

bankinter.



Broker Plus
Guía de programación
ProBacktest™

INDICE

	Pág.
1. Introducción al módulo ProBacktest™	4
2. Introducción	5
2.A) El acceso al módulo ProBacktest™	5
2.B) Zonas de configuración de ProBacktest	5
2.C) Resultados de ProBacktest™	14
2.C.1) Gráfico liquidez	14
2.C.2) Posiciones	15
2.C.3) Informe detallado	16
3. La programación de ProBacktest™	18
3.A) Entrada y salida del mercado	18
3.A.1) Instrucciones de apertura de posición	18
3.A.2) Instrucciones de STOP	22
3.A.3) Estrategia de cruces precio/SAR	22
3.B) Seguimiento de posiciones	23
3.B.1) Instrucciones de verificación del estado de posiciones	23
3.B.2) Contadores de posiciones	25
3.B.3) ENTRYINDEX	27
3.B.4) ENTRYQUOTE	28
3.B.5) PreviousTrade	29
4. Aplicaciones prácticas	31
4.A) Sistemas sobre indicadores	31
4.A.1) Sistema basado en los Heikin Ashi	31
4.A.2) Sistema basado en el Zig Zag	32
4.A.3) Sistema Breakout Range con Stop Dinámico	33
4.A.4) Sistema basado en el Estocástico Alisado	33
4.A.5) Swing Trading, ADX y Medias móviles	34
4.B) Sistemas de money management (gestión de capital)	36
4.B.1) Stop de protección (stop loss)	36
4.B.2) Objetivo de beneficio (take profit)	37
4.B.3) Stop de inactividad	37
4.B.4) Piramidación de una posición	38
4.B.5) Gestión dinámica del tamaño de órdenes	39
4.B.6) Consideración de los resultados históricos	40
4.B.7) La martingala clásica	41
4.B.8) La gran martingala	42
4.B.9) La Piquemouche	42
4.B.10) La Whittacker	44
4.B.11) La pirámide de Alembert	45
4.B.12) La contra de Alembert	46
5. Anexo: Configurar la gestión del capital	46
5.A) Capital	47

5.B) Gestión de riesgos (risk management)	47
5.B.1) Límite máximo del total invertido	47
5.B.2) Montante máximo por transacción y apalancamientos	48
5.B.3) Montante mínimo por transacción	48
5.C) Gestión de posiciones	48
5.D) Redondear la cantidad de valores a comprar / vender	49
5.E) Comisiones de operativa en acciones	49
5.E.1) Comisiones de operativa en futuros	50
6. Glosario	51

1. Introducción al módulo ProBacktest™

ProBacktest™ es la herramienta de **Broker Plus** para crear sistemas de inversión personalizados y probar su eficacia a lo largo de cualquier período del histórico disponible para un valor dado.

ProBacktest™ integra el lenguaje de programación ProBuilder™ (le aconsejamos la lectura previa de la *Guía de programación ProBuilder™*) con extensiones aplicables exclusivamente en el desarrollo de sistemas.

En este módulo puede simular aperturas de posiciones en función de condiciones personalizadas que integran:

- Posicionamiento en el mercado (orden pendiente o no, ...)
- Fechas de posicionamiento en el mercado (abrir la posición en la vela del día siguiente,...)
- Modalidades de posicionamiento (apertura de posición a precio de mercado, a precio límite X, ...)
- Posicionamiento de stops
- Resultados
- Precio de ejecución de la última orden

Los resultados de una simulación ProBacktest™ se presentan a través de los siguientes elementos:

- Curva de ganancias y pérdidas (o 'Equity Curve'), que indica el estado de su cartera virtual a medida que se aplica el sistema.
- Histograma de posiciones (verde en compras, rojo en ventas a crédito) de liquidación y de zonas sin transacciones en curso (sin velas).
- Informe detallado, que ofrece estadísticas generales de su sistema para el valor, período e históricos elegidos.


Este documento es una extensión del *Guía de programación ProBuilder™* pero también puede leerse de manera aislada.

Los usuarios más familiarizados con la programación pueden pasar directamente a la lectura del capítulo 3 o consultar el *glosario* para localizar rápidamente la explicación detallada de la función que busquen.

El contenido de toda la guía está dirigido a enseñarle a desarrollar y a probar sus propias ideas. En ningún caso se busca proporcionarle consejos de inversión.

2. Introducción

2.A) El acceso al módulo ProBacktest™

El espacio de programación ProBacktest™ es accesible desde el botón  de cualquier gráfico.

Tras pulsar ese botón, se abrirá por defecto la ventana de gestión en la pestaña 'Indicadores'. Active la pestaña 'ProBacktest' pulsando sobre la misma con el ratón. Con ello accederá a:

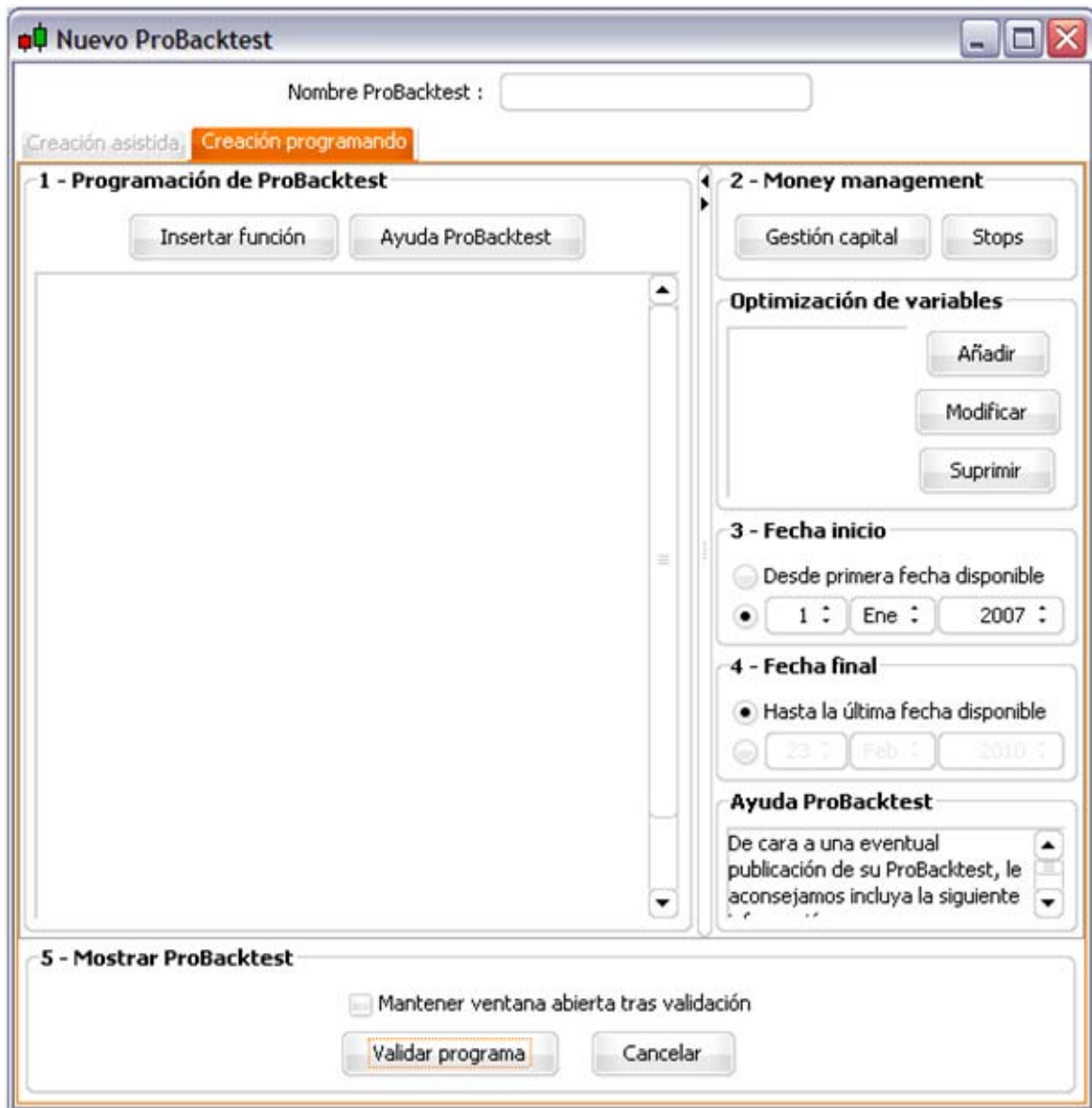
- La lista de los ProBacktests ya existentes (predefinidos o personalizados)
- Los controles de creación de un nuevo ProBacktest™
- Modificar o suprimir un ProBacktest™ existente.

Para crear un nuevo ProBacktest™, pulse 'Nuevo ProBacktest' y elija entre la posibilidad de crearlo mediante programación ('Creación programando') o con el asistente a la programación ('Creación asistida').

2.B) Zonas de configuración de ProBacktest™

Comenzaremos por examinar la creación de un backtest mediante programación. La ventana se compone de 5 zonas configurables:

- Zona de programación
- Gestión de capital (money management)
- Optimización de variables
- Fecha de inicio y de final
- Gestión de la ventana



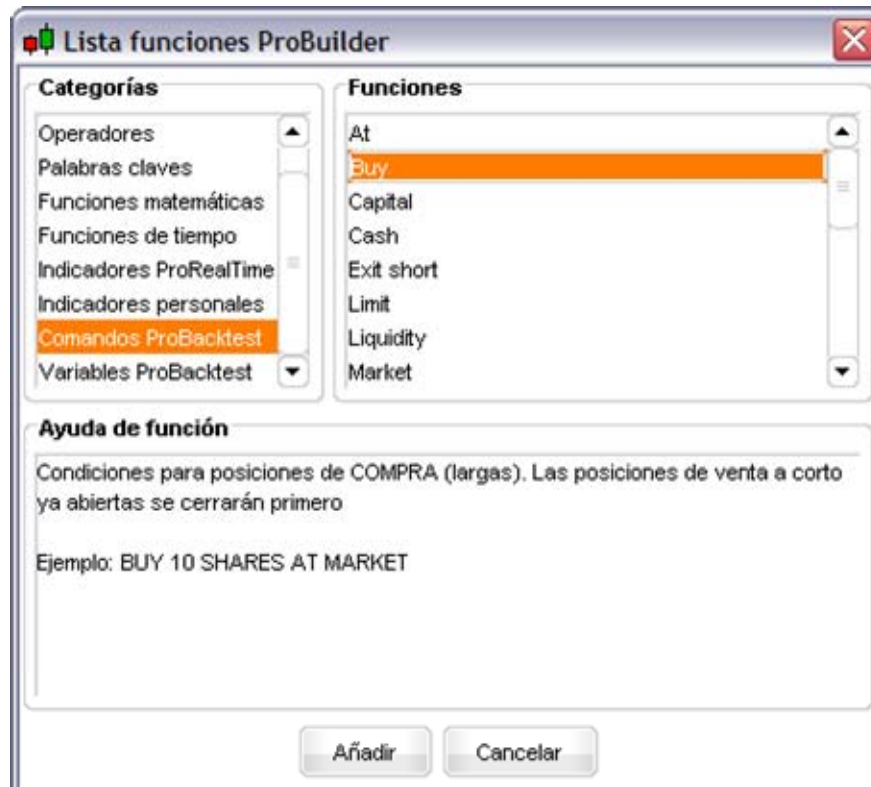
La **primera** sección le permitirá:

- Programar directamente un sistema en la zona de texto
- Utilizar la función de ayuda 'Insertar función' y acceder a la biblioteca de las funciones disponibles

La biblioteca de funciones las agrupa en 9 categorías distintas según el contexto en el que se aplican en la programación. Cada una de estas funciones se acompaña igualmente de un texto de ayuda.

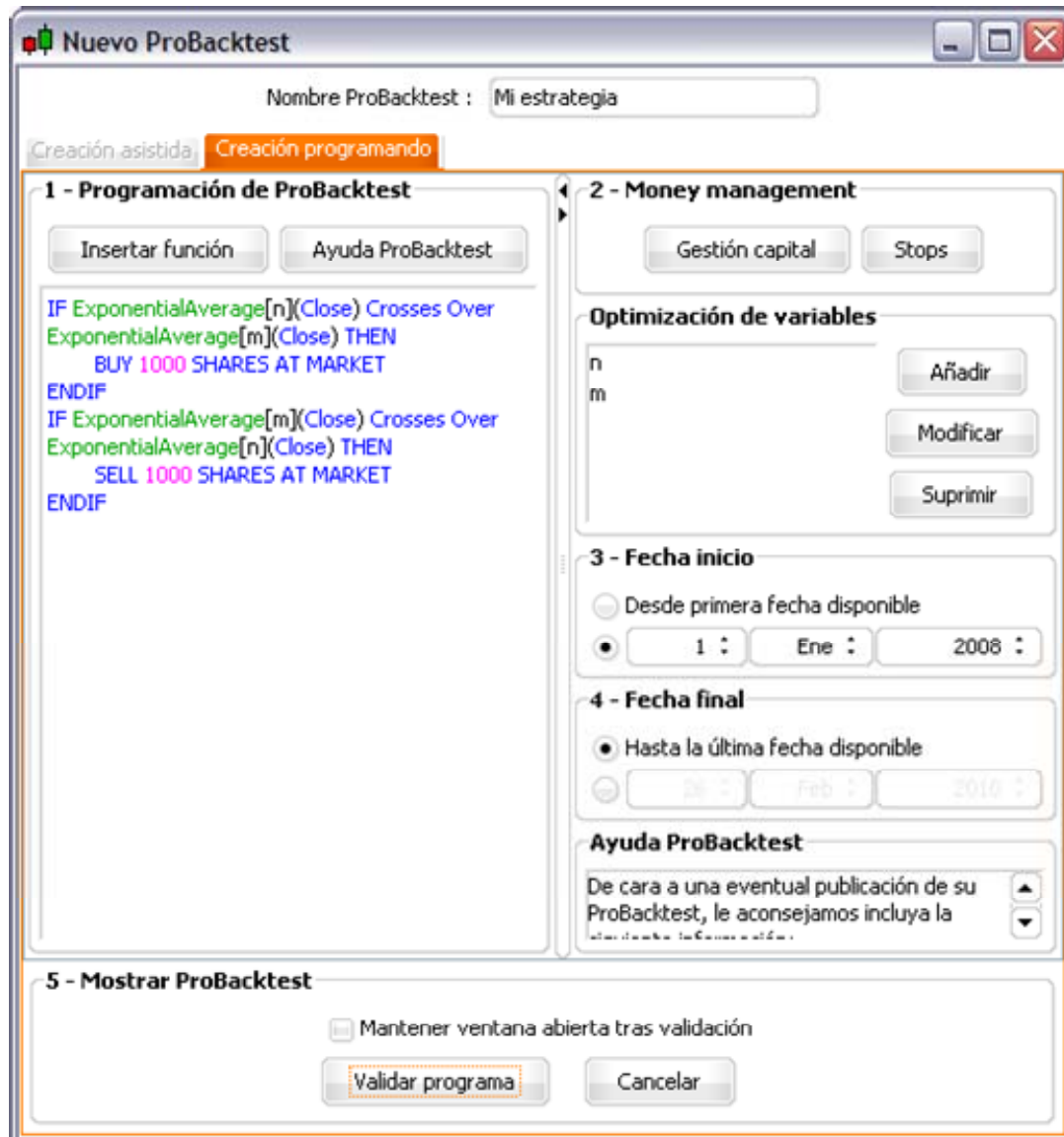
En nuestro caso utilizaremos la biblioteca pulsando 'Insertar función'.

Diríjase a la categoría 'Comandos ProBacktest' y pulse en '**BUY**'. A continuación, pulse el botón 'Añadir'. El comando se añadirá en la zona de programación.



Intentemos crear un backtest. Supongamos por ejemplo que deseamos comprar 10 valores a precio de mercado.

Para ello, realizaremos la operación descrita anteriormente para recuperar sucesivamente las instrucciones '**SHARES**', '**AT**' y '**MARKET**' (recuerde separar cada palabra de un espacio). La cantidad de valores a adquirir deberá insertarse entre los comandos '**BUY**' y '**SHARES**'. Por último, deberá atribuir un nombre a su sistema (en este ejemplo, se le denomina 'Mi Estrategia').



La **segunda** sección (Money management) permite definir la gestión del capital: coste de cada transacción, capital a invertir y stops de protección asociados.

- En 'Gestión capital' definiremos el capital inicial: la suma (ficticia) que deseamos invertir según el sistema que estamos creando, las comisiones de operativa, la gestión de riesgos y de posiciones.
- En 'Stops' podemos seleccionar cualquiera de estos 4 tipos de stop: de protección (stop loss), de beneficio (stop profit, take profit o target), dinámico (trailing stop) y de inactividad.

Gestión capital ✕

Capital

Capital inicial :

Comisión

Por orden
 Por contrato (futuros)

Comisión por orden :

 €, \$
 % operación

Comisión mínima : €, \$

Gestión de riesgos

Límite máximo de inversión : % Capital ▼

Inversión máxima por operación : % Capital ▼

Inversión mínima por operación : % Capital ▼

Gestión de órdenes

Reinvertir beneficios

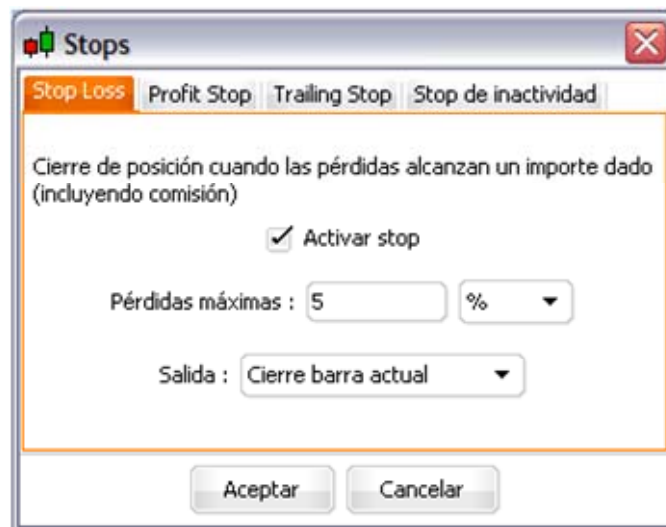
Acumular posiciones

 1 stop para todas las posiciones
 1 stop para cada posición

Se calcula un único nivel de stop. Si compra 1 unidad a 110 y luego otra a 120 con un stop loss de 20 puntos, el nivel de stop se fija al mismo nivel que si comprase ambas unidades a 115: el stop loss quedará fijado a 95 para ambas unidades

Redondear el num de valores a comprar / vender

Redondear a la unidad entera inferior
 Redondear a la unidad entera superior



Para más información acerca del ajuste de la gestión de capital, le sugerimos dirigirse al anexo disponible al final de esta guía.

La **tercera sección** le permite definir la optimización de variables. A través de esta función, podrá probar múltiples combinaciones de variables para encontrar aquellas que ofrezcan los mejores resultados.

El resultado de la optimización se presenta en un 'Informe de optimización'. En él se indican las estadísticas de cada valor para determinar así la combinación de variables a utilizar en la optimización de su sistema.

He aquí un ejemplo de sistema con optimización de los períodos de medias móviles n y m:

```
IF ExponentialAverage[n](Close) Crosses Over  
ExponentialAverage[m](Close) THEN  
    BUY 1000 SHARES AT MARKET  
ENDIF  
IF ExponentialAverage[m](Close) Crosses Over  
ExponentialAverage[n](Close) THEN  
    SELL 1000 SHARES AT MARKET  
ENDIF
```

Definimos las variables optimizadas n y m de este modo:

The dialog box 'Definición de variable' for variable 'n' contains the following fields:

- Nombre usado en programa: n
- Etiqueta en ventana propiedades: Periodos "n"
- Restricción: > 0
- Optimizando section:
 - Valor mínimo: 5
 - Valor máximo: 15
 - Paso: 1

Buttons: Aceptar, Cancelar

The dialog box 'Definición de variable' for variable 'm' contains the following fields:

- Nombre usado en programa: m
- Etiqueta en ventana propiedades: Periodos "
- Restricción: > 0
- Optimizando section:
 - Valor mínimo: 20
 - Valor máximo: 50
 - Paso: 1

Buttons: Aceptar, Cancelar

- **'Nombre usado en programa'** representa el nombre que toma la variable en nuestro código (n y m en este caso). Es importante distinguir entre mayúsculas y minúsculas.
- **'Etiqueta en ventana propiedades'** representa el nombre que se le atribuye a la variable para que ésta sea reconocible con mayor facilidad (por ejemplo 'Periodos "n" y Periodos "m").
- **'Valor mínimo y 'Valor máximo'** representan los extremos entre los que la variable puede oscilar durante las pruebas de optimización.
- **'Paso'** representa el intervalo de valores que la optimización respetará durante el análisis de los resultados.

He aquí un ejemplo de informe de optimización:

↑ Beneficios netos	Retorno de capital	Max draw down	Órdenes	% Operaciones ganadoras	Esperanza	Periodos "n"	Periodos "m"
109,0	+0,18%	516,0	4	+50,00%	27,2500	13.0	46.0
99,0	+0,16%	516,0	4	+50,00%	24,7500	11.0	50.0
58,0	+0,10%	516,0	5	+40,00%	11,6000	12.0	48.0
58,0	+0,10%	516,0	5	+40,00%	11,6000	12.0	49.0
58,0	+0,10%	516,0	5	+40,00%	11,6000	12.0	50.0
14,0	+0,02%	626,0	4	+50,00%	3,5000	13.0	49.0
14,0	+0,02%	626,0	4	+50,00%	3,5000	13.0	50.0
14,0	+0,02%	516,0	4	+50,00%	3,5000	14.0	40.0
14,0	+0,02%	516,0	4	+50,00%	3,5000	15.0	37.0
9,0	+0,01%	626,0	4	+50,00%	2,2500	15.0	46.0
9,0	+0,01%	626,0	4	+50,00%	2,2500	15.0	45.0
9,0	+0,01%	626,0	4	+50,00%	2,2500	14.0	48.0
9,0	+0,01%	626,0	4	+50,00%	2,2500	14.0	49.0
9,0	+0,01%	626,0	4	+50,00%	2,2500	14.0	47.0
4,0	+0,01%	516,0	4	+50,00%	1,0000	12.0	45.0
4,0	+0,01%	516,0	4	+50,00%	1,0000	11.0	48.0
4,0	+0,01%	631,0	4	+50,00%	1,0000	15.0	47.0
-1,0	-0,00%	626,0	4	+50,00%	-0,2500	14.0	45.0
-1,0	-0,00%	626,0	4	+50,00%	-0,2500	14.0	44.0
-1,0	-0,00%	626,0	4	+50,00%	-0,2500	13.0	48.0
-1,0	-0,00%	626,0	4	+50,00%	-0,2500	13.0	47.0
-6,0	-0,01%	626,0	4	+50,00%	-1,5000	15.0	43.0
-6,0	-0,01%	626,0	4	+50,00%	-1,5000	15.0	44.0
-6,0	-0,01%	626,0	4	+50,00%	-1,5000	14.0	46.0
-11,0	-0,02%	626,0	4	+50,00%	-2,7500	14.0	50.0

El informe del ejemplo ofrece 6 estadísticas para cada una de las combinaciones de variables estudiadas (en este caso: n y m). Veamos las estadísticas con más detalle:

- **Beneficios netos**, designa la plusvalía obtenida con la operativa realizada. Matemáticamente, se traduce en esta fórmula:

$$\text{Beneficios netos} = \text{Valor del capital tras operativa realizada} - \text{Valor inicial del capital}$$

Esta estadística permite evaluar de manera absoluta el potencial de ganancias con el sistema definido para cada valor de variables correspondientes.

Nota: las comisiones de operativa definidas en la sección 'Money management' se toman en cuenta en este cálculo.

- **Retorno de capital** es el beneficio expresado en %. Matemáticamente, se traduce por la fórmula

$$\text{Retorno de capital} = (100 \times \text{Beneficios netos}) / \text{Valor inicial del capital}$$

Indica el resultado relativo de este sistema, configurado con los valores respectivos de las variables.

- **Max drawdown** indica la pérdida máxima incurrida al aplicar el sistema. Dicho de otro modo, es la diferencia entre el punto máximo y el mínimo en la curva de pérdidas y ganancias.

Veamos un ejemplo de esta estadística:



El máx. drawdown es determinante dentro de su perfil de riesgo: si no desea exponerse a pérdidas importantes en una misma posición (incluso si el sistema en su conjunto pueda dar resultados muy positivos), deberá optar por una estrategia menos arriesgada.

- **Órdenes** indica la cantidad de órdenes enviadas desde el lanzamiento de este sistema.
- **% Operaciones ganadoras** indica el porcentaje de operaciones ganadoras dentro del sistema, y constituye una información complementaria a Max drawdown.

Matemáticamente se define así:

$$\% \text{ Operaciones ganadoras} = \frac{100 \times \text{Núm. de operaciones ganadoras}}{\text{Núm. total de operaciones}}$$

- **Esperanza** es la media de ganancias por transacción y puede ayudar a determinar la eficacia media de las órdenes enviadas. La esperanza es particularmente importante cuando no deseamos enviar un gran número de órdenes; en tal caso, la estadística mencionada se convierte en un criterio determinante para decidir si el sistema se aplica o no.

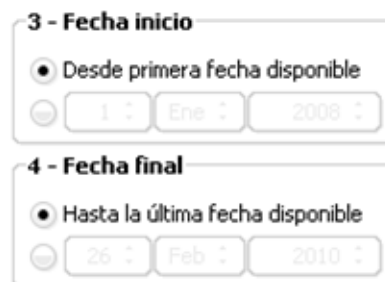
Matemáticamente, se define así:

$$\text{Esperanza} = \text{Total neto} / \text{Número de transacciones}$$

Nota: los valores óptimos de las variables de un sistema pueden cambiar para un mismo valor en función de las unidades de tiempo o el período histórico utilizados.

La **cuarta** sección sirve para definir el intervalo de tiempo sobre el que desee aplicar el sistema. Preste atención a la fecha final: en ella se cierran todas las posiciones que estén abiertas.

Esta función de ProBacktest™ se configura por defecto para probar el sistema a lo largo de la totalidad del histórico visible (las posiciones abiertas sólo se cierran cuando se da la condición de salida).



3 - Fecha inicio

Desde primera fecha disponible

1 : Ene : 2008 :

4 - Fecha final

Hasta la última fecha disponible

26 : Feb : 2010 :

2.C) Resultados de ProBacktest™

Además del informe de optimización ya presentado, los resultados de un ProBacktest se ofrecen bajo 3 aspectos complementarios:

2.C.1) Gráfico liquidez

También llamada '**Equity Curve**', representa la evolución del capital invertido (cuyo montante inicial se define en la gestión de capital o money management) bajo el supuesto de que el sistema se haya aplicado desde el inicio del histórico visible en el valor del gráfico. El gráfico de liquidez contiene dos informaciones complementarias: el relleno y el color de la curva.

- El **relleno del gráfico de liquidez** será de color verde si el resultado global es positivo (el valor actual del capital es superior al inicial) y de color rojo si el resultado global es negativo.

- La **línea del gráfico de liquidez** será de color verde en caso de una variación positiva respecto al nivel precedente, y de color rojo para indicar una variación a la baja.



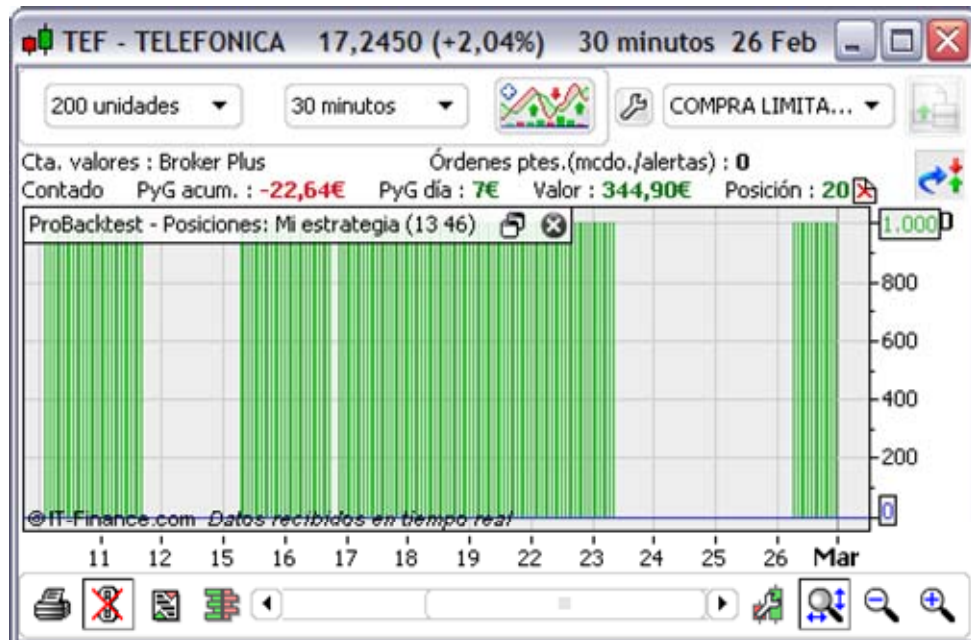
2.C.2) Posiciones

Esta información muestra un histograma con la evolución de las posiciones abiertas a raíz del sistema.

- Una vela verde indica la apertura de una posición a largo (compra)
- Una vela roja indica la apertura de una posición a corto (short selling)
- Si no hay ninguna vela visible, no hay ninguna posición abierta en el mercado.

La aparición de varias velas sucesivas de un mismo color indica que la(s) posición(es) son conservadas.

A lo largo del eje vertical visible en la zona derecha del gráfico se indica la cantidad de posiciones abiertas y acumuladas. En la ilustración siguiente constataremos que nos encontramos en una posición de compra de 1.000 acciones.



2.C.3) Informe detallado

El informe detallado le ofrece una tabla con las principales estadísticas de su sistema: resultados, duración y listado de las posiciones abiertas. El informe detallado se presenta en una ventana independiente de tres pestañas:

- En la pestaña **Estadísticas** hallará un presentación exhaustiva de los resultados del sistema (pérdidas y ganancias, número de operaciones ganadoras,...). Además de las informaciones más conocidas, señalaremos la relevancia de la 'Mayor ganancia' y 'Mayor pérdida', así como el DrawDown (máxima pérdida sucesiva). El análisis de estos dos valores le permitirá evaluar si la estrategia analizada es compatible con su propio perfil de riesgo.

Informe detallado: Mi estrategia

Estadísticas | **Lista de órdenes** | Seguimiento posiciones

	Todas las operaciones	Operaciones de compra	Operaciones de venta a corto
Beneficio neto	109,00	109,00	0,00
Beneficio bruto	336,00	336,00	0,00
Pérdidas brutas	-227,00	-227,00	0,00
Beneficio bruto / Pérdida bruta	1,48	1,48	n/d
Número de operaciones	4	4	0
% operaciones ganadoras	50,00%	50,00%	n/d
Operaciones con ganancias	2	2	0
Operaciones con pérdidas	2	2	0
Operaciones neutrales	0	0	0
Beneficio neto medio por operación	27,25	27,25	n/d
Beneficio medio en operaciones ganadoras	168,00	168,00	n/d
Pérdida media en operaciones perdedoras	-113,50	-113,50	n/d
Desviación estándar en pérdidas y ganancias	149,70	149,70	n/d
Beneficio máximo	239,00	239,00	0,00
Pérdida máxima	-126,00	-126,00	0,00
Tiempo medio en mercado (num barras)	37,25	37,25	n/d
Tiempo medio entre operaciones (num barras)	70,00	70,00	n/d
Tiempo medio entre oper. ganadoras (num barras)	46,50	46,50	n/d
Tiempo medio entre oper. perdedoras (num barras)	28,00	28,00	n/d
Tiempo medio entre oper. neutrales (num barras)	n/d	n/d	n/d
Porcentaje de tiempo en el mercado	41,67%	41,67%	41,67%
Comision total	56,00	56,00	0,00
Num. máximo de operaciones ganadoras consecutivas	0	0	0
Num. máximo de operaciones perdedoras consecutivas	2	2	0
Draw down (pérdida máxima en curva de liquidez)	516,00	516,00	0,00
Ganancia máxima consecutiva en efectivo	654,00	654,00	0,00
Retorno sobre capital inicial (Pérdidas y ganancias/C.inicial)	0,18%	0,18%	0,00%

Modificar ProBacktest | Cerrar |

- La pestaña '**Lista de órdenes**' le indica la hora, el sentido, la cantidad y el precio de las órdenes enviadas. Estas órdenes aparecen en el huso horario del mercado (es decir, expresadas en hora local).

Informe detallado: Mi estrategia

Estadísticas | **Lista de órdenes** | Seguimiento posiciones

Fecha	Compra/Venta	Precio	Cant	Valor actual	Comisión
26-feb-2010 11:30	Compra (Entrar)	17,1400	1000	17.140,0000	8,0000
23-feb-2010 12:30	Venta (Salir)	17,1450	1000	17.145,0000	8,0000
16-feb-2010 17:00	Compra (Entrar)	16,8900	1000	16.890,0000	8,0000
16-feb-2010 16:00	Venta (Salir)	16,8050	1000	16.805,0000	8,0000
15-feb-2010 11:30	Compra (Entrar)	16,8900	1000	16.890,0000	8,0000
11-feb-2010 15:30	Venta (Salir)	16,7000	1000	16.700,0000	8,0000
10-feb-2010 10:00	Compra (Entrar)	16,8100	1000	16.810,0000	8,0000

Modificar ProBacktest | Cerrar |

- Por último, la pestaña '**Seguimiento posiciones**' le da información sobre las posiciones (a largo o a corto, duración indicada en cantidad de velas, resultados absolutos y relativos de cada posición, fecha de entrada y de salida...).

Fecha entrada	Fecha salida	Tipo	Número barras	Res absoluto	Res relativo	Comisión
26-feb-2010 11:30	26-feb-2010 17:30	Comprar	12	97.0	0.56566	8.0
16-feb-2010 17:00	23-feb-2010 12:30	Comprar	81	239.0	1.4143705	16.0
15-feb-2010 11:30	16-feb-2010 16:00	Comprar	27	-101.0	-0.59770346	16.0
10-feb-2010 10:00	11-feb-2010 15:30	Comprar	29	-126.0	-0.7491946	16.0

3. La programación de ProBacktest™

3.A) Entrada y salida del mercado

Vamos a ilustrar dos categorías de instrucciones que permiten entrar y salir del mercado: las instrucciones de apertura de posición y de cierre de posición (stop).

3.A.1) Instrucciones de apertura de posición

Las instrucciones se diferencian en función del sentido de la posición:

- Posiciones largas
 - **BUY** instrucción de compra de valores (entrada a largo)
 - **SELL** instrucción de venta de valores adquiridos (salida de largo)
- Posiciones cortas
 - **SELLSHORT** instrucción de venta a crédito de valores (entrada a corto)
 - **EXITSHORT** instrucción de compra de valores vendidos a crédito (salida de corto)

ProBacktest™ no permite actualmente la simulación de 'hedging' (apertura simultánea de una posición y de otra en sentido inverso sobre un mismo valor). En la práctica, resulta posible cerrar una posición **BUY** con una **SELLSHORT**. No

obstante, se aconseja respetar los binomios descritos para cerrar una posición, utilizando la correspondiente a cada apertura.

Todos estos comandos pueden acompañarse de los elementos que describimos a continuación:

SELLSHORT 'Cantidad' 'Modo' AT 'Tipo' 'Fecha/hora de ejecución'

A) Cantidad

Consiste en el número de valores (volumen) que desea comprar o vender.

Atención: es posible no indicar ningún número. En tal caso, el programa considera por defecto la operativa con una acción o un lote en el caso de futuros.

B) Modo

Es posible definir el modo de inversión en términos absolutos o relativos.

- **SHARES** acciones, futuros y warrants
- **CASH** transacción en unidades monetarias (€ o \$)
- **% CAPITAL** transacción en porcentaje de capital (ex : 10% del capital)
- **% LIQUIDITY** transacción en porcentaje de liquidez restante

Ejemplo:

Cuando el RSI se halla en zona de sobreventa ($RSI < 30$) y el precio está situado por debajo de la banda de Bollinger inferior, se genera una orden de compra a precio de mercado por un 10% del capital.

Cuando el RSI se halla en zona de sobrecompra ($RSI > 70$) y el precio está situado por debajo de la banda de Bollinger inferior, se genera una orden de venta en idénticas condiciones.

```
IF RSI [14](Close) < 30 AND Close < BollingerDown[25](Close)
THEN
    BUY 10 %CAPITAL AT MARKET
ENDIF
IF RSI [14](Close) > 70 AND Close > BollingerUp[25](Close) THEN
    SELL 10 %CAPITAL AT MARKET
ENDIF
```

C) Tipo

Dispone de tres tipos de ejecución de órdenes:

- **AT MARKET** : La orden se envía a precio de mercado
- **AT (price) LIMIT** : Se ejecutará tras tocar el límite indicado y a precio que no lo sobrepase
- **AT (price) STOP** : La orden se ejecutará a precio de mercado tras tocar el stop indicado

Ejemplo: Volatility Breakout

Este sistema se basa en la volatilidad. Se posiciona una orden de compra y una de venta a crédito en cada vela, ambas a precio 'limite'.

La orden de compra se sitúa en el precio de cierre de la vela previa más un 50% del rango de la vela previa.

La orden de venta a crédito se sitúa en el precio de cierre de la vela previa menos un 50% del rango de la vela previa.

REM Volatility Breakout

BuyLimit = **Close[1]** + (**Range[1]** * **50 / 100**)

SellLimit = **Close[1]** - (**Range[1]** * **50 / 100**)

BUY 1 SHARES AT BuyLimit Stop

SELLSHORT 1 SHARES AT SellLimit Stop

D) Fecha/hora de ejecución

Por defecto, cada orden se ejecuta al cierre de la vela actual. Por lo tanto, en la vela **siguiente**.

Sin embargo, es posible modificar la fecha de ejecución en el caso de órdenes enviadas a precio de mercado a través de los comandos descritos a continuación (atención a su uso sin paréntesis ni corchetes):

- **NextBarOpen**: crea la orden en la apertura de la vela siguiente (configuración por defecto)
- **NextBarClose**: crea la orden en el cierre de la vela siguiente
- **ThisBarOnClose**: crea la orden en el cierre de la vela actual

- **TodayOnClose:** crea la orden en el cierre del día actual (relevante sólo si se emplea en intradía)
- **TomorrowOpen:** crea la orden en la apertura del día siguiente (relevante sólo si se emplea en intradía)
- **TomorrowClose:** crea la orden en el cierre del día siguiente (relevante sólo si se emplea en intradía)
- **RealTime:** crea la orden en tiempo real, en el tick actual

Las instrucciones de tipo 'Fecha/hora de ejecución' sólo pueden emplearse cuando son precedidas de la instrucción 'AT MARKET'.

Ejemplo: Rupturas de canal

El soporte y resistencia de un canal se refieren a los valores mínimo y máximo de las dos últimas velas. Si antes de las 16h el precio rompe al alza la resistencia del canal, se genera una orden de compra por un 70% del capital. La posición se cierra cuando el precio rompe a la baja el soporte del canal, siempre y cuando ello ocurra antes de las 16h.

REM Cierre de la segunda vela (indice 1)

IF IntradayBarIndex = 1 THEN

Resistencia = Highest[2](High)

Soporte = Lowest[2](Low)

ENDIF

REM Compra / Venta tras ruptura, si tiene lugar antes de las 16:00:00 (hora local)

IF IntradayBarIndex > 1 AND Time < 160000 THEN

REM Ruptura de resistencia

IF Close > Resistencia THEN

BUY 70 %CAPITAL AT MARKET THISBARONCLOSE

ENDIF

REM Ruptura de soporte

IF Close < Soporte THEN

**SELLSHORT 70 %CAPITAL AT MARKET
THISBARONCLOSE**

ENDIF

ENDIF

3.A.2) Instrucciones de STOP

Los Stops de un sistema pueden configurarse manualmente.

Además de los 4 tipos de stops disponibles en la sección 2 de la ventana de programación (ver sección 'Money Management'), también puede utilizar stops programados por usted mismo mediante el comando

SET STOP (precio)

En el que la constante 'precio' indica el nivel en el que deseamos cerrar nuestra posición.

3.A.3) Estrategia de cruces precio/SAR

El siguiente sistema lanza una orden de compra / venta a crédito en cuanto el precio cruza al alza / baja el SAR.

Además, se posiciona un stop dinámico de protección para cerrar la posición en caso de que el precio se aproxime a un nivel demasiado bajo (en el código, dicho stop se llama 'cut').

```
Indicator1 = Close
Indicator2 = SAR[0.02,0.02,0.2]
REM Compra
c1 = (Indicator1 Crosses Over Indicator2)
IF c1 THEN
    BUY 1 SHARES AT MARKET
ENDIF
REM Venta
c2 = (Indicator1 Crosses Under Indicator2)
IF c2 THEN
    SELLSHORT 1 SHARES AT MARKET
ENDIF
REM Posicionamiento de un Stop dinámico
IF Lowest[5](Close) < (1.2 * Low) THEN
    IF Lowest[5](Close) >= Close THEN
        Cut = Lowest[5](Close)
    ELSE
        Cut = Lowest[20](Close)
    ENDIF
ENDIF
SET STOP Cut
```

Atención al uso correcto de órdenes **STOP**:

- **AT (precio) STOP** define un nivel que lanza la orden en cuanto el precio lo toca.
- **SET STOP (precio)** define un stop de protección, que cierra una posición existente.

3.B) Seguimiento de posiciones

3.B.1) Instrucciones de verificación del estado de posiciones

ProBacktest™ permite subordinar la apertura de una posición a la existencia previa de otra posición en el mismo valor, tanto a largo como en venta a crédito. El control de las posiciones existentes se realiza directamente en el sistema, y el programa lo interpreta como una condición necesaria para que se abra una posición.

Los comandos que se utilizan para la gestión de estos casos son:

- **ONMARKET**: comprueba si hay posiciones abiertas
- **LONGONMARKET**: comprueba si hay posiciones abiertas a largo
- **SHORTONMARKET**: comprueba si hay posiciones cortas abiertas

Se utilizan sin necesidad de paréntesis ni corchetes, y suelen incluirse habitualmente en las secciones condicionales **IF**.

Las instrucciones de verificación del estado de posiciones son especialmente útiles en caso de acumulación (piramidación) de posiciones. Recordemos nuevamente que la acumulación de posiciones inversas sobre un mismo valor no es posible.

Veamos un ejemplo de uso de estos comandos:

Sistema en MACD (empleo de LONGONMARKET y de SHORTONMARKET):

El sistema que se muestra a continuación se basa en las inversiones del MACD respecto al nivel 0, y abrirá un cierto número de posiciones que se irán cerrando progresivamente. Esta evolución gradual presenta la ventaja de liquidar las ganancias y de limitar los riesgos.

REM Definición del MACD

Indicator1 = MACD[12,26,9](Close)

REM Observación de los cambios de signo del Histograma del MACD

c1 = (Indicator1 Crosses Over 0.0)

REM Compra: Si no hay una posición larga abierta y si $MACD > 0$,
compramos 3 títulos

IF NOT LONGONMARKET AND c1 THEN

BUY 3 SHARES AT MARKET ThisBarOnClose

Long = 0

Entry = Close

ENDIF

REM Cierre posición larga: los 3 valores se venden sucesivamente a
precios límite de 7%, 15% y 25%

REM de beneficio si la evolución del precio lo permite

REM Las posiciones abiertas restantes se cierran cuando el MACD
cruza el nivel 0

IF LONGONMARKET AND Long = 0 AND Close > (Entry * 1.07) THEN

SELL 1 SHARES AT MARKET ThisBarOnClose

Long = 1

ELSIF LONGONMARKET AND Long = 1 AND Close > (Entry * 1.15)
THEN

SELL 1 SHARES AT MARKET ThisBarOnClose

Long = 2

ELSIF LONGONMARKET AND Long = 2 AND Close > (Entry * 1.25)
THEN

SELL 1 SHARES AT MARKET ThisBarOnClose

Long = 3

ENDIF

REM: Si no hay una posición corta (venta a crédito) abierta y si
 $MACD < 0$,

REM: vendemos a crédito 3 valores

IF NOT c1 AND NOT SHORTONMARKET THEN

SELLSHORT 3 SHARES AT MARKET ThisBarOnClose

Short = 0

Entry = Close

ENDIF

REM Cierre posición a crédito: se compraran sucesivamente los 3 títulos a precios limite de

REM 7%, 15% y 25% de beneficio si la evolución del precio lo permite

REM Las posiciones abiertas restantes se cierran cuando el MACD cruza el nivel 0

```
IF SHORTONMARKET AND Short = 0 AND Close < (Entry * 0.93) THEN
```

```
    EXITSHORT 1 SHARES AT MARKET ThisBarOnClose
```

```
    Short = 1
```

```
ELSIF SHORTONMARKET AND Short = 1 AND Close < (Entry * 0.85) THEN
```

```
    EXITSHORT 1 SHARES AT MARKET ThisBarOnClose
```

```
    Short = 2
```

```
ELSIF SHORTONMARKET AND Short = 2 AND Close < (Entry * 0.75) THEN
```

```
    EXITSHORT 1 SHARES AT MARKET ThisBarOnClose
```

```
    Short = 3
```

```
ENDIF
```

3.B.2) Contadores de posiciones

Para crear sistemas que tomen en cuenta la cantidad de órdenes pasadas, ProBacktest incluye comandos que cuentan las posiciones:

- **COUNTOFPOSITION:** cantidad de posiciones abiertas desde el inicio del backtest
- **COUNTOFLONGSHARES:** núm. posiciones abiertas al alza (largas) desde el inicio del backtest
- **COUNTOFSHORTSHARES:** núm. posiciones abiertas a la baja (cortas) desde el inicio del backtest

Análogamente a las instrucciones de verificación del estado de posiciones, estos comandos se utilizan generalmente dentro de secciones condicionales.

Veamos a continuación un ejemplo que combina COUNTOFLONGSHARES y COUNTOFSHORTSHARES: Inverse Fisher Transform aplicado al RSI.

Este backtest se basa en el indicador 'Inverse Fisher Transform RSI' para dar órdenes de compra o de venta.

El sistema genera una orden de compra cuando el Inverse Fisher Transform RSI cruza al alza el nivel 50 y vende cuando el indicador cruza a la baja el nivel 80.

Inversamente, genera una orden de venta a crédito si el Inverse Fisher Transform RSI cruza a la baja el nivel 50 y la cierra (compra) cuando dicho indicador cruza al alza el nivel 20.

El backtest debe utilizarse con velas de 1h en el caso de futuros. En cambio, es aconsejable utilizarlo con velas diarias en el caso de acciones.

REM Inverse Fisher Transform aplicado al RSI

REM Parametros : n = Cantidad de velas a considerar en el cálculo del RSI con un paso de 1

n = 10

Ind=RSI[n](Close)

x = 0.1 *(Ind -50)

y = (EXP(2 * x)-1) / (EXP(2 * x)+ 1)

z = 50 * (y + 1)

miInverseFisherTransformsRSI = z[7]

IF (miInverseFisherTransformsRSI Crosses Over 50) THEN

BUY 1 SHARES AT MARKET

ENDIF

IF (miInverseFisherTransformsRSI Crosses Under 80) THEN

SELL COUNTOFLONGSHARES SHARES AT MARKET

ENDIF

IF (miInverseFisherTransformsRSI Crosses Under 50) THEN

SELLSHORT 1 SHARES AT MARKET ENDIF

IF (miInverseFisherTransformsRSI Crosses Over 20) THEN

EXITSHORT COUNTOFSHORTSHARES SHARES AT MARKET

ENDIF

3.B.3) ENTRYINDEX

El comando **ENTRYINDEX**[.] da acceso a la vela de la enésima transacción previa. Presenta características similares a las de las funciones `BarIndex` e `IntradayBarIndex` (presentadas en la *Guía de uso ProBuilder™*):

- La búsqueda se realiza analizando el gráfico de izquierda a derecha.
- La primera vela mostrada a la izquierda se entiende como vela 0 (p.ej., `ENTRYINDEX=3` sería la cuarta vela tras el inicio del histórico).

La sintaxis del comando es similar a la de una constante:

ENTRYINDEX[enésima orden previa]

Nota: ENTRYINDEX puede emplearse sin asociación a un número de vela definido entre corchetes. En tal caso, el programa tomará en consideración la vela de la última orden generada.

Ejemplo: Find Inside Bar

Este ejemplo es un sistema basado en un popular patrón de precios conocido como 'Inside Bar'. El sistema

- Evalúa si el rango de la tercera vela (contando la vela actual –considerada como vela 0) es superior al rango de la vela siguiente, y si ésta última es blanca (o verde). En tal caso, se abre una posición a largo (se genera una orden de compra)
- Evalúa si el rango de la tercera vela (contando la vela actual –considerada como vela 0) es inferior al rango de la vela siguiente, y si ésta última es negra (o roja). En tal caso, se abre una posición a corto (se genera una orden de venta a crédito).

La posición se cierra sistemáticamente 3 velas después de haber sido abierta.

```
Condicion1 = (High[2] >= High[1] AND Low[2] <= Low[1])
Condicion2 = (High[2] <= High[1] AND Low[2] <= Low[1])
Condicion3 = (Close[1] > Open[1])
Condicion4 = (Close[1] < Open[1])
IF (Condicion1 AND Condicion3) THEN
    BUY 10 %CAPITAL AT MARKET NextBarOpen
ENDIF
IF LONGONMARKET AND (BarIndex -ENTRYINDEX) = 3 THEN
    SELL 10 %CAPITAL AT MARKET ThisBarOnClose
ENDIF
IF (Condicion2 AND Condicion4) THEN
    SELLSHORT 10 %CAPITAL AT MARKET NextBarOpen
ENDIF
IF SHORTONMARKET AND (BarIndex -ENTRYINDEX) = 3 THEN
    EXITSHORT AT MARKET ThisBarOnClose
ENDIF
```

3.B.4) ENTRYQUOTE

El comando **ENTRYQUOTE[.]** llama al precio al que una orden se ha ejecutado. Es particularmente relevante cuando la duración entre cada operación es corta; es decir, en sistemas intraday. Su sintaxis es:

```
ENTRYQUOTE[enésima orden previa]
```

Como para todas las constantes, podemos indicar entre corchetes la orden a la que nos referimos. Si no se indica nada tras el comando ENTRYQUOTE, el programa tomará en consideración la vela de la última orden generada.

Ejemplo: Creación de un target (objetivo de beneficios o take-profit)

Definimos dos condiciones:

- Ausencia de posiciones abiertas
- RSI débil (<30)

La compra tendrá lugar cuando el precio sea superior a la media móvil de período 10.

La posición se cerrará cuando el precio en tiempo real supere (hemos usado una orden límite) al precio de compra en un 15%.

```
IF NOT ONMARKET AND RSI < 30 THEN
  IF Close > AVERAGE[10](Close) THEN
    BUY 100 %CAPITAL AT MARKET
  ENDIF
ENDIF
SELL 100 %CAPITAL AT ENTRYQUOTE * 1.15 LIMIT
```

3.B.5) PreviousTrade

Algunos traders consideran los resultados anteriores para construir sus sistemas. El comando **PreviousTrade(.)** sirve precisamente para llamar a estos resultados.

Representa los resultados en % de la enésima operación realizada. Su sintaxis de uso es análoga a la de una constante diaria de precio:

PreviousTrade(enésima operación previa)

PreviousTrade(1) = Nivel de la última operación realizada. ¡Cuidado con los paréntesis, son obligatorios!

Ejemplo:

Veamos un ejemplo basado en el cruce de las líneas del estocástico y del indicador RSI. Comenzamos por enviar una orden de compra a precio de mercado cuando las medias móviles exponenciales se cruzan; a continuación, se abrirá una posición

- De compra (acumulando posiciones) si:
 - La primera orden enviada está en resultado positivo
 - RSI es <30
- De venta si:
 - RSI es >70
 - La línea K cruza a la baja a la línea D

```
ONCE StochPeriod = 14
ONCE KPeriod = 3
ONCE DPeriod = 3
LineK = Stochastic[StochPeriod, KPeriod](Close)
LineD = Average[DPeriod](LineK)
// Envío de la primera orden a precio de mercado
IF ExponentialAverage[12](Close) Crosses Over
ExponentialAverage[20](Close) THEN
    BUY AT MARKET
ENDIF
//Envío de la segunda orden
IF LineK Crosses Over LineD THEN
    IF RSI < 30 THEN
        REM Compra si la orden anterior se encuentra en
        beneficios latentes
        IF PreviousTrade(1) > 0 THEN
            BUY 100 %LIQUIDITY AT MARKET
        ENDIF
    ENDIF
ENDIF
ENDIF
IF RSI > 70 AND LineK Crosses Under LineD THEN
    SELL 100 %CAPITAL AT MARKET
ENDIF
```

4. Aplicaciones prácticas

4.A) Sistemas sobre indicadores

4.A.1) Sistema basado en los Heikin Ashi

Este sistema genera una señal de compra cuando en estilo Heikin-Ashi aparece una vela roja seguida de una vela verde.

Inversamente, se genera una señal de venta a crédito si aparece una vela verde seguida de una roja.

El interés de este backtest se relaciona con el método de construcción de la vista Heikin Ashi, en la que no es posible aplicar sistemas. Por ello, este ProBacktest deberá aplicarse imperativamente en un gráfico con el precio en estilo velas (candlesticks).

```
ONCE PreviousStatus = 0  
IF BarIndex = 0 THEN  
    XClose = TotalPrice  
    XOpen = (Open + Close) / 2  
ELSE  
    XClose = TotalPrice  
    XOpen = (XOpen[1] + Xclose[1]) / 2  
ENDIF  
IF XClose >= XOpen THEN  
    IF PreviousStatus = -1 THEN  
        BUY 1 SHARES AT MARKET  
    ELSE  
        PreviousStatus = 1  
        IF PreviousStatus = 1 THEN  
            SELLSHORT 1 SHARES AT MARKET  
            PreviousStatus = -1  
        ENDIF  
    ENDIF  
ENDIF  
ENDIF
```

4.A.2) Sistema basado en el Zig Zag

Este backtest utiliza el indicador Zig Zag para identificar cuales habrían sido las mejores oportunidades e compra y venta. Los excelentes resultados (en mercados de acciones y futuros) se deben al carácter no predictivo del Zig Zag, que se recalcula a posteriori y no proporciona sistemáticamente señales válidas en tiempo real. A pesar de ello, este sistema interesa en tanto que permite comparar sus resultados con los de otros sistemas.

```
// En variable optimizada: periodos = 1 (de 5 à 10)
c11 = (myZigZag > myZigZag[1])
c12 = (myZigZag < myZigZag[1])
IF c11 AND (SHORTONMARKET OR NOT LONGONMARKET) THEN
    EXITSHORT COUNTOFSHORTSHARES SHARES AT MARKET
    BUY 50 %CAPITAL AT MARKET
ENDIF
IF c12 AND (LONGONMARKET OR NOT SHORTONMARKET) THEN
    SELL COUNTOFLONGSHARES SHARES AT MARKET
    SELLSHORT 50 %CAPITAL AT MARKET
ENDIF
```


4.A.3) Sistema Breakout Range con Stop Dinámico

Se trata de un sistema de tipo BreakOut: las señales se generan a partir de rupturas de máximos o de mínimos calculadas a lo largo de un período determinado. Este sistema abre únicamente posiciones largas que protege a continuación con un stop dinámico. El período sobre el cual se calcula el máximo debe determinarse como una variable optimizada.

REM periodo = variable optimizada (de 2 a 20 en pasos de 1)

ONCE MMentry = 5

ONCE Period = 14

REM Entrada a largo:

Condition = High > Highest[Period](High)[1]

IF Condition AND Summation[Period](Condition) = 1 THEN

BUY 1 SHARES AT MARKET

ENDIF

c2 = Lowest[10](Low[1])

**StopLoss = Highest[MMentry](High)[BarIndex -ENTRYINDEX + 1] /
Average[20](High / Low)**

SET STOP MAX(StopLoss,(c2 -0.01))

4.A.4) Sistema basado en el Estocástico Alisado

Este sistema reposa sobre el indicador 'Estocástico alisado' aplicado al precio mediano y su media móvil.

La señal de compra aparece cuando el indicador es superior a su media móvil exponencial. Se define un objetivo de tipo Límite, superior en un 10% al precio de compra. La señal de venta se da cuando el indicador desciende nuevamente por debajo de su media móvil exponencial.

REM Compra**Indicator1 = SmoothedStochastic[9,9](MedianPrice)****Indicator2 = ExponentialAverage[9](Indicator1)****// Inicio variable****StopLimit = 10 c1 = (Indicator1 >= Indicator2)****IF c1 THEN****BUY 1 SHARES AT MARKET RealTime****ENDIF****IF LONGONMARKET THEN****SELL AT ENTRYQUOTE * (1 + StopLimit / 100) Limit****ENDIF****IF SHORTONMARKET THEN****EXITSHORT AT ENTRYQUOTE / (1 + StopLimit / 100) Limit****ENDIF****REM Venta****IF NOT c1 THEN****SELL AT MARKET RealTime****ENDIF****REM Venta a crédito****IF NOT c1 THEN****SELLSHORT 1 SHARES AT MARKET RealTime****ENDIF****REM Recompra****IF c1 THEN****EXITSHORT AT MARKET RealTime****ENDIF**

4.A.5) Swing Trading, ADX y Medias móviles

Este backtest se basa en el indicador ADX y en su posicionamiento respecto al valor 30, con el objetivo de reducir las señales falsas y minimizar los riesgos.

Se trata de un sistema que presenta un gran número de condiciones, lo que restringe el número de señales.

```
MyADX12 = ADX[12]
ADXperiods = 5
MyMM20 = Average[20](Close)
IsLow30 = 0
FOR Count = 0 TO ADXperiods DO
    IF MyADX12[Count] < 30.0 THEN
        IsLow30 = 1
        BREAK
    ENDIF
NEXT
// Compra
// ADX 12 está por encima de 30 desde al menos entre 5 y 10 sesiones
Condition1 = NOT IsLow30
// Si los precios vuelven a apoyarse en la MME 20 días durante 1 a 4
sesiones consecutivas
Condition2 = High > MyMM20 AND Low < MyMM20 AND High[1] <
MyMM20[1] AND Low[1] < MyMM20[1]
// Si el máximo del día sobrepasa el máximo de la víspera
Condition3 = Dhigh(0) > Dhigh(1)
IF Condition1 AND Condition2 AND Condition3 THEN
    BUY 1 SHARES AT MARKET ThisBarOnClose
ENDIF
// Venta a corto
// ADX 12 está por encima de 30 desde al menos entre 5 y 10 sesiones
Condition4 = NOT IsLow30
// Si los precios vuelven a apoyarse en la MME 20 días durante 1 a 4
sesiones consecutivas
Condition5 = High > MyMM20 AND Low < MyMM20 AND High[1] >
MyMM20[1] AND Low[1] > MyMM20[1]
// Si el mínimo del día sobrepasa el mínimo de la víspera
Condition6 = Dlow(0) < Dlow(1)
IF Condition4 AND Condition5 AND Condition6 THEN
    SELLSHORT 1 SHARES AT MARKET ThisBarOnClose
ENDIF
```

4.B) Sistemas de money management (gestión de capital)

Los resultados de un backtest pueden mejorarse sustancialmente adoptando reglas sofisticadas de gestión de capital (money management).

Estas estrategias de money management suelen formalizarse en las martingalas, destinadas a optimizar la esperanza matemática de un sistema (ganancia o pérdida media por operación). Esto implica poder estimar previamente la probabilidad de que una operación resulte en ganancia, y el importe del beneficio.

Puede ser interesante codificar directamente órdenes stop (de protección), take profit (objetivo) e inactividad, así como subsistemas que permitan la gestión dinámica del tamaño de posiciones.

4.B.1) Stop de protección (stop loss)

El código presentado a continuación integra una orden stop loss de tipo stop limit en su sistema. Recuerde definir las condiciones de su stop; en este caso se les denomina StopLossLong y StopLossShort.

```
ONCE Level = 0.05
```

```
REM Seleccion del nivel de perdida maxima determinante del cierre de posicion (0.05 equivale a 5%)
```

```
REM Si la posicion es a largo, el cierre se da cuando el precio varia de -nivel% (bajada de -nivel%)
```

```
IF LONGONMARKET AND (Close -ENTRYQUOTE) / (ENTRYQUOTE) < Level THEN
```

```
SELL AT MARKET StopLossLong
```

```
ENDIF
```

```
REM Si la posicion es a corto, el cierre se da cuando el precio varia de nivel% (subida de nivel%)
```

```
IF SHORTONMARKET AND (Close -ENTRYQUOTE) / (ENTRYQUOTE) > Level THEN
```

```
EXITSHORT AT MARKET StopLossShort
```

```
ENDIF
```

4.B.2) Objetivo de beneficio (take profit)

Este código integra un objetivo (take profit) de tipo StopLimit en su sistema. Recuerde definir las condiciones de su stop; en este caso se les denomina TakeProfitLong y TakeProfitShort.

```
ONCE Level = 0.05
```

```
REM Seleccion del nivel de ganancia determinante del cierre de  
posicion (0.05 equivale a 5%)
```

```
REM Si la posicion es a largo, el cierre se da cuando el precio varía  
de nivel% (subida de nivel%)
```

```
IF LONGONMARKET AND (Close -ENTRYQUOTE) / (ENTRYQUOTE) >  
Level THEN
```

```
    SELL AT MARKET TakeProfitLong
```

```
ENDIF
```

```
REM Si la posicion es a corto, el cierre se da cuando el precio varía  
de -nivel% (bajada de -nivel%)
```

```
IF SHORTONMARKET AND (Close -ENTRYQUOTE) / (ENTRYQUOTE)  
< Level THEN
```

```
    EXITSHORT AT MARKET TakeProfitShort
```

```
ENDIF
```

4.B.3) Stop de inactividad

Este código permite integrar un stop de inactividad en su sistema. Recuerde definir las condiciones de su stop; en este caso se les denomina InactivityStopLong e InactivityStopShort. En el ejemplo descrito, el stop se activa tras 10 velas.

```
ONCE Count = 10
```

```
REM Seleccionar el numero de velas a contar desde la apertura de  
posicion, tras el cual dicha
```

```
REM posición se cerrará sistemáticamente
```

```
IF ONMARKET AND (BarIndex -ENTRYINDEX + 1) > Count THEN
```

```
    IF LONGONMARKET THEN
```

```
        SELL AT MARKET InactivityStopLong
```

```
    ENDIF
```

```
    IF SHORTONMARKET THEN
```

```
        EXITSHORT AT MARKET InactivityStopShort
```

```
    ENDIF
```

```
ENDIF
```

4.B.4) Piramidación de una posición

Para acumular posiciones es necesario activar la casilla 'Acumular posiciones' en la ventana 'Gestión capital' del backtest. La piramidación consiste en el envío de varias órdenes sucesivas en la misma dirección, con el objetivo de aumentar el tamaño de la posición. El piramidaje se concreta en el momento en que varias órdenes se validan simultáneamente, tal y como se muestra en este sencillo backtest:

REM Compra 1 cuando el RSI es inferior a 30

IF RSI[14](Close) < 30 THEN

BUY 1 SHARES AT MARKET

ENDIF

REM Si se ha abierto una posición larga (compra) y el precio al cierre de la vela anterior esta REM por debajo del precio de apertura de la vela actual, se compra cada vez 1 valor suplementario

IF LONGONMARKET AND Open > Close[1] THEN

BUY 1 SHARES AT MARKET

ENDIF

REM Cuando el precio cruza a la baja una media móvil simple, se cierra la totalidad de la posición

IF Close Crosses Under Average[14](Close) THEN

SELL 100 %CAPITAL AT MARKET

ENDIF

4.B.5) Gestión dinámica del tamaño de órdenes

Para modificar dinámicamente el tamaño de las posiciones sin pasar obligatoriamente por el piramidaje, es posible utilizar una variable que indica la cantidad de partes del activo subyacente que se deseen negociar.

```
ONCE OrderSize = 1
REM Las ordenes son inicialmente sobre 1 parte del activo
subyacente
BUY OrderSize SHARES AT MARKET
REM la posicion se cierra tras 2 velas
IF BarIndex -ENTRYINDEX >= 2 THEN
    SELL AT MARKET
ENDIF
REM Si le RSI es inferior a 30, se aumenta la posicion en cada vela
REM Este tamaño se limita a un máximo de 50
IF RSI[14](Close) < 30 THEN
    OrderSize = MAX(OrderSize,(OrderSize + 1.50))
ENDIF
REM Si el RSI es superior a 70, se disminuye en cada vela el
tamaño de la posicion a abrir
REM Este tamaño no puede ser inferior a 0
IF RSI[14](Close) > 70 THEN
    OrderSize = MIN(OrderSize,(OrderSize -1.0))
ENDIF
```

4.B.6) Consideración de los resultados históricos

Mediante la variable PreviousTrade(i) se puede alterar el comportamiento del sistema presentado a continuación, independientemente de que resulte en pérdidas o en beneficios. Retomando el backtest previo, es posible mejorar su comportamiento reduciendo el tamaño de las posiciones en caso de pérdida (dado que en tal caso no estaría siguiendo al mercado correctamente), o al contrario, aumentándolo cuando las ganancias se repiten.

```
ONCE OrderSize = 1
REM Las ordenes son inicialmente sobre 1 parte del activo subyacente
BUY OrderSize SHARES AT MARKET
REM la posicion se cierra tras 2 velas
IF BarIndex -ENTRYINDEX >= 2 THEN
    SELL AT MARKET
ENDIF
REM Si le RSI es inferior a 30, se aumenta la posicion en cada vela
REM Este tamaño se limita a un máximo de 50
IF RSI[14](Close) < 30 THEN
    OrderSize = MAX(OrderSize,(OrderSize + 1.50))
ENDIF
REM Si el RSI es superior a 70, se disminuye en cada vela el tamaño
de la posicion a abrir
REM Este tamaño no puede ser inferior a 0
IF RSI[14](Close) > 70 THEN
    OrderSize = MIN(OrderSize,(OrderSize -1.0))
ENDIF
REM Modificamos el comportamiento del backtest en función de los
resultados pasados
REM analizando sucesivamente las 3 últimas transacciones.
REM Si una operación da perdidas, el tamaño de la posicion cae en 1.
Si da ganancias, aumenta de 1
FOR i = 1 TO 3 DO
    IF PreviousTrade(i) >= 0 THEN
        OrderSize = MAX(OrderSize,(OrderSize + 1.50))
    ELSIF PreviousTrade(i) < 0 THEN
        OrderSize = MIN(OrderSize,(OrderSize -1.0))
    ENDIF
NEXT
```


Con estas herramientas se pueden integrar varias martingalas en los sistemas de ProBacktest™. He aquí algunos ejemplos de técnicas populares de gestión del capital susceptibles de integrar otras estrategias.

4.B.7) La martingala clásica

La martingala clásica consiste en doblar la posición cuando se afronta una pérdida, con la idea de reembolsar en la siguiente transacción y ganar tras recuperar la posición de salida. El mayor inconveniente de esta gestión del capital es que si se producen varias pérdidas consecutivas, la posibilidad de duplicar la posición se hace cada vez menos factible. Así, partiendo de un capital de p.ej. 1000€, cinco pérdidas sucesivas requerirían 1000×25 euros = 32000€ para continuar con el sistema.

Los sistemas derivados de esta martingala pueden ser más adaptables a la operativa con acciones que con futuros, ya que la inversión inicial y el apalancamiento suele ser más importante en estos últimos mercados. Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del
sistema*****//

ONCE OrderSize = 1

REM Inicio con una posicion de 1
//*****//

//*****Código a insertar tras las instrucciones que cierran
una posicion*****//

IF PreviousTrade(1) < 0 THEN
    OrderSize = OrderSize * 2

    REM Si la ultima transacción resulta en perdidas, se dobla el
    tamaño de la posicion

ELSIF PreviousTrade(1) > 0 THEN
    OrderSize = 1

    REM Si la ultima transacción resulta en ganancias, se retorna
    a una posicion de tamaño 1

ENDIF

//*****//

REM En el backtest, el tamaño de la posicion queda definido por la
variable OrderSize
```

4.B.8) La gran martingala

La gran martingala es similar a la clásica, con la única diferencia que además de doblar la posición en cada pérdida, se añade una unidad suplementaria.

Esta martingala es más arriesgada que la clásica en caso de pérdidas sucesivas, pero permite aumentar significativamente las ganancias en caso contrario. Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del
sistema*****//

ONCE OrderSize = 1

REM Inicio con una posición de 1

//*****//

*****Código a insertar tras las instrucciones que cierran
una posición*****//

IF PreviousTrade(1) < 0 THEN

    OrderSize = OrderSize * 2 + 1

// Si la última transacción resulta en pérdidas, se dobla el tamaño
de la posición y se añade una unidad

ELSIF PreviousTrade(1) >= 0 THEN

    OrderSize = 1

// Si la última transacción resulta en ganancias, se retorna a una
posición de tamaño 1

ENDIF

//*****//

REM En el backtest, el tamaño de la posición queda definido por la
variable OrderSize
```

4.B.9) La Piquemouche

La Piquemouche es otra variante de la martingala clásica. En caso de pérdida, se aumenta el tamaño de la posición en 1 si se han dado menos de 3 pérdidas consecutivas. Cuando se acumulan más de 3 pérdidas seguidas, se duplica el tamaño de la posición. La primera ganancia reinicia el tamaño de la posición, fijándolo nuevamente a 1.

Este sistema de gestión de posiciones es menos arriesgado que los dos anteriores, ya que retrasa el aumento del tamaño de la posición hasta pasadas las 3 pérdidas

sucesivas. Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del
sistema*****//

ONCE OrderSize = 1

// Inicio con una posicion de 1

ONCE BadTrades = 0

// Inicialización del contador de transacciones sucesivas con
resultado de perdidas

//*****//

//*****Código a insertar tras las instrucciones que
cierran una posicion*****//

IF PreviousTrade(1) < 0 THEN
    BadTrades = BadTrades + 1
    IF BadTrades < 3 THEN
        // Si la ultima operación resulta en perdida y no se
        superan 3 seguidas,
        // Se aumenta en 1 unidad el tamaño de la próxima
        posicion
        OrderSize = OrderSize + 1
    ELSIF PreviousTrade(1) < 0 AND BadTrades MOD 3 = 0
    THEN
        // Si la última operación resulta en perdida y se
        acumulan mas de 3 seguidas,
        // Se duplica el tamaño de la próxima posición
        OrderSize = OrderSize * 2
    ENDIF
ELSIF PreviousTrade(1) >= 0 THEN
    // Si la última operación resulta en ganancia, la próxima
    posición volverá a tamaño 1
    OrderSize = 1
    BadTrades = 0
ENDIF

//*****//

REM En el backtest, el tamaño de la posicion queda definido por la
variable OrderSize
```

4.B.10) La Whittacker

La Whittacker consiste en configurar el tamaño de la próxima posición igual a la suma del tamaño de las dos posiciones previas en caso de pérdida. Si hay una ganancia, se reinicia a 1 el tamaño de la posición. Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del
sistema*****//
ONCE OrderSize = 1
// Inicio con una posicion de 1
//*****Código a insertar tras las instrucciones que
cierran una posicion*****//
IF PreviousTrade(1) < 0 THEN
    OrderSize = OrderSize + OrderSize[1]
ELSIF PreviousTrade(1) >= 0 THEN
    OrderSize = 1
    // Si la última operación resulta en ganancia, la próxima
    posición volverá a tamaño 1
ENDIF
//*****//
REM En el backtest, el tamaño de la posicion queda definido por la
variable OrderSize
```

4.B.11) La pirámide de Alembert

Concebida por d'Alembert (matemático francés del siglo XVIII), esta martingala aumenta la posición de 1 en caso de pérdida; en caso de ganancia, la disminuye de 1. Esta técnica sólo es pertinente cuando se considere que las ganancias sucesivas disminuyen la probabilidad de que la siguiente operación resulte en ganancia (y recíprocamente, que una pérdida aumente la posibilidad de que la siguiente operación resulte en ganancia). Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del sistema*****//  
ONCE OrderSize = 1  
  
// Inicio con una posición de 1  
//*****Código a insertar tras las instrucciones que cierran una posición*****//  
IF PreviousTrade(1) < 0 THEN  
    OrderSize = OrderSize + 1  
ELSIF PreviousTrade(1) >= 0 THEN  
    OrderSize = MAX(OrderSize,(OrderSize -1.1))  
ENDIF  
  
//*****//  
  
REM En el backtest, el tamaño de la posición queda definido por la variable OrderSize
```

4.B.12) La contra de Alembert

Esta es la estrategia recíproca de la pirámide epónima, ya que se disminuye el tamaño de la posición de 1 en caso de pérdida y se aumenta de 1 en caso de ganancia.

La técnica implica la consideración de que los resultados históricos son representativos de los resultados futuros.

Para aplicarla, se integrará este código en las condiciones de entrada y de salida:

```
//*****Código a insertar al principio del
sistema*****//
ONCE OrderSize = 1
// Inicio con una posicion de 1
//*****//
//*****Código a insertar tras las instrucciones que cierran
una posicion*****//
IF PreviousTrade(1) < 0 THEN
    OrderSize = MAX(OrderSize,(OrderSize -1.1))
ELSIF PreviousTrade(1) >= 0 THEN
    OrderSize = OrderSize + 1
ENDIF
//*****//
```

5. Anexo: Configurar la gestión del capital

La 'gestión de capital' se encuentra disponible en la sección Money Management de la ventana de programación.

Es un elemento clave que puede influir decisivamente sobre el resultado final del sistema; así, una disminución de las comisiones de operativa, o una política más relajada de gestión del riesgo, podría resultar en una sensible variación del resultado neto.

Se compone de 5 secciones configurables:

- Capital
- Comisiones de operativa
- Gestión de riesgos (risk management)
- Gestión de posiciones

- Gestión del redondeo de la cantidad de valores a comprar/vender

Existen reglas específicas aplicables a la sección de comisiones de operativa al trabajar sobre distintos tipos de valores (acciones o futuros). Por ello, trataremos estas comisiones al final del presente anexo.

5.A) Capital

En esta sección, basta con indicar la suma que desea invertir mediante el sistema.

Nota: salvo mención explícita (ver la configuración del límite máximo del importe invertido), el ProBacktest™ dejará de abrir posiciones cuando el capital inicial se haya agotado.

Capital

Capital inicial :

Si su sistema no abre posiciones, pruebe a aumentar el importe del Capital inicial

5.B) Gestión de riesgos (risk management)

La sección permite definir el límite máximo de inversión en el mercado a través de 3 secciones:

Gestión de riesgos

Límite máximo de inversión :	<input type="text" value="100"/>	% Capital ▼
Inversión máxima por operación :	<input type="text" value="100"/>	% Capital ▼
Inversión mínima por operación :	<input type="text" value="0"/>	% Capital ▼

5.B.1) Límite máximo del total invertido

Este campo limita las pérdidas y gestiona el apalancamiento. Empiece seleccionando en el menú desplegable el modo de inversión del capital. Preste atención a la elección de la escala: valor absoluto (en efectivo), % capital o % liquidez.

5.B.2) Montante máximo por transacción y apalancamientos

La inversión máxima por operación le permite limitar la cantidad invertida en el envío de una orden. Funciona de modo similar a la descrita previamente.

Gestione el apalancamiento mediante la combinación de ambas funciones. Veamos un ejemplo:

- Límite máximo de inversión: 500 % Capital
- Inversión máxima por operación: 500 % Capital

A través de esta configuración, cada orden enviada tendrá un apalancamiento de 5.

Inversamente, si define un % de 100, limitará las pérdidas al valor total de su cartera.

5.B.3) Montante mínimo por transacción

La principal utilidad del montante mínimo por transacción es de evitar que las posiciones abiertas sean demasiado pequeñas, previniendo comisiones de operativa desmesuradas respecto al importe invertido.

Ejemplo: la compra de una acción XXX que cotiza a 5€ con comisiones de 5€ por orden, implicaría que las comisiones representan el 100% del importe invertido. Ello implica comenzar la inversión con una brecha relativa de -100%, que se transformaría en -200% si se consideran además las comisiones de una orden de venta al mismo precio.

5.C) Gestión de posiciones

En esta sección tiene un máximo de 3 casillas a seleccionar.

Reinvertir beneficios configura la gestión de las ganancias. Tras el cierre de una posición, por defecto el sistema no aumenta el capital inicial con los beneficios que se hayan podido obtener. En cambio, si selecciona esta opción, los beneficios aumentarán el capital disponible para invertir.

Acumular posiciones define si las posiciones abiertas se deben acumular entre sí en caso de que las condiciones de entrada se cumplan repetidamente, pero sin permitir el cierre de una posición ya abierta. Por defecto se abre una sola posición cada vez, a menos que en el código haya una indicación explícita para actuar de otro modo. En tal caso, la activación de esta opción no influiría sobre los resultados del sistema.

Si ha creado stops predefinidos en la interfaz, puede asociarlos a las posiciones según estos parámetros:

- Un stop para todas las posiciones (aplicable en el caso de acumulación de posiciones)
- Un stop por posición

Esta sección es particularmente interesante si se aplica a un stop dinámico. Tratándose de un stop que sigue la evolución del precio y que se define habitualmente como la distancia en % del último precio, en el caso de acumulación de posiciones el sistema podrá elegir entre aplicar un único stop dinámico (aplicado al precio medio ponderado de las posiciones abiertas) o aplicar un stop dinámico por cada posición abierta.

5.D) Redondear la cantidad de valores a comprar / vender

Esta sección es relativamente simple a rellenar. Permite definir cómo se hará el redondeo de la cantidad de valores a comprar o a vender, y se realiza mediante la selección de una de estas alternativas:

Redondear el num de valores a comprar / vender

Redondear a la unidad entera inferior

Redondear a la unidad entera superior

5.E) Comisiones de operativa en acciones

Para aplicar un ProBacktest™ a instrumentos de tipo acción o similar (warrants, ETFs,...), deberá activarse la pestaña 'Por orden' en la sección 'Comisión'. Tal y como se ilustra en la figura a continuación, se pueden configurar las comisiones de operativa en precio real o en %.

Las comisiones de operativa por orden se aplican en la apertura y cierre de la posición. Una operativa en ambos sentidos (compra/venta a largo o venta a crédito/recompra a corto) se corresponde con dos órdenes -y por lo tanto, con el doble de comisiones por orden.

La casilla 'Comisión por orden' define la cantidad mínima de las comisiones.

Comisión

Por orden Por contrato (futuros)

Comisión por orden : €,\$ % operación

Comisión mínima : €,\$

5.E.1) Comisiones de operativa en futuros

Para configurarlas se debe pulsar en la pestaña 'Por contrato (futuros)' de la sección 'Comisión'. A continuación, podrá definir:

- La comisión por contrato
- El depósito por contrato
- El valor de un punto

A) Comisión por contrato

En los mercados de futuros, la comisión va por contrato.

B) Depósito por contrato

El depósito por contrato es la garantía a depositar para poder realizar la operación.

El depósito por contrato varía según el valor negociado y según el broker. Para conocer el importe preciso, puede consultarlo en la pestaña "Futuros" del bróker de Bankinter.

No obstante, si desea simular un cierto apalancamiento, le ofrecemos la fórmula que le permitirá calcular el depósito por contrato respecto a dicho apalancamiento:

$$\text{Depósito por contrato} = \text{Capital inicial} / \text{Apalancamiento}$$

Supongamos que dispone de un capital inicial de 1000€ y que desea disponer de un apalancamiento de 5. Su depósito deberá ser de: $1000 / 5 = 200$ €.

C) Valor de un punto

En la plataforma, este valor indica el apalancamiento aplicable sobre las ganancias o pérdidas realizadas. Depende del futuro en el que se aplica el backtest, y se expresa en unidades monetarias por unidad de valor del futuro. Se calcula mediante la siguiente fórmula:

$$\text{Valor de un punto} = \text{Valor monetario de un tick} / \text{Tamaño de un tick}$$

Siendo el tick la variación mínima del futuro autorizada por el mercado donde cotiza (también llamado ticksize). Puede consultar dichos ticksize en la pestaña "Futuros" del bróker de Bankinter.

6. Glosario

A

Código	Implementación	Función
Abs	Abs(a)	Función matemática "Valor Absoluto"
AccumDistr	AccumDistr(price)	Designa la Acumulación Distribución clásica
ADX	ADX[N]	Indicador Average Direccional Index
ADXR	ADXR[N]	Indicador Average Direccional Index Rate
AND	a AND b	Operador lógico Y
AroonDown	AroonDown[P]	Designa el Aroon Down
AroonUp	AroonUp[P]	Designa el Aroon Up
Atan	Atan(a)	Función matemática "Arctangente"
AS	RETURN Result AS"ResultName"	Instrucción utilizada para atribuir un nombre a una curva
AT	AT (precio)	Asociación a un precio
Average	Average[N](price)	Media Móvil Aritmética
AverageTrueRange	AverageTrueRange[N](price)	Designa la Media móvil con alisado de Wilder del True Range

B

Código	Implementación	Función
BarIndex	BarIndex	Cuenta el número de velas en el gráfico
BollingerBandWidth	BollingerBandWidth[N](price)	Ancho de banda de Bollinger
BollingerDown	BollingerDown[N](price)	Soporte de la banda de Bollinger
BollingerUp	BollingerUp[N](price)	Resistencia de la banda de Bollinger
BREAK	BREAK (FOR...DO...BREAK....NEXT) o (WHILE...DO...BREAK...WEND)	Instrucción de salida forzada de bucle FOR o WHILE
BUY	BUT (n) SHARES	Instrucción apertura posición a largo (compra)

C

Código	Implementación	Función
CALL	MyResult = CALL MyFunction	Llamada de función del usuario
CCI	CCI[N](price) o CCI[N] aplicado por defecto en TypicalPrice	Commodity Channel Index
ChaikinOsc	ChaikinOsc[Ch1, Ch2](price)	Designa el oscilador de Chaikin
Chandle	Chandle[N](price)	Designa el Chande Momentum Oscillator
ChandeKrollStopUp	ChandeKrollStopUp[Pp, Qq, X]	Stop de protección según Chande y Kroll en posición compradora
ChandeKrollStopDown	ChandeKrollStopDown[Pp, Qq, X]	Stop de protección según Chande y Kroll en posición vendedora
Close	Close[N]	Designa el precio de cierre de la vela actual o de n días previos
COLOURED	RETURN COLOURED(R,G,B)	Result Colorea una curva de un cierto color según la convención RGB
COS	COS(a)	Función Coseno
Crosses Over	a Crosses Over b	Operador booleano que verifica que una curva pase por encima de otra
Crosses Under	a Crosses Under b	Operador booleano que verifica que una curva pase por debajo de otra
CUMSUM	CUMSUM(price)	Sumatorio de un precio desde el inicio del histórico mostrado
CurrentDayOfWeek	CurrentDayOfWeek	Designa el día actual
CurrentHour	CurrentHour	Designa la hora actual
CurrentMinute	CurrentMinute	Designa el minuto actual
CurrentMonth	CurrentMonth	Designa el mes actual
CurrentSecond	CurrentSecond	Designa el segundo actual
CurrentTime	CurrentTime	Designa Hora Minuto actual
CurrentYear	CurrentYear	Designa el año actual
CustomClose	CustomClose[N]	Constante configurable en la ventana de propiedades al mostrar el gráfico (por defecto, cierre)
Cycle	Cycle(price)	Indicador Ciclo (en precio)

D

Código	Implementación	Función
Date	Date[N]	Designa la fecha de cierre de la vela actual
Day	Day[N]	Día de cierre de la vela actual
Days	Days[N]	Contador de días desde 1900
DayOfWeek	DayOfWeek[N]	Designa el día de la semana durante el cual la vela actual se ha cerrado
Dclose	Dclose(N)	Precio de cierre del enésimo día previo al de la vela actual
DEMA	DEMA[N](price)	Doble Media Móvil Exponencial
Dhigh	Dhigh(N)	Precio máximo del enésimo día previo al de la vela actual
DI	DI[N](price)	Designa el Demand Index (Índice de la Demanda)
DI minus	Diminus[N](price)	Designa el DI-
Diplus	Diplus[N](price)	Designa el DI+
Dlow	Dlow(N)	Precio mínimo del enésimo día previo al de la vela actual
DO	Ver FOR et WHILE	Instrucción opcional de los bucles FOR y WHILE para introducir la acción de inicio de bucle
Dopen	Dopen(N)	Precio de apertura del enésimo día previo al de la vela actual
DOWNTO	Ver FOR	Instrucción que se aplica sobre el bucle FOR para ordenar una lectura decreciente
DPO	DPO[N](price)	Designa el Detrended Price Oscillator

E

Código	Implementación	Función
EaseOfMovement	EaseOfMovement[I]	Designa el indicador Ease of Movement
ELSE	Ver IF/THEN/ELSE/ENDIF	Instrucción de llamada de la segunda condición a defecto de la primera salida de IF
ELSEIF	Ver IF/THEN/ELSE/ENDIF	Contracción de ELSE IF (imbricación en un conjunto de instrucciones condicionales de otro conjunto de instrucciones)
EMV	EMV[N]	Designa el indicador Ease of Movement Value
ENDIF	Ver IF/THEN/ELSE/ENDIF	Instrucción a introducir al final del conjunto de instrucciones condicionales
EndPointAverage	EndPointAverage[N](price)	Ultimo punto de Media Móvil
Exp	Exp(a)	Función matemática "Exponencial"
ExponentialAverage	ExponentialAverage[N] (price)	Media Móvil Exponencial

F-G

Código	Implementación	Función
FOR/TO/NEXT	FOR i=a TO b DO a NEXT	Bucle "Para" (Toma todos los valores del inicio al final designado, o vice versa (DOWNTO))
ForceIndex	ForceIndex(price)	Da el indicador Force Index, que determina quién controla el mercado (vendedor, comprador)

H

Código	Implementación	Función
High	High[N]	Designa el precio máximo alcanzado durante el período N
Highest	Highest[N](price)	Designa el máximo de un período sur un horizonte temporal dado
HistoricVolatility	HistoricVolatility[N](price)	Designa la volatilidad histórica (o volatilidad estadística)
Hour	Hour[N]	Designa la hora de cierre de cada vela del histórico

I-J-K

Código	Implementación	Función
IF/THEN/ENDIF	IF a THEN b ENDIF	Conjunto de instrucciones condicionales sin segunda condición
IF/THEN/ELSE/ENDIF	IF a THEN b ELSE c ENDIF	Conjunto de instrucciones condicionales
IntradayBarIndex	IntradayBarIndex[N]	Cuenta el número de velas en el gráfico intraday

L

Código	Implementación	Función
LinearRegression	LinearRegression[N](price)	Regresión lineal
LinearRegressionSlope	LinearRegressionSlope[N]	Pendiente de la regresión lineal (price)
Log (logarithm)	Log(a)	Función matemática "logaritmo neperiano"
Low	Low[N]	Designa el mínimo alcanzado durante el período
Lowest	Lowest[N](price)	Designa el mínimo de un período dentro de un horizonte temporal dado

M

Código	Implementación	Función
MACD	MACD[S,L,Si](price)	Designa al Moving Average Convergence Divergence (MACD) en histograma
MACDline	MACDLine[S,L](price)	Designa la línea del MACD
MassIndex	MassIndex[N]	Indicador Mass Index aplicado en N velas
Max	Max(a,b)	Función matemática "Máximo"
MedianPrice	MedianPrice	Media del precio máximo y del mínimo
Min	Min(a,b)	Función matemática "Mínimo"
Minute	Minute	Designa el minuto del instante del cierre de cada vela del histórico
Mod	a Mod b	Función matemática "Resto del cociente euclídeo"
Momentum	Momentum[I]	Designa el Momentum (precio de cierre actual – precio de cierre de la enésima vela precedente)
MoneyFlow	MoneyFlow[N](price)	Designa el MoneyFlow entre -1 y 1
MoneyFlowIndex	MoneyFlowIndex[N]	Designa el MoneyFlowIndex
Month	Month[N]	Designa el mes de cierre de cada vela del histórico

N

Código	Implementación	Función
NEXT	Ver FOR/TO/NEXT	Instrucción a introducir al final del bucle "Para" (FOR)
NOT	NOT a	Operador lógico NO

O

Código	Implementación	Función
OBV	OBV(price)	Designa el "On-Balance-Volume"
ONCE	ONCE VariableName = VariableValue	Instrucción que precede a otra que deseamos ejecutar una sola vez
Open	Open[N]	Designa el precio de apertura de la vela actual o celle de n días previos
OpenOfNextBar	OpenOfNextBar	Designa el precio de apertura de la vela siguiente
OR	a OR b	Operador lógico O

P-Q

Código	Implementación	Función
PriceOscillator	PriceOscillator[S,L](price)	Indicador Percentage Price oscillator
PositiveVolumeIndex	PriceVolumeIndex(price)	Designa el indicador Positive Volume Index
PVT	PVT(price)	Designa el indicador "Price Volume Trend"

R

Código	Implementación	Función
R2	R2[N](price)	Designa el coeficiente R Cuadrado (error de precios en la regresión lineal)
Range	Range[N]	Devuelve el Range (rango, diferencia entre el precio máximo y mínimo de la vela actual)
REM	REM comentario	Precede un comentario (el código no lo toma en cuenta, pero facilitan la lectura al usuario)
Repulse	Repulse[N](price)	Devuelve el indicador Repulse (mide la fuerza alcista y bajista de cada vela)
RETURN	RETURN	Resultado Instrucción que envía el resultado
ROC	ROC[N](price)	Designa el "Price Rate of Change"
RSI	RSI[N](price)	Designa el oscilador "Relative Strength Index"
Round	Round(a)	Función matemática "Redondeo a la unidad" (parte entera)

S

Código	Implementación	Función
SAR	SAR[At,St,Lim]	Designa el Parabólico SAR
SARatdmf	SARatdmf[At,St,Lim](price)	Designa el Parabólico SAR en el módulo ATDMF
SCREENER	SCREENER[c](price)	Muestra los resultados
Sin	Sin(a)	Función matemática "Seno"
Sgn	Sgn(a)	Función matemática "Signo de"
SMI	SMI [N,SS,DS](price)	Designa el índice Estocástico Momentum (Stochastic Momentum Index)
SmoothedStochastic	SmoothedStochastic[N,K](price)	Designa un estocástico alisado
Square	Square(a)	Función matemática "Cuadrado" (potencia 2)
Sqrt	Sqrt(a)	Función matemática "Raíz cuadrada"
STD	STD[N](price)	Función estadística "Desviación Típica"
STE	STE[N](price)	Función estadística "Error típico"
Stochastic	Stochastic[N,K](price)	Línea %K del Estocástico
Summation	Summation[N](price)	Suma de un cierto precio de las N últimas velas
SuperTrend	SuperTrend[STF,N]	Designa el Super Trend

T

Código	Implementación	Función
Tan	Tan(a)	Función matemática "Tangente"
TEMA	TEMA[N](price)	Media Móvil Exponencial Triple
THEN	Ver IF/THEN/ELSE/ENDIF	Instrucción que sigue la primera condición del conjunto condicional "IF"
Time	Time[N]	Da la evolución de la HoraMinutoSegundo y permite llamar la hora en el programa
TimeSeriesAverage	TimeSeriesAverage[N](price)	Media Móvil de las series temporales
TO	Ver FOR/TO/NEXT	Instrucción "hasta " en el bucle "Para" (FOR)
Today	Today	Fecha actual
TotalPrice	TotalPrice[N]	(cierre + apertura + máximo + mínimo) / 4
TR	TR(price)	Designa el True Range
TriangularAverage	TriangularAverage[N](price)	Media Móvil Triangular
TRIX	TRIX[N](price)	Triple Media Móvil Exponencial
TypicalPrice	TypicalPrice[N]	Designa el precio Típico (Media de máximo, mínimo y cierre)

U

Código	Implementación	Función
Undefined	a = Undefined	Permite dejar una variable indefinida (es un tipo de variable)

V

Código	Implementación	Función
Variation	Variation(price)	Da la diferencia entre el cierre de la víspera y el cierre actual en %
Volatility	Volatility[S, L]	Designa la volatilidad de Chaikin
Volume	Volume[N]	Designa el volumen
VolumeOscillator	VolumeOscillator[S,L]	Designa el oscilador de volumen
VolumeROC	VolumeROC[N]	Designa el volumen del Rate Of Change (ROC)

W

Código	Implementación	Función
WeightedAverage	WeightedAverage[N](price)	Designa la Media Móvil Ponderada
WeightedClose	WeightedClose[N]	Da la Media entre el precio de cierre, máximo y mínimo, con ponderaciones respectivas de 2, 1 y 1
WEND	Ver WHILE/DO/WEND	Instrucción a introducir al final del bucle While/Do/Whend (mientras)
WHILE/DO/WEND	WHILE (condición) DO (acción) WEND	Bucle "Mientras"
WilderAverage	WilderAverage[N](price)	Da la Media Móvil de Wilder
Williams	Williams[N](close)	Calcula el %R de Williams
WilliamsAccumDistr	WilliamsAccumDistr(price)	Indicador Acumulación/Distribución de Williams

X

Código	Implementación	Función
XOR	a XOR b	Operador lógico O exclusivo

Y

Código	Implementación	Función
Year	Year[N]	Da la evolución de los años y permite llamar a los años en el programa
Yesterday	Yesterday[N]	Da la evolución del día de ayer y permite llamar a este último en el programa

Z

Código	Implementación	Función
ZigZag	ZigZag[Zr](price)	Designa los Zig-Zag de la teoría de las ondas de Elliott
ZigZagPoint	ZigZagPoint[Zp](price)	Designa los Zig-Zag de la teoría de las ondas de Elliott calculadas a Z puntos

OTROS

Código	Función
+	Operador de adición
-	Operador de sustracción
*	Operador de multiplicación
/	Operador de división decimal
=	Operador de igualdad
<>	Operador de diferencia
<	Operador de inferioridad estricta
>	Operador de superioridad estricta
<=	Operador de inferioridad
>=	Operador de superioridad