Sistema de trading automático y backtesting de estrategias algorítmicas de inversión

mschapiro@doctorsapix.com

Algunas definiciones…

Algoritmo: conjunto de instrucciones o reglas bien definidas, ordenadas y finitas que permiten realizar una actividad mediante pasos sucesivos que no generen dudas a quien lo ejecute (wikipedia).

Automatizar: Traspasar trabajo de un hombre a una máquina o autómata.

Observaciones

1) Un algoritmo no necesariamente debe ejecutarse mediante una computadora o Implementarse mediante un lenguaje de programación. La definición de algoritmo no Implica existencia de tecnología.

2) La automatización de una tarea no implica que el 100% de la misma sea ejecutada por una computadora y no exista participación humana alguna. La tarea puede estar parcialmente automatizada, o los humanos involucrados pueden estar cumpliendo el papel de autómatas por diferentes razones (costos, leyes, etc.)

Luego…

Trading algorítmico: Llevar adelante una estrategia de trading en la que la toma de decisiones (de compra o venta) se lleva adelante basándose en reglas predefinidas, exactas, etc.

Trading automático: Llevar adelante una estrategia de trading en la que el proceso de compra-venta minimiza la participación de humanos

Trading de alta frecuencia: La frecuencia es mayor que en el trading de baja Frecuencia ☺

Aunque la definición parezca poco seria, la realidad es que “Alto” es un adjetivo difuso, difícil de categorizar sin un contexto…

Hablar de alta frecuencia en el Merval es muy diferente a hablar de alta frecuencia en el S&P, y esto impacta de lleno en la viabilidad tecnológica y financiera de un proyecto de estas características.

Mientras que hacer trading automático de alta frecuencia en el NYSE demanda millones de dólares y lleva a luchas inmobiliarias para estar 100 metros más cerca de un servidor de datos, en Argentina se podría hacer trading algorítmico, automático y de alta frecuencia con una computadora 386 y Windows 3.1 (no es broma!)…

Simplemente el hecho de que nadie lo esté haciendo (o al menos no muchos) hace que aprovechar una oportunidad sea mucho más factible que en otros mercados.

Oportunidades: Una oportunidad no implica necesariamente la existencia de una posibilidad de arbitraje, puede tratarse de un punto de entrada o salida basándonos en nuestro algoritmo de trading, teóricamente "ganador" (sino no lo implementaríamos).

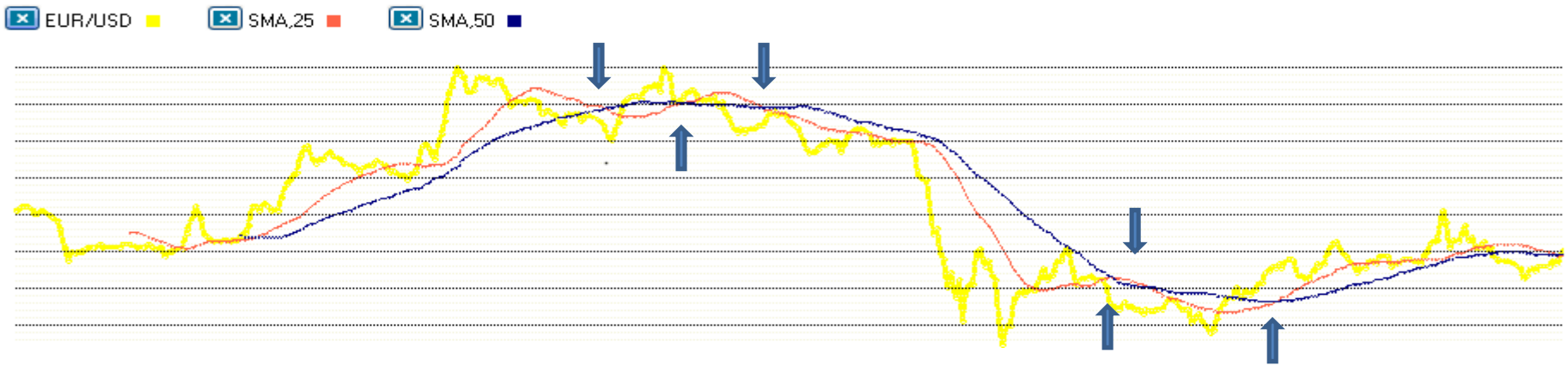
Dicha oportunidad estará disponible por mucho más tiempo (y seguramente en más ocasiones durante el mismo día) en un mercado emergente ilíquido que en un mercado muy desarrollado.

Ejemplo: Realizar un arbitraje estadístico entre MSFT y GOOG (por decir algo…) es un plan bastante más ambicioso que realizar lo mismo con TVPA y TVPY

Importante: Otro tema diferente es si existe profundidad de mercado para aprovechar esas oportunidades o simplemente visualizamos precios a los que no podemos acceder. Es posible que veamos oportunidades que en realidad no lo son, dado que no hay nadie dispuesto a comprarnos o vendernos.

Ejemplo de una estrategia algorítmica

Cruce de SMA de 25 y 50 observaciones



Alguna gente puede pensar que cuando se cruza un SMA de 50 observaciones con uno de 25, esto sugiere una señal de compra o venta (dependiendo de cómo se cruzan).

No importa si esto tiene sentido o no a nivel financiero, es una estrategia algorítmica porque tiene reglas bien precisas:

SI SMA(50,lag:-2) < SMA(25,lag:-2)
SI SMA(50,lag:-1) > SMA(25,lag:-1)
→ VENDER

SI SMA(50,lag:-2) > SMA(25,lag:-2)
SI SMA(50,lag:-1) < SMA(25,lag:-1)
→ COMPRAR

Ejemplo de una estrategia algorítmica…

Esta estrategia va a ser algorítmica independientemente del modo en que se ejecute.

Aún recibiendo el ámbito financiero todos los días en mi casa y tomando las cotizaciones de este, calculando los SMA con un ábaco y enviando las órdenes de compra y venta por telegrama, esto es trading algorítmico.

Y dado que la estrategia es algorítmica, puedo…

-Testearla empíricamente usando datos pasados

- Automatizarla para que sea factible su ejecución sin demasiado esfuerzo humano

Importante!

En algunas publicaciones se habla de trading algorítmico como el trading implementado por medio de computadoras., y se lo mezcla con trading automático. Personalmente no comparto esta idea.

Ejemplo wikipedia: In electronic financial markets, algorithmic trading or automated trading, also known as algo trading, black-box trading or robo trading, is the use of computer programs for entering trading orders with the computer algorithm deciding on aspects of the order such as the timing, price, or quantity of the order, or in many cases initiating the order without human intervention

Objetivos funcionales

1) Desarrollar un sistema que sea capaz de testear empíricamente **cualquier** estrategia algorítmica de inversión definida formalmente..

2) Que el uso de ese sistema no requiera **programación procedural** por parte del usuario.

3) Que la programación de la estrategia sea más bien una **definición formal de las Reglas de trading** (programación declarativa, basada en el qué y no el cómo).

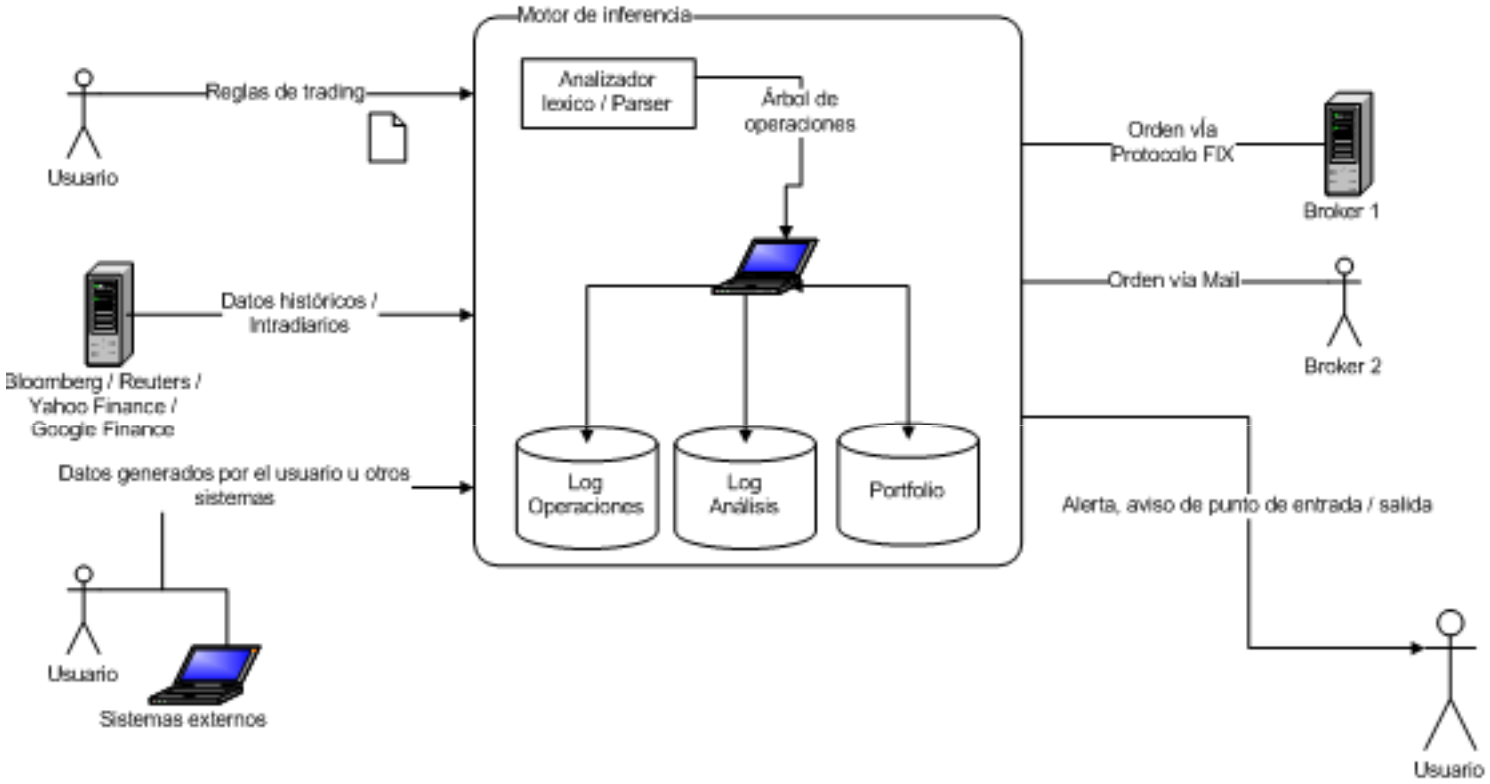
4) Que el sistema pueda demostrar qué hizo y por qué lo hizo (trazabilidad).

Indirectamente…

5) Logrando los objetivos anteriores, no se necesita mucho más para tradear Automáticamente, pues si somos capaces de controlar la ejecución de la estrategia con datos históricos, lo podremos hacer con datos en tiempo real (que de hecho son datos históricos que se van generando segundo a segundo)

Las diferencias radicarán exclusivamente en temas externos: protocolos de comunicación con brokers, retraso en los datos intradiarios que recibimos, etc.

Arquitectura de un sistema de trading automático (1)



Arquitectura de un sistema de trading automático (2)

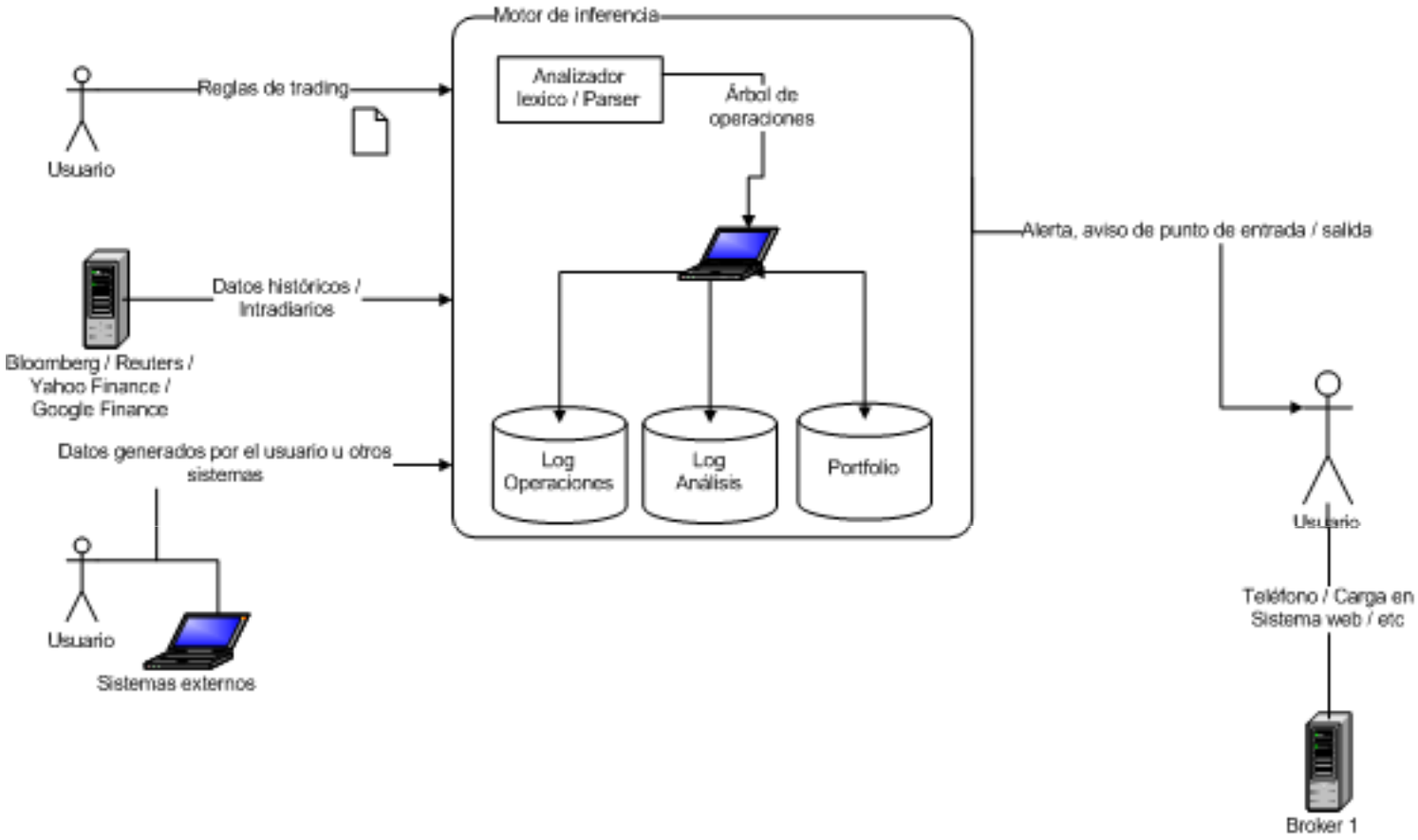
Proveedor de datos: notar que se puede partir de algo tan simple como un proveedor de datos gratuito (yahoo, google, bolsar, etc). Obviamente esto no será óptimo, pero no viola el concepto de automatización

Datos generados por el usuario u otros sistemas: El sistema debería poder nutrirse de información externa fuera de la info típica de cotizaciones de instrumentos financieros.

Ej: Una empresa puede estar interesada en tradear automáticamente considerando como parámetro el nivel de riesgo aceptable que puede tomar, y el riesgo que ya tiene tomado en otras carteras. (ver interoperabilidad)

Carga de órdenes: Todo vale si lo que queremos es hacerle llegar a nuestro broker una orden de compra o venta. Hoy por hoy, en Argentina, será nuestro broker el que nos imponga un mecanismo. Puede variar desde algo tan rústico como un Mail enviado automáticamente por nuestro sistema, a algo más sofisticado como la implementación de un protocolo estandarizado (típicamente FIX: Finance Information Exchange).

Arquitectura de un sistema de trading automático (3)



En esta solución hay trabajo manual por parte del usuario. Aún así automatizamos un proceso (análisis de datos y detección de los puntos de e/s) y podría considerarse trading automático. Al margen del nombre, la pregunta es si resuelve un problema o no.

Hoy por hoy para un inversor chico en Argentina puede ser bastante complejo encontrar un broker que le permita cargar ordenes automaticamente, mientras que tener un motor de inferencia que tome datos de yahoo finance y envíe avisos ante el cumplimiento de reglas es algo accesible a cualquiera.

Problemática…

Evaluar una o varias condiciones a lo largo de un lote de datos no es nada muy complicado…

Por ejemplo, si quisiéramos correr un algoritmo que simule la compra de acciones de MSFT cuando esta vale menos de 70 dólares y la venta cuando vale más, tendríamos una estructura similar a la siguiente:

```
For i = 1 To totalDias
        If msft.precio > 70 Then
           venderMsft()
        Else
           comprarMsft()
        End If
Next
```

Si luego quisiéramos modificar esta condición de compra, por ejemplo, para también vender en caso de que aumente la volatilidad por encima de determinado nivel….

```
For i = 1 To totalDias
        If msft.precio > 70 or msft.volatilidadDiaria > x Then
           venderMsft()
        Else
           comprarMsft()
        End If
Next
```

Problemática…

Sin embargo, el problema cambia radicalmente si lo que queremos es evaluar **Cualquier condición y ejecutar cualquier acción**, convirtiendo al problema en algo asi:

```
For i = 1 To totalDias
        If condición() Then
                ejecutarAccion()
        End If
Next
```

La problemática se origina en el hecho de que el código de un programa de computadora es estático, no cambia mientras un usuario lo utiliza.

El usuario ingresa datos (condiciones que hacen a su algoritmo de trading) que NO ingresan como código, sino que ingresan como datos.

Es necesario transformar esos datos en comportamiento, por lo que de alguna manera estamos creando un lenguaje de programación.

Analogía 1

Se puede pensar en este problema del siguiente modo.

Supongamos que se tiene una macro de excel que realiza determinadas acciones. Dentro de esa macro hay mucho código, dentro de este un IF que evalúa determinadas condiciones.

Imaginemos que esa condición requiere ser dinámica, y que el usuario (que no sabe nada de Macros ni de VBA) sea capaz de modificarla.

Ese IF podría ser escrito del siguiente modo.

```
IF cells(1,1) = True then
End if
```

Luego, el usuario puede escribir cualquier operación lógica/matemática en la celda A1 y la macro funcionará en base a esa expresión.

Es posible realizar esto con excel dado que este tiene un potente analizador de expresiones lógicas y matemáticas

Lenguaje declarativo orientado al trading algorítmico.

Reglas

- Existen operaciones y acciones. Las acciones indican **qué** hacer, y las operaciones **cuándo**.

- Una operación puede ser lógica , matemática, o una mezcla de ambas

- Las acciones de trading se ejecutan en base al cumplimiento de operaciones lógicas.

- Una operación lógica devuelve verdadero o falso. Las operaciones lógicas están asociadas a operadores lógicos (>, <, =, etc)

- Una operación matemática retorna un número, los operadores asociados a una operación matemática son (+, -, /, *, ^, raiz, etc)

- Todas las operaciones están formadas por dos operandos.

- Una acción puede estar asociada a varias operaciones lógicas y viceversa.

- Todas las operaciones lógicas que se declaran independientemente se componen mediante operadores AND (si pasa esto Y pasa esto Y esto otro…)

Sintaxis

Acciones

→ esta flecha le indica al sistema que lo que sigue es una acción.

BUY ***cant*** *comisión*: realiza la compra de CANT acciones pagando COMISIÓN como costo de transacción

SELL ***cant*** *comisión*: realiza la venta de CANT acciones pagando COMISIÓN como costo de transacción

> **(*) cant:** notar que la cantidad a comprar o vender puede ser una operación matemática. Esto permite, entre otras cosas, generar estrategias long-short en las que queremos estar long y short en un mismo monto en dólares (no en cantidad de acciones)

MAIL *direcciónMail:* envía un mail avisando el cumplimiento de la condición

SMS número: envía un SMS avisando el cumplimiento de la condición

SOUND: realiza un sonido para que el operador del sistema se entere del cumplimiento de la condición

Notar que MAIL, SMS y SOUND sólo tienen sentido en el trading automático y no en el back testing

Sintaxis

Operaciones

Recordemos que el sistema sólo conoce operaciones de dos operandos.

Además, el analizador léxico no sabe separar en términos (por lo que el usuario es responsable de esto.

Luego…

oper1 = …

Siempre una operación se debe definir con un identificador (oper1 en este caso) seguido de un signo igual. Ej:

oper1 = (operando1 operador operando2)**;**

oper2 = ((operando1 operador operando2) operador2 operando3);

Todas las operaciones deben finalizarse con un punto y coma (;)

Sintaxis

Objetos

Existen dos objetos fundamentales en el sistema. Ticker y MyPortfolio.

Ticker representa cualquier instrumento financiero, aunque realmente podemos abstraernos de esto y representar cualquier serie de tiempo (pues si quisiéramos tradear en función del SMA o la volatilidad de las lluvias en Buenos Aires y tuviéramos esa serie de tiempo podríamos considerar que tenemos un Ticker "LluviaBA".

El objeto Ticker tiene las siguientes funciones.

Open, Close, High, Min, Volume, StDev, Change, SMA, EMA, etc.

El objeto MyPortfolio representa el portfolio que se tiene en cada momento y que va variando en función de la estrategia algorítmica cargada.

Mediante este objeto podemos controlar pérdidas o ganancias, políticas (de máxima ponderación por ejemplo), controlar la volatilidad del portfolio, etc.

Algunas funciones de este objeto son: Money, StDev, Have, Change.

Sintaxis

Opciones

Las opciones permiten definir características que impactan en la estrategia en si misma, pues no dependen ni de la acción ni de la condición.

Por el momento existen 3 opciones:

MaxExecutions: Cantidad máxima de veces que se puede ejecutar una estrategia. A priori todas las estrategias son infinitas, pues se consideran oportunidades de entrada o salida que siempre se aprovechan. Dándole un valor a este parámetro se puede lograr un comportamiento diferente.

StopLoss: Indica un stop loss en % o en dólares que se cargará por cada posición tomada.

Limit: Idéntico al stop loss pero para limitar la ganancia.

Ejemplos…

El siguiente ejemplo muestra como cargar una estrategia de compra de acciones de BPAT Y venta de FRAN siempre que el ratio de precios aumente, y permite desarmar la posición siempre que suceda la opuesto.

Son dos estrategias diferentes, dado que cada grupo de acciones está asociada a diferentes operaciones

Estrategia 1:

cond = (bpat.close[-1] / fran.close[-1]) > (bpat.close[-2] / fran.close[-2]);
→ BUY FRAN 100 0.6;
→ SELL BPAT 100 0.6;

Estrategia 2:

cond = (bpat.close[-1] / fran.close[-1]) < (bpat.close[-2] / fran.close[-2]);
→ SELL FRAN 100 0.6;
→ BUY BPAT 100 0.6;

Notar que la cantidad de acciones a comprar o vender también podría ser una operación matemática (de modo que quede en función del propio ratio de precios)

Ejemplos…

En este ejemplo la cantidad comprada de FRAN depende del precio de FRAN y de BPAT.

Estrategia 1:

```
cond = (bpat.close[-1] / fran.close[-1]) > (bpat.close[-2] / fran.close[-2]);
→ BUY FRAN ((BPAT.close[-1] / FRAN.close[-1] ) * 100 0.6;
→ SELL BPAT 100 0.6;
```

Al considerar la cantidad de acciones a comprar como una operación matemática, este valor puede quedar ligado a cualquier variable financiera que queramos.

Así como en este caso se compra una cantidad de acuerdo al precio de dos acciones, podríamos incluir volatilidades, correlaciones, funciones de análisis técnico, etc.

Ejemplo put-call parity

=== callMasPlata = callMSFT.close + (callMSFT.strike * E^-rt);
=== putMasStock = putMSFT.close + MSFT.close;

cond = callMasPlata < putMasStock;
→ SELL putMSFT;
→ SELL MSFT;
→ BUY callMSFT;
→ BUY zeroCoupon callMSFT.strike;

Si quiero puedo cargarle a la condición mis costos de transacción…

=== callMasPlata = callMSFT.close + (callMSFT.strike * E^-rt);
=== putMasStock = putMSFT.close + MSFT.close;

cond = (callMasPlata * 1.2%) < putMasStock;

Algunos ejemplos útiles de operaciones

Ratio de volumen de dos stocks: stock1.volume[lag:0] / stock2.volume[lag:0]

Ratio de volumen de dos stocks en dólares: (stock1.volume[lag:0] * stock1.close[lag:0]) / (stock2.volume[lag:0] * stock2.close[lag:0])

Cantidad de acciones tradeadas en la última hora para un stock: stock1.volume[lag:0] - stock1.volume[lag:-3600] (lag en segundos)

Change semanal: stock1.close[lag:0] / stock1.close[lag:-5]; (lag en días)

Volatilidad diaria de un stock medida en los últimos 30 días: stock1.stdev[lag:0|observations:22];

Ratio de misma volatilidad contra el mes anterior: stock1.stdev[lag:0|observations:22] / stock1.stdev[lag:-22|observations:22];

Diferencial de volatilidad entre dos stocks: stock1.stdev[lag:0|observations:22] - stock2.stdev[lag:0|observations:22];

> **Notar que siempre que aparece "stock" puede ser cualquier instrumento, una opción, un futuro, una divisa.** Si es importante de todos modos notar que por ejemplo si bien el sistema puede aceptar una serie de tiempo de una opción, no tiene una función propia para calcular la volatilidad implícita de la misma… o sea que si dijeramos opcion1.stdev nos estaríamos refiriendo a la volatilidad de la prima de la opción.

Promedio entre el high y el low intradiario: (stock1.high[lag:0] + stock1.low[lag:0]) / 2;

Promedio de correlaciones semanales entre dos stocks: ((stock1.correlation[ticker: stock2|observations:5|lag:0] + stock1.correlation[ticker: stock2|observations:5|lag:-5]) + stock1.correlation[ticker: stock2|observations:5|lag:-10]) / 3;

Interoperabilidad

Algo bastante deseable de un sistema de este tipo es que pueda tomar datos (y operar con estos) de otros sistemas.

En lo que respecta a backtesting esto es sencillo y se realiza mediante la importación de series de tiempo.

En lo que respecta a trading automático el sistema es capaz de tomar información en tiempo real de archivos de texto o páginas web, y esos datos pueden ser operados del mismo modo que se operan los valores que maneja el sistema.

Esto implica básicamente que el usuario puede tradear cualquier dato que disponga

Interoperabilidad – Ejemplo

Supongamos que una empresa tiene una regla basada en el pago de dividendos, pues cuando se anuncia el pago de dividendos de determinada acción se compra inmediatamente una cantidad de acciones en función del dividendo anunciado por esa compañía.

El sistema de trading automático no maneja esta información, pero la empresa es capaz de conseguirla.

La empresa puede administrar un archivo de texto indicando el monto de dividendo por acción, el cuál permanece en cero siempre que no haya anuncio.

Cuando hay anuncio, inmediatamente un empleado que se ocupa de esto actualiza el archivo.

Por otro lado tenemos una regla de trading algorítmico programada del siguiente modo:

```
oper1 = archivoDividendosMSFT.txt > 0;
→ BUY MSFT (1000 / archivoDividendosMSFT.txt);
with options
        maxExecutions = 1;
```

Interoperabilidad – Ejemplo

Esa misma interoperabilidad se puede lograr con un script de página web (técnica bastante utilizada para interoperar).

Esto suma la capacidad de que haya un proceso automático del cual el sistema obtiene su resultado y utiliza este para operar.

Siguiendo el ejemplo anterior, pero suponiendo que ni siquiera queramos mantener un empleado de la empresa monitoreando los anuncios de dividendos, sino que queremos programar un script que monitoree la página de la comisión nacional de valores y verifique si hay algún anuncio de dividendos para ERAR.

Esto es bastante más simple de lo que parece dado que esos anuncios, aún cuando se hacen en medios digitales, llevan normas de formato muy estrictas (visitar http://www.cnv.gov.ar/InfoFinan/Zips.asp?CodiSoc=218&DescriSoc=YPF%20S.A.&TipoDocum=7&TipoArchivo=1&TipoBalance=0&Lang=0)

Luego sólo se trata de acceder al documento y verificar si cumple con ese formato y menciona a nuestra compañía. Luego…

oper1 = http://localhost/dividendosERAR.php > 0;
→ BUY ERAR (1000 / http://localhost/dividendosERAR.php);
with options
maxExecutions = 1;

Debilidad

Una debilidad del sistema es que todavía no permite esto:

→ BUY *operación1* 100

Cuando lo permita se podrán comprar acciones dependiendo de una operación. En el caso anterior podíamos comprar acciones de Siderar si se anunciaban dividendos. Pero no podíamos comprar acciones de la empresa que anuncie dividendos (cualquiera sea esta).

Cuando esto sea posible podremos meter la siguiente regla:

oper1 = http://localhost/dividendosTicker.php <> “”;
→ BUY http://localhost/dividendosTicker.php
with options
maxExecutions = 1;

O podríamos hacer estrategias del siguiente tipo

→ SELL maximo(volatilidadImplicita, opcion1, opcion2,…,opcionN)
→ BUY minimo(volatilidadImplicita, opcion1, opcion2,…,opcionN)

Ejemplos…

Supongamos que queremos realizar una estrategia basada en correlaciones diarias.

Opción 1: Conozco la correlación "normal" entre las dos acciones (supongamos 0.6), y cargo instrucciones de trading en base a ese número:

cond1 = TVPA.correlation(TVPY, 50,-1) < 0.6;
→ BUY TVPA 100 0.6;

Olvidémonos por un momento de si esta estrategia tiene o no sentido financiero…

Opción 2: Ni siquiera quiero molestarme en establecer la correlación "normal", quiero que sea dinámica y el sistema la considere de esa forma.

=== promCorr = ((TVPA.correlation(TVPY, 50, -2) + TVPA.correlation(TVPY, 50, -3)) + TVPA.correlation(TVPY, 50, -4)) / 3;
cond1 = TVPA.correlation(TVPY, 50) < **promCorr**;
→ BUY TVPA 100 0.6;

Como se vé, en la variable promCorr calculé la correlación promedio de las últimas 3 observaciones. (en el futuro la idea es incluir funciones PROMEDIO, MÁXIMO, MÍNIMO, etc… para facilitar este tipo de instrucciones, dado que si bien se pueden construir mediante instrucciones de dos operandos, es un poco incómodo.

Analizador Léxico – Árboles binarios – ¿Cómo interpretar operaciones?

```
cond = (bpat.close[-1] / fran.close[-1]) > (bpat.close[-2] / fran.close[-2]);
```

Notar que cuando el usuario ingresa una condición de este tipo mediante una cadena de texto debemos ser capaces de comprender el significado de la misma y llevarlo a una estructura de datos que podamos evaluar.

Como cadena de texto no nos sirve de nada, pues sólo después de entender el pedido del usuario el sistema podrá determinar que debe obtener los retornos de BPAT y de FRAN, que debe calcular el ratio de ambos precios para cada día y para el día anterior, y que debe comparar esos ratios.

Para simplificar el texto voy a escribir la misma operación del siguiente modo:

```
cond = ( A / B ) > ( C / D );
```

Notar que la “forma condicional” o “forma funcional” es completamente variable, siempre respetando la idea de un operador y dos operandos. Podríamos tener algo asi:

```
cond2 = ( ( ( A + B ) / ( ( C + D ) + E ) ) * ( 2 + 1 ) ) > ( 5 / ( 2 – F ))
```

Analizador Léxico – Árboles binarios – ¿Cómo interpretar operaciones?

La operación 2 (cond2) está compuesta por muchas operaciones y no podemos ni pensar en resolverla como un todo.

Como dijimos antes, sólo conocemos operaciones simples con dos operandos, y la única forma que tenemos de resolver (A + B) / ((C + D) + E), es resolviendo primero (C + D), para obtener luego (A + B) / (CDresuelto + E), y así sucesivamente.

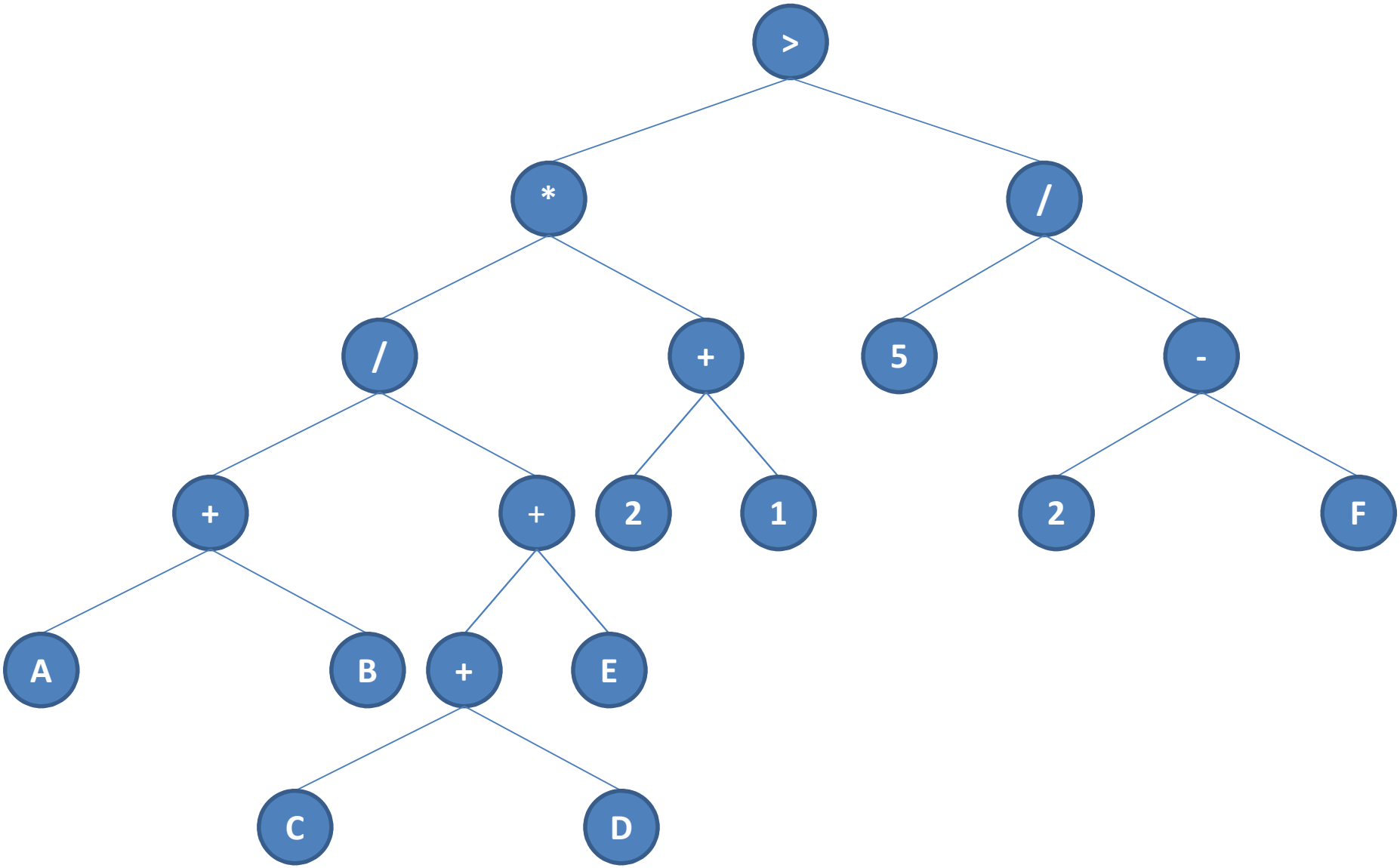
Esto se puede representar mediante un árbol binario.

Un árbol binario es una estructura computacional (al igual que un vector de datos), pero en lugar de ser una cantidad de celdas de memoria contiguas, cada celda padre tiene 2 celdas hijas (a las cuáles conoce y puede acceder).

Esta estructura permite representar muy bien problemas recursivos como el que tenemos aquí, pues una operación se compone de un operador (nodo padre) y dos operandos (nodos hijos), cada operando puede ser a su vez una operación, que debe ser resuelta antes que la operación padre a la cual pertenece.

Gráficamente…

cond2 = (((A + B) / ((C + D) + E)) * (2 + 1)) > (5 / (2 – F))



La lectura del árbol se realiza del siguiente modo: Hijo izquierdo, Padre, Hijo derecho.

Analogía 2

Una analogía que se me ocurre para ejemplificar el problema de procesar cualquier tipo de operación es comparar una calculadora científica de las que permiten ingresar una cadena de operaciones entera, ej:

(2 + 1 + 3 – 100 + 20) * 2

De las que funcionan siempre respetando el órden Operando1, Operador, Operando2, Enter.

2 + 1, Enter
Resultado + 3, Enter
Resultado – 100, Enter
Resultado + 20, Enter
Resultado * 2, Enter.

La segunda calculadora es mucho más simple, sólo necesita memoria para 2 operandos y un operador, mientras que la primera debe ser capaz de entender la operación, poder representarla mediante alguna estructura computacional, resolverla en conjunto, etc.

La primer calculadora suma del mismo modo que la segunda (sólo sabe sumar dos números), pero además sabe estructurar operaciones complejas en un árbol.

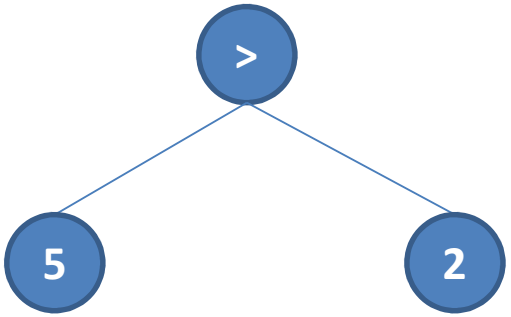
Ejemplos…

Al tener la estrategia de inversión modelada en un árbol binario tenemos algunas ventajas importantes.

Recordar que partimos de un modelo el el cuál teníamos un algoritmo codificado, totalmente estático, por ejemplo:

```
if  ( ( ( A + B ) / ( ( C + D ) + E ) ) * ( 2 + 1 ) ) > ( 5 / ( 2 – F )) then
                    comprar()
end if
```

Ahora tenemos un árbol binario, al cual podemos pensar como una estructura parametrizable, pues podemos acceder a cada nodo y modificar un valor.

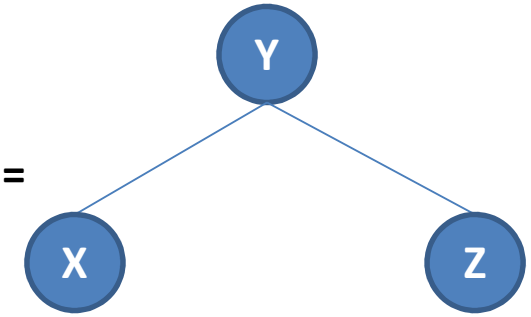



```
if 5 > 2 then
                comprar()
end if
```

Al fragmento de código (if 5 > 2…) no puedo ni pensar en modificarlo programáticamente… no puedo, en tiempo de ejecución, cambiar el 2 por un 6 y ver como reacciona mi estrategia.

Ejemplos…

F(x,y,z) =



En cambio el árbol binario es fácilmente modificable, en cualquier momento puedo acceder a cualquiera de los nodos y modificar un valor.

Puedo pensar en un árbol como una función donde cada nodo representa una variable.

Luego, **puedo maximizar esa función**, puedo analizar su sensibilidad a diferentes variables, etc.

Ahora que tengo una estrategia de inversión definida como F(x,y,z), la podría analizar (hacer backtesting) variando X, Y y Z…

F(1,>,4)
F(1,<,3)
F(5,=,0)

Etc…

Ejemplos…

Ahora que tengo toda mi estrategia parametrizada, puedo iterar fácilmente.

Ejemplo.

```
for x = 0 to 100
        for z = 0 to 100
                for y = > to <    (*) pensemos en esta linea como válida, de hecho lo es…
                        resultado = ejecutarEstrategia(x,y,z)
                next

        next
next
```

Por lo que puedo pensar en maximizar mi estrategia mediante algún método numérico, o en el peor de los casos, haciendo un barrido por todo el conjunto numérico de cada variable (lo cuál es lento, pero es mejor que nada).

Cuando hablamos de maximizar, no necesariamente se trata del resultado en dólares de la estrategia… podemos pensar en minimizar la volatilidad de los cashflows que produce, buscar la estrategia algorítmica que mejor correlaciona con otro instrumento que tenemos en nuestro portfolio, etc.

Optimización

Sintaxis

El sistema permite optimizar estrategias dejando parámetros de la misma sin definir.

Ejemplo:

```
cond1 = MSFT.CLOSE < opt[20,40,40,precioMSFT];
→ BUY MSFT 100 0.6;
```

cond1 = MSFT.CLOSE < opt[*desde*,*hasta*,*cantValores*,*nombre*];

opt: Indica que ese valor se debe iterar para buscar diferentes parametrizaciones y sus resultados.

desde: indica el primer valor del conjunto numérico en el que vamos a iterar.

hasta: indica el último valor del conjunto numérico en el que vamos a iterar.

cantValores: sirve para definir indirectamente el salto en la iteración. En el ejemplo , al tener 40 valores y un rango de 20, se puede ver que saltaremos de a 0.5 dólares

Optimización

Esta capacidad de optimizar estrategias de inversión le provee al sistema indirectamente la capacidad de “aprender”.

Una estrategia de inversión podría basarse en optimizar una estrategia para el comportamiento de determinado instrumento financiero todos los días por la mañana (backtesting) y ejecutarla a la tarde (trading automático).

Notar que esta optimización no necesariamente debe basarse en encontrar cualquier parametrización que retorne buenos resultados, sino que puede usarse para optimizar un modelo conocido.

Por ejemplo, la gente que opera según análisis técnico podría buscar para cada instrumento la parametrización óptima de sus funciones (SMA con 50 observaciones ? 55 ? Aplicado sobre observaciones minuto a minuto ? O por hora ? O por día ?)

Universo de estrategias

A la vez que definimos una función estrategia parametrizable, indirectamente definimos un universo de estrategias de inversión, dado por el producto cartesiano de todos los conjuntos numéricos de cada variable.

Así, ahora somos capaces de evaluar realmente si una estrategia es buena o no, la podemos rankear en un universo de estrategias posibles.

Limitaciones

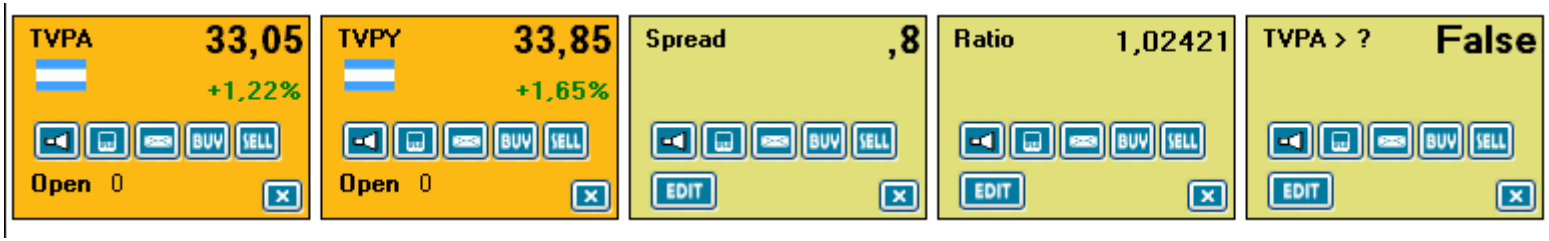
Notar que ese universo está formado únicamente por las estrategias de idéntica "forma condicional", pues la función dada por un árbol de estas características:

oper = operando1 operador operando2;

Será diferente a la función dada por un árbol de estas otras:

oper = (operando1 operador operando2) operador1 operando3;

Capturas (1)



Esta es una captura del tablero de control de la herramienta de trading automático.

Los primeros dos cuadros muestran la cotización de TVPA y TVPY.

Luego tengo cargadas operaciones matemáticas y lógicas.

La primera (spread) es una resta entre TVPY y TVPA. La segunda es la división entre ambos.

La tercera es la siguiente operación: oper = TVPA.close > TVPY.close;

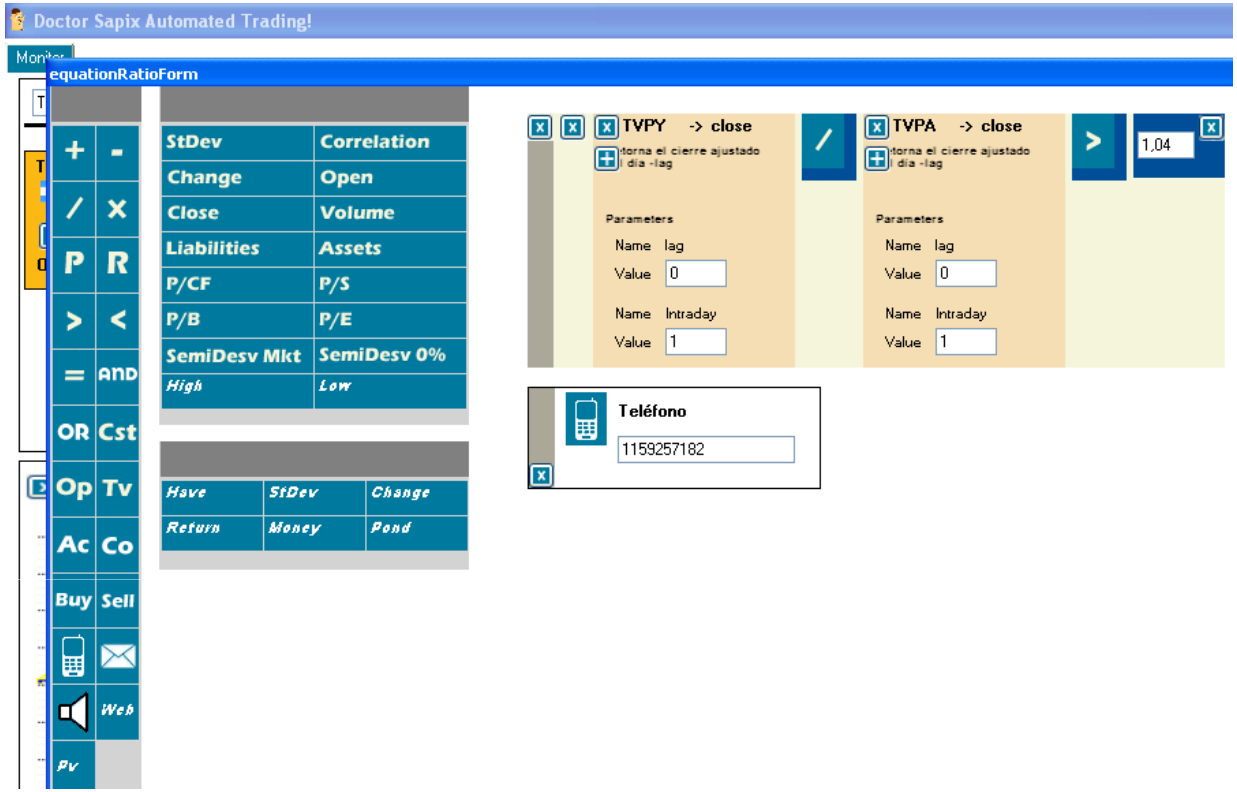
En cada recuadro tengo las acciones posibles (comprar, vender, mail, sms, sonido) que podrían realizarse en base al valor alcanzado por cada una de estas operaciones.

Capturas (2)

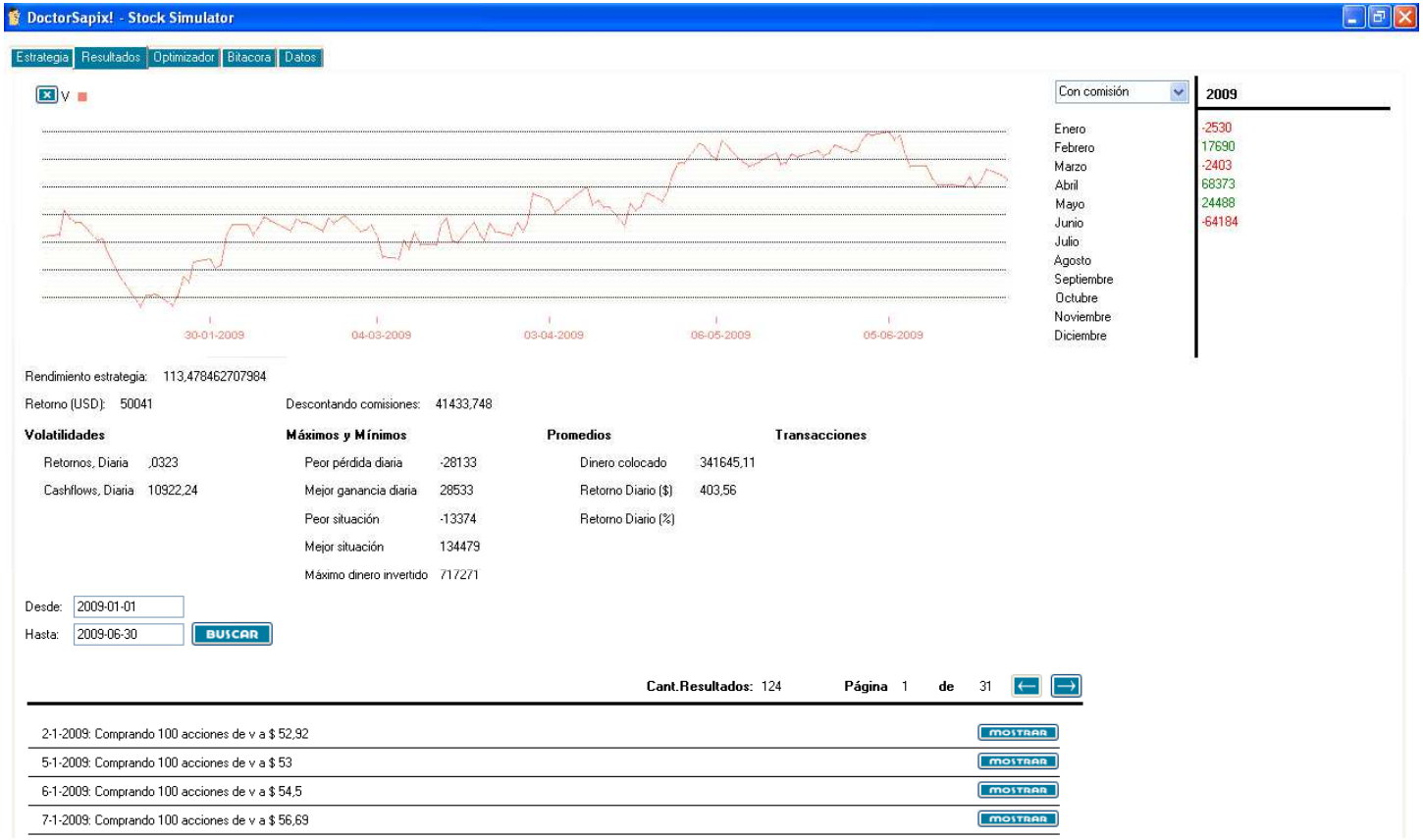
Este es el editor de estrategias en modo gráfico.

En este caso se ve que si el ratio entre TVPY y TVPA supera el valor 1.04 se enviará un SMS al 59257182

Las botoneras de la izquierda representan las disintas operaciones, funciones y objetos que conoce el sistema.



Capturas (3)



Resultado del backtesting de una estrategia determinada