

Todas estas categorías son importantes y su utilidad depende del tipo de indicador o sistema que estemos programando, no obstante podemos destacar las siguientes como las más comunes a la hora de desarrollar una estrategia de trading:

Quote Fields:

Contiene todos los comandos y funciones relacionados con el precio del activo que están a nuestra disposición para programar una estrategia, por ejemplo, el comando "*DailyOpen*" devuelve un valor numérico que indica el precio de apertura para la actual sesión de trading.

11.2. Programando una estrategia básica con PowerLanguage

En este apartado vamos a realizar paso a paso el proceso de programar una sencilla estrategia de trading con Power Language Editor. Concretamente vamos a programar el sistema "IBEX DIARIO Canal R"

Para comenzar, una vez que hayamos instalado la plataforma Multicharts, ejecutamos el programa Power Language Editor desde el escritorio o bien a través de C:\Program Files\TS Support\MultiCharts\ PLEditor.exe (o en la carpeta donde se haya instalado).

Cuando se haya ejecutado, el Power Language Editor nos mostrará la interfaz de programación que ya mencionamos en el punto 11.1. Nos centraremos en la ventana de programación (Ver Figura 64).

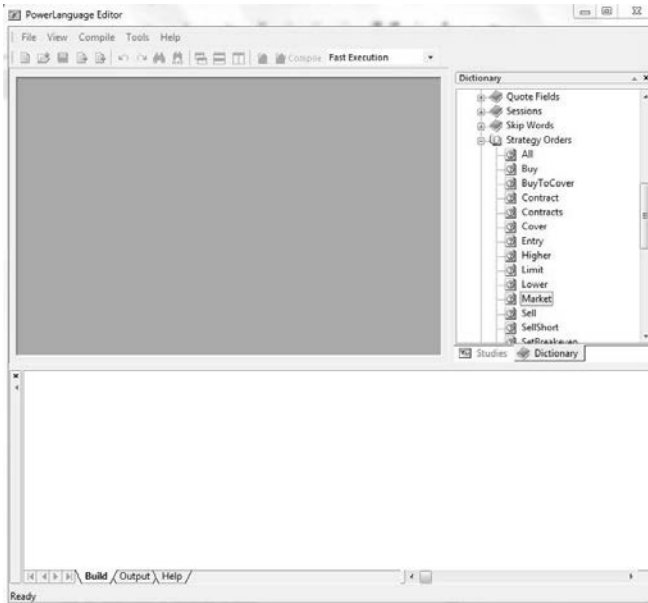



Figura 64. Programando una estrategia en PowerLanguage. Paso 1

Procedemos a preparar el archivo en el que vamos a escribir el código del sistema. Para ello vamos al menú superior y hacemos clic en “File” y luego “New”. También podemos utilizar el acceso directo haciendo clic en el icono:  que se encuentra también en la parte superior izquierda. Al hacer clic se nos mostrará la siguiente ventana (Figura 65).

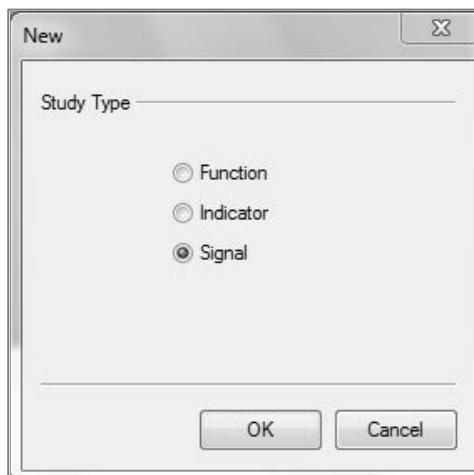


Figura 65. Programando una estrategia en PowerLanguage. Paso 2

Seleccionamos la opción “Signal” ya que nuestro objetivo es programar un sistema y no un indicador o una función. En ese mismo instante se abrirá una nueva ventana requiriéndonos ponerle nombre a nuestra estrategia. Escribimos: “Sistema IBEX DIARIO Canal R_L&C” y hacemos clic en “OK”

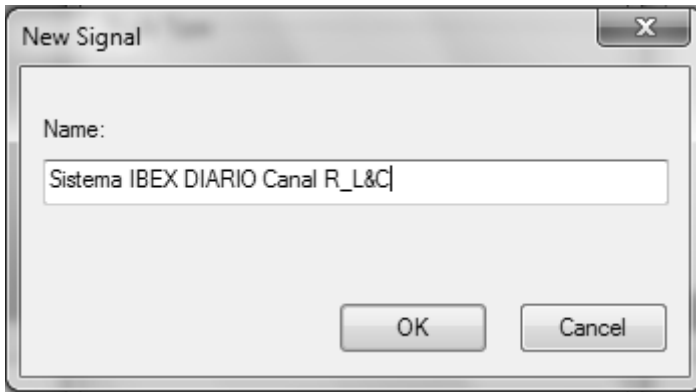


Figura 66. Programando una estrategia en PowerLanguage. Paso 3

Ahora ya tenemos lista la “pizarra” para comenzar a escribir nuestro código. Veremos una ventana con el fondo blanco y el cursor situado en la primera línea.

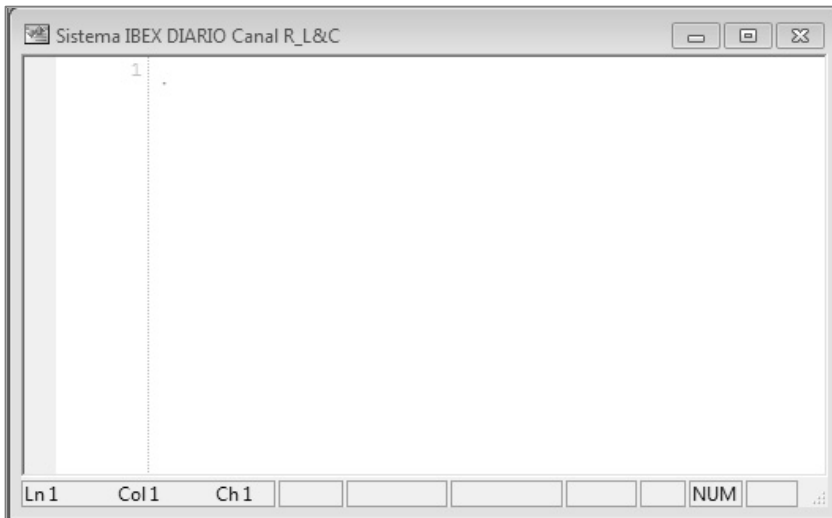


Figura 67. Programando una estrategia en PowerLanguage. Paso 2

Antes de continuar vamos a comentar brevemente la estructura básica que debe tener cualquier sistema en PowerLanguage, es decir, los componentes básicos que debe tener; estos son:

Parámetros

Variables

Regla o reglas de entrada

Regla o reglas de salida

En realidad los parámetros son también variables, la diferencia estriba simplemente en que los parámetros son aquellas variables que queremos que sean visibles posteriormente para optimizar o modificar. Esto quedará más claro cuando lo vayamos programando.

Comenzamos definiendo los parámetros que vamos a utilizar en este sistema, estos son:

- Objetivo
- Stop
- Porcentaje de *Trailing*
- Número de contratos
- Banda R Superior
- Banda R Inferior
- Número de barras desde la entrada
- Largo
- Corto

El objetivo y el stop no los comentaremos por razones más que obvias. El porcentaje de *trailing* es un componente del objetivo dinámico y representa el porcentaje que estaremos dispuestos a arriesgar una vez que hayamos alcanzado el objetivo, con el fin de incrementar nuestros beneficios pero con un riesgo reducido; es decir, es una pequeña herramienta de gestión monetaria que vamos a automatizar.

El número de contratos también es conveniente tratarlo como un parámetro ya que en el proceso de optimización del sistema puede ser interesante comparar los resultados que obtendríamos comprando un solo contrato con dos o tres o más.

Las Bandas R, como mencionamos en varios apartados de esta guía, son parámetros que una vez que son detectados se fijan y se comportan como constantes, pero al comienzo, cuando no se conoce su valor, se deben realizar miles de pruebas hasta descubrir el valor exacto para el producto sobre el que nos interesa. En el caso que nos ocupa, para el futuro del IBEX ya tenemos estos valores pre-definidos, pero en todo caso vamos a incluirlos como parámetros por si quisiéramos algún día probar a optimizarlos.

El número de barras desde la entrada es un parámetro que nos ayuda a controlar que una posición permanezca durante mucho tiempo en mercado sin definirse hacia el objetivo y sin ejecutarse el stop.

Finalmente, vamos a ponerle a este sistema una especie de “botones” con los que podemos decirle al sistema que opere solo en largo, solo en corto o de las dos maneras; para ellos agregamos los parámetros “largo” y “corto”.

Aclarado lo anterior nos ponemos frente a la ventana de programación y escribimos los parámetros como primera línea.

Lo primero que haremos será escribir un comentario con la información del sistema. El único propósito de hacer estos es por organización y claridad. Los comentarios son muy útiles en programación ya que nos pueden ayudar a recordar y entender en el futuro lo que hemos escrito en el pasado. Funciona como notas que no son tenidas en cuenta para el sistema. Otra cosa importante es que no pueden marcar tildes ni utilizar la eñe (ñ).

Para hacer un comentario en PowerLanguage utilizamos las llaves { }. Todo lo que va escrito entre llaves no va a ser interpretado por el sistema y el color de la letra se volverá verde. Así, cualquier texto que observemos que sea de color verde cuando abramos un sistema en powerlanguage nos indica que

se trata de un comentario que hizo quien lo programó. Escribiremos nuestro primer comentario:

```
{Sistema IBEX DIARIO Canal R (Largo y Corto)
}
```

A continuación procederemos con los parámetros. En PowerLanguage los parámetros se denominan inputs, así que para comenzar escribimos:

Inputs: Largo(1), Corto(1),Barras(10), BSup(80), BInf(78), NumContratos(1), Objetivo(500), Stop1(700),TrailingPercent(10);

Que en nuestra ventana de programación se verá más o menos como se ilustra en la siguiente imagen.

```
1 {Sistema IBEX DIARIO Canal R (Largo y Corto)
2 }
3
4 Inputs: Largo(1), Corto(1),Barras(10), BSup(80), BInf(78), NumContratos(1),
5 Objetivo(500), Stop1(700),TrailingPercent(10);
```

Figura 68. Programando una estrategia en PowerLanguage. Paso 3

Observaremos que cuando escribimos la palabra “Inputs:”, ésta al igual que los valores de los parámetros entre paréntesis se ponen de color azul, en tanto los nombres de los parámetros se ponen de color rosa. El PowerLanguage Editor dispone en su sintaxis de una serie de palabras clave que actúan como funciones que tienen un propósito determinado y que facilita la escritura del código, entre ellas se encuentra la palabra “Inputs” la cual siempre se mostrará de color azul y nos indica que después de los dos puntos seguidos (:) se escribirán los parámetros (variables que se pueden optimizar). Existen así mismo otras palabras claves comunes tales como *Open, Close, And, Entryprice, End, If, Ago, etc.* Todas ellas son útiles para armar el puzle de un sistema. Para saber más de estas palabras podemos consultar la pestaña “Diccionario” (*Dictionary*) ubicada en la “barra de navegación” (Ver Figura 61) y allí seleccionamos la primera opción de la lista titulada: “Índice alfabético de palabras clave” (*Keyword Alphabetical Index*) y se nos abre en

la “Ventana de Resultados” (la ventana No 3 que se indica en la Figura 61). Otra manera alternativa y más directa de abrir este listado de palabras claves es haciendo clic en la pestaña inferior “Help” de la “Ventana de Resultados”

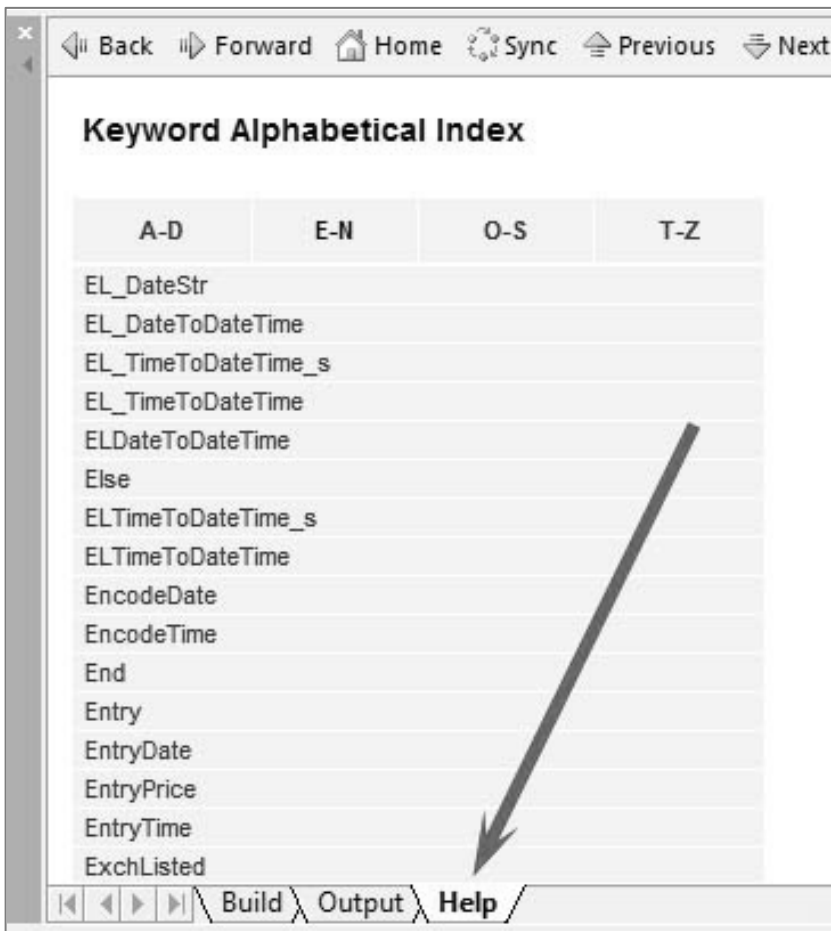


Figura 69. Programando una estrategia en PowerLanguage. Paso 4

Como ya hemos visto en este ejemplo en curso, no es necesario conocer todas las palabras claves para programar y la utilidad del diccionario es que sabemos que lo tenemos ahí para programar cualquier idea que se nos venga a la cabeza. Imagine por ejemplo que a usted se le ocurre la idea que conociendo el precio de cierre del 5to día hacia atrás con respecto a la fecha actual, usted puede estimar la dirección del mercado. Si el cierre de 5 días atrás es mayor que el cierre

actual es una señal de compra. PowerLanguage le facilita el camino para que usted programe fácilmente esta idea de trading con las palabras clave: “*If*”, “*then*”, “*close*”. Con una simple ojeada al diccionario se dará cuenta que para programar esta idea sólo tiene que escribir:

If Close[5] > Close[0] then Buy;

Los números entre corchetes [] le indican al sistema el número exacto de barras hacia atrás en donde debe consultar el valor de cierre. El valor cero [0] hace siempre referencia a la barra actual.

Continuamos con la escritura de nuestra estrategia básica sobre el futuro del IBEX. Veamos nuevamente el código que hemos escrito hasta ahora:

```
{Sistema IBEX DIARIO Canal R (Largo y Corto)
}
Inputs: Largo(1), Corto(1), Barras(10), BSup(80),
BInf(78), NumContratos(1),
Objetivo(500), Stop1(700), TrailingPercent(10);
```

Obsérvese que los Inputs se separan entre sí mediante comas (,) y que el último input termina con un punto y coma (;). Esta es una regla que se debe seguir rigurosamente y que aplica tanto a las variables como a los inputs (parámetros) los cuales como ya dijimos anteriormente, son también variables.

Bien, ahora procedemos con el siguiente bloque de código, es decir, las “variables”, para lo cual dejamos una línea en blanco de por medio después de los inputs (no es técnicamente necesario, es sólo con el propósito de separar y organizar nuestro código) y escribimos:

variables: bandasup(0), bandainf(0), Objetivo2(0), Stop2(0);

Consecuentemente nuestro código debería quedar así: