil. Tendrá banda superior, banda inferior y, en el medio, la banda central. Su interpretación es la siguiente:

→La media central define el VALOR que debería tener dicho mercado., es decir, si el precio actual se encuentra por encima de la media central el producto está sobre-comprándose, y si el precio está por debajo de la media central está sobre-vendiéndose. Si el precio toca la banda superior estamos en una situación clara de sobre-compra, pero si el precio toca la banda inferior de Bollinger estaremos en situación real de sobre-venta. También puede interpretarse una situación de lateralidad financiera si el precio toca rápidamente la banda inferior y superior de Bollinger alternando ambas y pasando rápidamente por la banda central: se mueve entre un máximo y un mínimo sin establecer una tendencia clara.

Cuando estemos seguros de estar en una tendencia clara, demos comprar cuando el precio toca la banda superior y vender cuando toca la banda inferior.

8. Velas Japonesas: la idea se basa en expresar lo más visualmente posible el valor que el precio ha ido tomando en una franja temporal finita. En función de la forma que toma una vela japonesa usted va a poder conocer la siguiente información del precio:

- Si ha subido o bajado el precio en ese periodo temporal.
- Precio al inicio del periodo.
- Precio al cierre del periodo.
- Precio máximo.
- Precio mínimo.

Estas formaciones o patrones de las velas pueden indicar que haya probabilidad de que el mercado se dé la vuelta, que continúe si avanza, que está en un nivel de soporte o resistencia, que experimentó el precio un fuerte rechazo en un nivel anterior, etc. Los tipos de vela japonesa, más usados es el tipo “Dojhi”, “martillo”, “ahorcado” y “envolvente”.

9. Niveles de Fibonacci: estos niveles son trazados en la gráfica de precios de un máximo a un mínimo elegido. El potencial de esta herramienta radica en que se irán desplegando las zonas aproximadas de los niveles del precio de alto interés para la mayoría de los inversores. Podría decirse entonces que con los niveles de Fibonacci podemos hacer una proyección de en qué nivel el precio va a encontrar niveles de soporte y en cuáles niveles de resistencia.

Los niveles más importantes son los de 100%, 61,8%, 50%, 23,6% y 0%. A mayor retroceso experimentado por el precio mayor dificultad habrá para que continúe la tendencia tras este retroceso. Se puede decir que es de gran utilidad porque no solo lee la intención del mercado, sino que además tiene gran tasa de eficacia porque la comunidad de traders mundial lo utiliza bastante y todo el mundo va mover el precio al mismo sitio.

3.9- Ejemplos de Análisis Financiero

1) **Analista Técnico:** la lectura y proyección de futuros valores de los mercados financieros se realizan utilizando indicadores técnico, es decir, con funciones matemáticas. Algunos indicadores técnicos más comunes son:

Medias Móviles



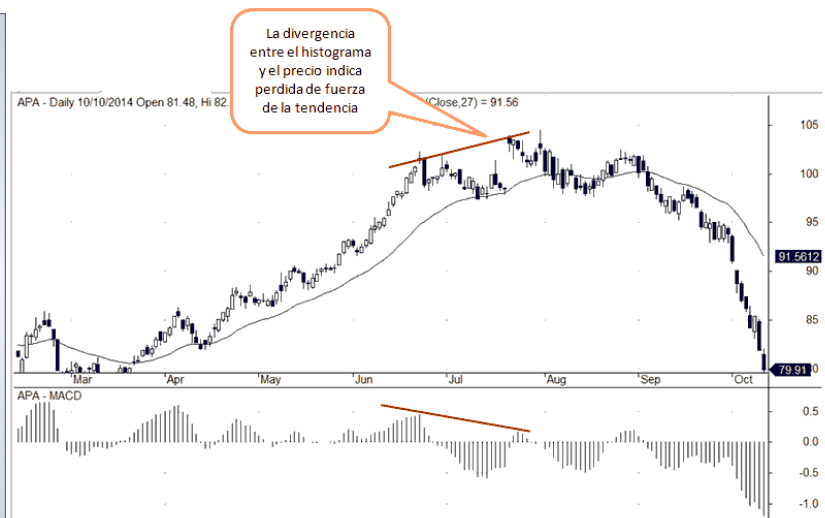
Bandas de Bollinger



RSI

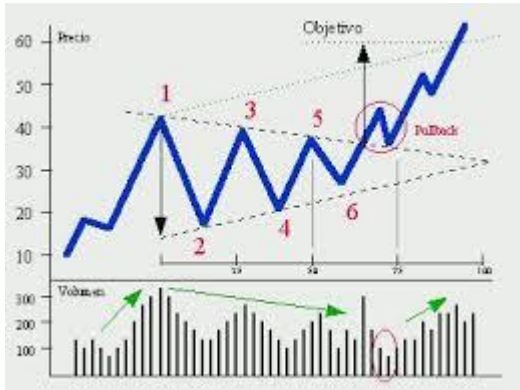


Divergencias con MACD



2) **Analista Gráfico:** (Chartista): operar los mercados estudiando las formas gráficas que forma el precio: triángulos, hombro-cabeza-hombro, etc.

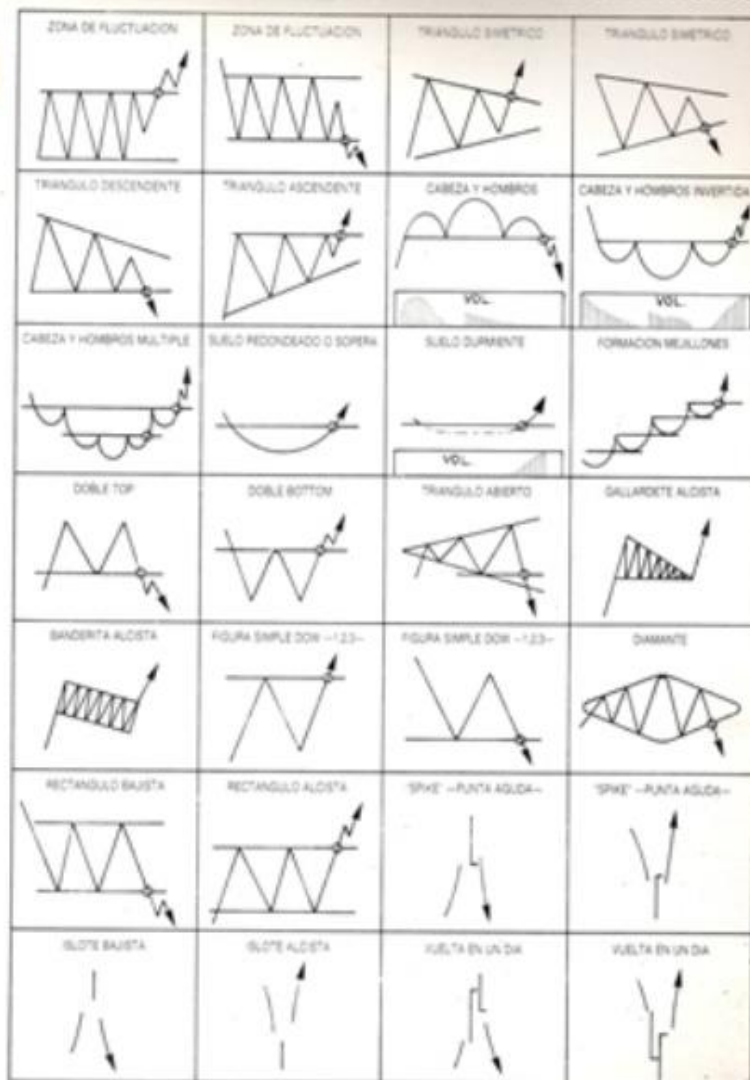
Triángulo alcista:



Hombro-Cabeza-Hombro:



ESQUEMA DE LAS MAS USUALES FIGURAS CHARTISTAS



(Escapes al alza deberían estar acompañados de volumen)

3) Analista Fundamental: es el análisis de noticias fundamentales. Podemos ver en algunas webs las noticias que se esperan tener. Cada noticia va acompañada de un índice de importancia, y como es de esperar los mercados se mueven más ante noticias d alta importancia.

Un calendario de noticias económicas útil para realizar este tipo de trading podemos encontrarlo en <https://es.investing.com/economic-calendar/>. Aparece un semejante a este, viniendo representada la importancia de cada noticia por la cantidad de toros que le acompañan (entre 1 y 3):

Calendario Económico	Festivos	Calendario de resultados	Conversor de divisas	Calculadora de Fibonac
00:50	JPY		Reservas en moneda extranjera (USD) (Feb)	1.232,3B 1.231,6B
01:01	GBP		Índice de ventas al por menor del BRC (Anual) (Feb)	-0,4% 0,2% -0,6%
04:30	AUD		Decisión de tipos de interés (Mar)	1,50% 1,50%
04:30	AUD		Decisión sobre tipos del Banco de la Reserva de Australia	
04:45	JPY		Subasta de deuda a 30 años (JGB)	0,907%
Aprox.	CNY		Reservas en moneda extranjera (USD)	2,969T 2,998T
06:00	JPY		IPC subyacente del BoJ (Anual)	0,1%
06:00	IDR		Reservas en moneda extranjera (USD) (Feb)	116,90B
07:00	ZAR		Reservas de divisa extranjera (USD) (Feb)	46,73B 46,67B
07:00	ZAR		Reservas netas de divisas (USD) (Feb)	41,250B 41,373B
08:00	EUR		Pedidos de fábrica de Alemania (Mensual) (Ene)	-2,5% 5,2%
08:45	EUR		Balance presupuestario de Francia (Ene)	-69,0B
09:30	GBP		Índice Halifax de precios de la vivienda (Mensual) (Feb)	0,3% -0,9%
09:30	GBP		Índice Halifax de precios de la	5,3% 5,7%

3.10- Gestión del riesgo

En cualquier sistema de contratación o cualquier intercambio de bienes y productos debe de tenerse en cuenta 3 factores:

- 1-El pronóstico del precio que tendrá en un futuro próximo (saber qué hacer).
- 2-Cálculo del momento adecuado de la compra, para maximizar beneficios (**saber cuándo hacerlo**).
- 3-Gestión monetaria adecuada y establecimiento de stop de pérdidas en caso de llegar a una cantidad determinada (**cantidad que comprometo por operación**). Ésta última requiere gran disciplina.

La parte más difícil de una transacción financiera en mercados financieros no es la estrategia que desarrolles, sino tener una eficiente gestión monetaria. El tamaño de la cuenta y el porcentaje de la cartera que queda comprometido en el mercado cada instante afecta a los resultados finales, y mucho. Por tanto, la gestión del riesgo o gestión monetaria incrementa la probabilidad de que el operador sobreviva en los mercados y llegue al largo plazo.

Algunos de los **pilares principales** de la gestión del riesgo podrían ser:

- 1) Solo podrás disponer del 50% de la cartera que tengas en tu bróker para invertir. La otra mitad no se utilizará bajo ningún concepto, excepto en periodos de mucha adversidad.
- 2) El máximo invertido simultáneamente en cada mercado será del 5% de nuestra cartera, para cada mercado.
- 3) Como en bolsa hay inversión inicial nominal (fija) luego hay una pérdida o ganancia marginal (extra y variable), podemos necesitar fondos extras para cubrir pérdida. Por eso en cada mercado se debe reservar un depósito de margen de 10% a 15% de la cantidad de mi cartera para cubrir pérdidas.
- 4) El capital invertido en un grupo de mercados relacionados debe limitarse al 20%. Ejemplo: EUR/USD, EURGBP y EUR/JPY.
- 5) La cantidad invertida por operación debe ser $\leq 2\%$ de mi cartera de inversión.

La idea más extendida para reducir riesgo es la de “**diversificación**”. En realidad funciona muy bien para reducir riesgo, pero en ocasiones podría ser excesiva. Esto ocurre cuando tenemos muchos mercados abiertos y las operaciones ganadoras se ven diluidas por una mayoría de operaciones perdedoras. Estudiando las gráficas podemos ver los mercados que tienen mayor correlación entre sí, y a MENOR correlación mayor diversificación obtendremos operándolos.

Hasta aquí se ha definido la gestión monetaria preventiva y más básica. Existe también lo que llamamos “gestión de riesgo en la operativa”, que también afecta directamente al riesgo monetario. Estamos hablando de la colocación de stop loss y stop de beneficios de forma dinámica sobre los propios mercados. Para realizar una buena **colocación de Stops** el operador debe tener siempre en cuenta que la **relación B/R** (Beneficio / Riesgo) entre precio objetivo y retroceso sea > 2 , y de esta manera poner las probabilidades de su parte cuando decida entrar en una operación. Esta tasa se suele llamar “coeficiente B/R”.

Las gráficas de beneficios de los traders suelen describir picos y valles de valor, como ocurren en las gráficas de los mercados. Entonces, ¿cuánto invertir después de pérdidas o beneficios? **Lo ideal es NO invertir siempre lo mismo por operación, sino tomar esa decisión según en la fase en que está mi rendimiento.** En otras palabras, si vemos que mi gráfica de beneficios, como trader, ha superado un máximo anterior debo de ir reduciendo la cantidad invertida y no aumentarla. Los traders inexpertos no tienen en cuenta este análisis y no sólo invierten siempre la misma cantidad por operación, sino que según van ganando operaciones reinvierten los beneficios de forma exponencial: a la larga arruinan sus cuentas. Por el contrario, si estás en pérdidas pasado un cierto tiempo podrás ir aumentando la cantidad invertida por operación. Si no realizas un trading emocional terminarás obteniendo rápidamente beneficios de nuevo.

Respecto a técnicas de contratación múltiple combinadas con la gestión del riesgo también hay mucho que decir. Existen técnicas para “exponenciar” beneficios y cubrir pérdidas en un mismo mercado que se encuentre tendencial claro (alcista o bajista). Por ejemplo se trataría de hacer lo siguiente: un mercado esté alcista y me genere 10€ de beneficios, por lo que en ese caso colocaríamos otra orden alcista de 5 Euros y acercaríamos el nivel de STOP-LOSS al nuevo precio. De esta forma colocamos los StopLoss de ambas operaciones próximos al nuevo precio, y si sigo generando beneficios podré sacar 30€ o si pierdo sacar solo 3€ de ganancias. En ambos casos ganaría algo, y el B/R sería de orden > 2.5 .

Tener en cuenta para todo lo anterior los siguientes factores del análisis técnico:

- 1-Líneas de ruptura del precio.
- 2-Ructura de líneas de tendencia.
- 3-Uso de soportes y resistencias.
- 4-Localización de posibles niveles de retroceso.
- 5-Considerar GAP's por futuras noticias fundamentales imprevistos incontrolables.

Por último se describen los 5 tipos de órdenes posibles de realizar en el mercado:

1-Orden al mercado: orden que se realiza de compra o venta al precio actual con el mejor precio que se pueda y lo más instantáneo posible.

2-Orden limita: orden programada para que se ejecute cuando l precio llegue al valor definido por el cual el operador está dispuesto a aceptar una orden. Ejemplo: el Eur/Usd está a 1,05074 y programo una orden de compra cuando el Eur/Usd valga 1,05000.

3-Stop: una orden de stop sirve para establecer los niveles en los cuales deberá ejecutarse una orden de salida del mercado automáticamente. Sirve para establecer stop de pérdidas o stop de beneficios (asegurar beneficios).

4-Stop limitada: combina una orden de Stop con una orden limitada. Funciona como una orden STOP, pero cuando se ejecuta se convierte en limitada, es decir, sirve para controlar stop de beneficios y stop de pérdidas simultáneamente.

5-Orden por mejor precio si toca.

Entonces, en los mercados gana el trader más rápido, no el más inteligente. Si combinas la rápida anticipación ante los sucesos fundamentales de mercados con una buena gestión monetaria irás por el buen camino hacia el éxito financiero en el mundo del trading. Si no logras el éxito, al menos será muy probable que sobrevivas en los mercados y que no pierdas tu dinero.

3.11- Tipos de comisiones presentes operando CFD's, Opciones Binarias y Acciones.

Acciones:

En acciones tienes un precio fijo por transacción para el acceso al mercado. Esto quiere decir que vale lo mismo en comisiones comprar 8 acciones de Danone de un vez que comprar sólo una. Un caso semejante ocurre con el banco INGDIREC.COM. Por lo demás no tienes que preocuparte, ya que sólo puedes perder lo que te gastaste en la acción, aunque la empresa quiebre.

Opciones Binarias:

En acciones binarias las comisiones son mucho más elevadas. Sabemos que te pagan, en media, el 75% de lo que inviertes en caso de que aciertes el sentido del mercado, y que te quitan el 100% si te equivocas. Como hay un 50% de probabilidades de que acertar, la esperanza matemática resultante termina siendo de -12,5%. Podríamos decir entonces que las comisiones en opciones binarias son del 12,5%. ¡WAUH!

CFD:

Aquí las comisiones son diversas, y algunas son explícitas y otras implícitas, es decir, unas las impone el bróker y otras el mismo mercado. Veamos un resumen de estas:

→ **Spread:** Es la diferencia entre el precio real y el que te da tu bróker, es decir, el margen de beneficios que se lleva tu bróker al comprarles o venderles una operación en bolsa. Ejemplo: tú realizas una operación en Euro/Dólar alcista y compras un LOTE a 1'05099, pero el precio real que había era de 1'05000 y por eso tu bróker se embolsará 0,1% en comisiones de este tipo.

→ **Swap**: es el tipo de comisiones extra que te cobran o te “pagan” por hacer trading por la noche, es decir, mantener tu operación financiera 24 horas al día. Esta comisión va en función de la liquidez que tenga el mercado operado en cada instante. El valor del SWAP suele rodar entre el 2 a 2'5%. En principio te cobran esa comisión si estás posicionado con una operación alcista (compra), pero también te **pagan** esta comisión si estás posicionado bajista (venta). Sin embargo, no suelen pagar esta comisión muchos bróker, aprovechando la opacidad y confusión de rentabilidades y comisiones existente. Logran así pagar al cliente lo menos posible.

→ **Apalancamiento financiero**: es un préstamo que el cliente establece con su bróker y suele ser elevado. Este préstamo no es opcional, pues la entidad reguladora británica FCA permite que los brokers fijen préstamos de orden 10 de la cantidad que invierta el cliente. Esto significa que si yo invierto como cliente 1€ mi bróker que presta 9€ para que llegue a invertir en el mercado un bloque de 10€, de los cuales 9 de esos 10 no los tengo. Se ha demostrado que los traders con más rentabilidad son los que corren menos riesgo, y esta práctica potencia el riesgo. Los brokers pinta este servicio como algo favorable para el cliente, pero nada más lejos de la realidad.

→ **GAP**: los movimientos del mercado no son lineales, y tienden a ser impulsivos, es decir, repentinos, bruscos e inesperados. Un GAP es un salto repentino y brusco del precio del mercado. Este fenómeno se considera como una comisión implícita del mercado, pues está en la naturaleza del mismo y representa un riesgo muy real. Dicho esto es importante mencionar el uso de STOP LOSS (o stop de pérdidas), que si ocurre un GAP se saltará este STOP LOSS y correremos un riesgo más elevado que el deseado inicialmente. En brokers como IG.com se permiten StopLoss GARANTIZADOS a cambio de altas comisiones, pero casi no sale rentable utilizar este servicio.

→ **Retrasos de Operación:** la operación que un usuario coloca en la plataforma de su bróker no le llega al servidor instantáneamente, es evidente. Este retraso puede deberse a la conexión lenta de tu wifi, a la plataforma con baja eficacia de tu bróker o por querer colocarte tu bróker retrasos extra adrede. Esto lo hacen para tratar de morder más cantidad de comisiones del cliente. Todo es una carrera contra las mordidas de capital antes de ver un solo euro. ¿Cómo lo logran en este caso? Simple, así: si retraso tu operación la ganancia que esperabas será menos porque el recorrido del precio hasta tu objetivo ya fijado también será menor (en caso de tener razón el mercado ya se empezará a mover en el sentido que indicaste). Si pierdes la operación perderás menos, pero como el 95% de la gente pierde dinero eso no supone un problema. De esta manera tienen un 95% de gente perdedora, y del 5% de ganadores muerden un cantidad suficiente para no permitir que sea fácil de mantenerse mucho tiempo.

3.12- Comparativa de rentabilidad mensual: Opciones Binarias Vs CFD's.

Para realizar el análisis comparativo de rentabilidad he creado diversos programas en Java, los cuales permiten realizar capturas de pantalla automáticas, cada cierta temporalidad. Estos programas se explican en la sección 5.1 y 5.2. Se han realizado más programas del estilo, pero esos 2 programas son los más relevantes.

La temporalidad elegida con mayor relevancia en la rentabilidad es la de 1 hora o la de 5 minutos. Sin embargo, si tuviera que elegir entre una de ellas yo elegiría la temporalidad de 1 hora, pues a mayor rango temporal considerado el bróker podrá manipular menos el precio real en su plataforma.

Por tanto, una vez al día analizaba las fotos realizadas. El tiempo muestral del análisis fue menor a un mes, pues fue suficiente para obtener un patrón común de comportamiento en las gráficas de las divisas Eur/Usd, Eur/Gbp y Eur/Jpy. Una vez localizados varios patrones de comportamiento periódicos se ha podido optimizar los puntos de entrada y de salida del mercado.

Las consideraciones realizadas, según la estrategia de mercado generada tras este análisis, fueron las siguientes:

- 1) Se utilizará una estrategia propia de la web www.vivirdeopcionesbinarias.com.
- 2) Esa estrategia será utilizada en gráficas de velas japonesas de timeFrame de 1 hora.
- 3) Deberemos entrar al mercado, tanto en una operación de CFD's como de Opciones Binarias, cuando veamos que recién se ha cerrado la vela anterior con una flecha encima indicando un sentido alcista (verde) o bajista (roja).
- 4) Hacer caso a la flecha si además de eso hay una probabilidad mayor al 75% del mismo color que la flecha, en el indicador de probabilidades ubicado en la parte superior izquierda de cada mercado.
- 5) El punto de salida será temporal, es decir, cuando cierre la vela de 1 hora actual deberá ejecutarse una orden Stop Profit. Con ello saldré del mercado en ese instante.

Nótese que la estrategia buscaba ser válida para CFD y Opciones binarias simultáneamente, y que se creó para este apartado en concreto. Esto requiere una añadida comprensión de los mercados financieros, pues no es simple llegar a una estrategia financiera de éxito siendo novato. Ni si quiera en DEMO.

Veamos ahora las cifras semanales:

- ➔ 66% de rentabilidad semanal en operaciones de **CFD's**.
- ➔ 40% de rentabilidad semanal en **Opciones Binarias** (suponiendo que cuando gano me retornan un 75% de beneficio, más mi inversión).

Procedemos ahora a la demostración de estas cifras con capturas de pantalla de los mercados durante la semana del 28 de noviembre de 2016 al 2 de diciembre de 2016:

DÍA 1:

En opciones binarias: 1 operación hecha, de la cual 1 ganada.

En CFD: 1 operación, con un total de 109 pips ganados.

A tiempo pasado la cosa queda así:



DÍA 2:

En opciones binarias: 3 operaciones hechas, de las cuales 3 ganadas.

En CFD: 3 operaciones, con un total de 195 pips ganados.

A tiempo pasado la cosa queda así:



DÍA 3:

En opciones binarias: 2 operaciones hechas, de las cuales 2 ganadas.

En CFD: 3 operaciones, con un total de 370 pips ganados.

A tiempo pasado la cosa queda así:



DÍA 4:

En opciones binarias: 1 operación hecha, de la cual 1 perdida.

En CFD: 1 operación, con un total de -97 pips perdidos.

A tiempo pasado la cosa queda así:

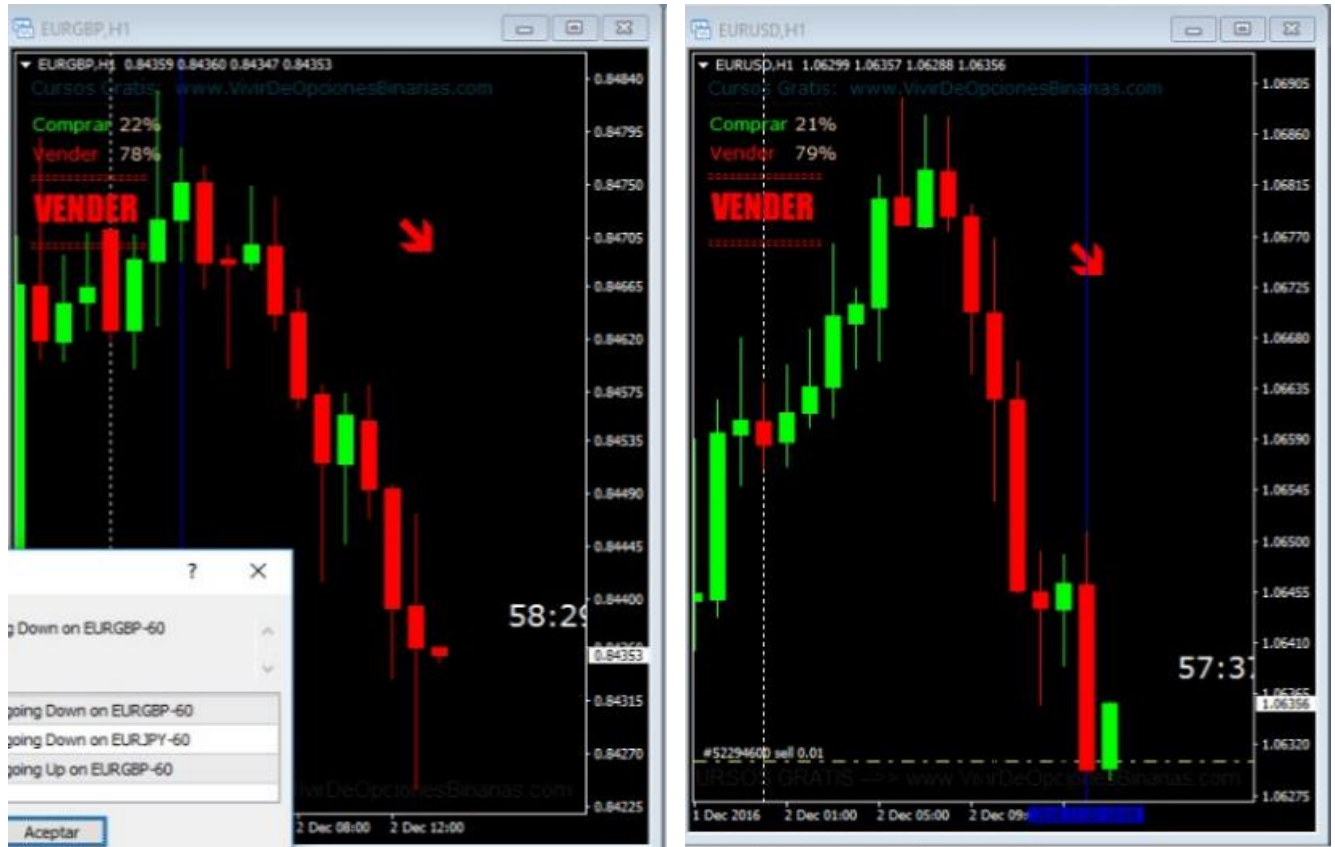


DÍA 5:

En opciones binarias: 3 operaciones hechas, de la cual 2 ganadas y 1 perdida.

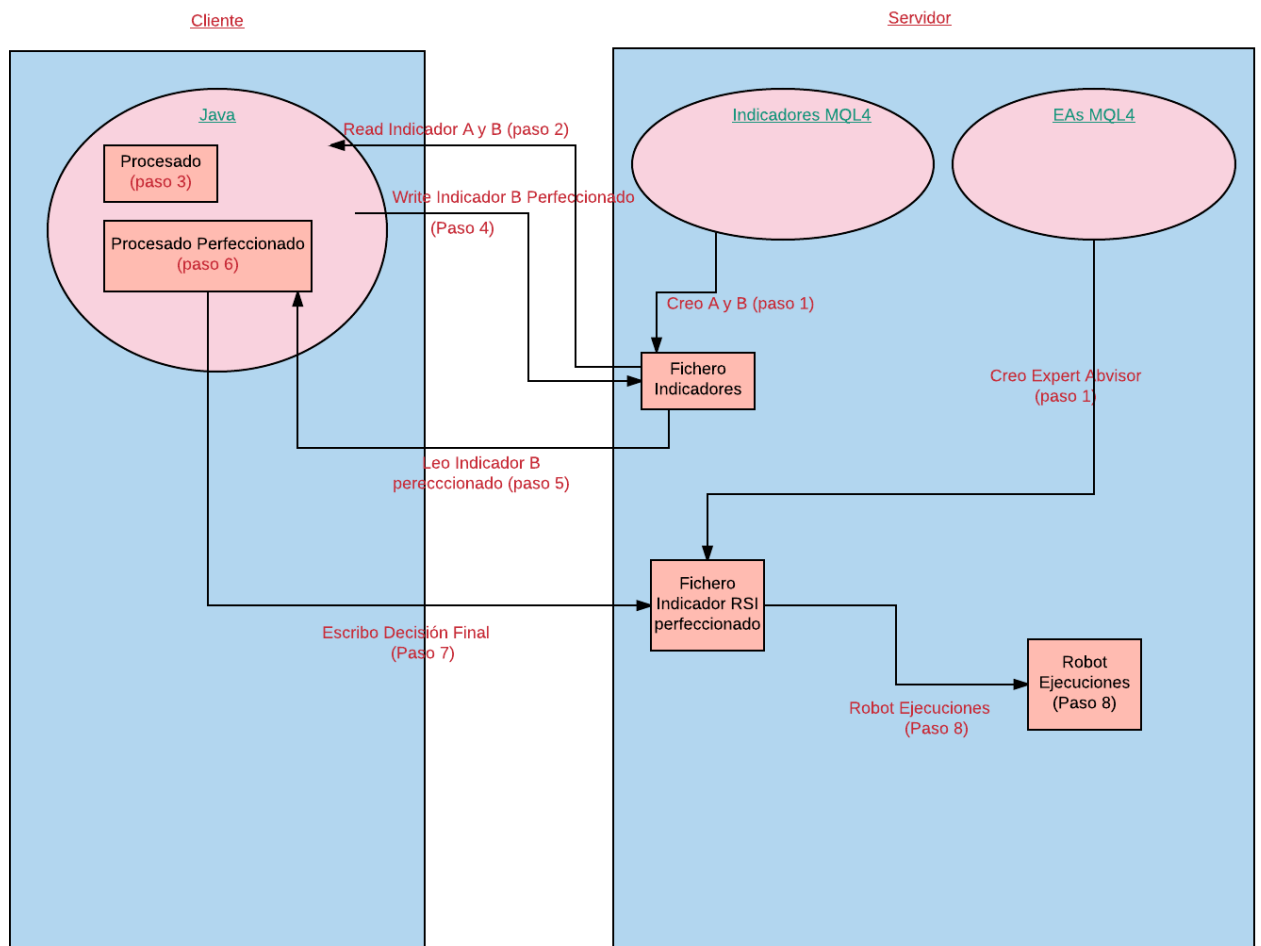
En CFD: 3 operación, con un total de +66 pips ganados.

A tiempo pasado la cosa queda así:



4. Diseño general de la APP

4.1- Visión General



En esta imagen vemos la clara separación entre el cliente Java y el servidor MQL4, lo cual puede apreciarse por 2 cuadrados azules bien separados: el cuadrado de la izquierda representa al cliente y el de la derecha representa al servidor.

El paso 1 deberá ser ejecutar el lado del servidor. Esto se hará cargando los indicadores y el Expert Advisor que ejecutará MetaTrader4. Automáticamente se escribirán los ficheros que estos indicadores y EA han generado como resultado de su ejecución.

El paso 2 es ahora abrir el lado del cliente con el programa Java. Cuando todos los parámetros de la ventana Java se carguen, internamente Java leerá los ficheros de los indicadores A y B escritos por el servidor. Los indicadores A y B representan a los indicadores MACD y RSI utilizados en cada mercado deseado.

El paso 3 ocurre cuando el programa Java recibe el texto leído: recibe un String del indicador MACD ("ROJO" o "VERDE") y un dato DOUBLE del indicador RSI. Ahora el programa Java, desde el lado del cliente, procesa los datos recibidos (un string y un double).

Como es mejor comparar 2 datos del mismo tipo para tomar una decisión definitiva para operar automática en los mercados financieros, en el paso 4 se escribe un String sobre los ficheros del directorio del servidor. Esos ficheros se llamarán "String_RSI_mercado_X". Si Java ha leído un Double menor o igual a 30.00 escribimos "SOBRE-VENTA"; si ha leído un Double mayor o igual a 70.00 escribimos "SOBRE-COMPRA"; si ha leído un Double mayor que 30.00 y menor o igual que 70.00 escribimos "QUIETO".

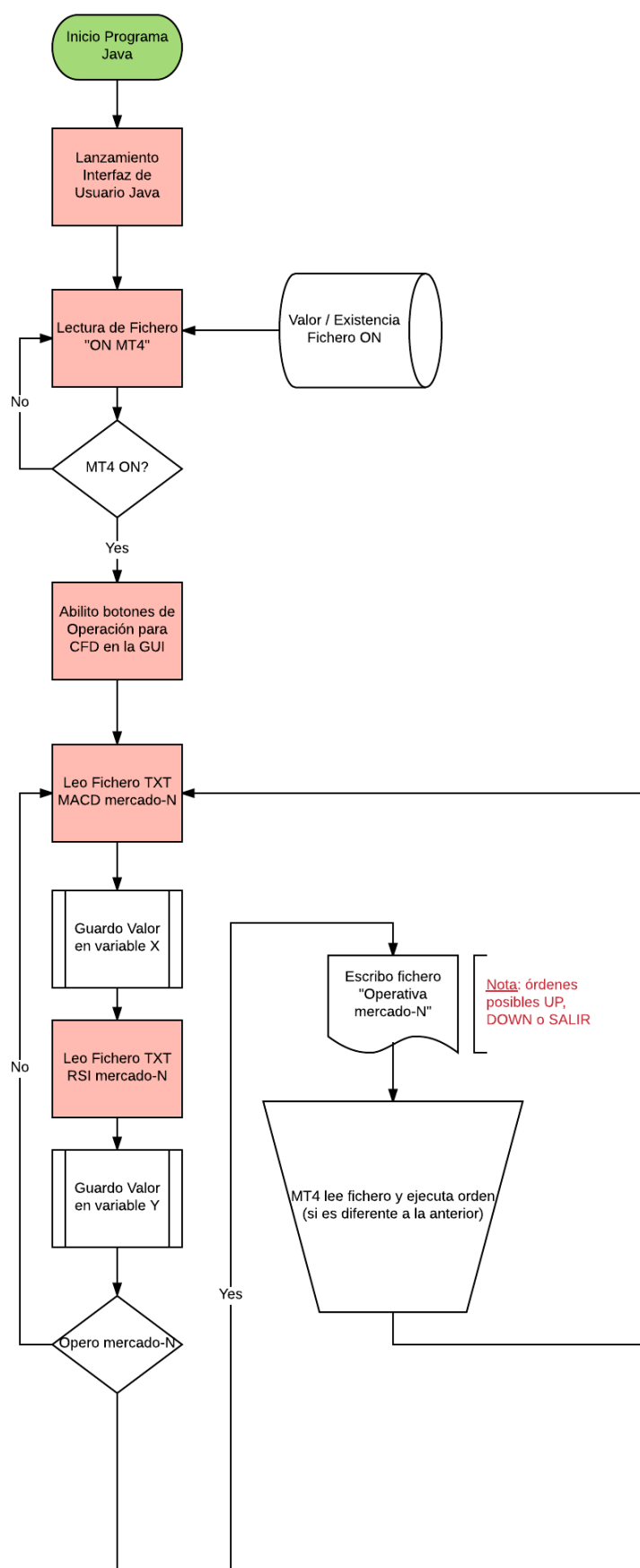
En paso 5 leo lo recién escrito por Java y lo guardo en una variable, es decir, es string escrito. Así compruebo si se han cerrado correctamente los manejadores de lectura y escritura anteriormente, y sino preparo una exception que maneje la excepción y reintente. La idea es evitar que el programa aborte sea como sea.

En paso 6 ya vamos a comparar el String de MACD (verde o rojo) con el String del RSI. De la comparativa sacamos una acción definitiva, resultado de las entradas leídas del MACD y el RSI. Podré sacar "COMPRAR", "VENDER" o "SALIR". Esta salida procesada va a ir en forma de String. Queda en una variable lista para utilizar.

En el paso 7 escribo el valor de la variable que contiene el String que muestra la decisión definitiva. Lo voy a escribir en un fichero que solo leerá el EA.

Por último, en el paso 8 desde la parte del servidor leo la orden que acabo de escribir en los ficheros de comunicación con el EA. Quien leerá la orden al mercado no será Java, sino el programa del EA que está a la espera continua de recibir datos. El EA está en la parte del servidor, recuerden. Cuando servidor reciba una orden leída por el EA, éste mismo EA se encargará de ejecutar la orden leída automáticamente.

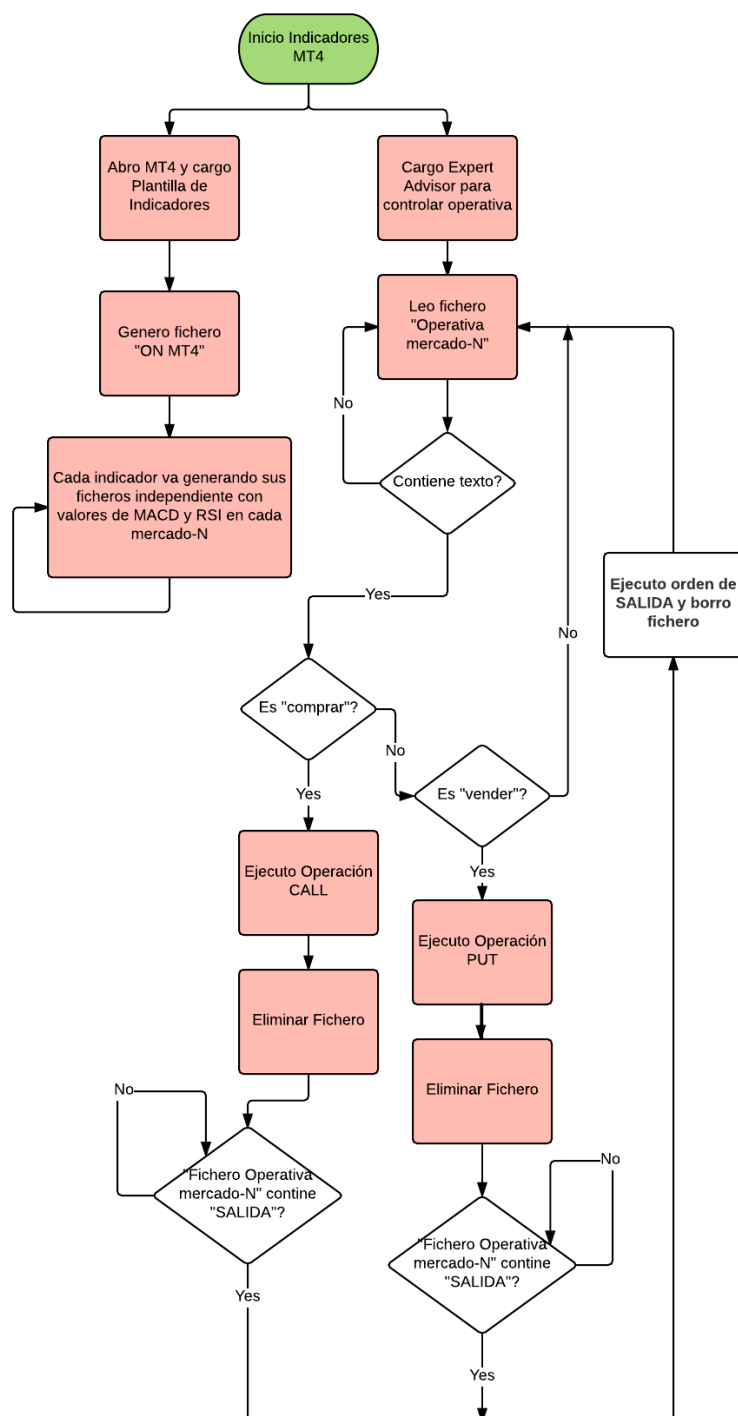
4.2- Flujo-Grama del Front-End (Java)



4.3- Flujo-Grama del Back-End (MetaTrader4)

PROGRAMA MT4

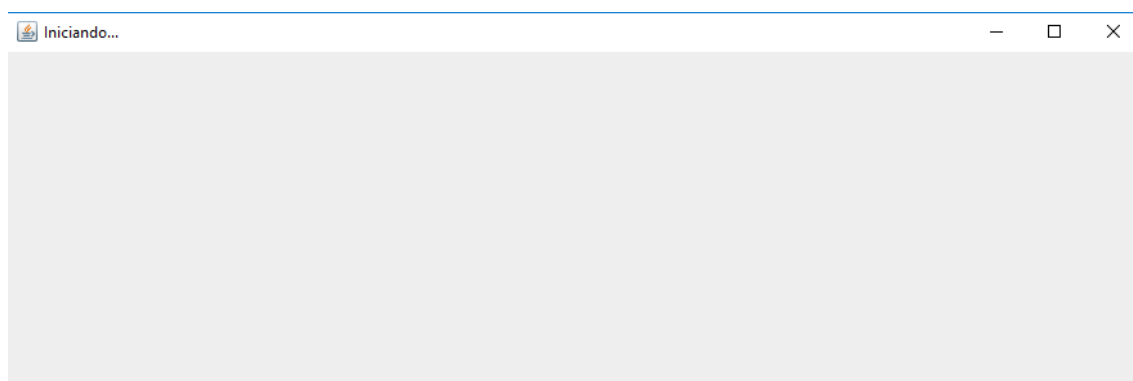
juan anto



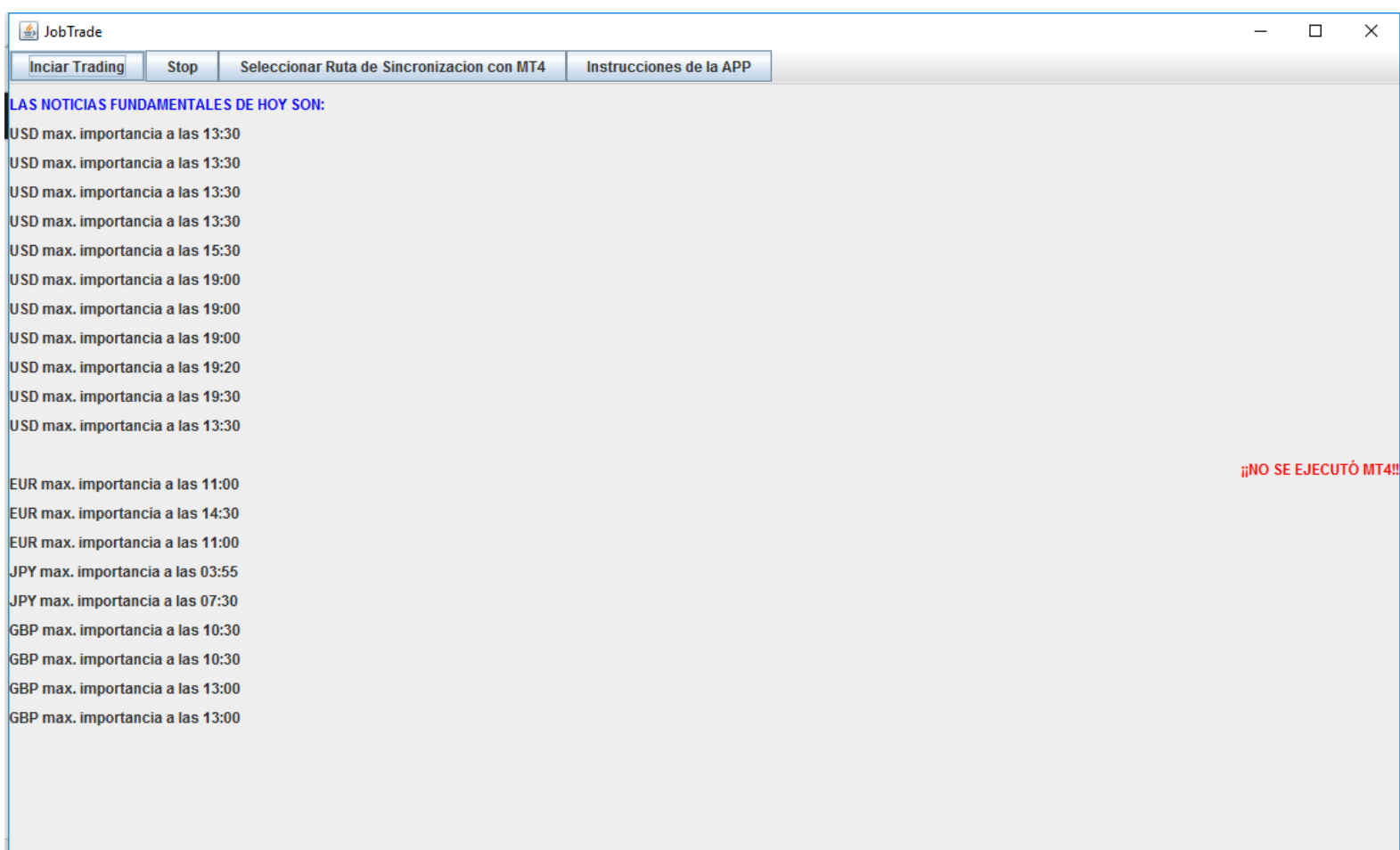
4.4- Apariencia real de la aplicación Final

La interfaz gráfica que verá el usuario de la aplicación Java será la siguiente:

1º) VENTANA INICIAL:



2º) VENTANA PRINCIPAL:



En la imagen anterior podemos ver las noticias fundamentales QUE hay durante toda una semana, aunque como indica ahí esa lista fue diseñada para ver tan sólo las noticias del día actual (HOY). La lista de noticias está programada para un máximo de 25 noticias. Puede pensar que 25 noticias es poco, pero le aseguro que ya es bastante raro el día que hay 10 noticias de máxima importancia como para que hayan 25. Si fuera habitual tener 25 noticias de máxima importancia diariamente (referente al Euro, Libra y Yen) los precios de los productos no llegarían a estabilizarse, y la inflación superaría el 8%. Eso produciría una inestabilidad económica importante.

Después de seleccionar un mercado de los 3 posibles deberemos de indicar la ruta donde buscar el archivo que genera MetaTrader4, un fichero para cada mercado que tenga los indicadores y el Expert Advisor activo. Si el programa Java encuentra el fichero, la etiqueta que ahora esta **ROJA** a la derecha de la interfaz deberá de ponerse a **VERDE**, y podrá iniciarse el programa automático. Para iniciar el programa debe pulsar **“Iniciar Trading”**.

Si pulsa el botón **“Pausar Trading”** el programa de MetaTrader4 cerrará las operaciones abiertas automáticamente, estén como estén, y se quedará en un modo de espera hasta que el usuario decida reiniciar el trading.

Si el usuario pulsa **“Stop”** el programa cerrará el mercado.

Puede consultar también un texto y leer algunas instrucciones de cómo funciona la APP pulsando el botón **“Instrucciones de la APP”**.

5. MetaTrader 4

5.1- Toma de datos en Tiempo Real con MetaTrader 4

Actualmente hay multitud de herramientas que te proporcionan información en tiempo real de los mercados financieros, es decir, el valor de las cotizaciones en tiempo real y las gráficas de los mercados correspondientes.

Como ingeniero, yo siempre busco desarrollar mis propias herramientas para lograr ahorrar gastos de servicios de terceros. Para ello, llegar a concretar la información y discernir la información útil de la que solo es “basura” me ha tomado aproximadamente un mes de estudio dedicado. En este apartado se detallan las conclusiones de la herramientas encontradas y los motivos que me han llevado a elegir las, como la relación entre costos y calidad de servicio deseado.

Una vez ya metidos de lleno en el tema de las diversas herramientas de streaming (para obtener cotizaciones de mercado en tiempo real) existen 2 vías principales: usar librerías o Frameworks y usar programas que realicen todo el trabajo software por el usuario a cambio de un costo. La primera opción requiere conocimientos de nuevos comandos de programación para conectarnos a según que servidor de mercado, y hay demasiados servidores diferentes con software propietario cada uno. La segunda opción solo requiere localizar un buen programa que nos entregue los datos de las cotizaciones lo más rápido, eficiente y barato posible. Todos los servidores principales de mercados los encontramos en 2 plataformas: Forex y YAOOH. Veamos como conectarnos al servidor:

- A) **El uso de librerías** no me convenció, pues me llevaba demasiado tiempo de aprendizaje conseguir utilizar una librería de programación de forma eficiente, encontrar como entrar a cada servidor si tengo que batir al firewall, etc. Recuerde que se necesita enviar peticiones de cotizaciones reales periódicamente (varias por segundo a ser posible). El mejor framework gratis lo ofrece Yahoo, pero con cierto retraso y con solo 5 decimales por valor cotizable (se opera con 6). En el mundo financiero “la información de calidad se paga”. Frameworks o librerías de programación quedan entonces descartados.

- B) **Conectarnos a un servidor Forex desde un programa** asequible y rentable. El programa que guarda la mejor relación Usabilidad/Precio es MetaTrader4 (o MT4). Tras un mes de estudio, el software que mejor se ajusta a esta relación es ese, a pesar de existir una gran gama de herramientas parecidas. El programa MetaTrader4 es un software minoritario y propiedad de la empresa MetaQuotes. El programa ha sido altamente aceptado entre la comunidad de traders de todo el mundo en el año 2009, con mucha rapidez. De hecho, el 90% de traders novatos actualmente empiezan utilizando MT4, lo cual generó que muchos brókers finalmente hayan migrado sus plataformas de trading a MetaTrader4.

El MT4 puede correr sobre Window o MAC, y una vez instalado el MT4 en el ordenador del cliente software se conectaría fácilmente a los servidores del bróker (explicado en el siguiente apartado). Los brokers ahora tenían que pagar a los desarrolladores del programa MT4 por su uso, pero su mentalidad fue la siguiente: “cuanto menos cambios tenga que hacer el cliente, más fácil es que haga trading conmigo, y este programa lo usa casi todo el mundo”.

Pero, ¿por qué MetaTrader4 fue tan aceptado por la comunidad? En mi opinión se debió a 2 razones. Son las siguientes:

- 1) La primera razón fue que **crearon una interfaz gráfica muy sencilla con multitud de herramientas para hacer análisis financiero rápido y eficaz** sobre la gráfica del mercado. Los clientes podían contratar MetaTrader4 para desarrollar un estudio del mercado eficaz, y posteriormente operar en una plataforma ajena, pero resultaba incómodo: la de su bróker. En esta fase del programa, en torno al año 2010, los bróker competían por tener las comisiones más bajas, pero no se preocupaban por ofrecer al cliente unas facilidades asequibles para analizar los mercados.
- 2) Posteriormente surge una segunda razón, a partir del año 2014-2015. En este momento **toma más fuerza la plataforma de MetaTrader4 cuando empiezan a circular por la web estrategias de gente desarrollado en lenguaje MQL4**, lenguaje propietario de esta empresa. Daba la opción de realizar trading automático con esos programas desarrollados por el usuario a su gusto. Lo que marcó la diferencia real que los posicionó por delante del resto al lenguaje de programación fue que: el lenguaje era simple, fácil, rápido, MQL4 es una variante de C++, con posibilidad de usar objetos de programación, simuladores, clases, scripts, etc. Además MQL4 tiene sus propias palabras reservadas, como cualquier otro lenguaje de programación, y evidentemente se crea como lenguaje de alto nivel.

Una vez comprendida la estructura de MT 4, yo personalmente prefería conectarme a servidores DEMO ajenos a mi bróker para evitar cualquier tipo de control, lo cual también es posible en MT4 si sabes cómo hacerlo.

Con el tiempo fui creando mis propias estrategias, probando y mejorando las de otros traders, o simplemente comprando operaciones de CFD's en esta plataforma de forma fácil y rápida. Actualmente la empresa propietaria ha sacado del mercado MetraTrader4 y está actualizando sus aplicaciones a **MetaTrader5**. Esta nueva actualización se va a mencionar en el proyecto levemente para indicar por qué no es tan atractiva como la 4. La versión 5 no gusta, principalmente porque es casi al completo de pago y no facilita la creación de estrategias software de forma fácil, y el sistema de archivos cambia considerablemente respecto a la versión anterior.

5.2- Uso de la GUI de Meta Trader 4

Para utilizar una cuenta DEMO en MetaTrader4 vamos a usar un bróker llamado Hanseatic Brokerhouse, y su web puedes encontrarla en <http://www.hansetrader.es/>. Es un bróker alemán con una de sus sedes en Valencia (España). Recomendado por sus bajas comisiones, su profesionalidad, servicio técnico disponible y atento, NO tratan de engañarte para que ingreses dinero ni te presionan. Eso a la larga hace a uno confíe en ellos y, cómo no, ingresar dinero y sentirte cómodo.

Una lista de servidores disponibles DEMO de la propia empresa MetaQuotes serían los siguientes:

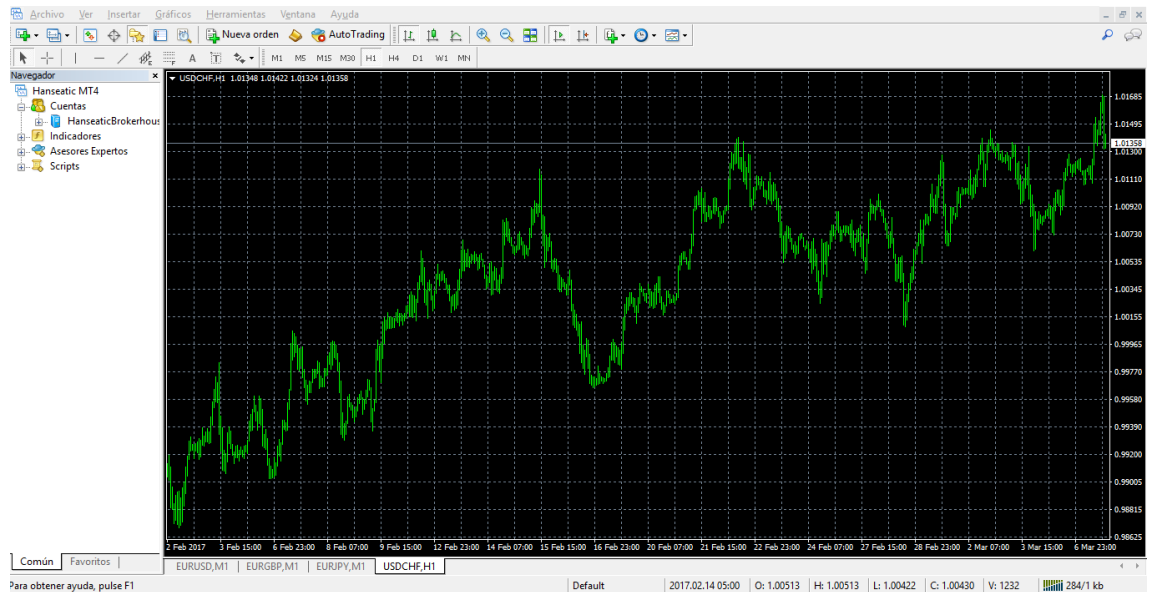
FOREX-Server
ICMTrader-Demo
AForex-Demo
GO4X-Demo
FXCM-USDDemo01
IngotBrokers-Server
Forex.com-Demo 3
ICMBrokers-Demo
MoneyAge-Server

También tenemos el propio servidor DEMO de Hanseatic:

HanseaticBrokerhouse-Demo

Para este proyecto, por motivos de privacidad y seguridad, no realizaremos nada en cuenta real. Todo será en cuenta DEMO. Comencemos entonces con el tutorial de la plataforma MetaTrader4:

- 1) Lo primero que nos encontramos al instalar la aplicación MetaTrader4 es la interfaz usuario semejante a esta...



- 2) Una vez estemos ante esta interfaz gráfica debemos seleccionar el servidor adecuado. Para ello debemos de ir a **Archivo** → **Abrir Cuenta** → **Clicar en el “+”** (de la ventana que emerge para añadir un servidor cualquiera). Introducir el nombre del servidor, por ejemplo, **FOREX-Server** y pulsar ENTER.

Abrir cuenta ? X

Servidores comerciales
Seleccione un servidor comercial más rápido:

Dirección	Empresa	Ping
HanseaticBrokerhouse-Demo	Hanseatic Brokerhouse Global Markets Ltd	514.22 ms
HanseaticBrokerhouse-Live	Hanseatic Brokerhouse Global Markets Ltd	85.66 ms
introduzca el nombre del corredor 'BrokerName' o la dirección del tipo mt.company.com:443		

↑
Aquí

Escanear

Siguiente > Cancelar

Abrir cuenta ? X

Datos personales
Para abrir la cuenta, rellene los siguientes campos:

Nombre: asdasd

E-Mail: ejemplo@gmail.com

Teléfono: +34 123123123

Tipo de cuenta: FXEUR

Depósito: 100000 EUR

Apalancamiento: 1:100

☒ Estoy de acuerdo con recibir las noticias a mi e-mail

< Atrás Siguiente > Cancelar

Marcar Opción

Le damos a "Siguiente" y nos sale la siguiente ventana, de la cual guardamos los datos de Usuario y Contraseña:

Abrir cuenta ? X

Registro
Espere, por favor

Nombre: asdasd

Tipo de cuenta: demoFXEUR (1:100)

Depósito: 100000

Servidor: FOREX-Server

Usuario: [REDACTED]

Contraseña: [REDACTED]

Inversor: [REDACTED] (contraseña de sólo lectura)

Guarde su nombre de usuario y contraseña en un lugar seguro.

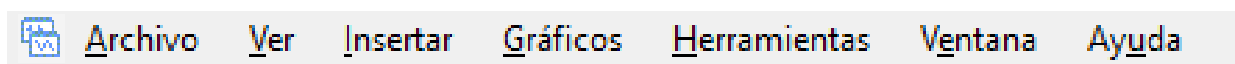
< Atrás Finalizar Cancelar

3) Ejecutamos la conexión con el servidor.

Finalizamos el proceso y deberá de sonar una locución sonorfa que dice: "I'm coming mail". De esta forma sabremos que nos hemos conectado eficazmente al servidor seleccionado. No obstante yo prefiero operar en el servidor DEMO de Hanseatic, pues es mejor para operar un las comisiones son idénticas a mi cuenta Real. De esta manera pruebo en demo mi estrategia en igualdad de condiciones que mi cuenta REAL. En serio, recomendado este bróker, pues es uno de los pocos en los que no he visto falseo en las cotizaciones del mercado. Uno de los mejores de los que opera en España.

4) Visualizamos en la parte superior de la ventana 3 líneas de botones, muy prácticas y potentes. Las explicamos a continuación:

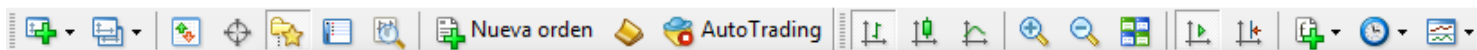
En la línea 1 de botones tenemos:



- a) ARCHIVO: para abrir nuevo gráfico de Divisa o materia prima, abrir una cuenta diferente o conectarnos a diferente servidor desde "Abrir una Cuenta" o abrir carpeta de Datos para acceder a los códigos fuente de nuestras estrategias (entre otras opciones).
- b) VER: para modificar la barra de herramientas superior, cambiar el idioma del programa (por defecto puesto a "Spanish"), configurar los símbolos de las monedas que podemos abrir como nuevo gráfico (desde archivo) y, debajo del desplegable de este menú VER, una serie de opciones que muestran si están abiertas ventanas o abrirlas desde este menú (Ventana de Datos, Navigator, Terminal, ...). En el botón VER podemos configurar entonces qué ver y cómo verlo dentro del programa: ver otros desplegables, tamaño del gráfico completo o parcial, etc.

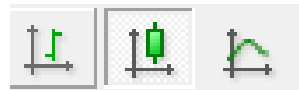
- c) INSERTAR: en este botón se despliega un menú donde podemos seleccionar entre una multitud de herramientas para incorporar a nuestra gráfica y que nos sirva para realizar un análisis técnico o gráfica eficaz. Entre otras herramientas, aquí podemos seleccionar: líneas horizontales, verticales, oblicuas, indicadores técnicos de tendencia (Bandas de Bollinger), Osciladores (MACD), etc.
- d) GRÁFICOS: en este botón encontramos un desplegable útil para realizar una configuración completa sobre la visualización general de los gráficos, tal como el tipo de gráfico deseado (de líneas, de velas japonesas o de barras), el zoom de la gráfica o el color del fondo y de gráfica.
- e) HERRAMIENTAS: contiene las opciones de ejecución de órdenes que podemos lanzar al mercado en cada instante.
- f) VENTANA: ofrece la posibilidad de posicionar automáticamente los diversos mercados o gráficas abiertas en toda la ventana del programa. Hay opciones para el orden de visualización de cada gráfica tales como “Cascada”, Mosaico”, etc. Además muestra las ventanas abiertas en la parte inferior del desplegable.
- g) AYUDA: contiene todos los servicios que puede resultar de ayuda al operador de mercados. Algunos servicios pueden ser: artículos relacionados, preguntas frecuentes, enlace con el sitio oficial de MQL5 y todo lo relacionado con el lenguaje MQL, etc.

En la línea 2 de botones tenemos:



- a) **GRÁFICOS:** permite cargar un nuevo gráfico o materia prima (commodities), metales preciosos, índices (SP500), etc.
- b) **PERFILES** (no usaremos este botón).
- c) **OBSERVACIÓN DE MERCADO:** carga un panel con una serie de divisas con los precios de comprar y venta actualizados en tiempo real. Estos son los precios Bid y Ask, que son los reales del mercado. Al comprar o vender una opción en bolsa compraremos más caro del precio ASK en nuestro bróker, o venderemos más barato del precio BID en nuestro bróker. Esto representa la comisión directa de nuestro bróker.
- d) **VENTANA DE DATOS:** esta opción saca un panel que nos da información referente a la parte del mercado que indiquemos con el ratón. Esto significa que dará el precio de apertura, cierre, mínimo, máximo, fecha y tiempo de cada punto del mercado indicado con el puntero.
- e) **NAVEGADOR:** emerge un panel que nos permite visualizar las cuentas asociadas al terminal MetaTrader4, Indicadores, ExpertAdvisors y Scripts. Más adelante explicaremos los Indicadores, EAs y Scripts.
- f) **TERMINAL:** nos muestra un panel en la parte inferior de la gráfica donde se ve todo lo referente a la operativa e historial financiero de nuestra cuenta de trading. Muestra operaciones abiertas en la sección “**Operaciones**”, el % de capital y tiempo operado en cada divisa clicando en “**Exposición**”, historial completo de operaciones si clicamos en “**Historial de Cuenta**”, noticias, alertas, señales, etc. No obstante, los 3 primeros botones subrayados en negrita son los más importantes de esta sección.

- g) **PROBADOR DE ESTRATEGIA:** sirve como simulador de cualquier ExpertAdvisor (EA) que hayas desarrollado en MQL4 y desees probarlo en fase de simulación. La simulación puedes hacerla con la gráfica de temporalidad deseada desde antes al año 2000, hasta la actualidad.
- h) **NUEVA ORDEN:** permite ejecutar orden PUT o CALL con un tamaño de lote entre 0.01 hasta 8.
- i) **EDITOS MetaQuotesLenguaje:** se visualiza como una carpetilla de archivos color marrón, y si clicas en ella te lleva al editor oficial de MetaQuotes. Ahí podrás desarrollar tu software para las estrategias financieras deseadas.
- j) **AUTO-TRADING:** permite hacer trading automático.
- k) **SELECCIÓN DE TIPO DE GRÁFICA:**



- l) **ZOOM DE GRÁFICA:**



- m) **GRÁFICAS EN MOSAICO:**



- n) **DESPLAZAMIENTO DE GRÁFICAS:**



- o) **LISTA DE INDICADORES**



- p) **TEMPORALIDADES**



- q) **PLANTILLAS**



En la línea 3 de botones tenemos:



- a) CURSOR: sirve para obtener un puntero de razón en forma de “flecha”. Es el tipo de puntero por defecto.
- b) CURSOR EN CRIZ: este tipo de cursor te permite, al ir moviéndote por el gráfico, ver el valor del precio en cada parte de cualquier representación temporal, además del instante temporal horario que representa cada vela o línea del gráfico. Si clicas con el botón derecho del ratón con este tipo de puntero puede medir más fácil la caída o subida del precio existente y el número de velas que estás midiendo.

El indicador que verás vendrá dando los siguientes valores:

velasDeDistancia / PIPSdeDistancia / ValorPrecio

- c) SECCIÓN DE LÍNEAS: permite insertar en las gráficas líneas horizontales, verticales y oblicuas. Las líneas suelen utilizarse en el trading para definir niveles del precio, objetivos, resistencias y soportes. Los niveles, por tanto, podrán ser horizontales u oblicuos. El panel de líneas lo vemos a continuación:



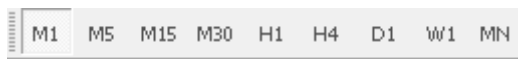
- d) CANALES Y NIVELES TÉCNICOS: en el siguiente panel podemos insertar en la gráfica un canal tendencial donde limitar el precio inferior y superiormente. También podemos insertar pentagramas de Fibonacci, el cual se considera un indicador mixto (ni gráfico completamente ni técnico completamente).



- e) ESCRITURA EN GRÁFICA: panel con 3 opciones para escribir símbolos, letras o etiquetas sobre la gráfica. Este panel permite escribir un pequeño texto sobre una zona de la gráfica o una “manito arriba” o “manito abajo” en símbolos. Estos símbolos son especialmente útiles para hacer backTesting o análisis de mercados con datos del pasado. Así podemos indicar pequeñas notas o validación de operación OK o perdedora y simplificar el recuento de operaciones cuando acabemos el análisis. Veamos el panel al que nos referimos:



- f) TEMPORALIDAD DE GRÁFICA: este menú permite variar la temporalidad que representa cada vela o barra de la gráfica (en caso de usar velas o barras). Si usamos diagrama de líneas las líneas van a ser el resultado de la unión lineal de muchos puntos, y cada uno de estos puntos estarán donde cierre cada intervalo temporal. Las temporalidades son: 1 minuto, 5 minutos, 15 minutos, 30 minutos, 1 hora, 4 horas, 1 día, 1 semana y 1 mes. Por tanto, en gráficas de temporalidad de 1 mes cada vela indicará lo que ha hecho el precio de ese activo en un mes completo.



5.3- Expert Advisors

Es un tipo de módulo de programación, encargado de realizar Trading en mercados de forma automática. Los códigos Expert Advisors (llamados también “EA”), permiten desarrollar robots personalizados, los cuales realicen operaciones financieras automáticas. Tendrán cierta inteligencia artificial. Leen la gráfica principal del mercado y actúan sobre ella obedeciendo los parámetros definidos por el desarrollador del programa.

Las funciones clave predefinidas en un EA son diferentes a un indicador, script y plantilla. Al fin y al cabo cada tipo de código cumple una funcionalidad bien diferenciada entre sí. Veamos entonces la estructura básica de un EA:

- 1º) **Propiedades del código software**, tales como URL propietaria y licencia de Copy Right. Irán precedidas por “**#property**”.
- 2º) **Variables de entrada por defecto**, seguidas de su tipo, nombre y valor que tendrá por defecto.
- 3º) **Función principal**, definida como **void OnTick(void){ ... }**. Dentro de esta función se encuentra el código esencial del programa, y no necesitamos más métodos para crear un robot eficiente. De hecho, que sea corto y simple no significa que sea ineficaz.

5.4- Indicadores

Los indicadores son programas encargados de generar otras gráficas adicionales a la gráfica principal (la que muestra las cotizaciones de precios en tiempo real). Los indicadores que vienen por defecto en MetaTrader4 son los indicadores del MACD y el RSI. Stos indicadores son los más usados por los analistas técnicos, y considerando eso vienen incluidos en el programa. Un ejemplo de cómo queda una pantalla con gráfica principal, MACD y RSI (más adelante veremos que su nombre es “plantilla”) es la siguiente imagen:



Nótese que la gráfica de precios viene arriba y expresa en velas japonesas de colores verdes (alza) o rojo (a la baja). La segunda gráfica es la que expresa el indicador del MACD, y la de más abajo es la representante del RSI.

Cada gráfica se genera a partir de la gráfica de precios, y en mi opinión este tipo de códigos es más complejo que los de tipo EA. Mientras que un EA lee los valores del mercado que recibe del servidor en cada momento y decide entonces qué hacer, un indicador técnico crea una gráfica a partir de los instantes pasados. Se debe hacer énfasis en esto último, pues es más complejo de programar de lo que se cree en primer momento. La gráfica de precios es generada al recibir los datos del servidor, pero el servidor entrega los valores de precios de cada minuto del pasado. Cuando estos datos son recibidos el mercado los coloca visualmente de manera que se ajuste a la representación que el usuario solicitó en la temporalidad indicada de su gráfica, indicada en la GUI de usuario. Aquel lector que tenga

conocimientos de programación software sabrá que la información recibida por un servidor se almacena en un buffer de entrada, y aquí ocurre lo mismo, con la pequeña variante de que para cada vela de 1 minuto habrán 6 buffers:

Time (instante de inicio del periodo):

Ejemplo → 2017.03.08 16:33

Open (precio apertura):

Ejemplo → 1.0544

High (precio max. del periodo):

Ejemplo → 1.0580

Low (precio min. del periodo):

Ejemplo → 1.0501

Close (precio de cierre del periodo):

Ejemplo → 1.0550

Volume (volumen de operaciones del periodo):

Ejemplo de operaciones cerradas → 23

Entrando en detalle en la estructura del desarrollo software de un indicador técnico, la estructura software de un indicador es la siguiente:

- 1º) **Propiedades del código software**, tales como URL propietaria y licencia de Copy Right. Irán precedidas por “**#property**”.
- 2º) **Sentencias INCLUDE**: son archivos ajenos al actual que decidimos incluir o importar en este desde el principio del desarrollo. Esto es útil para usar funciones ya definidas en el programa que estamos importando, por ejemplo.
- 3º) **Propiedades de ventana de indicador**: suelen definir el color por defecto que tomará la gráfica que define el propio indicador, el color de fondo del indicador, si queremos que la gráfica del indicador se defina fuera de la gráfica principal, etc.

- 4º) **Variables de entrada por defecto**, seguidas de su tipo, nombre y valor que tendrá por defecto.
- 5º) **Función principal inicial**, definida como **int OnInit(void){...}**. Dentro de esta función se encuentra el código que se ejecutará primero nada más empezar el programa, incluso antes de recibir datos del servidor.
- 6º) **Función principal "calculation"**, donde está el código que define el indicador propiamente dicho.

La cabecera del método sería algo así:

```
int OnCalculate (const int rates_total,  
                const int prev_calculated,  
                const datetime& time[],  
                const double& open[],  
                const double& high[],  
                const double& low[],  
                const double& close[],  
                const long& tick_volume[],  
                const long& volume[],  
                const int& spread[])  
{ ...}
```

5.5- Scripts

Bueno, ya se cuenta con una noción básica de lo que es un EA y un Indicador Técnico y como desarrollarlo en software. Ahora es el turno de los archivos Scripts, y no son muy diferentes a los anteriores.

Explicado de modo general todo lo visto hasta ahora diríamos que le EA es un robot automático que pone operaciones en bolsa sin la supervisión del usuario; el indicador genera automáticamente una gráfica alternativa a la principal para dar información al usuario sobre la principal, pero necesitando al usuario para ejecutar la operaciones manualmente; y por último, los scripts que solo se ejecutan una vez y finalizan.

El lector debe darse cuenta de que se ha explicado desde el tipo de código más general o complejo hacia otros cada vez más simple. Nótese también que el término “script” en software de bolsa no significa lo mismo que en lenguajes de programación de alto nivel convencionales. En bolsa un script suele ser lo que es aquí: un código que sirve para ejecutarse una sola vez y finalizar.

La estructura es muy parecida a la que tenían los códigos de indicadores:

1-Propiedades del creador software.

2-Entradas por defecto.

3-Método onStart(): en él está el código que se ejecuta en la fase de funcionamiento. Tiene un principio y un fin. Sin bucles ni llamadas reiteradas ni recursivas.

4-Método onDenit(): en él podemos colocar un pequeño código responsable de finalizar el programa. Puede dejarse vacío si se desea.

5.6- Plantillas

Los códigos de las plantillas son bastante diferentes a los anteriores. En el desarrollo software de EAs, indicadores y scripts tenemos que lidiar con un entorno de programación muy parecido al C++, por lo que un desarrollador de lenguajes de alto nivel no tendrá gran dificultad para lograr grandes cosas y en poco tiempo. Sin embargo, en las plantillas nos enfrentamos ante un código tipo CSS, es decir, desarrollable en un documento tipo “bloc de notas”, con una sentencia de asignación por línea de código y posibilidad de agrupar las propiedades por bloques.

Respecto a la funcionalidad de una plantilla, el trabajo de una plantilla es llamar a indicadores, diferentes ventanas, etc y colocarlos simultáneamente de manera ordenada en toda la ventana del programa. Esto quiere decir que si hemos desarrollado un indicador MACD y un RSI podemos crear una plantilla que los agrupe en un solo clic de ratón en nuestra ventana. Por tanto, la función de esta herramienta es la de agrupar “gráficas” generadas por EAs, Indicadores o Scripts.

El código ejemplo de esta sección viene a continuación, y representa un ejemplo sobre propiedades de la ventana completa.

```
<window>
  height=151
  <indicator>
    name=main
  </indicator>
  <indicator>
    name=Moving Average
    period=13
    shift=0
    method=0
    apply=0
    color=255
    style=0
    weight=1
    period_flags=0
    show_data=1
  </indicator>
</window>
```

Vemos que se indica la altura donde se colocará la ventana de la gráfica, se abre una sección de indicador y se indica el indicador al que nos vamos a referir como “main” y que se refiere a la gráfica de precios principal. Una vez se ha obtenido la gráfica principal vemos a llamar a un indicador de medias móviles (llamado “Moving Average”), y sobre el cual se definen una serie de parámetros como son: periodo de media móvil, color, estilo de letra, tabulaciones, etc.

5.7- Estructura de archivos de Meta Trader 4

MetaTrader4 en Window en:

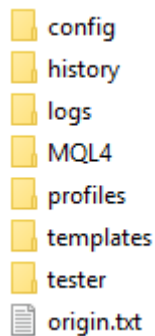
C: \ User \ Usuario \ AppData \ Roaming \ MetaQuotes \ Terminal \ **NUMBER** \

Nótese que el subdirectorio “NUMBER” apareció en negrita, y lo que simboliza es que dicho subdirectorio recibirá siempre un nombre con un número hexadecimal. Por ejemplo, el subdirectorio del final de la ruta se podría llamar con el siguiente número hexadecimal:

101E64ACD3D159E1692EC4E406D1AD8.

Dentro de este subdirectorio guardará MetaTrader todos sus archivos, pero la instalación se realiza en otro sitio. Lo que esto quiere decir es que en C: / Archivos de Programa / MetaTrader / va a hacerse la instalación de la aplicación, y los datos se van a guardar también en el disco C, pero en la parte de usuarios y de programas.

Los archivos que encontramos en la raíz de los archivos tratados por MetaTrader son los siguientes:



Las 2 primeras carpetas son usadas solamente por el programa, es decir, la carpeta **config** y **history**. El resto de carpetas serán usadas para guardar o manipular datos por el usuario. El último archivo obviamente es un archivo de texto con unas palabras del fabricante, sin más que decir (este último archivo no se encuentra en todas las versiones).

En la carpeta **config** se almacenan datos de configuración de terminal, cuentas, usuarios, servidores, etc. En otras palabras: datos del cliente software que corren sobre la aplicación o programa.

En la carpeta **history** se almacenarán datos de históricos, tales como valores de señales usadas, descargas de valores de algún mercado, etc. Sólo almacena los datos de cotizaciones que el usuario ha considerado relevantes, no todas.

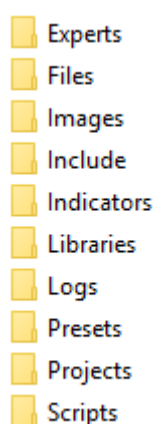
En la carpeta **logs** se almacenará información referente a los errores de programas desarrollados por el usuario. Estos programas pueden ser de tipo Script, Expert Advisor e Indicadores. Los datos vendrán almacenados en archivos de texto plano (.txt) y cada uno contendrá la hora exacta de ejecución (dada por el servidor de cotizaciones) y ruta donde se encuentra el programa.

La carpeta **tester** normalmente estará vacía, sin usar por el usuario. En ella deberían colocar los programas en testing, es decir, que se han probado en el simulador de MetaTrader4.

La carpeta **profiles** contiene toda la información de perfil, es decir, referente al botón “perfiles” que encontramos en la GUI de MetaTrader4 (segunda fila de botones).

En la carpeta **templates** se alojan los códigos y programas de tipo “plantilla”, como se explicaba en la sección anterior (6.6). Los archivos siempre serán de extensión .tpl, y excepcionalmente de extensión .htm.

Y llegamos al fin a la carpeta **MQL4**, donde podemos encontrar los ficheros de toda la programación MQL4 (como su nombre indica). El árbol de subdirectorios que aloja en su interior sería el siguiente:



Los programas que permiten crear robots para operar en bolsa se encuentran en la carpeta **Experts**, pues son Expert Advisors. Los indicadores de forma semejante van a ir en la carpeta del mismo nombre: carpeta **Indicators**, así como los scripts en la carpeta **Scripts**. Volvemos a ver aquí otra vez la carpeta **Logs** o registros, la cual funciona idénticamente a la que estaba fuera: muestra el estado, errores, etc de los códigos ejecutados. También hay carpetas especiales, como es el caso de la carpeta **Include**: en esta carpeta colocaremos manualmente los archivos que podemos necesitar importar alguna vez en algún programa del tipo de sea mediante la sentencia #include. El resto de carpetas no se han usado o tienen poca importancia, por lo que merece explicarlas.

6. MQL4: servidor Jast

6.1- Mis primeros programas creados en MQL4

En este apartado se van a mostrar algunos de los programas Expert Advisors que creé al principio de mi aprendizaje del lenguaje MQL4. Este tipo de programa, los EAs, fueron los programas que primeramente desarrollé porque conceptualmente me resultaban más fáciles de entender: tenían una parte visual y una parte de procesamiento de datos como Java. Es evidente que en estos momentos aun no entendía ni cómo funcionaban los indicadores, y mucho menos las plantillas.

La utilidad que le di a los Expert Advisor en mis comienzos de programador de software de bolsa no fue el correcto, pues la utilidad real de un Expert Advisor es la de crear robots de operaciones en bolsa. Eso quiere decir que un Expert Advisor funcional debe de ser capaz de colocar una orden de compra o de venta según convenga, y después salirse del mercado en el mejor momento posible.

En la actualidad existen multitud de traders que crean estrategias de bolsa en lenguaje MQL4 y en la plataforma de MetaTrader4, programando estos robots y ganando dinero anualmente por dejar este programa corriendo y generando entradas y salidas al mercado automáticamente. Obviamente es arriesgado, y no resulta fácil: lo normal es que te termines arruinando con los primeros Expert Advisors que desarrolles.

Los programas Expert Advisor permiten importar ficheros (como es el caso del fichero del cálculo de medias móviles), generar alertas sonoras con una pequeña descripción textual (como ocurre con un indicador), preguntar por el precio actual (bid), de venta (ask), de cierre del periodo (Close) y de apertura del periodo (Open). Adicionalmente, y ningún otro código podrá hacerlo, un EA puede mandar órdenes de compra/venta al mercado y después cerrarlas: para tomar esta decisión puede calcularse un valor del precio fijo para entrar al mercado y salirse cuando haya una diferencia de 50 pips (variaciones más pequeñas del precio del mercado) respecto al precio de entrada por ejemplo, pero también pueden generarse esas órdenes de entrada y salida al mercado utilizando instantes temporales. La segunda opción para generar órdenes a los mercados es la que he utilizado en el software del proyecto, esa tan solo la explicaré brevemente en los apartados siguientes por motivos de propiedad intelectual.

Veamos ahora diversos códigos propios creados en mis inicios, que en este caso serán 3 códigos:

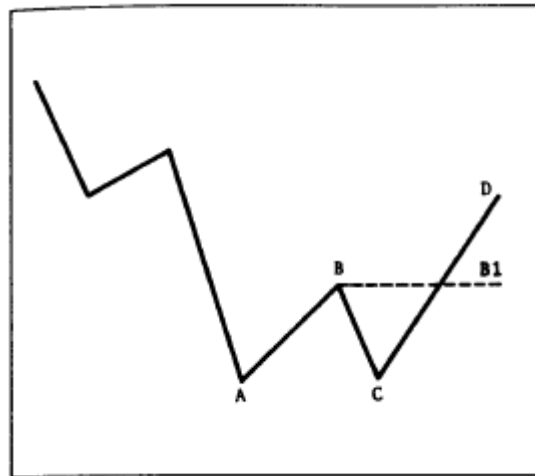
1) Compra/venta al mercado considerando las Bandas de Bollinger (cuando el precio toca B.Bollinger Up vendo, y cuando el precio toca B.Bollinger Down compro):

```
1 #property copyright "juan antonio tora canovas"
2 #property link      "www.juanantoniotor.com"
3
4
5 extern int periodo=20;
6 extern int desviacion=2;
7 extern int nOrdenes=1;
8 extern double Lote=0.01; //para controlar el lotaje
9 extern double StopLoss=500;
10 extern double TakeProfit=50;
11
12 static int aux_hora=Hour(); //inicia en la hora que estamos al comienzo de ejecutar
13 static int aux_minuto=Minute(); //inicia en la hora que estamos al comienzo de ejecutar
14 static bool permiso=true;
15
16 int init(){
17     Alert("Algoritmo de B.Bollinger arrancado en ",Symbol());
18     return (0);
19 }
20
21
22 int start(){
23     double bbsup3 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_UPPER, 3);
24     double bbinf3 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_LOWER, 3);
25     double mediabb3 = iMA(Symbol(), 0, periodo, 0, MODE_SMA, PRICE_CLOSE, 1);
26
27     double bbsup2 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_UPPER, 2);
28     double bbinf2 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_LOWER, 2);
29     double mediabb2 = iMA(Symbol(), 0, periodo, 0, MODE_SMA, PRICE_CLOSE, 1);
30
31     double bbsup1 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_UPPER, 1);
32     double bbinf1 = iBands(Symbol(), 0, periodo, desviacion, 0, PRICE_CLOSE, MODE_LOWER, 1);
33     double mediabb1 = iMA(Symbol(), 0, periodo, 0, MODE_SMA, PRICE_CLOSE, 1);
34
35     double cierre3 = Close[3];
36     double apertura3 = Open[3];
37
38     double cierre2 = Close[2];
39     double apertura2 = Open[2];
40
41     double cierre1 = Close[1];
42     double apertura1 = Open[1];
43
44     if(cierre1>=bbsup1){ //si la ultima vela esta por encima o tocando BBollinger superior meto OP_BUY
45         OrderSend(Symbol(), OP_BUY, 0.01, Ask, 3, Ask - StopLoss*Point, Ask + TakeProfit*Point, NULL, 0, 0, Green);
46     }
47     else{
48         if(cierre1<=bbinf1){
49             Alert("Cruce media bajista");
50             OrderSend(Symbol(), OP_SELL, 0.01, Bid, 3, Bid + StopLoss*Point, Bid - TakeProfit*Point, NULL, 0, 0, Red);
51         }
52     }
53
54     int minuto=Minute();
55     int hora=Hour();
56     //REINICIO DE VARIABLES DE COMPROBACIÓN
57     if(minuto==0 || minuto==5 || minuto==10 || minuto==15 || minuto==20 || minuto==25 || minuto==30 || minuto==35)
58         if(aux_minuto!=minuto){ //para que no repita en un mismo minuto el proceso indefinidamente
59             aux_minuto=minuto;
60             permiso=true;
61         }
62     }
63     return(0);
64 }
65
66 int deinit(){
67     Alert("Fin en ", Symbol());
68     return (0);
69 }
```

2) **Generación de alertas sonoras cuando el precio esté definiendo una tendencia alcista o bajista según indica la teoría de Dow:**

La teoría de Dow indica que en tendencia alcista el precio debe de definir un máximo, un mínimo y luego un máximo superior al máximo anterior. Si el precio marcara ahora un mínimo superior al anterior y empezara a crecer hasta niveles superiores al último máximo, entonces deberíamos entrar alcistas porque es muy probable que el precio esté en tendencia alcista clara. Veamos una imagen que aclara lo que trato de explicar, y esto mismo es lo que se ha programado en este Expert Advisor.

Comprar cuando el precio supere el nivel **B1**:



El programa debería de ir bien si el código se definió correctamente, pero aun así no funciona como debería y se generan muchas falsas señales. A la hora de programar el software que a continuación se va a mostrar no se tuvo en cuenta que el mercado no se mueve en línea recta como vemos en la imagen ejemplo, sino haciendo muchas interferencias como si de una onda se tratara.

Veamos el código que definí, algo más largo que lo anterior:


```

1 //+-----+
2 //|                                     EA_Prueba.mq4 |
3 //|                                     Juan Antonio |
4 //|                                     www.juanantoniatora.com |
5 //+-----+
6 #property copyright "Juan Antonio"
7 #property link      "www.juanantoniatora.com"
8
9 //+-----+
10 //| Expert initialization function |
11 //+-----+
12 extern int VG_barras= 30; //Variable global externa que toma del gráfico 3 velas
13
14 static int aux_hora      = 0;
15 static int aux_minuto    =0;
16
17 static double diferencia =0.00;
18 static double array_diferencia[30]; //array de String de 3 posiciones
19 static double cierre     =0.00;
20 static double apertura   =0.00;
21
22 static double max_1      =0.00;
23 static double max_2      =0.00;
24 static double min_1      =0.00;
25 static double min_2      =0.00;
26
27 static bool   tono_Alcista=false;
28 static bool   tono_Bajista=false;
29
30 int init() {
31     Alert("Mi Exper Advice ha comenzado. YEAH!!");
32     diferencia=0.0;
33     return;
34 }
35
36 int start(){
37     string divisa=Symbol(); //obtiene NOMBRE DE DIVISA
38     double precioActual=Bid;
39     int hora= Hour(); //obtiene HORA DEL SERVIDOR
40     int minuto=Minute(); //obtiene MINUTO DEL SERVIDOR
41
42     bool comprobacion_max_1 = false;
43     bool comprobacion_max_2 = false;
44     bool comprobacion_min_1 = false;
45     bool comprobacion_min_2 = false;
46     int vela_max_1          = 0;
47     int vela_max_2          = 0;
48     int vela_min_1          = 0;
49     int vela_min_2          = 0;
50
51     //guardo en array las diferencias de cada vela
52     for(int i=0; i<VG_barras; i++){ // desde 0 a VG_barras
53         //miro el valor de la vela actual
54         if(i==0){
55             cierre=Bid;
56             apertura=Open[i];
57             diferencia=cierre - apertura;
58             array_diferencia[0]=diferencia;
59         }
60         else{ //mira estado de la vela siguiente
61             if(i>0){
62                 cierre=Close[i];
63                 apertura=Open[i];
64                 diferencia=cierre - apertura;
65                 array_diferencia[i]=diferencia;
66             }
67         }
68     }
69
70     //YA HE GUARDADO TODAS LAS DIFERENCIAS EN EL ARRAY DESDE VELA ACTUAL 0 A VELA 29
71     //Ahora busco los mínimos y los máximos de la función
72
73     if(array_diferencia[0] > 0 && array_diferencia[1]<0){ //acaba de hacerse un MAX
74         for(int j=1; j<VG_barras; j++){
75             if(array_diferencia[j]>0 && comprobacion_min_2==true && comprobacion_max_2==true && comprobacion_min_1==true &&
76                 max_1=Close[j];
77                 comprobacion_max_1 = true;
78                 vela_max_1=j;
79             }
80             if(array_diferencia[j]<0 && comprobacion_min_2==true && comprobacion_max_2==true && comprobacion_min_1==false){
81                 min_1=Close[j];
82                 comprobacion_min_1 = true;
83                 vela_min_1=j;
84             }
85             if(array_diferencia[j]>0 && comprobacion_min_2==true && comprobacion_max_2==false){
86                 max_2 = Close[j];
87                 comprobacion_max_2 = true;
88                 vela_max_2 = j;
89             }
90             if(array_diferencia[j]<0 && comprobacion_min_2==false){
91                 min_2 = Close[j];
92                 comprobacion_min_2 = true;
93                 vela_min_2 = j;
94             }
95         }
96     }

```

```

97 else{
98 //si vela 0 está bajista hay un MAXIMO
99 if(array_diferencia[0]<0 && array_diferencia[1]>0){
100 for(int k=1; k<VG_barras; k++){
101 if(array_diferencia[k]<0 && comprobacion_max_2==true && comprobacion_min_2==true && comprobacion_max_1==true &
102 min_1=Close[k];
103 comprobacion_min_1 = true;
104 vela_min_1=k;
105 }
106 if(array_diferencia[k]>0 && comprobacion_max_2==true && comprobacion_min_2==true && comprobacion_max_1==false)
107 max_1=Close[k];
108 comprobacion_max_1 = true;
109 vela_max_1=k;
110 }
111 if(array_diferencia[k]<0 && comprobacion_max_2==true && comprobacion_min_2==false){
112 min_2 = Close[k];
113 comprobacion_min_2 = true;
114 vela_min_2 = k;
115 }
116 if(array_diferencia[k]>0 && comprobacion_max_2==false){
117 max_2 = Close[k];
118 comprobacion_max_2 = true;
119 vela_max_2 = k;
120 }
121 }
122 }
123 }
124
125 //Lanzo alerta alcista
126 if(array_diferencia[0]>0 && array_diferencia[1]<0){ //detecto MINIMO
127 if(min_1<min_2 && max_1<max_2){
128 if(tono_Alcista==false){
129
130 tono_Alcista=true;
131 tono_Bajista=false;
132 Alert(divisa," ALCISTA a las ",hora,":",minuto);
133 }
134 }
135 }
136 //Lanzo alerta bajista
137 if(array_diferencia[0]<0 && array_diferencia[1]>0){ //detecto MAXIMO inmediato
138 if(min_1>min_2 && max_1>max_2){
139 if(tono_Bajista==false){
140
141 tono_Bajista=true;
142 tono_Alcista=false;
143 Alert(divisa," BAJISTA a las ",hora,":",minuto);
144 }
145 }
146 }
147
148 //REINICIO DE VARIABLES DE COMPROBACIÓN
149 if(minuto==0 || minuto==15 || minuto==30 || minuto==45){
150 if(aux_minuto!=minuto){ //para que no repita en un mismo minuto pitidos
151 tono_Alcista=false;
152 tono_Bajista=false;
153 aux_minuto=minuto;
154 }
155 }
156
157 } //del metodo

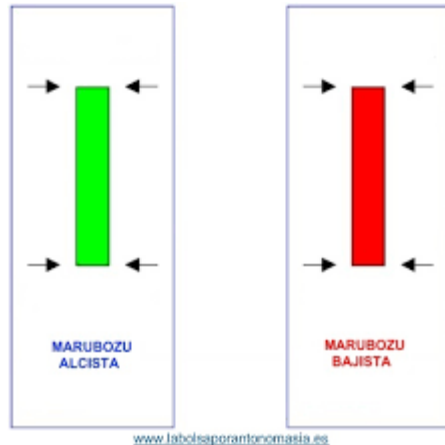
160 int deinit(){
161 Alert("Expert Advice finalizado.");
162 return;
163 }
164

```

3) Generación de alertas sonoras cuando el precio forme una vela japonesa de tipo “mazubozu”:

En primer lugar expliquemos bien lo que es una vela japonesa de tipo “mozubozu”: es una vela de cuerpo largo, alcista o bajista, que no presenta mechas ni por arriba ni por abajo.

Es decir, una vela alcista abrirá en su mínimo y cerrará en su máximo, mientras que una vela bajista abrirá en su máximo y cerrará en su mínimo.



La idea subyacente es que si una vela alcista abre a un nivel y cierra a otro superior si que haya habido retroceso del precio en el periodo elegido esto puede ser un gran indicativo de que el mercado está con decisioes fuertemente alcistas. Enviaríamos al mercado una operación BUY o compra tras una vela mazubozu alcista.

Veamos ahora el código que creé en su momento,

```
4 //| Juan Antonio |
5 //| www.juanantoniotor.com |
6 //+-----+
7 #property copyright "Juan Antonio"
8 #property link      "www.juanantoniotor.com"
9
10 //+-----+
11 //| Expert initialization function |
12 //+-----+
13 extern int VG_barras= 30; //Variable global externa que toma del gráfico 3 velas
14
15 static int aux_hora      = 0;
16 static int aux_minuto    =0;
17
18 static double diferencia =0.00;
19 static double array_diferencia[30]; //array de String de 3 posiciones
20 static double cierre[3];
21 static double apertura[3];
22 static double maximo[3];
23 static double minimo[3];
24
25 static int minutoAUX =0;
26
27 static bool permiso =true;
28 static double mitad_Vela2 =0;
29
30
31
32 int init() {
33
34     string divisa2=Symbol();
35     Alert("EA \"Morubozu\" operativo en ",divisa2,".");
36     permiso=true;
37     diferencia=0.0;
38     return;
```

```

39 }
40
41 int start(){
42     string divisa=Symbol(); //obtiene NOMBRE DE DIVISA
43     double precioActual=Bid;
44     int hora= Hour(); //obtiene HORA DEL SERVIDOR
45     int minuto=Minute(); //obtiene MINUTO DEL SERVIDOR
46
47     if(minuto==6 || minuto==11 || minuto==16 || minuto==21 || minuto==26 || minuto==31 || minuto==36 || mir
48         if(permiso==true){
49
50             minutoAUX = minuto;
51             permiso=false;
52
53             //guardo en array las diferencias de cada vela desde la actual
54             for(int i=0; i<VG_barras; i++){ // desde 0 a VG_barras
55                 //miro el valor de la vela actual
56                 if(i==1){
57                     cierre[i]=Close[i];
58                     apertura[i]=Open[i];
59                     diferencia=cierre[i] - apertura[i];
60                     array_diferencia[i]=diferencia; //alcista-bajista
61                 }
62                 else{ //mira estado de la vela siguiente
63                     if(i==2){
64                         cierre[i]=Close[i];
65                         apertura[i]=Open[i];
66                         maximo[i]=High[i];
67                         minimo[i]=Low[i];
68                         diferencia=cierre[i] - apertura[i];
69                         array_diferencia[i]=diferencia; //alcista-bajista
70                         if(array_diferencia[i]<0){ //cambio de BAJISTA A ALCISTA
71                             mitad_Vela2 = ((apertura[i] - cierre[i])/2) + cierre[2];
72                         }
73                         else{ //cambio de alcista a bajista
74                             if(array_diferencia[i]>0){
75                                 mitad_Vela2 = ((cierre[i] - apertura[i])/2)+apertura[2];
76                             }
77                             else{
78                                 mitad_Vela2=0; //valor inalcanzable
79                             }
80                         }
81                     }
82                     else{
83                         if(i>2){
84                             cierre[i]=Close[i];
85                             apertura[i]=Open[i];
86                             maximo[i]=High[i];
87                             minimo[i]=Low[i];
88                             diferencia=cierre[i] - apertura[i];
89                             array_diferencia[i]=diferencia; //alcista-bajista
90                         }
91                     }
92                 }
93             }
94
95             //PATRON 2: PAUTA PENETRANTE
96             if( array_diferencia[2]>0 && array_diferencia[3]>0 && array_diferencia[1]<0 && cierre[1]<mitad
97                 if( cierre[2] < apertura[1] ){ //comprobacion de GAP alcista para futura trend bajista
98                     if(cierre[1]<mitad_Vela2){
99                         Alert("Bajista en ",divisa," ",hora,":",minuto);
100                     }
101                 }
102             }
103             else{
104                 if( array_diferencia[2]<0 && array_diferencia[3]<0 && array_diferencia[1]>0 && cierre[1]>mi
105                     if( cierre[2] > apertura[1] ){ //comprobacion de GAP bajista para futura trend alcista
106                         if(cierre[1]>mitad_Vela2){
107                             Alert("Alcista en ",divisa," ",hora,":",minuto);
108                         }
109                     }
110                 }
111             }
112         }
113     }
114     if(minuto==4 || minuto==9 || minuto==14 || minuto==19 || minuto==24 || minuto==29 || minuto==34 || minuto==
115         permiso=true;
116     }
117 } //del metodo start()
118
119 int deinit(){
120     Alert("Expert Advice finalizado.");
121     return;
122 }

```

6.2- Paradigma de programando para operar automáticamente en los mercados desarrollando herramientas MQL4

Un paradigma de programación es una propuesta tecnológica adoptada por una comunidad de programadores y desarrolladores, que buscan resolver uno o varios problemas claramente delimitados. La resolución de estos problemas debe suponer consecutivamente un avance significativo en algún parámetro de la ingeniería software.

Por otra parte, un paradigma de programación representa un enfoque distinto al actual, una filosofía distinta para diseñar soluciones respecto a lo actual. El paradigma de programación empleado guarda una estrecha relación con el lenguaje de programación que se utiliza, así como el momento en que el lenguaje de programación fue creado.

El paradigma de programación más utilizado actualmente es el llamado “Orientación a Objetos”, siendo el núcleo central de este paradigma la unión de datos y su procesamiento, todo ello agrupado en una entidad que denominamos “objeto”.

En el lenguaje MQL4 el paradigma de programación anterior, el paradigma OO, se mantiene mediante la posibilidad de creación de clases de alto nivel, integración de objetos, herencia e importación de ficheros. No obstante, el objetivo de MQL4 no es desarrollar programas complejos de alto nivel, desarrollar complejas o bonitas interfaces gráficas, ni establecer conexiones socket en tiempo real con otra aplicación en tiempo real. La funcionalidad de MQL4 para la cual fue creada es la de desarrollar códigos que permitieran establecer estrategias al trader a partir de las cotizaciones actuales del mercado. Queda claro entonces que MQL4 está pensado para que el procesado de datos sea a nivel interno: los datos llegan del servidor, los muestra la gráfica de MetaTrader4 y después la estrategia toma esos datos guardándolos en buffers y procesándolos. ¡Punto!

Una de las dificultades con las que he tenido que lidiar, al programar en MQL4 y tratar de integrar los lenguajes Java y MQL4, ha sido sacar los datos de MQL4 de una manera lo más flexible y estandarizada posible, y posteriormente leerlos en Java.

Algunas diferencias de la estructura de archivos en Java y MQL4 son:

1) Forma de importar ficheros dentro del fichero actual:

En java se crea un grupo de clases java y podemos importar o heredar en cualquier otra una clase anterior que esté dentro del mismo paquete, pero en MQL4 importar un archivo que necesitemos para utilizar funciones clave se hará de forma algo distinta.

Pongamos el ejemplo del caso de querer pedir el valor de los volúmenes de transacciones para cada periodo temporal actual y pasado. Primero importaremos el fichero **Volumes.mqh**, el cual deberá estar dentro de la carpeta "includes" que viene por defecto. Por tanto para cada fichero que queramos importar a mi indicador, EA, etc primero deberé de meterlo en la carpeta Include, y después importarlo a mi código con la sentencia **#include**. De esta forma podremos utilizar palabras clave extra a las básicas, las cuales son reconocidas porque he importado el código adicional de volúmenes.

2) Lectura y escritura de fichero:

La lectura y escritura de ficheros se hace semejante a Java pero en menos líneas. En java se crea un objeto de tipo File, luego de tipo BufferedWriter y después escribimos.

```
String ruta1 = "C:/Users/juan anto
File archiv1 = new File(ruta1);
BufferedWriter bw1;

bw1 = new BufferedWriter(new FileWriter(archiv1));
bw1.write("Acabo de crear este fichero de texto.");
}
bw1.close();
```

En el caso de MQL4 solo es necesario crear una variable manejadora de ficheros, creo un FileWrite() y cierro el fichero liberando el manejador asociado con FileClose().

```
int filehandle=FileOpen("mi.txt",FILE_WRITE|FILE_TXT); //
if(filehandle!=INVALID_HANDLE) {
    FileWrite(filehandle,"\nRSI ",a,"% a las ",Hour(),":",
    FileClose(filehandle);
    Print("FileOpen OK");
}
```

3) No es posible elegir el directorio sobre el que queremos escribir o leer:

Lo que pasará es que el fichero que leerá o escribirá debe de estar bajo el directorio **C:/ ... /MQL4/ Files/ ...**, es decir, que si en la ruta que indicamos en la imagen anterior solo pusimos el fichero **“mi.txt”** ese fichero se creará en **C:/user/juanantonioAppData/Roaming/MetaQuotes/Terminal /675720BBF2A300C / MQL4 / Files / mi.txt**. No se puede utilizar rutas absolutas. Las rutas de archivos siempre serán relativas al directorio raíz **c:/.../MQL4/FILES/** como dije.

Esto que acabo de mencionar pone de manifiesto lo que se mencionaba en el apartado 6.1 respecto a la baja flexibilidad de sacar archivo de la propia estructura de archivos de MetaTrader. El terminal MT4 tiene una estructura de archivo muy poco flexible.

Recuerde que MetaTrader no fue creado con la idea de estandarizar su código, sino que se creó para dar la flexibilidad necesaria para dar al trader la posibilidad de crear estrategias muy eficientes de bolsa, de manera rápida y simple. Piense que este software tiene algunas funciones extras de pago, y si se excediera la flexibilidad tal vez no podrían asegurar los beneficios.

6.3- Estrategia de Trading subyacente del Proyecto

Si usted ha leído algo sobre el trading, mercados financieros, etc sabrá de sobra que los mercados caen con muchas más velocidad que con la que suben. El principio de este hecho está basado en un principio psicológico: la confianza. Es más fácil perder la confianza de un inversor que ganarla, e igualmente ocurre en los mercados y esto es reflejado en las gráficas.

Este trabajo se basa en ese principio, y para medir la confianza del mercado en cada momento se ha utilizado 2 indicadores propios: MACD modificado y RSI modificado. El MACD mide la fuerza que tiene el mercado para continuar la tendencia actual, y el RSI la fuerza relativa de que cambie la tendencia actual (que el mercado entre en lateralidad o cambie bruscamente el sentido de la tendencia).

Entonces, ¿cómo operaremos cada tendencia? La respuesta es operar en LARGO tendencias alcistas (comprar) y en CORTO tendencias bajistas (vender). Los términos LARGO y CORTO se refieren a la duración que una operación estará en el mercado. Una operación alcista se opera en LARGO por el hecho de que el precio tardará más en subir, y una tendencia bajista se opera en CORTO porque el precio tardará casi nada en caer. No obstante he modificado el MACD y el RSI un poco, de forma que las medias móviles, los periodos de muestreo, los colores y los niveles son otros diferentes a los básicos para lograr mejores resultados.

Haremos una operación en LARGO de COMPRA cuando el MACD esté en verde y el RSI esté por debajo del nivel 20 (sobre-venta).

Sin embargo, haremos una operación en CORTO de VENTA cuando el MACD está en rojo y el RSI esté por encima del nivel 80 (sobre-compra).

Por defecto, cada operación se sacará del mercado cuando el indicador MACD cambie de color respecto al color que tenía cuando la operación de venta/compra se envió.

Además, colocaremos Stops Loss y Stop Profits fijos adicionales, ajenos a los indicadores MACD y RSI personalizados. Como stop de pérdidas (Stop Loss) se coloca un margen de pérdidas de 50 pips (puntos o variaciones del mercado): cuando el mercado esté 50 pips en contra del nivel del precio al que entré la operación se cerrará con esas pérdidas máximas. Como stop de beneficios colocaremos un stop profit a los 150 pips para salirnos del mercado con beneficios a los 150 pips. Esto es necesario para protegernos en el caso de haber tendencias fuertes, porque el MACD o RSI podrían reaccionar tarde en tal caso, y sacarnos en pérdidas o con muy pocos beneficios.

6.4- Definimos nuestra Plantilla para cargar los indicadores creados

He creado, además de los indicadores, una plantilla que incorpora los indicadores MACD y RSI modificados (periodos de señales y medias móviles). Lo que hemos hecho es tomar como base una plantilla del programa y cambiar lo necesario para que se carguen nuestros indicadores en lugar de lo que se cargan de fábrica con las variaciones mencionadas.

Por motivos de Copyright no se mostrarán el código de la plantilla desarrollada, pero sí el resultado. Note que el color del MACD es siempre el mismo en todo el espectro de la señal, cosa que no nos interesa.



Debido a la falta de documentación sobre cómo desarrollar plantillas en MQL4 no supe crear alarmas y todo lo que ya tenía desarrollado en los indicadores (lectura/escritura de ficheros, colores modificados, alertas, ect).

A pesar de la simplicidad de cargar todos los indicadores dando un solo clic en la plantilla desea, descartamos utilizar plantillas software en el proyecto. No vi rentable el tiempo necesario para aprender a programar plantillas a pesar de la simplicidad de usuario.

6.5- Indicadores propios implementados (utilizados en el proyecto)

Una vez descartada la opción de utilizar plantillas MQL4 pasamos a perfeccionar los indicadores MACD y RSI, ambos modificados altamente. Los motivos fueron 2, bastante evidentes: evitar tener problemas de copyright y obtener unos indicadores que se adaptaran mejor a mi forma de trading.

Una captura de cómo entrega el programa la gráfica del EUR/USD por defecto (básica) es la siguiente:



Vimos en el apartado anterior que realizando una plantilla la cosa mejora un poco, pero no me pareció suficiente. Veamos ahora el resultado de la programación de los indicadores propios:

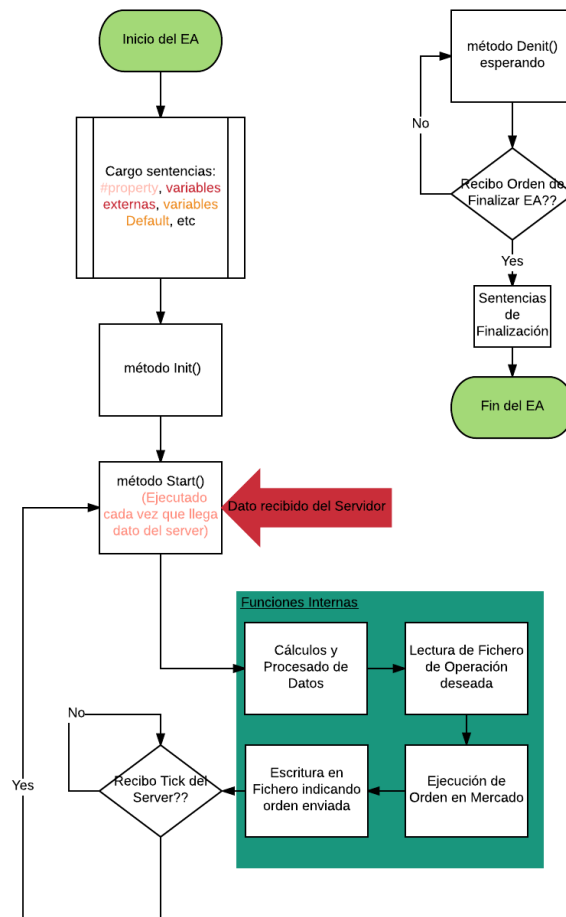


Esta opción es preferible a todas las anteriores, porque adapto el RSI y el MACD a valores que encajan mejor con mi código. Por otra parte, visualmente aparece el color ROJO y VERDE en el MACD cuando debo vender o comprar, además de mi nombre dentro de la gráfica.

Como indica el cartel de letras color rosa "MACD Juan Antonio Tora", es decir, en el MACD se genera ese letrero. Fue decidido así porque los cambios de código más significativos se realizaron en el MACD, aunque en el RSI también se realizaron cambios. Los cambios realizados en el RSI fueron más de tipo de procesamiento de los datos previamente definidos por el fabricante, añadiendo la posibilidad de guardarlos en un fichero (escribo una fichero txt) y cambiando los parámetros de medias móviles (de periodo 14 a periodo 7).

6.6- Desarrollando el Expert Advisor que opera CFD's automáticamente al recibir la orden del programa Java

El flujo-grama funcional de Expert Advisor es el siguiente:



Nótese que hay 3 método en el programa, uno de los cuales está aislado del flujo principal del programa: el método Init() se ejecuta 1º, después el método Start() y, sólo cuando se se reciba una orden de FINALIZAR PROGRAMA, el método Denit() se ejecutará como secuencia de cierre del programa.

El módulo rectangular de color verde trata de ilustrar las 4 fases desarrolladas dentro del método Start(). La fase 2 y 4 del método **Start()** son las relacionadas con Lectura y Escritura de fichero (escribimos en ficheros de texto plano). Los ficheros que lee el EA los han escrito el programa Java, es decir:

1º: Los indicadores MACD_mio y RSI_mio escriben sus valores actuales en ficheros (llamados fichero A y fichero B).

2º: El programa Java lee fichero A y fichero B, y genera una línea de texto según el resultado tras procesar la información reída.

3º: Java escribe el resultado de la orden en un fichero (fichero C).

4º: El EA de MetaTrader4 lee el fichero C cada cierto tiempo, y ejecuta la orden que se indique en dicho fichero. Cuando haya realizado la orden lo comunicará escribiendo OK o ERROR según el caso, en un fichero D. De esta forma el programa Java podrá estar informado de que ocurrió una excepción interna en la gestión de archivos, e informar al usuario para que pare el trading automático.

6.7- Simulador: probando los EAs

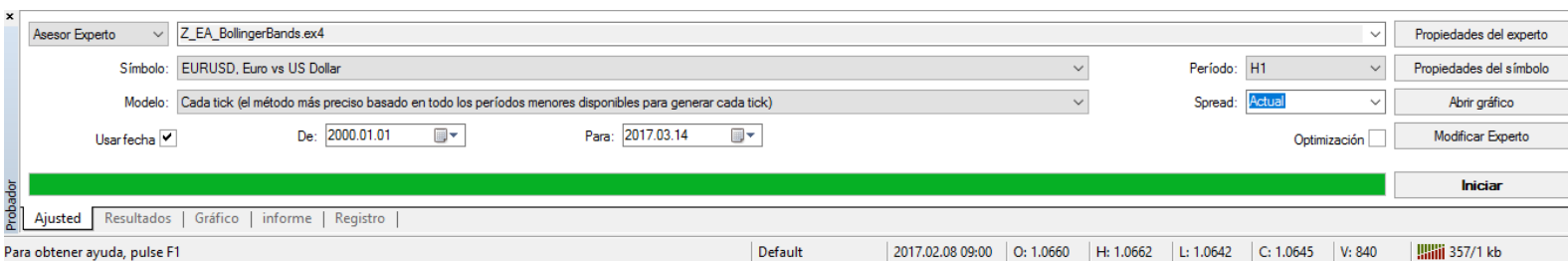
El terminal MetaTrader4 (o MT4) tiene la herramienta de usar un simulador de estrategias de trading automático, es decir, de utilizar un EA (Expert Advisor) determinado. Con esta herramienta el usuario puede probar la rentabilidad que le habría generado la estrategia en modo automático si hubiera estado funcionando en un periodo temporal del pasado determinado (en el mercado concreto seleccionado).

Para ello, primero debemos de crear un Expert Advisor en la carpeta “Expert” dentro de la ruta adecuada (en el apartado 5-7 ya quedó explicado el sistema de archivo de MT4).

Poteriormente, Pulsar el botón que parece una lupa (“probador de estrategias – Ctrl+R”), en la Interfaz de Usuario Gráfica del MT4.



Una vez abierta la herramienta, en la parte inferior del programa MetaTrader4 aparecerá algo como esto (si la ventana está comprimida por favor expándala hacia arriba):



Por defecto está configurado para que pregunte en el primer menú por un Expert Advisor (“Asesor Experto”) para ejecutar. Cuando haya seleccionado el EA que tiene usted que probar (en mi caso es **Z_EA_BollingerBands.ex4**), vaya a la opción siguiente (menú “**Símbolo**”) para elegir el mercado sobre el que desea probarlo. Si ha realizado todo correcto hasta el momento, el tercer paso sería elegir un método de ejecución para su estrategia (menú “**Método**”). Por último, fije una fecha del pasado para simular su estrategia (marque la opción “**Usar fecha**” y después fije la deseada).

Cuando haya entendido y configurado correctamente el simulador, entonces deberá darle al botón que tiene a su derecha justo donde finaliza la barra verde que se ve en la imagen: botón “**Iniciar**”. Este botón iniciará la simulación, que podría tardar varios minutos o sólo algunos segundos. Todo dependerá de la complejidad de su estrategia y de la fecha seleccionada.

Usted sabrá que la ejecución de su estrategia ha finalizado cuando oiga un pitido y la barra verde sea rellena completamente. En ese momento, vaya entonces a las siguientes opciones:

Resultados

Gráfico

Informe

Registro

Basándome en mi experiencia, yo le recomiendo al lector qu use sobretodo la opción “gráfico” y la opción “Informe”: en la primera se podría ver una gráfica de Beneficio o Pérdida a lo largo del tiempo y en la segunda habrá un informe detallado de cómo de buena es su estrategia algorítmicamente.

En mi caso no apareció gráfica, pero veamos el resultado de mi informe de estrategia (es un 43,8% de buena solamente y termina en pérdidas):

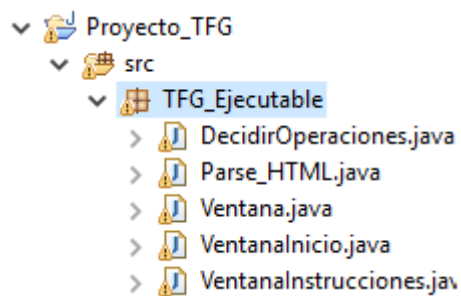
Barras en el historial	2151	Ticks modelados	1936148	Calidad del modelado	43.87%
Errores de desalineación de gráficos	2				
Depósito inicial	10000.00			Spread	Actual (3)
Beneficio neto total	0.00	Beneficio bruto	0.00	Pérdida bruta	-0.00
Factor de beneficio		Beneficio esperado	0.00		
Drawdown absoluto	0.00	Drawdown máximo	0.00 (0.00%)	Drawdown relativo	0.00% (0.00)
Total de transacciones	0	Posiciones cortas (% ganadoras)	0 (0.00%)	Posiciones largas (% ganadoras)	0 (0.00%)
		Transacciones rentables (% del total)	0 (0.00%)	Transacciones no rentables (% del total)	0 (0.00%)
	Mayor	transacción rentable	0.00	transacción no rentable	-0.00
	Media	transacción rentable	0.00	transacción no rentable	-0.00
	Número máximo	ganancias consecutivas (beneficio en dinero)	0 (0.00)	pérdidas consecutivas (pérdidas en dinero)	0 (-0.00)
	Máx.	beneficio consecutivo (número de ganancias)	0.00 (0)	pérdidas consecutivas (número de pérdidas)	-0.00 (0)
	Promedio	ganancias consecutivas	0	pérdidas consecutivas	0

7. Cliente JAST: programa desarrollado en Java

En esta sección se va a exponer el funcionamiento general de la interfaz gráfica de usuario (GUI de la APP), así como el cliente software (también desarrollado en Java). Por tanto, el “Back-End” de la APP completa sería MT4 y el “Front-End” el programa Java (ver sección 4 para más información).

7.1- Arquitectura programa Java y llamadas recursivas

En este primer apartado nos centramos en explicar las clases utilizadas en el programa cliente que desarrollamos en Java. El conjunto de archivos correspondiente al cliente Java estará originalmente agrupado en un Proyecto Java, y este en un package. Veamos el esquema de árbol por si puede ayudar a entenderlo mejor:



Como vemos en la imagen hemos creado 5 clases Java, y entre sí van a ir lanzándose llamadas recursivas entre las diferentes clases. El objetivo es realizar una comunicación eficaz entre cliente y mercados financieros eficazmente, y cerrar el programa sólo cuando el cliente decida cerrar el programa.

El archivo que genera la ventana de interfaz de usuario principal es el llamada “**Ventana**”, que lanza a su vez el control primeramente a una ventana inicial de espera: archivo “**Ventanalncio**”. Cuando pasa un tiempo predefinido por el software, este mismo programa carga las noticias que debe de tenerse ya en la carpeta “Descargas” del computador Window. Para realizar esto el programa oculta la “Ventanalncio” y devuelve el control a “Ventana”. En ventana se lanza el control al archivo “Parse_HTML” y es este el encargado de traducir el HTML devuelto de la página de noticias. La noticias las guardará en un String muy largo separado por “_” cada noticia. Este string será devuelto a la clase “**Ventana**”. En ese momento se cargará la interfaz gráfica con una lista de las noticias reconocidas como de alta importancia.

Hasta este instante se ha utilizado solamente el paradigma de programación de “Orientación a Objetos” y programación secuencias. A partir del momento en que se carga la interfaz gráfica Java completa se incorporará el paradigma de “inversión de control”, es decir, el control de la APP ya no lo tiene el programa sino el propio usuario.

La APP tendrá 4 botones, y si pulsa alguno de los botones el programa modificará su funcionamiento o emergerán ventanas con instrucciones. El funcionamiento es simple:

Cuando la etiqueta lateral que hay en la APP cambie de **ROJO** a **VERDE** estarán disponibles los botones de INICIO y STOP para controlar el trading de los mercados EUR/USD, EUR/GBP y EUR/JPY. La etiqueta se pondrá a VERDE cuando al menos uno de los 3 mercados estén con los indicadores MACD_mio y RSI_mio cargados y el EA Expert_mio funcionando.

La etiqueta podrá pasar de verde a roja en cualquier momento que detecte que los mercados han sido cerrados, parando el trading en ese momento y después invalidando el uso de botones INICIO y STOP. No obstante, mientras el programa esté iniciado y el trading esté activo el control recaerá sobre el archivo “**DecidirOperaciones**”.

Para finalizar la sección, se muestra una imagen esquemática de cómo se llaman entre sí las diferentes clases del programa Java unas a otras:

7.2- Programas que captura la pantalla cada hora y cada 5 minutos

El programa Java que realiza las capturas de pantalla lo he desarrollado en parte gracias a partes de código publicado en www.StackOverFlow.com. Mi programa está separado en 2 métodos: el método MAIN y el método CAPTURE_SCREEN. Evidentemente la parte del código que genera las capturas de pantalla está en el método CAPTURE_SCREEN.

El código de stacOverFlow.com que nos ayuda a capturar la pantalla del ordenador es algo así:

```
static public void captureScreen(String fileName) throws Exception {
    Dimension screenSize = java.awt.Toolkit.getDefaultToolkit().getScreenSize();
    Rectangle screenRectangle = new Rectangle(screenSize);
    Robot robot = new Robot();
    BufferedImage image = robot.createScreenCapture(screenRectangle);
    ImageIO.write(image, "png", new File(fileName));
}
```

Desglosando el código de la imagen, vemos primero que mediante el objeto "Dimension" pedimos al sistema las dimensiones de la pantalla, después se crea un objeto "Rectangle" del tamaño igual a la dimensión de la pantalla, guardamos la imagen capturada en un Buffer de imágenes y escribimos la imagen del buffer en un archivo de extensión ".PNG".

El método principal, MAIN, es mucho más extenso. Se ha instanciado una variable "i" de tipo entero para usarla de contador de imágenes, una variable de tipo String para guardar la ruta donde guardaremos cada foto y 3 variables temporales inicializadas a valor "-1" usadas para guardar los segundos, minutos y hora auxiliares del sistema.

Tras inicializar y declarar las variables anteriores se ha creado una clausula try/catch por si ocurrieran problemas al crear el objeto Thread, A.capturaDePantalla y el de tipo Calendar.

El proceso es el siguiente:

- 1- Creo un objeto de tipo Calendar, y variables de tipo INT para la hora, minuto y segundo.
- 2- Obtengo la hora, minuto, segundo del sistema operativo llamando al objeto Calendar.
- 3- Guardamos los valores de las variables temporales en las 3 variables auxiliares que definimos antes del Try/Catch.

- 4- Utilizamos el método CurrentThread del objeto Thread para meter un retardo de 10 segundos a la ejecución del programa.
- 5- Llamamos al objeto CAPTURE_SCREEN de esta propia clase y ejecutamos la primera captura de pantalla. Deberá haber dado tiempo en esos 10 segundos a colocar la pantalla principal del computador en el programa MetaTrader4 para visualizar las cotizaciones de bolsa.
- 6- Tras esta primera captura vuelvo a calcular el tiempo que falta para la siguiente hora exacta (calculado en segundos siempre). Esto se hace mediante diferencias entre Minuto actual y el minuto 59 de la siguiente hora, considerando también los segundos exactos, etc.
- 7- Esperamos los segundos necesarios como anteriormente con el método CurrentThread y cuando el tiempo se cumple entra en el bucle While, que siempre estará a TRUE.
- 8- Todo el proceso anterior se repite desde el punto 5, siendo el punto 7 colocado ahora al final del bucle While.

El programa equivalente para capturas de pantalla reiteradas cada 5 minutos se puede hacer igual que el programa de capturas cada hora

exacta, y solo cambiaría la parte del código "CALCULO DEL TIEMPO

RESTANTE...", pudiéndose hacer no con

diferencias entre minutos sino con

operaciones de módulo 5. Para que sirva de

ayuda proporciono una pequeña parte del

código propio:

```
public static void main(String[] args) {
    int i=0;
    String FILENAME="C:/Users/juan antonio/Desktop/UPCT/CURSAND...";
    //inicialmente negativo o no valida
    int hora_aux=-1, minuto_aux=-1, segundos_aux=-1;

    try{
        System.out.println("Capturas iniciadas...");

        /**TIEMPO DEL SISTEMA*/
        Calendar calendario1 = Calendar.getInstance();
        int hora1, minutos1, segundos1;
        hora1 =calendario1.get(Calendar.HOUR_OF_DAY);
        minutos1= calendario1.get(Calendar.MINUTE);
        segundos1 = calendario1.get(Calendar.SECOND);

        //Guardamos tiempo del sistema en auxiliares
        hora_aux= hora1;
        minuto_aux= minutos1;
        segundos_aux=segundos1;

        /**PRIMERA CAPTURA DE PANTALLA TRAS 10 SEGUNDOS DE EJECUCIÓN*/
        Thread.currentThread().sleep(1000*10); //ESPERA DE 10 SEGUNDOS
        A_CapturaDePantalla.captureScreen(FILENAME); //El objeto que llama al método static es la propia clase
        System.out.println("Captura de pantalla S"+i+" hecha a las "+hora1+"-"+minutos1+"-"+segundos1);
        i++; //incremento para el contador de FOTO

        /**CÁLCULO DE TIEMPO RESTANTE HASTA LA SIGUIENTE HORA EXACTA*/
    }
}
```

7.3- Programa controlador CFD's (explicado)

La longitud del código desarrollado en esta clase Java es relativamente extenso, pero la idea es bastante simple conceptualmente (eso es lo realmente importante). La idea o paradigma de programación subyacente tomada en cuenta es la siguiente: mientras la interfaz sigue escuchando y visible todo el tiempo a y hacia el usuario, el programa de inteligencia artificial de trading seguirá activo en segundo plano. El usuario no será consciente de este funcionamiento más allá del hecho de que cuando se cierre algún indicador de MT4 la etiqueta que indica si el MT4 está funcionando correctamente cambiará.

Por tanto, en esta clase Java se va a comunicar los deseos del usuario con la ejecución de las operaciones en el terminal MetaTrader4 (MT4).

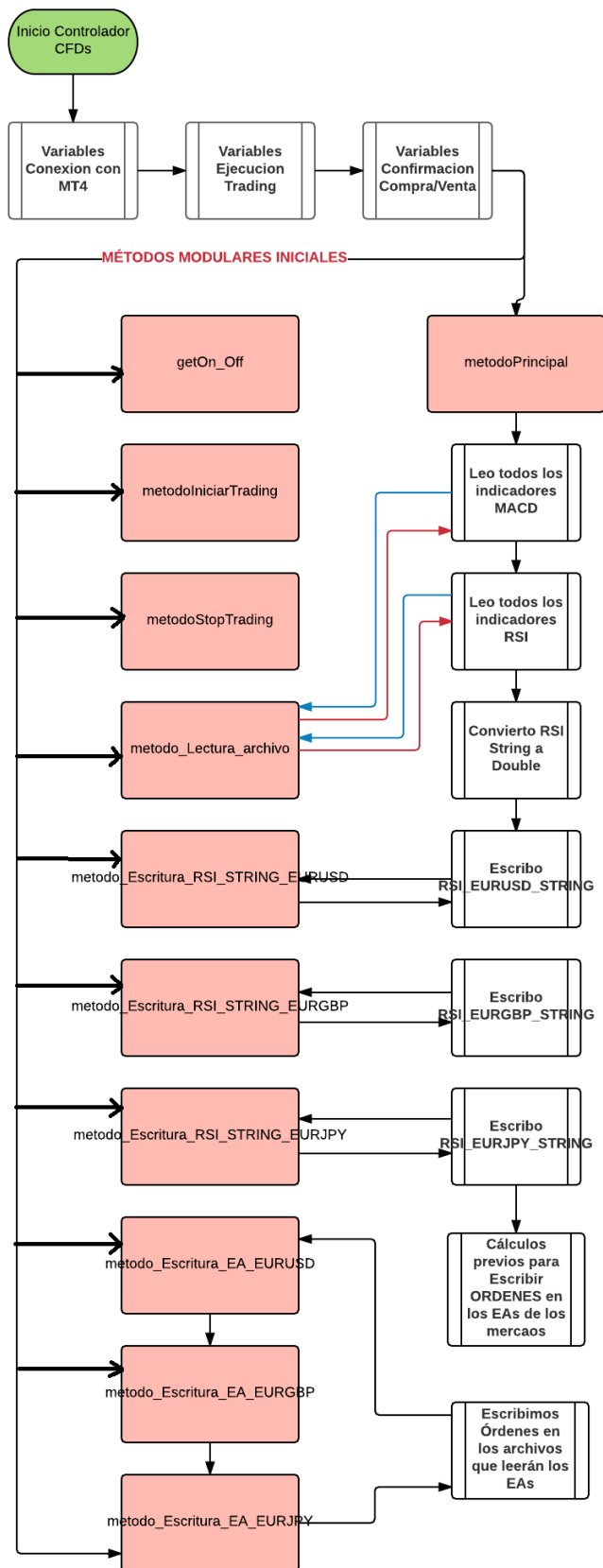
Ahora bien, ¿Cómo se ha implementado esta idea y cómo se ha logrado llevar a la práctica? Se ha seguido estos pasos:

- 1º) La clase tiene un método GET que devuelve un valor "-1" si no encuentra indicios de que el MT4 esté funcionando correctamente. En caso de funcionar correctamente o para un caso Default devolverá el valor "1". Este valor será leído periódicamente por la clase "Ventana" que es la encargada de crear la Window o interfaz de usuario, la cual modificará la etiqueta lateral derecha a VERDE ("MT4 ON") o a ROJO ("MT4 OFF").
- 2º) En caso de tener la etiqueta en verde, el programa deberá de ejecutar las estrategias que estén disponibles de las 3 máximas que hay posibles. El programa Java traslada la decisión que deberá ejecutar en el mercado al terminar MT4 escribiendo un breve texto en el fichero adecuado, y posteriormente el Expert Advisor de MT4 respectivo para cada mercado ejecutará dicha orden. Se trata tan solo de hacer una escritura/lectura de archivos sincronizada temporalmente (para evitar que colisionen los manejadores de lectura/escritura de archivos entre sí).
- 3º) Este programa, encargado de realizar las operaciones en los mercados financieros, no solo deja la respuesta de MT4 ON/OFF estática a la espera de ser leída. El propio código podrá consultar a sí mismo por valor de sus variables y tener consciencia de si ejecutar una orden (porque está MT4 a ON) o no (porque MT4 esté a OFF).

Acompaño la explicación con el siguiente flujo-grama:

PROGRAMA CONTROLADOR CFD'S

juan antonio | M



7.3- Lectura de noticias macro-económicas con ERROR 403

Las noticias de los mercados financieros se van a tomar estableciendo una conexión HTTPS a la URL <http://es.investing.com/economic-calendar/>. En esta web diariamente se emite un calendario económico de macro-economía, pero si se desea puede verse semanal utilizando la opción conveniente. En nuestro caso descargaremos el HTML de la página que se carga por defecto, la cual muestra las noticias del día actual.

Las noticias, por ejemplo programadas para el lunes 13 de marzo de 2017 eran estas:

Hora	Div.	Importancia	Evento	Actual	Previsión	Anterior
Lunes, 13 de marzo de 2017						
Todo el día		Festividad	India - Holi - Festival de la Primavera			
00:50	JPY	▼▼▼	Pedidos de maquinaria (subyacente) (Anual) (Ene)	-3,3%	6,7%	
00:50	JPY	▼▼▼	Pedidos de maquinaria (subyacente) (Mensual) (Ene)	0,5%	6,7%	
00:50	JPY	▼▼▼	Índice de precios de bienes corporativos (Mensual) (Mar)	-0,1%	0,6%	
00:50	JPY	▼▼▼	Índice de precios de bienes vorporativos (Anual) (Mar)	1,0%	-1,2%	
05:30	JPY	▼▼▼	Índice de actividad del sector servicios (Mensual)	0,2%	-0,4%	
07:30	GBP	▼▼▼	Índice Investing.com GBP/USD		47,1%	
07:30	USD	▼▼▼	Índice Investing.com Oro		75,5%	
07:30	USD	▼▼▼	Índice Investing.com S&P 500		56,7%	
07:30	CAD	▼▼▼	Índice Investing.com USD/CAD		27,4%	
07:30	CHF	▼▼▼	Índice Investing.com USD/CHF		54,5%	
07:30	AUD	▼▼▼	Índice Investing.com AUD/USD		60,7%	
07:30	JPY	▼▼▼	Índice Investing.com USD/JPY		57,0%	
07:30	NZD	▼▼▼	Índice Investing.com NZD/USD		59,0%	
07:30	EUR	▼▼▼	Índice Investing.com EUR/USD		42,8%	
10:00	EUR	▼▼▼	Producción industrial de Italia (Mensual) (Ene)	-0,8%	1,4%	
10:00	EUR	▼▼▼	Producción industrial de Italia (Anual) (Ene)	3,3%	6,6%	
11:40	EUR	▼▼▼	Subasta de deuda italiana a 3 años ligada a la inflación (BTP)		0,25%	
11:40	EUR	▼▼▼	Subasta de deuda italiana a 7 años ligada a la inflación (BTP)		1,59%	
12:00	EUR	▼▼▼	Balanza comercial de Portugal (Ene)		-3,05B	
13:45	EUR	▼▼▼	ECB's Lautenschlaeger Speaks			
14:30	EUR	▼▼▼	Declaraciones de Mario Draghi, presidente del BCE			
15:00	USD	▼▼▼	Índice de tendencias del empleo de The Conference Board (Feb)		130,00	
15:00	USD	▼▼▼	Índice de la Fed sobre las condiciones del mercado laboral		1,3	
15:00	EUR	▼▼▼	Subasta de deuda francesa a 12 meses (BTF)		-0,540%	
15:00	EUR	▼▼▼	Subasta de deuda francesa a 3 meses (BTF)		-0,730%	
15:00	EUR	▼▼▼	Subasta de deuda francesa a 6 meses (BTF)		-0,715%	
15:00	EUR	▼▼▼	ECB's Constancio Speaks			
16:30	USD	▼▼▼	Subasta de deuda a 3 meses (T-Bill)		0,745%	
16:30	USD	▼▼▼	Subasta de deuda a 6 meses (T-Bill)		0,835%	
17:30	EUR	▼▼▼	ECB's Praet Speaks			

El programa Java que se desarrolla inicialmente para lograr la conexión HTTP resulta fallido, ya que su ejecución genera un error 403 (acceso prohibido). El programa en su método principal llama a un método que se encarga de hacer la conexión HTTP, y este método recibe la URL como string de entrada. Después se intenta devolver el resultado por consola. Veamos el error que se genera y el código correspondiente:

```
1 package TFG_Pruebas;
2 import java.io.*;
3
4 public class E_Lectura_HTML_1ERROR_403 {
5     public static String getHTML(String urlToRead) throws Exception {
6         //StringBuilder es una secuencia de caracteres mutable
7         StringBuilder result = new StringBuilder();
8         URL url = new URL(urlToRead);
9         HttpURLConnection conn = (HttpURLConnection) url.openConnection();
10        conn.setRequestMethod("GET");
11        BufferedReader rd = new BufferedReader(new InputStreamReader(conn.getInputStream()));
12        String line;
13        while ((line = rd.readLine()) != null) {
14            result.append(line);
15        }
16        rd.close();
17        return result.toString();
18    }
19    public static void main(String[] args) throws Exception{
20        String htmlDevuelto=getHTML("https://es.investing.com/economic-";
21        System.out.println(htmlDevuelto);
22    }
23 }
```

Console Problems @ Javadoc Declaration History

terminated> E_Lectura_HTML_1ERROR_403 [Java Application] C:\Program Files (x86)\Java\jre7\bin\javaw.exe (12/03/2017 19:02:50)

Exception in thread "main" java.io.IOException: Server returned HTTP response code: 403 for URL: https://es.investing.com/economic-
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(Unknown Source)
at TFG_Pruebas.E_Lectura_HTML_1ERROR_403.getHTML(E_Lectura_HTML_1ERROR_403.java:11)
at TFG_Pruebas.E_Lectura_HTML_1ERROR_403.main(E_Lectura_HTML_1ERROR_403.java:20)

7.4- Eliminando ERROR 403, pero sin lograr HTML todavía

En este segundo intento se intenta hacer una conexión HTTP mediante sockets, objetos de programación “URL”, “URLConnection” y Buffers de entrada. En el código siguiente se puede visualizar el programa generado.

Nótese que se introduce como cadena de caracteres en el objeto “URL” la propia URL del sitio web, que es la entrada del objeto “URLConnection” y se intenta establecer la conexión. Después se empieza a leer el código HTML que devuelve el servidor de la conexión: todo el HTML se va guardando en una variable String llamada “**contenidoHTML**”, y esta es la encargada de devolver el HTML compacto en un único strin al método principal.

Lo que ocurre en este segundo intento es lo mismo que anteriormente. A diferencia del caso anterior, ahora no utilizamos el objeto Java **URLConnection**, sino el objeto **URLConnection**. Esto es así con la esperanza de que **URLConnection** no sea tan restrictivo en lo que se refiere a protocolos de capa 7 en internet, pero vuelve a no ser suficiente efectiva la solución ideada. Veamos el error generado:

```
package TFG_Pruebas;

import java.io.BufferedReader;

public class E_Lectura_HTML_2soloJAVASCRIPT {
    //instanciamos el socket por si hace falta
    Socket ss;

    /**CONECCIÓN CON EL SERVIDOR WEB Y OBTENCIÓN DEL HTML*/
    static public String getContenidoHTML() throws IOException {
        URL url = new URL("https://es.investing.com/economy");
        URLConnection uc = url.openConnection();
        uc.connect();
        //Creamos el objeto con el que vamos a leer
        BufferedReader in = new BufferedReader(new InputStreamReader(uc.getInputStream()));
        String inputLine;
        String contenidoHTML = "";
        while ((inputLine = in.readLine()) != null) {
            contenidoHTML += inputLine + "\n";
        }
        in.close();
        return contenidoHTML;
    }
}
```

```
java.io.IOException: Server returned HTTP response code: 403 for URL: https://es.investing.com/economy
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(Unknown Source)
at sun.net.www.protocol.https.HttpsURLConnectionImpl.getInputStream(Unknown Source)
at TFG_Pruebas.E_Lectura_HTML_2soloJAVASCRIPT.getContenidoHTML(E_Lectura_HTML_2soloJAVASCRIPT.java:15)
at TFG_Pruebas.E_Lectura_HTML_2soloJAVASCRIPT.main(E_Lectura_HTML_2soloJAVASCRIPT.java:10)
```


7.5- Código Python que devuelve el HTML deseado

En este punto del desarrollo software había perdido la esperanza de encontrar una forma de acceder remotamente a un servidor HTTPS. Decidí entonces utilizar el lenguaje **Python** para tratar de descubrir si se podría simular una conexión con el servidor y hacerle creer a este servidor que lo realizaba el navegador GoogleChrome. Justo antes de tirar la toalla di con este código, ya que crear un Parse de JavaScript que tradujera a HTML era algo impensable.

Véase las siguientes líneas del código **Paython** que se encuentran en color **ROJO**: en estas líneas se puede ver que se coloca la URL, se preparan una serie de cabeceras para enviar la petición al servidor (sí nos haremos pasar por el navegador GoogleChorme), abrimos la conexión, leemos el HTML y guardamos todo el contenido en donde queramos. El código siguiente efectivamente pude comprobar que realizaba lo que estaba buscando, por lo que ahora tenía que intentar realizarlo con Java.

Veamos el código Paython:

```
import urllib2

url = "http://es.investing.com/economic-calendar/"

hdr = {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.11 (KHTML,
like Gecko) Chrome/23.0.1271.64 Safari/537.11',

'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',

'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.3',

'Accept-Encoding': 'none',

'Accept-Language': 'en-US,en;q=0.8',

'Connection': 'keep-alive'}

req = urllib2.Request(url, headers=hdr)

try:

    page = urllib2.urlopen(req)

except urllib2.HTTPError, e:

    print e.fp.read()

content = page.read()

print content
```

7.6- Solución al problema de acceso Prohibido 403 (explicado)

Esta clase se llamará "Parse_HTML" y la describimos a continuación:

- 1- Importamos 4 clases de programación del package JAVA AWT: Dimension, Robot, Toolkit y KeyEvent. Además necesitamos importar la clase IOException del package JAVA IO.
- 2- Abrimos la clase Java y conseguimos la resolución de la pantalla con el objeto Toolkit al colocarlos como variables de instancia de la clase.
- 3- Abro el método PRINCIPAL de la clase, y lo primero que hace el método principal es llamar al método externo "**OpenURL**". Como entrada al método se le introduce una cadena de caracteres String que corresponde a la URL del sitio web deseado: <https://es.investing.com/economic-calendar/>.
- 4- Entramos al método llamado "**OpenURL**", al cual se le introduce como entrada una URL a la que conectarse.
- 5- Ahora dentro del método pedimos el nombre del sistema operativo, y si el nombre obtenido es igual a "**Windows**" entonces arranca el programa el navegador **GoogleChrome**. Esto hará que **GoogleChrome** se abra y cargue la página web de la URL que introducimos al método. Esto es gracias a que Java ha accedido a los archivos DLL del sistema operativo para abrir el archivo correspondiente al navegador **GoogleChrome**.
- 6- Si no fuera posible abrir el navegador saltaría una exception por consola. Si se pudo abrir el navegador esto sería ¡¡fantástico!!
- 7- Es el momento de retomar el control del programa por el método principal, y lo primero que hace el método principal es cargar las coordenadas donde mover el mouse del ordenador mediante la utilización de un objeto de la clase Robot.
- 8- Colocado el ratón del ordenador sobre las coordenadas deseadas vamos a utilizar la clase robot para simular también la pulsación del botón 2 del ratón (botón derecho), después botón 1 (botón izquierdo) y guardar el código HTML del sitio web en mi ordenador mediante movimientos de ratón, clic de

ratón y pulsaciones de teclado (todo simulado por el programa).

9- Accedemos tras 10 segundos de espera al archivo .html recién descargado.

10-Guardamos el contenido HTML del archivo descargado, tras leerlo, en una variable de tipo String. El contenido HTML va a ir como una cadena única de caracteres de texto plano. El contenido HTML de esta variable se enviará a la clase que llamó a esta, y se enviará como return del método principal de la clase actual.

Este código no se mostrará por motivos de propiedad intelectual en este informe, pero considero que con la explicación anterior es más que viable entender el desarrollo hecho y fácilmente realizable por el lector de esta sección.

8. Team Viewer

Team Viewer es un programa que permite a cualquier usuario controlar computadores desde cualquier parte del mundo y sin estar presente físicamente. Me ha parecido buena idea incorporar esta sección a modo de propuesta de control remoto de trading, puesto que se ha creado la aplicación software solamente para sistema operativo Window (ordenadores laptop y sobremesa).

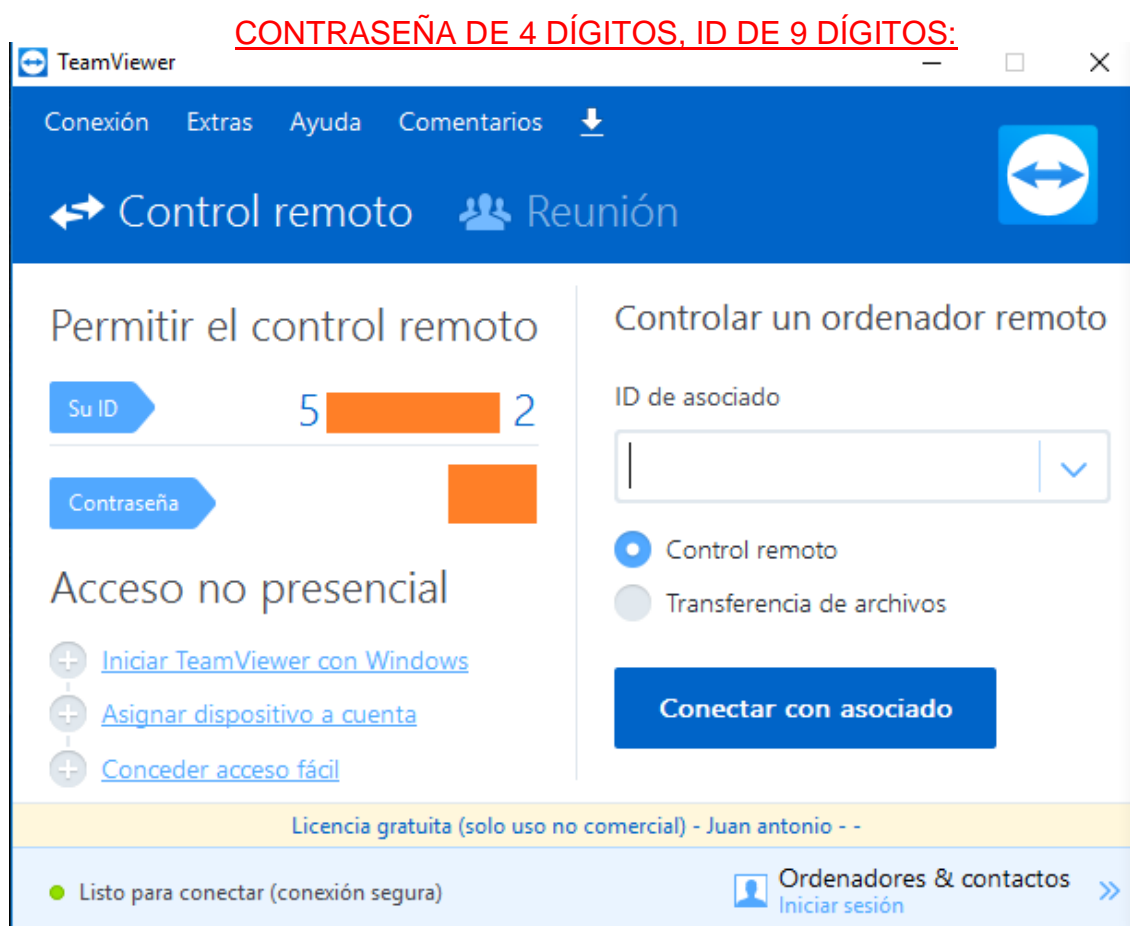
Utilizando Team Viewer el usuario podría obtener la movilidad idónea al poder controlar su operativa de trading desde su teléfono móvil (siempre disponible y portable en su bolsillo), pero conservando la potencia de ejecución o computación que nos ofrece un ordenador. ¿Acaso no es cierto que en un teléfono móvil la potencia de procesamiento y ejecución de procesos simultáneos es menor que la obtenida en un ordenador? Efectivamente, así es.

De esta forma podemos colocar un antivirus y diversas herramientas en nuestro ordenador para evitar programas malware y spyware, al igual que en nuestro smartPhone. Puesto que nuestro ordenador tendrá generalmente más recursos computacionales que el smartPhone el antivirus que corre en él será más eficiente y dará más seguridad: las estrategias de trading estarán corriendo sobre el ordenador, y el control remoto ejecutado con conexión socket entre el PC y el SmartPhone. Las estrategia software de bolsa, que es lo más importante del proceso, queda protegida eficazmente.

La comunicación entre SmartPhone y laptop también será protegida mediante protocolo SSL en capa de aplicación. Esto lo incorpora la aplicación Team Viewer por sí misma.

La forma de conectar el dispositivo A y el dispositivo B es la siguiente:

- 1- Abrimos el programa Team Viewer en el PC, y se nos da un ID de usuario y una contraseña (cada vez que cierre el programa y lo vuelva a abrir la contraseña cambiará).
- 2- ID de usuario y contraseña actual deben ser memorizados o anotados en algún lugar. Tener precaución para proteger la contraseña, porque no se cambiará hasta que en el ordenador no se cierre la APP.



- 3- Dejar la APP ejecutándose en el ordenador, como se ve en la figura anterior. Podemos cerrar la tapa del ordenador, apagar la pantalla, etc. Solo es necesario que el ordenador se encuentre siempre conectado a internet y ejecutándose la APP Team Viewer todo el tiempo.
- 4- Abrir en el Smart Fhone la APP Team Viewer correspondiente, y colocar en la parte de “Dispositivo Asociado” el ID que aparecería en la foto anterior del ordenador, así como la contraseña que se da debajo del ID (tapado ID por seguridad).
- 5- Una vez se establece la conexión entre PC y SamrtPhone podemos controlar el PC desde el teléfono móvil, a base de pulsaciones táctiles hechas en el terminal móvil: si mueves el dedo por la pantalla del SmartPhone la flecha dl ratón se moverá en el PC, si mantienes pulsada la pantalla 2 segundo se interpretará en el PC como un doble clic de ratón, etc.

- 6- Una vez acabes de colocar tu operativa, abrir el programa deseado en tu PC desde tu teléfono móvil, cierra la conexión socket en Team Viewer y estará listo para volver a realizar otra conexión más tarde (con misma ID y password). No confunda cerrar la conexión de la aplicación con apagar el ordenador, pues si cierra el ordenador después no podrá retomar el control de su ordenador remotamente, ni encenderlo ni nada.

De esta manera queda tratada un área que no era un punto del proyecto inicial, pero que consideraba importante: el acceso remoto y accesibilidad global. Sin duda es una solución que sólo aporta “seudo-control” remoto, pero reitero que esta sección no se consideró como objetivo a cumplir en este proyecto. La idea principal del proyecto estaba basada en el “diseño de una APP en Java en local combinada con estrategias programables en software de bolsa MQL4”. Esta sección es un extra, y se espera que sirva como idea para futuros Trabajos Fin de Estudios de otros alumnos en caso de decidir utilizar la base anterior: la idea de realizar el tutorial en el apartado 5 y explicar MQL4 en profundidad en sección 6 era aportar continuidad al proyecto.

9. Conclusiones y trabajos futuros

El objetivo que se estaba buscando alcanzar con este trabajo, como ya se explicó en la introducción, era obtener formación de finanzas combinada con ingeniería de software. Una vez completada la formación era cuestión de terminar desarrollando una herramienta para mi propio uso que significara un ingreso extra a mi trabajo, de manera pasiva. Sin lugar a dudas he cumplido el objetivo.

No solo he logrado mis objetivos generales de manera satisfactoria, sino que en el proceso he aprendido procedimientos que no conocía y que desde ahora me serán de gran utilidad. Aquí va alguno de la experiencia adquirida: por primera vez resolví problemas software de la vida real y obtuve soluciones que reportaban ingresos reales sobre mis finanzas, aprendí a conectarme a servidores de cotizaciones FOREX en tiempo real, he generado un programa PARSE de HTML para localizar noticias financieras en una URL indicada, etc. En otras palabras: he logrado aprender 2 lenguajes de programación a fondo y conceptos de inversión en bolsa sobre inversiones de alto riesgo, para finalmente desarrollar programas de estrategias financieras de alto riesgo.

A nivel del área de gestión de proyectos software también he aprendido bastante: principalmente a sintetizar y modular el trabajo de proyectos más grandes en módulos más pequeños de manera eficaz. He utilizado herramientas rápidas para realizar Flujo-Gramas funcionales. La modularidad me ha permitido aportar la gama completa de soluciones software que lleva todo un proyecto software real en cualquier trabajo en una empresa privada, efectiva y eficazmente.

Pese a todos la formación que me reporta este Trabajo Fin de Grado yo me quedo con satisfacción de haber crecido tanto personalmente (leí varios libros de Coach financiero) como consolidar mis conocimientos técnicos. Sin duda este proyecto ha marcado un punto de inflexión en mi forma de ver mi profesión, muy positivo, de aquí en adelante.

Siendo crítico con mi trabajo debo decir que me hubiera gustado programar una plataforma web y ejecutar el software en máquinas remotas. Por falta de tiempo decidí hacer un proyecto basado en el estudio el análisis de mercados y los fundamentos software de programación de bolsa. Deseaba más. Puede ampliarse ente proyecto también desde la parte de arquitectura de redes de telecomunicaciones: uso de máquinas virtuales distintas dentro de un mismo PC con MetaTrader4 instalado en una máquina y la aplicación Java instalada en otra.

Desde un punto de vista más empresarial, podría desarrollarse un sistema de distribución de señales de entrada al mercado a partir del sistema que dejo finalizado por mi parte. Podría crearse algún tipo de servicio de pago de SMS de pago mensual, pertenecer a grupos de wasap privados previo pago de cuota mensual o técnicas parecidas.

Por último diré que, el sistema definido en este proyecto fin de grado es casi de tipo “Maestro/Esclavo” (recibe la orden de Java para operar un mercado, pero solo realiza la orden automática MetaTrader4 si el mercado está abierto). Podría mejorarse el sistema realizándose un programa de tipo Maestro/Esclavo puro, ya que la complejidad algorítmica podría disminuir considerablemente.

10.Bibliografía

Libros técnicos:

- 1) Deitel & Deitel: Cómo programar en Java.
- 2) Análisis técnico de los mercados financieros.
- 3) Vivir del Trading.

Libros de Coaching recomendados:

- 1) ¿Quién se ha llevado mi queso?
(Autor: Spencer Johnson M.D., Ed. Empresa Activa; ISBN 9788416715008).
- 2) Padre rico, padre pobre.
(Autor: Robert T.Kiyosaki; Ed. Punto de Lectura; ISBN 9788466332125).
- 3) Los 7 hábitos de la gente altamente efectiva.
(Autor: Stephen R. Covey; Ed. Booked; ISBN 9788449324949).
- 4) -Los 7 paradigmas para liberarse del estrés de la gente altamente efectiva.
(Autor: Dr. Michael Olpin Sam Braken; Ed. Booked).
- 5) El ahorrador inteligente.
(Autor: Emilio Ontiveros y David Cano; Ed. Booked).
- 6) Los secretos de la mente millonaria.
(Autor: T.Harv Eker; Ed. Sirio).
- 7) Influencia.
(Autor: Robert Caldini; Ed. ilustra; ISBN 978849614850).
- 8) The Obstacle is the Way.
(Autor: Ryan Holiday; Ed. Triumph).
- 9) Snow Ball.
(Autor: Ryan Holiday; Ed. Triumph).

Enlace de internet:

Estrategia opciones binarias	https://www.youtube.com/watch?v=JuomQVk3sr8
Información servidores gratis FOREX	http://www.vivirdeopcionesbinarias.com/2014/12/descargar-metatrader-4.html
Figuras Chartistas	https://www.youtube.com/watch?v=C4Sf4POOCn4
Tutoriales programación MQL4	https://quivofx.com/school/learn-mql4/
Estrategia RSI+MACD	https://www.rankia.com/blog/ahorrador-compulsivo/628258-indicadores-tecnicos-iii-rsi-tactica-acechador
Tutorial sobre EAs	https://www.youtube.com/watch?v=NJkalyWlfrE
Tutorial Indicador Técnico	https://www.youtube.com/watch?v=Lu7sRQBmVsQ
Funcionamiento de indicadores de medias móviles	https://www.youtube.com/watch?v=PiCPFPE-rMQ
Estrategia "CERO" para trading intradiario	http://www.novatostradingclub.com/formacion/la-estrategia-cero/

