

[Home](#) > [Módulo de Estratégias](#) > Introdução

Introdução

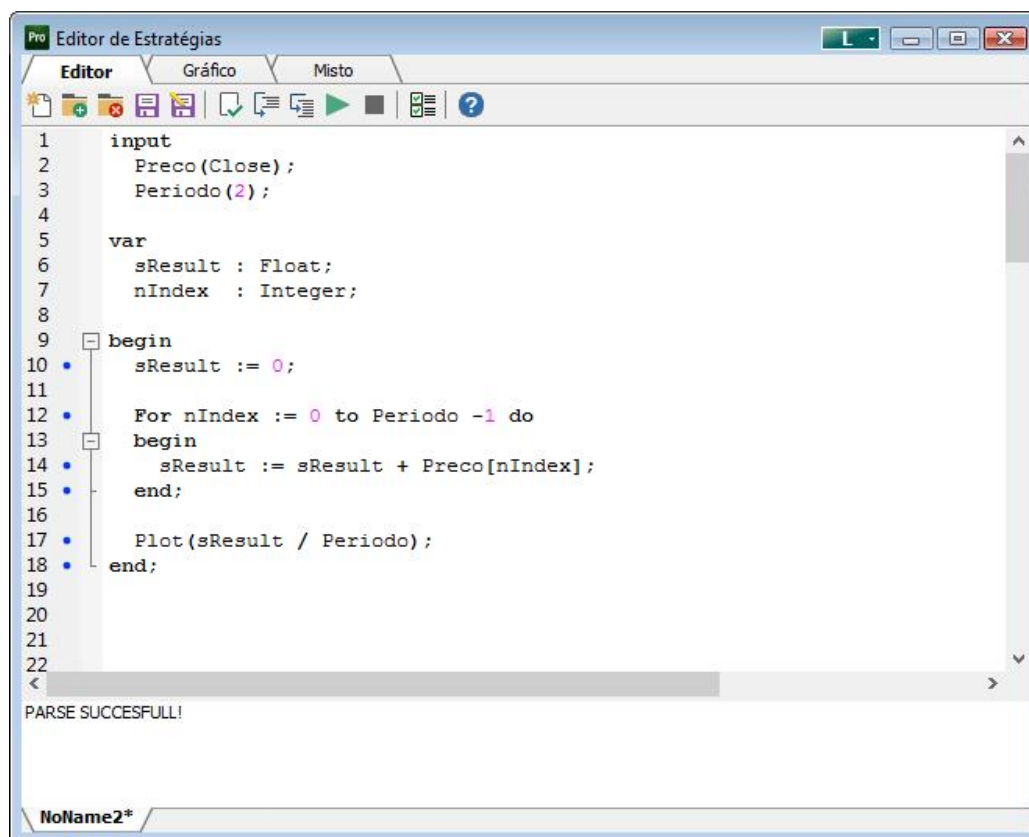
A NTSL (Nelógica Trading System Language) é uma poderosa linguagem criada com um único propósito: permitir o desenvolvimento dos melhores e mais eficientes sistemas algorítmicos de operação. A Nelógica oferece juntamente com a NTSL um ambiente de criação pioneiro e revolucionário chamado AlgoTools. Com o AlgoTools é possível codificar, testar e simular com grade agilidade qualquer estratégia de operação.

Na NTSL, o usuário encontra uma grande facilidade no momento de criação de suas estratégias, a possibilidade de criar toda ela em português, havendo assim, a facilidade e fácil entendimento da estratégia criada.

[Home](#) > [Módulo de Estratégias](#) > Estrutura de uma Estratégia

Estrutura de uma Estratégia

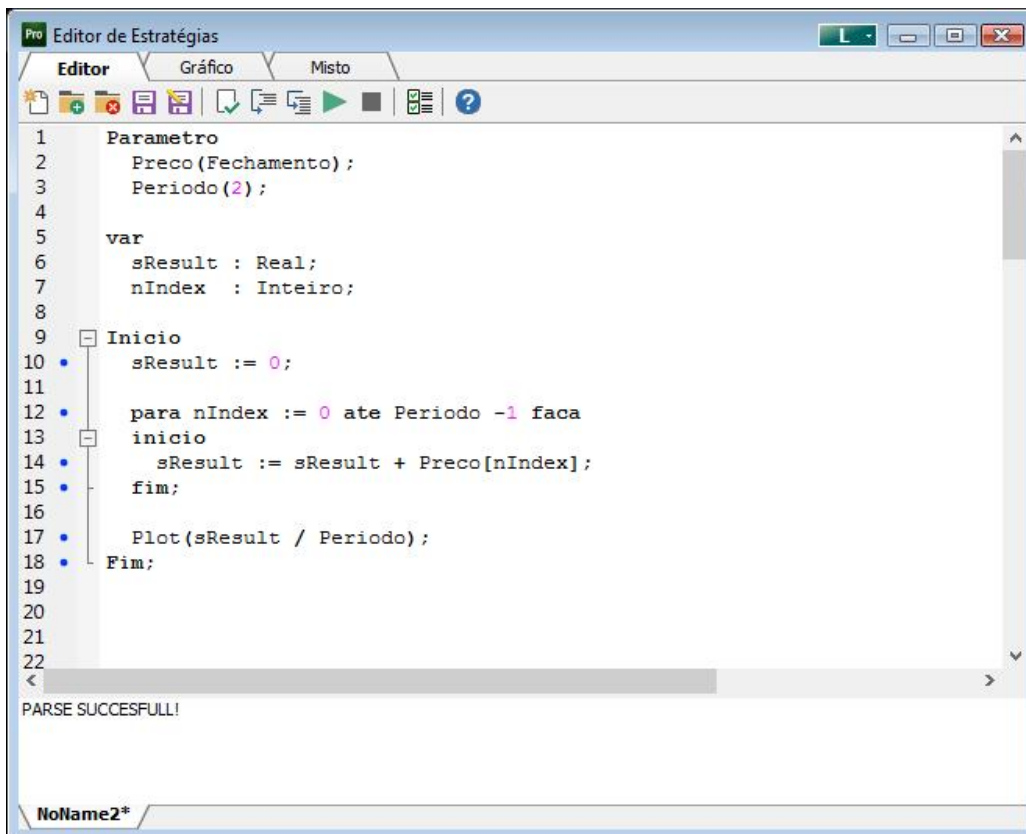
Observe o trecho de código abaixo que representa o indicador média móvel. Esse código apresenta as três áreas que definem a estrutura de uma estratégia. São elas: área de declaração de parâmetros de entrada, área de declaração de variáveis e funções e área de código.



```
1  input
2    Preco(Close);
3    Período(2);
4
5  var
6    sResult : Float;
7    nIndex  : Integer;
8
9  begin
10     sResult := 0;
11
12     For nIndex := 0 to Período -1 do
13     begin
14         sResult := sResult + Preco[nIndex];
15     end;
16
17     Plot(sResult / Período);
18 end;
```

PARSE SUCCESFULL!

NoName2*



A área de parâmetros de entrada compreende toda a região entre a palavra reservada **input (parametro)** e a palavra reservada **var**. A área de variáveis e funções começa com a palavra **var** e estende-se até a palavra **begin (inicio)**. Finalmente, a região de código inicia-se com a palavra reservada **begin (inicio)** e finaliza na palavra **end (fim)**, conforme imagem abaixo:

Entrada	• input
Variáveis e funções	• var
Código	• begin • end.

Área de declaração de parâmetros de entrada

Na área de declaração de parâmetros de entrada informamos todos os parâmetros externos que a estratégia usará. Esses parâmetros são fundamentais, pois:

- Definem a interface com o mundo externo, ou seja, é onde usuário poderá alterar e o que servirá como parâmetro de chamada caso a estratégia seja utilizada em outra interface/estratégia.
- Define os itens que serão analisados no processo de otimização.

Para definir um parâmetro: NOME_DO_PARÂMETRO (VALOR_DE_INICIALIZAÇÃO)

Exemplo

Input	Definição
Preco (Close);	Define um parâmetro inicializado com o valor de fechamento de preços da série de dados.
Período (2);	Define um parâmetro chamado Período com o valor 2.

Área de declaração de variáveis e funções

Na área de declaração de variáveis informamos todas as variáveis que serão utilizadas na estratégia. Nesta região também descrevemos funções que desejamos usar no código. As funções devem ser sempre codificadas após a declaração das variáveis.

Para declarar uma variável: NOME_DA_VARIAVEL: TIPO

Exemplo:

Variável	Definição
sResult : Float;	Cria uma variável chamada sResult do tipo ponto flutuante.
nIndex: Integer;	Cria uma variável chamada nIndex para armazenar números inteiros.

Área de código

Nesta parte descreve-se o código propriamente dito, ou seja, as regras que utilizam parâmetros, variáveis e outros dados para calcular sinais e indicadores.

[Home](#) > [Módulo de Estratégias](#) > Fluxo de Execução de uma Estratégia

Fluxo de Execução de uma Estratégia

O código de uma estratégia, naturalmente, é executado de maneira sequencial. Dentro desse contexto, observe novamente o indicador de média móvel na figura 1. Não existe explicitamente um laço que faça com que o sistema percorra toda a base de dados de um ativo. No entanto, essa é exatamente a ação realizada pelo software.

Assim, a sequência de comandos existente na área de código será executada uma vez para cada elemento de dados existente. Em um exemplo hipotético, mas ilustrativo imagine uma série de dados diários composta unicamente pelos 21 pregões de um determinado mês de janeiro. Para cada dado, será executado o código da estratégia em um loop implícito. A regra em pseudocódigo é, portanto:

- Tomando uma série de dados chamada aqui de X de tamanho N;
- Para cada item X[i], com i variando de 0 até N-1, executa-se a área de código da estratégia.

```

1  input
2    Preco(Close);
3    Periodo(2);
4
5  var
6    sResult : Float;
7    nIndex  : Integer;
8
9  begin
10   sResult := 0;
11
12   For nIndex := 0 to Periodo -1 do
13   begin
14     sResult := sResult + Preco[nIndex];
15   end;
16
17   Plot(sResult / Periodo);
18 end;
19
20
21
22

```

PARSE SUCCESFULL!

NoName2*

```

1  Parametro
2    Preco(Fechamento);
3    Periodo(2);
4
5  var
6    sResult : Real;
7    nIndex  : Inteiro;
8
9  Inicio
10   sResult := 0;
11
12   para nIndex := 0 ate Periodo -1 faca
13   inicio
14     sResult := sResult + Preco[nIndex];
15   fim;
16
17   Plot(sResult / Periodo);
18 Fim;
19
20
21
22

```

PARSE SUCCESFULL!

NoName2*

[Home](#) > [Módulo de Estratégias](#) > Variáveis e séries de dados

Variáveis e séries de dados

As variáveis são utilizadas para armazenar valores a serem usados dentro do fluxo do código. Elas podem ser **três tipos**:

- **Float (Real):** Representa números de ponto flutuante (reais)
- **Intege (Inteiro)r:** Representa números inteiros;
- **Boolean (Booleano):** Representa um tipo booleano (true [verdadeiro] ou false [falso])

As variáveis na linguagem oferecem uma flexibilidade muito maior do que na maioria dos sistemas de programação conhecidos. Na verdade, todas as variáveis são séries de dados, o que significa que é possível navegar entre valores atuais e passados.

Acessando dados anteriores

O dado atual de uma variável pode ser descrito por Variavel[0]. O número entre colchetes indica o dado de quantos períodos anteriores deseja-se acessar (sendo 0, portanto, da barra corrente). Para ilustrar melhor, vamos a um exemplo:

```
sResult := Preco[1];
```

A linha de código acima está atribuindo o valor da variável Preco[1] para a variável sResult. Imagine que Preço corresponde a todos os valores de fechamento da série de dados de um certo ativo, como na tabela abaixo:

Data	Posição	Valor
05/10/2010	Preco[0]	12,43
04/10/2010	Preco[1]	12,40
01/10/2010	Preco[2]	11,39
30/09/2010	Preco[3]	12,51
29/09/2010	Preco[4]	12,38
dd/mm/aaaa	Preco[n]	xx,xx

Portanto, Preco[1] refere-se ao valor de ontem do preço de fechamento (sResult vale então 12,40 em nosso exemplo). Dessa forma, o número inteiro que especificarmos entre colchetes indica ao sistema quantos períodos no passado deve-se acessar a informação.

Séries de dados

As séries de dados, são os valores decorrentes de cada candle do gráfico e são denominados da seguinte maneira:

- **Open (Abertura):** Retorna o valor de abertura da barra atual.
- **Close (Fechamento):** Retorna o valor de fechamento da barra atual.
- **Low (Mínima):** Retorna o valor mínimo da barra atual.
- **High (Fechamento):** Retorna o valor máximo da barra atual.
- **Quantity (Quantidade):** Retorna o valor da quantidade da barra atual.

[Home](#) > [Módulo de Estratégias](#) > Controle de Fluxo

Controle de Fluxo

As instruções de fluxo são utilizadas para administrar a sequência de execução das instruções dentro de um programa. A NTSL apresenta três tipos desse tipo:

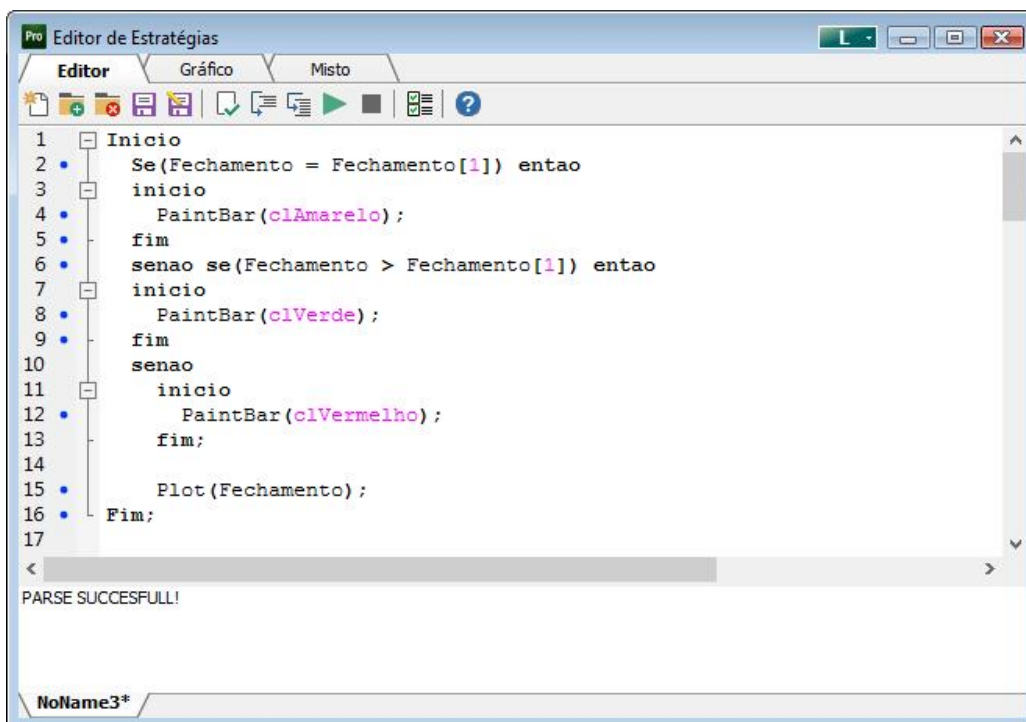
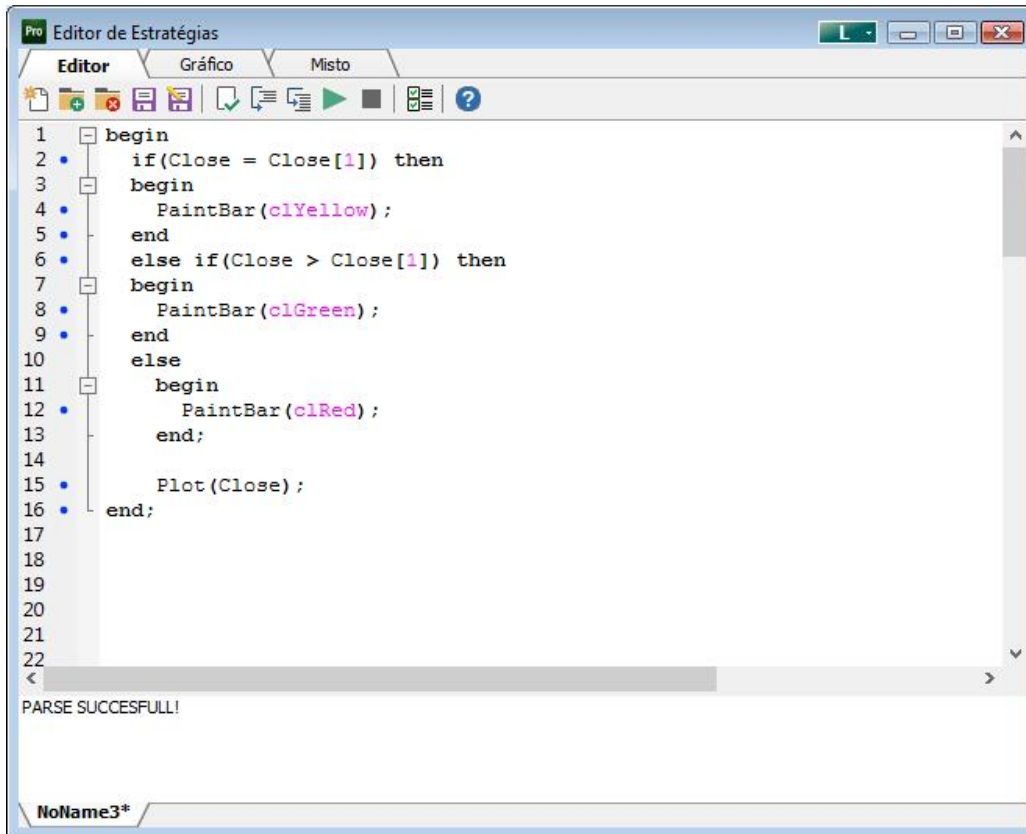
IF THEN ELSE (SE ENTÃO SENÃO)

Se (condição) for verdadeiro executa-se a listagem (comandos 1), caso (condição) seja falso executa-se (comandos 2). Tanto (comandos 1) quanto (comandos 2) podem ser sequências de instruções.

A expressão (condição) pode ser qualquer tipo de teste que resulte em verdadeiro ou falso, alguns exemplos:

- IF (Close > Close[1])
- IF (nIndex = 10)
- IF ((bCond = false) and (Volume > 1000)

A seguir um exemplo de coloração de candles de acordo com a condição presente no **If then else**.



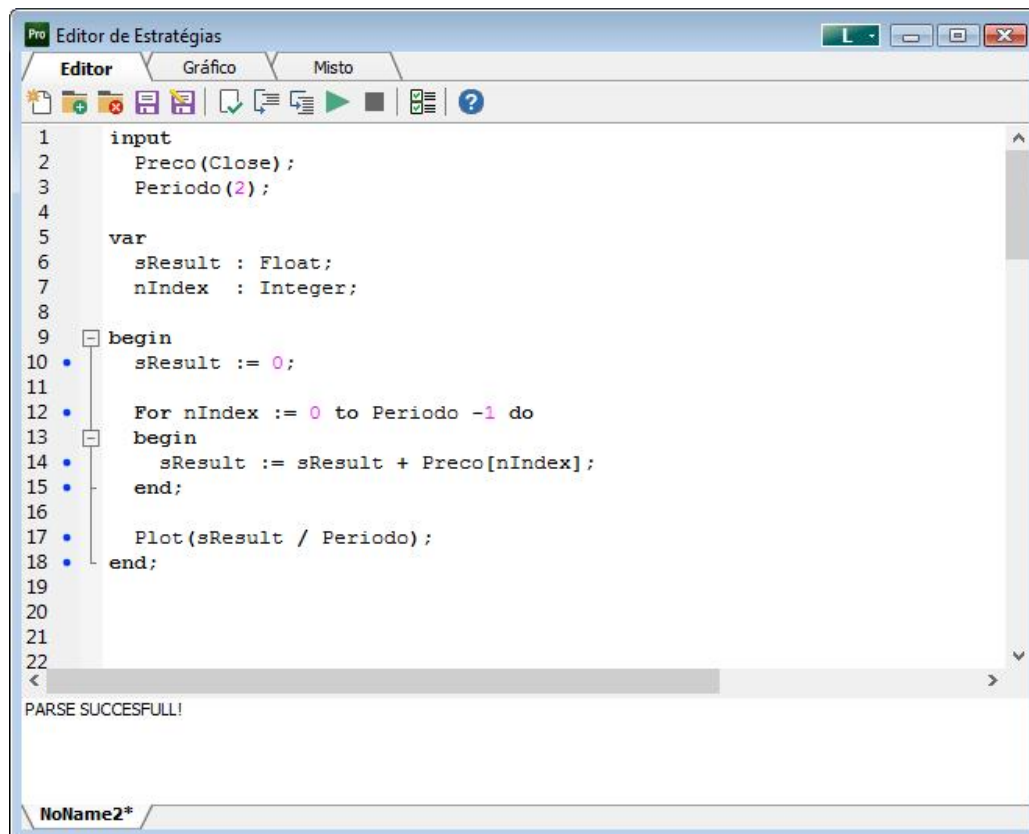
Se o valor de fechamento da barra atual (Close equivale a Close[0]) for igual ao fechamento da barra anterior (Representado por Close [1]) executa-se o código que segue a palavra reservada **THEN (ENTÃO)**. Caso contrário, o sistema executa o código subsequente até chegar na palavra reservada **ELSE (SENÃO)**.

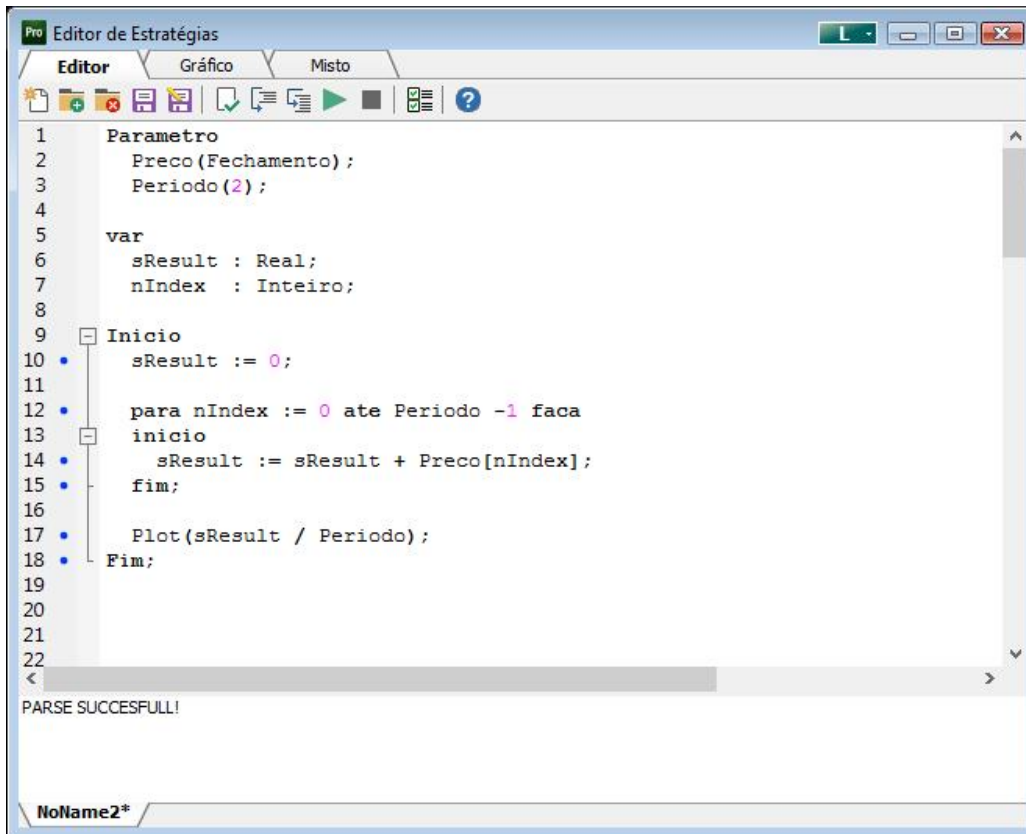
FOR TO (PARA ATE)

O comando FOR é utilizado para definir um loop controlado, ou seja, (comandos) é executado repetidamente até que a (variável de contagem) saia do (valor inicial) e atinja (valor final). A cada iteração a (variável de contagem) é incrementada em 1.

Exemplo:

Observe o código da média móvel na figura 1. A variável de contagem nIndex começa valendo 0 e deve chegar ao valor de Período - 1. Período é um parâmetro de entrada, ele é usado para definir o tamanho da média. Assim, se Período valer 9, o comando FOR criará um loop de 9 iterações (de 0 até 8) para cada barra, calculando assim o valor médio para a posição atual.





WHILE DO (ENQUANTO FAÇA)

A execução da estratégia ao chegar no comando WHILE testa o resultado de (condição). Caso (condição) seja verdadeiro (true) a listagem (comandos) é executada. Após a execução a (condição) volta a ser testada, assim, o loop apenas irá se encerrar quando (condição) deixar de ser verdadeira.

Exemplo

No código a seguir, reescrevemos o indicador média móvel utilizando a instrução **WHILE (ENQUANTO)** ao invés de **FOR (PARA)**.


```

1  input
2    Preco(Close);
3    Perodo(2);
4
5  var
6    sResult : Float;
7    nIndex  : Integer;
8
9  begin
10   sResult := 0;
11   nIndex  := 0;
12
13   while(nIndex <= Perodo -1) do
14   begin
15     sResult := sResult + Preco[nIndex];
16     nIndex  := nIndex + 1;
17   end;
18
19   Plot(sResult / Perodo);
20 end;
21
22 <

```

PARSE SUCCESFULL!

NoName2 MM2*

```

1  Parametro
2    Preco(Fechamento);
3    Perodo(2);
4
5  var
6    sResult : Real;
7    nIndex  : Inteiro;
8
9  Inicio
10   sResult := 0;
11   nIndex  := 0;
12
13   enquanto(nIndex <= Perodo -1) faca
14   inicio
15     sResult := sResult + Preco[nIndex];
16     nIndex  := nIndex + 1;
17   fim;
18
19   Plot(sResult / Perodo);
20 Fim;
21
22 <

```

PARSE SUCCESFULL!

NoName2 MM2*

[Home](#) > [Módulo de Estratégias](#) > Operadores

Operadores

Os operadores constituem os símbolos matemáticos e lógicos usados em cálculos e comparações.

Operadores matemáticos

Os operadores matemáticos são:

Operador	Descrição	Exemplo	Resultado
+	Adição	5+4	9
-	Subtração	5-4	1
*	Multiplicação	5*4	20
/	Divisão	5/4	1,25

O operador de divisão possui a maior força de precedência, seguido por multiplicação. Assim, como consta na imagem abaixo:

Expressão	Resultado
10*10/5+2	22
10*10+2	102
50+100*10/2-1	549

Operadores lógico

Os Operadores lógicos são utilizados principalmente para comparações.

"E" lógico

Representado pela palavra reservada **and (e)**, retornará TRUE somente quando as duas condições de teste forem verdadeiras conforme Tabela Verdade abaixo:

Condição 1	AND	Condição 2	Resultado
FALSE	AND	FALSE	FALSE
FALSE	AND	TRUE	FALSE
TRUE	AND	FALSE	FALSE
TRUE	AND	TRUE	TRUE

"OU" lógico

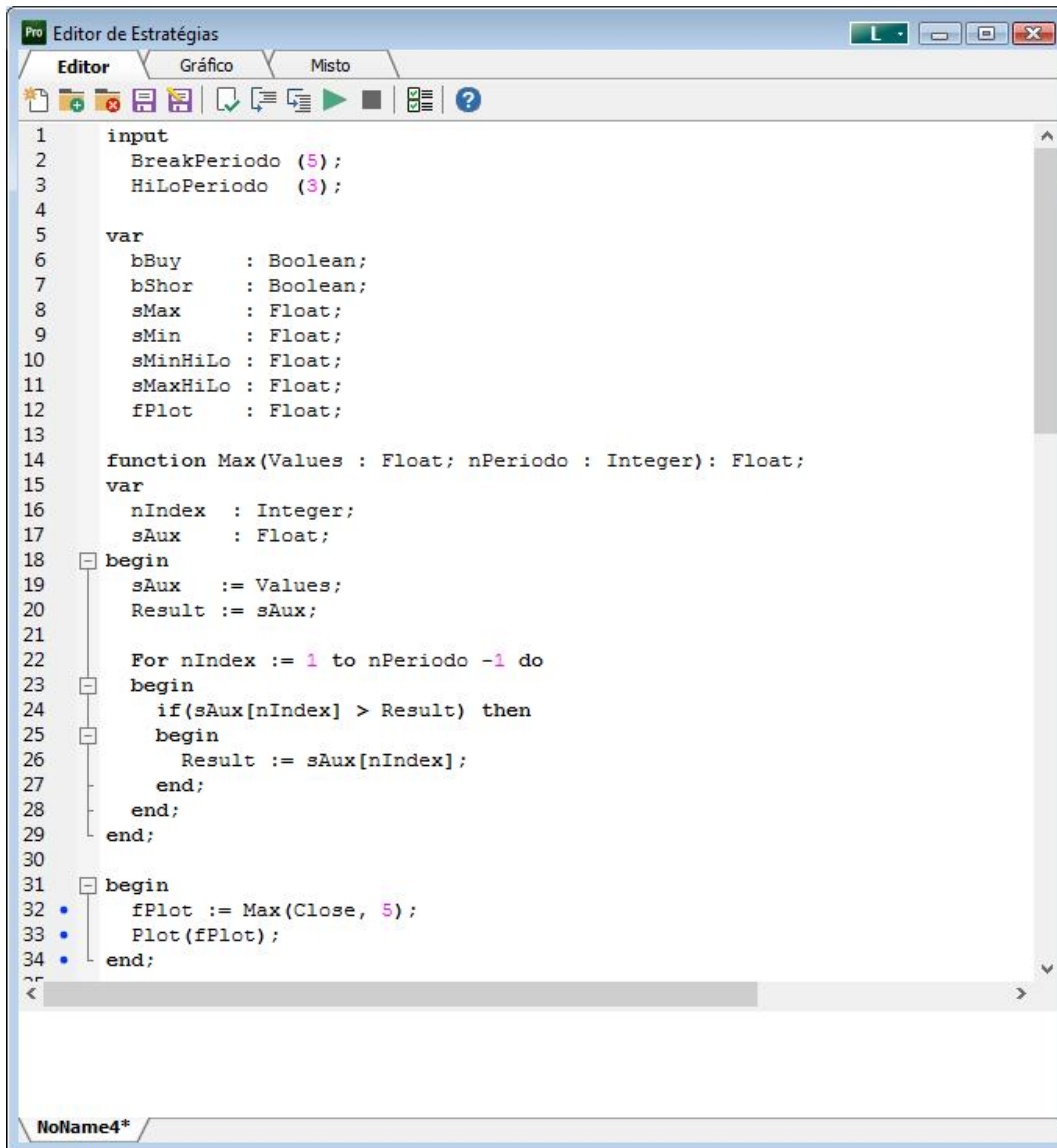
Representado pela palavra reservada **or (ou)**, retornará TRUE (verdadeiro) sempre que pelo menos uma das condições de teste for verdadeira, conforme Tabela Verdade abaixo:

Condição 1	OR	Condição 2	Resultado
FALSE	OR	FALSE	FALSE
FALSE	OR	TRUE	TRUE
TRUE	OR	FALSE	TRUE
TRUE	OR	TRUE	TRUE

[Home](#) > [Módulo de Estratégias](#) > Funções

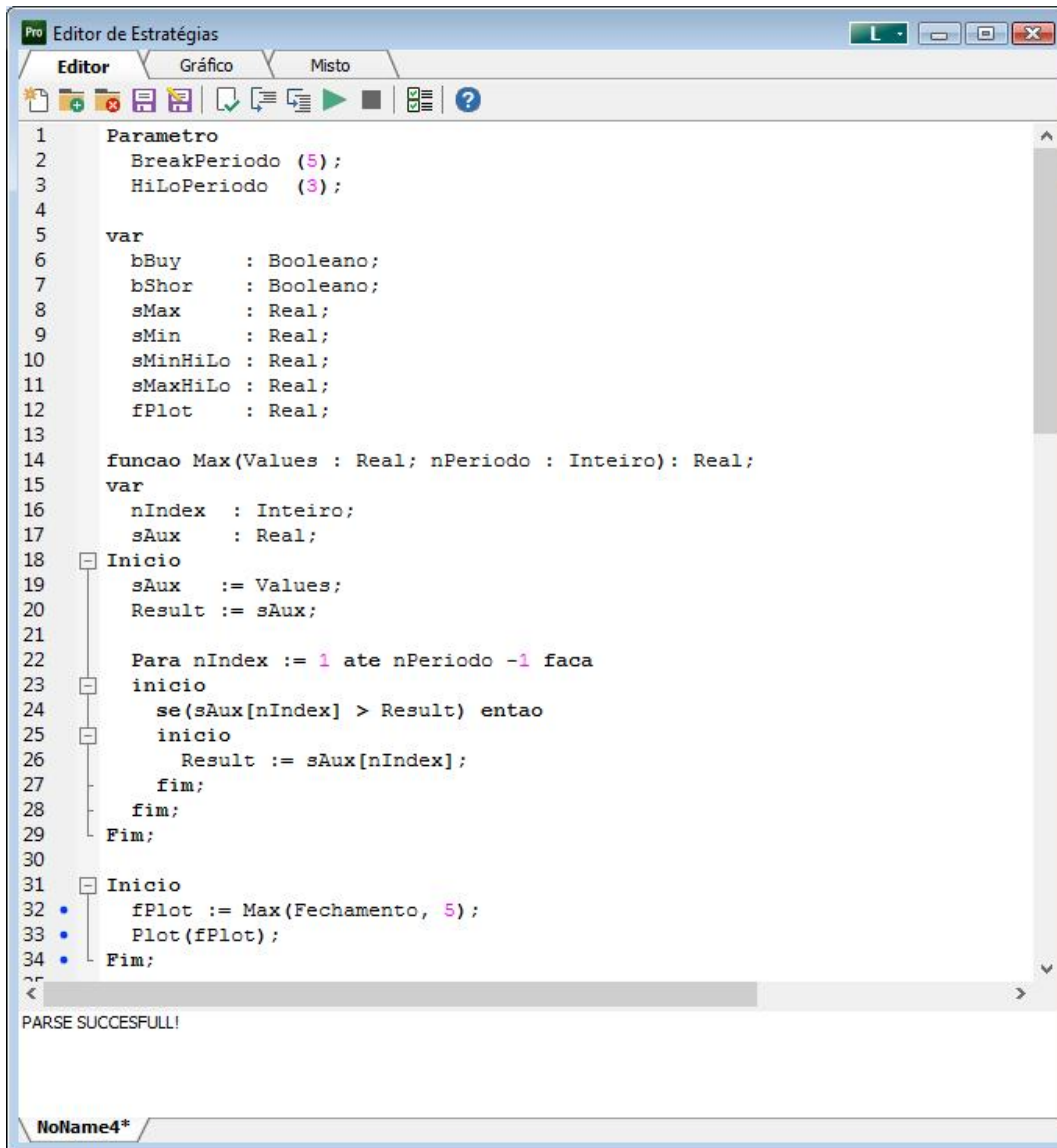
Funções

Conforme visto, funções são declaradas e descritas na área de declaração de variáveis e funções, abaixo um exemplo de funções:



```
1  input
2      BreakPeriodo (5);
3      HiLoPeriodo  (3);
4
5  var
6      bBuy      : Boolean;
7      bShor     : Boolean;
8      sMax      : Float;
9      sMin      : Float;
10     sMinHiLo  : Float;
11     sMaxHiLo  : Float;
12     fPlot     : Float;
13
14     function Max(Values : Float; nPeriodo : Integer): Float;
15     var
16         nIndex : Integer;
17         sAux    : Float;
18     begin
19         sAux := Values;
20         Result := sAux;
21
22         For nIndex := 1 to nPeriodo -1 do
23         begin
24             if(sAux[nIndex] > Result) then
25             begin
26                 Result := sAux[nIndex];
27             end;
28         end;
29     end;
30
31     begin
32         fPlot := Max(Close, 5);
33         Plot(fPlot);
34     end;
```

NoName4*



Observe que primeiro são declaradas as seis variáveis usadas na área principal. É sempre interessante manter o código o mais claro e organizado possível e as funções desempenham um papel fundamental nessa tarefa.

Criando Funções

Sintaxe:

Function (funcao) Nome da Função ((parâmetro 1 : TIPO); (parâmetro 2: TIPO:);(parâmetro n: TIPO)): Tipo de Retorno

Begin(inicio)

Comandos

End (fim);

[Home](#) > [Módulo de Estratégias](#) > Funções de biblioteca

Funções de biblioteca

Além do usuário poder criar seus próprios indicadores, é possível utilizar a biblioteca do sistema, ou seja, o usuário pode utilizar estratégias já criadas em novas.

Dentro das funcionalidades de bibliotecas, o usuário poderá colorir os gráficos de acordo com as condições determinadas pelo seu indicador.

Funções Gráficas

Para criar um gráfico de linha o usuário deverá utilizar a função **Plot**, onde o sistema irá efetuar a interligação dos pontos criados pelo indicador.

Funções matemáticas

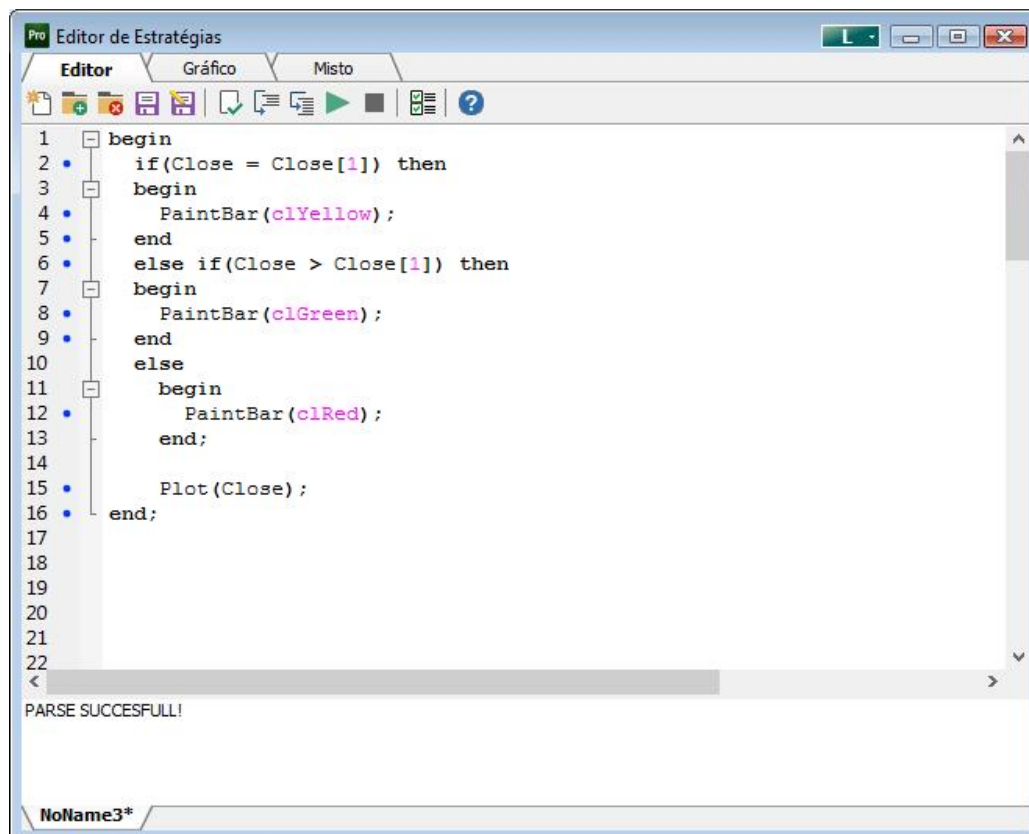
As funções matemáticas tem como finalidade implementar as seguintes funcionalidades:

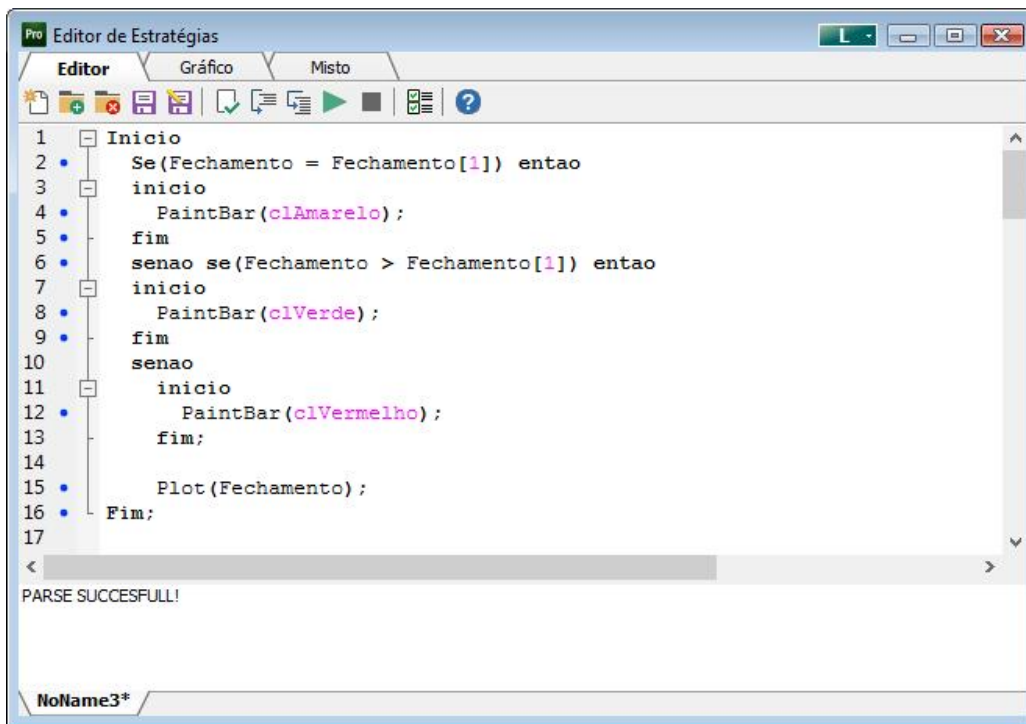
- **Power(valor,potência):** Tem como funcionalidade, gerar valores elevados em determinada potência;
- **Round(valor):** Tem como funcionalidade, arredondamento de números quebrados, caso o valor após a vírgula seja menor do que cinco, arredonda para baixo, caso contrário, arredonda para cima;
- **Sqrt(valor):** Tem como funcionalidade mostrar a raiz quadrada de valores desejados pelo usuário;

Funções Gráficas

Como visto anteriormente, a função **Plot** realiza a ligação dos valores gerados na estratégia e cria gráficos de linhas, mas caso haja a necessidade, o usuário também poderá colorir o gráfico de acordo com o desejado.

Esta funcionalidade denominada **PaintBar(cor)** permite ao usuário, colorir o gráfico com cores em determinadas situações do indicador, como na imagem abaixo:





As cores disponíveis para coloração e seus respectivos códigos são possíveis ser visualizados na imagem abaixo:



- clPreto;
- clMarrom;
- clVerde;
- clOlive;
- clNaval;
- clPurpura;
- clTeal;
- clCinza;
- clPrata;
- clVermelho;
- clLima;

- clAmarelo;
- clAzul;
- clFuschia;
- clAqua;
- clBranco;
- clVerdeClaro;
- clAzulClaro;

[Home](#) > [Módulo de Estratégias](#) > Lista de Funções

Lista de Funções

A seguir de forma descritiva, segue a lista de funções presentes no ProfitChart:

Back-testing

- **BuyAtMarket:** Realiza envio de ordem de compra à mercado;
- **BuyPrice:** Retorna o valor da última operação de compra;
- **BuyStop(Stop, Limite):** Realiza envio de ordem de compra stop;
- **BuyToCoverStop(Stop, Limite):** Realiza o envio de ordem de compra stop para fechar posição;
- **BuyToCoverAtMarket:** Realiza o fechamento de uma operação de venda;
- **IsBought:** Retorna verdadeiro caso o cliente esteja posicionado na compra;
- **IsSold:** Retorna verdadeiro caso o cliente esteja posicionado na venda;
- **SellPrice:** Retorna o valor da última operação de venda;
- **SellToCoverAtMarket:** Realiza uma ordem de venda à mercado caso exista uma posição de compra;
- **SellToCoverStop(Stop, Limite):** Envia uma ordem stop caso exista uma posição de compra;
- **SellShortAtMarket:** Envia ordem de venda à mercado para abrir posição;
- **SellShortStop(Stop, Limite):** Abre uma posição de venda enviando uma ordem do tipo Stop.

Calendário

- **CalcDate(Data referência, Data de alteração):** Retorna o cálculo de adição ou subtração de datas
- **CalcTime(Hora referência, Tempo de alteração):** Retorna o cálculo de adição ou subtração de horas;
- **CloseD(Número de dias anteriores):** Retorna o valor de fechamento de determinado dia anterior;
- **CloseM(Número de meses anteriores):** Retorna o valor de fechamento de determinado mês anterior;
- **CloseW(Número de semanas anteriores):** Retorna o valor de fechamento de determinada semana anterior;
- **CloseY(Número de anos anteriores):** Retorna o valor de fechamento de determinado ano anterior;
- **DaysToExpiration(Mês, Ano desejado - 1900):** Retorna o número de dias que existem entre data específica e dia atual;
- **HighD(Número de dias anteriores):** Retorna o valor de máxima de determinado dia retroativo;
- **HighM(Número de dias anteriores):** Retorna o valor de máxima de determinado mês retroativo;

- **HighW(Número de meses anteriores):** Retorna o valor de máxima de determinada semana anterior;
- **HighY(Número de anos anteriores):** Retorna o valor de máxima de um candle de determinado ano anterior;
- **LowD(Número de dias anteriores):** Retorna o valor de mínima de determinado dia retroativo;
- **LowM(Número de meses anteriores):** Retorna o valor de mínima de um candle mensal anterior determinado;
- **LowW(Número de semanas anteriores):** Retorna o valor de mínima de um candle semanal anterior determinado;
- **LowY(Número de anos anteriores):** Retorna o valor de mínima de um candle anual anterior determinado;
- **LastCalcDate:** Retorna a data do último candle completo do gráfico;
- **LastCalcTime:** Retorna o tempo de conclusão do último candle;
- **LastDayOfMonth(Mês de referência):** Retorna o valor do último dia do mês de referência;
- **Next3rdFriday(Mes):** Retorna quantos dias úteis faltam para a terceira sexta feira do determinado mês;
- **OpenD(Número de dias anteriores):** Retorna o valor de abertura de determinado dia retroativo;
- **OpenM(Número de meses anteriores):** Retorna o valor de abertura de determinado mês retroativo;
- **OpenW(Número de semanas anteriores):** Retorna o valor de abertura de determinada semana retroativa;
- **OpenY(Número de anos anteriores):** Retorna o valor de abertura de determinado ano retroativo;

Gráficas

- **Alert(Cor):** Aciona um alarme quando ocorrer situação e colore candle atual;
- **AvgPrice:** Retorna a média entre Abertura, Máxima, Mínima, Fechamento de determinado candle;
- **CurrentBar:** Retorna o índice atual da barra (candle);
- **C_Doji (Percentual):** Retorna valor de 0 ou 1 caso o candle seja um Doji com o percentual determinado;
- **C_ShootingStar(Tamanho, Fator):** Retorna o valor de 0 ou 1 caso o candle seja um Shooting Star de acordo com o Tamanho e o Fator;
- **Highest(Série, Período):** Retorna o maior valor da série dentro do período;
- **HighestBar(Série, Período):** Retorna o índice do maior valor da série no período;
- **LastBarOnChart:** Retorna se é a última barra do gráfico;
- **Leader:** Retorna se ponto médio é maior que mínima ou maior que máxima de candle anterior;
- **Lowest(Série, Período):** Retorna o menor valor da série dentro no período;
- **LowestBar(Série, Período):** Retorna o índice da barra com o menor valor da série no período;
- **MedianPrice:** Retorna a média entre a máxima e a mínima de determinado candle;
- **PaintBar(cor):** Colore as barras do gráfico;
- **Plot:** Desenha o indicador de acordo com o gráfico;
- **Range:** Retorna o valor de Máxima menos Mínima do determinado candle;
- **RGB(RED, GREEN, BLUE):** Colore de acordo com os valores de RGB;
- **TypicalPrice:** Retorna o valor médio entre a máxima, mínima e fechamento de determinado candle.

Indicadores

- **2MVPower**: Retorna status do indicador 2MV Power;
- **AVGTrueRange(Período)**: Retorna o valor do indicador True Range;
- **AccuDistr**: Retorna o valor do indicador Acumulação/Distribuição;
- **AccuDistrW**: Retorna o valor do indicador Acumulação/Distribuição Williams;
- **ADX(Período, Tipo de Média)**: Retorna o valor do indicador ADX;
- **ArmsEaseOfMov(Média, Tipo de Média)**: Retorna o valor do indicador Arms Ease of Movement;
- **AroonLin(Período) | Linha Desejada |**: Retorna o valor do indicador Aroon Linha, de acordo com o período e de acordo com a linha desejada;
- **AroonOsc(Período)**: Retorna o valor do indicador Aroon Oscilador;
- **BalanceOfPower(Média, Tipo de Média)**: Retorna o valor do indicador Balança de poder;
- **BearPower(Período, Média, Tipo de Média)**: Retorna o valor do indicador Bear Power;
- **BollingerBands(Desvio, Média, Tipo de Média) | Linha Desejada |**: Retorna o valor da linha da Banda de Bollinger de acordo com a linha desejada;
- **BollingerBandW(Desvio, Média, Tipo de Média)**: Retorna o valor do indicador Bollinger Band Width;
- **BollingerBPerc(Desvio, Média, Tipo de Média)**: Retorna o valor do indicador Bollinger b%;
- **BullPower(Período, Média, Tipo de Média)**: Retorna o valor do indicador Bull Power;
- **CCI(Período)**: Retorna o valor do indicador CCI;
- **ChaikinMoneyFlow(Período)**: Retorna o valor do indicador Chaikin Money Flow;
- **ChaikinOsc(Média Longa, Média Curta)**: Retorna o valor do indicador Oscilador Chaikin;
- **C_Doji(Percentual)**: Identifica se o candle é um Candlestick "Doji";
- **DarvasBox**: Retorna o valor do indicador Darvas Box;
- **DiDiIndex(Média de Referência, Tipo de média de referência, Média1, Tipo de Média1, Média2, Tipo Média2) | Linha Desejada |**: Retorna o valor da linha selecionada do indicador DiDi Index;
- **DiPDiM(Período) | Linha Desejada |**: Retorna o valor do indicador DI+/DI- de acordo com a linha desejada;
- **DonchianCH(Período) | Linha Desejada |**: Retorna o valor do indicador Canal Donchian de acordo com a linha desejada;
- **Envelope(Percentual, Média, Tipo de Média) | Linha Desejada |**: Retorna o valor do Retorna o valor da linha do indicador Envelope;
- **FastStochastic(Período)**: Retorna o valor do indicador Estocástico Rápido;
- **FinancialVol(VolumeProjetado)**: Retorna o Valor do volume financeiro com ou sem o Volume Projetado;
- **ForceIndex(Média, Tipo de média)**: Retorna o valor do indicador Force Index;
- **FrassonATR(Fator, Período MaxMin, Período ATR) | Linha Desejada |**: Retorna o valor do indicador Frasson ATR;
- **FrassonVH(Fator, Período MaxMin, Período VH) | Linha Desejada |**: Retorna o valor do indicador Frasson VH;
- **FullStochastic(Período)**: Retorna o valor do indicador Estocástico Pleno;
- **FuraChao(Coeficiente, Deslocamento)**: Retorna o valor do indicador Fura Chão;

- **FuraTeto(Coeficiente, Deslocamento):** Retorna o valor do indicador Fura Teto;
- **HeikinAshi(Média, Tipo de Média) | Dados desejados | :** Retorna o valor do indicador HeikinAshi de acordo com a linha desejada;
- **HiLoActivator(Período) | Linha Desejada | :** Retorna o valor do HiLo Activator de acordo com a linha desejada;
- **HistoVolatility(Média, Tipo da média):** Retorna o valor do indicador Volatilidade Histórica;
- **IchimokuCloud(Tenkan-Sen, Kijun-Sen, Senkou Span B) | Linha Desejada | :** Retorna o valor do indicador Ichimoku Cloud de acordo com a linha desejada;
- **ImpliedVolatility(Black&Scholes, Americana):** Retorna o cálculo do indicador Volatilidade Implícita, dependendo do tipo de cálculo utilizado;
- **KeltnerCH(Desvio, Média, Tipo de Média) | Linha Desejada | :** Retorna o valor do indicador Keltner Chaneles, de acordo com a linha desejada;
- **Kvo(Média Longa, Média Curta, Sinal) | Dado Desejado | :** Retorna o dado desejado do indicador KVO Linha & Histograma;
- **MACD(Média Longa, Média Curta, Sinal) | Dado Desejado | :** Retorna o dado desejado do indicador KVO Linha & Histograma;
- **MFI:** Retorna o valor do indicador Market Facilitation Index;
- **Momentum(Período, Média, Tipo da média):** Retorna o valor do indicador Momentum;
- **MomentumStochastic(Período):** Retorna o valor do indicador Momento Estocástico;
- **MoneyFlow:** Retorna o valor do indicador Money Flow;
- **MoneyFlowIndex(Período):** Retorna o valor do indicador Money Flow Index de acordo com o período utilizado;
- **NelogicaBottomFinder | Linha Desejada | :** Retorna o valor do indicador Nelogica Bottom Finder de acordo com a linha Desejada;
- **NelogicaPullBackFinder | Linha Desejada | :** Retorna o valor do indicador Nelogica PullBack Finder de acordo com a linha desejada;
- **OBV:** Retorna o valor do indicador OBV;
- **OnBalanceTR:** Retorna o valor do indicador On Balance True Range;
- **OpenInterest:** o valor do indicador Contratos em aberto;
- **ParabolicSAR(Fator de Aceleração, Limite de aceleração):** Retorna o valor do indicador SAR Parabolic;
- **Pivot(Normal,TrêsLinhas) | Linha Desejada | :** Retorna o valor do indicador Pivot, de acordo com a linha selecionada;
- **PriceOsc(Média 1, Tipo de média 1, Média 2, Tipo de média 2):** Retorna o valor do indicador Oscilador de preço, de acordo com as médias utilizadas;
- **PriceVolumeTrend:** Retorna o valor do indicador Tendência Preço/Volume;
- **PriorCote(Tipo):** Retorna o valor do indicador Prior Cote, de acordo com o tipo utilizado;
- **QuantityVol:** Retorna o valor do indicador Volume Quantidade;
- **Rafi:** Retorna o valor do indicador Rafi;
- **Ravi(Média Longa, Média Curta):** Retorna o valor do indicador Ravi de acordo com o período das médias desejadas;
- **Rsi(Período, Tipo):**Retorna o valor do indicador IFR;
- **RsiStochastic(Período):** Retorna o valor do indicador IFR Estocástico;
- **Roc(Período, Média, Tipo de Média):** Retorna o valor do indicador Roc;

- **SafeZoneDownTrend(Multiplicador, Média, Deslocamento):** Retorna o valor do indicador Stop SafeZone DownTrend;
- **SafeZoneUpTrend(Multiplicador, Média, Deslocamento):** Retorna o valor do indicador Stop SafeZone UpTrend;
- **SlowStochastic(Período):** Retorna o valor do indicador Estocástico Lento;
- **StopATR(Desvio, Média, Tipo de Média) | Linha Desejada | :** Retorna o valor do indicador Stop ATR de acordo com a linha desejada;
- **Tilson(Fator, Média):** Retorna o valor do indicador Tilson's T3 Moving Average;
- **Trix(Média, Tipo de Média):** Retorna o valor do indicador TRIX;
- **TrixM(Média, Tipo de Média):** Retorna o valor do indicador TRIXM;
- **TopBottomDetector(Período):** Retorna o valor do indicador Detector de Topos e Fundos;
- **Trades:** Retorna o valor do indicador Negócios;
- **VSS(Multiplicador, Média, Deslocamento):** Retorna o valor do indicador VSS;
- **VWAP(Tipo):** Retorna o valor do indicador VWAP de acordo com o tipo desejado;
- **Williams(Período):** Retorna o valor do indicador Williams de acordo com o período desejado;

Matemáticas

- **ABS(Valor ou variável):** Retorna o valor absoluto (sem sinal);
- **Combination(Numero, Quantidade de números):** Retorna o número de combinações possíveis em determinado utilizando o determinado número;
- **COS(Valor ou variável):** Retorna o valor de um Cosseno;
- **EXP(Valor ou variável):** Retorna o valor de um Exponencial;
- **Factorial(Número):** Retorna o fatorial do valor estabelecido ($1*2*...n$);
- **GCD(Valor1, Valor2)** Retorna o maior denominador comum de dois números;
- **MinutesToTime(Minutos):** Retorna o valor de minutos desde a meia noite em horas;
- **Permutation(Número de Candles considerados, Número de Objetos):**
- **Power(Base, Expoente):** Eleva valores nas determinadas potências;
- **Random(Limite):** Retorna um valor aleatório de 0 até o limite;
- **Round(Valor ou variável):** Arredondamento de número;
- **Round2Fraction(Valor Decimal):** Arredonda o número para o valor mais próximo de um múltiplo do incremento mínimo de um ativo;
- **SIN(Valor ou variável):** Retorna o valor de Seno;
- **Sqrt(Valor ou variável):** Retorna raiz quadrada dos valores.

[Home](#) > [Módulo de Estratégias](#) > Back-Testing

Back-Testing

A funcionalidade de Back-testing permite ao usuário avaliar uma determinada estratégia, teoria ou modelo através de uma análise de dados históricos.

Lista de funcionalidades utilizadas para Back-Testing:

- [Lista de funções](#);
- [Criar regra de execução](#);
- [Execução da estratégia](#)

Após criada a estratégia de Back-Testing, para ser adicionado diretamente no gráfico, clique no botão direito sobre o mouse e selecione a opção "Inserir Regra de Execução".

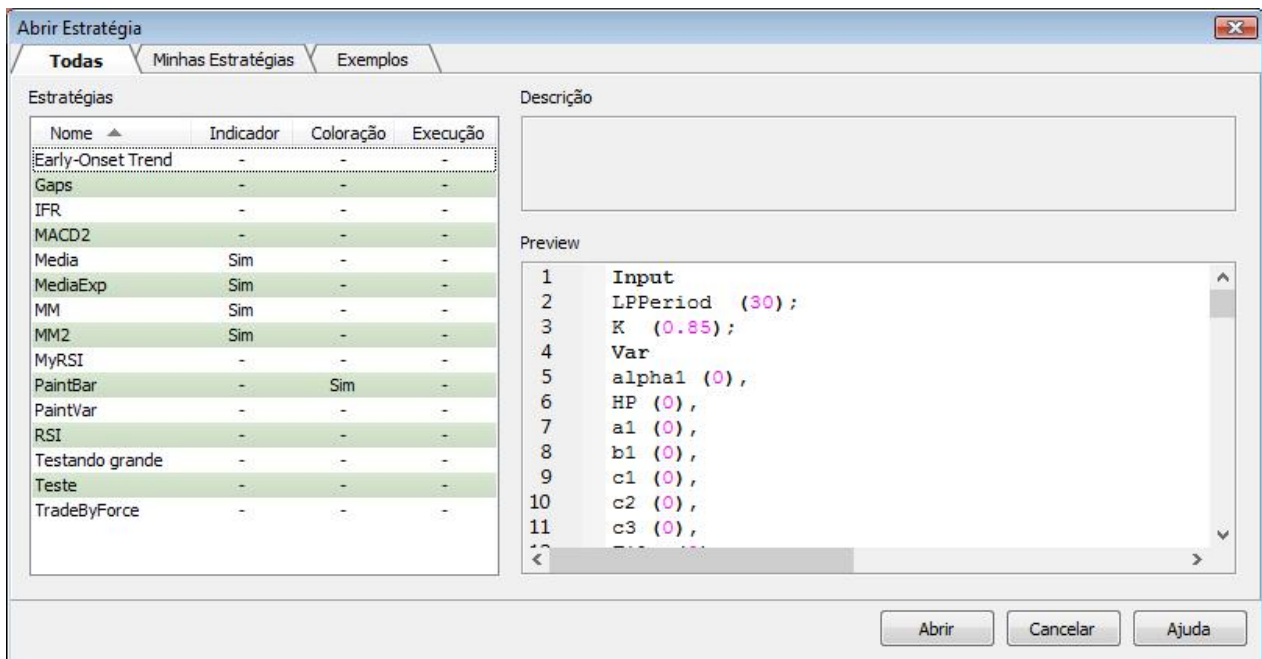
[Home](#) > [Módulo de Estratégias](#) > Abrir Estratégias

Abrir Estratégias

Na opção de "Abrir Estratégias", o usuário terá acesso a três abas, elas são:

- **Todas:** O usuário poderá ver todas as estratégias dentro do seu ProfitChart;
- **Minhas Estratégias:** O usuário irá filtrar para somente exibir todas as estratégias criadas por ele dentro do ProfitChart;
- **Exemplos:** O usuário irá filtrar para exibir exemplos de estratégias que já vem como padrão no ProfitChart.

Além das abas, o usuário também poderá pré-visualizar o seu código de estratégia para confirmar informações.

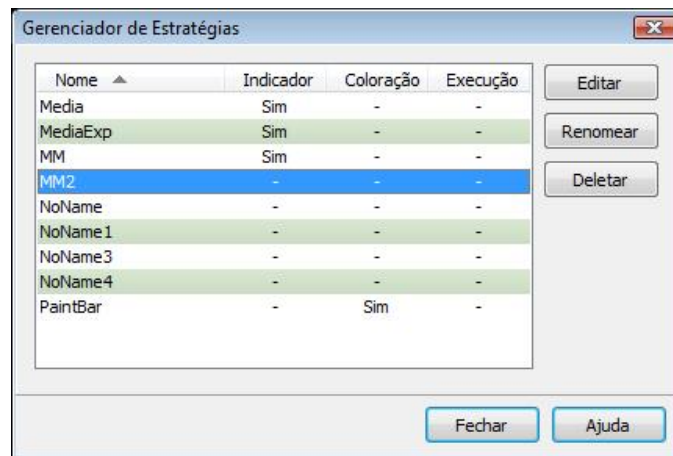


[Home](#) > [Módulo de Estratégias](#) > Gerenciador de Estratégias

Gerenciador de Estratégias

A opção de gerenciador de estratégias, permite ao usuário escolher uma determinada estratégia criada para edição, fazendo com que o Editor de estratégias carregue a estratégia determinada, ao clicar em "Editar".

O usuário também poderá excluir as estratégias desejadas, selecionando as mesmas e clicando no botão "Excluir", além da funcionalidade de renomear a estratégia através do botão "Renomear".

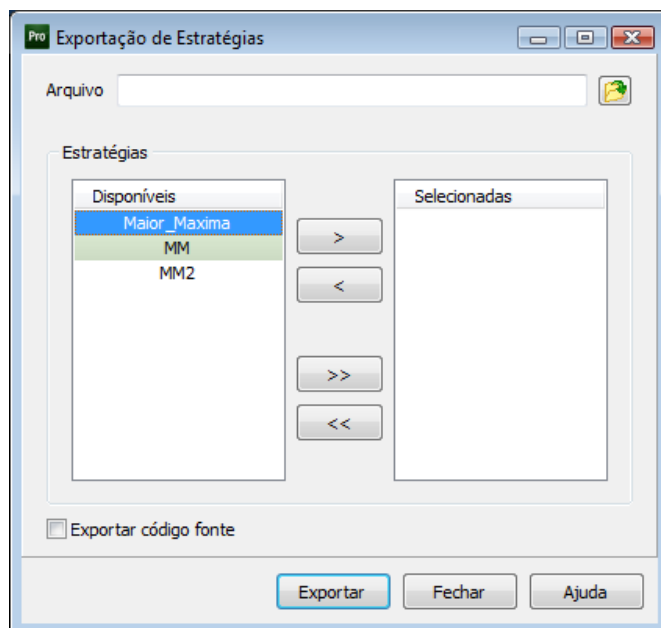


[Home](#) > [Módulo de Estratégias](#) > Exportar/Importar Estratégia

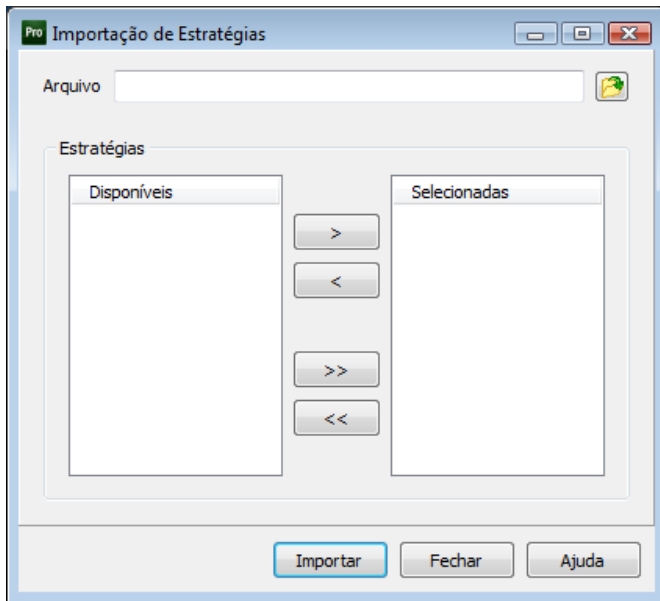
Exportar/Importar Estratégias

Nesta funcionalidade, permite ao usuário exportar as estratégias criadas por eles para que possam ser importadas novamente.

O usuário também tem a possibilidade de querer exportar o código fonte da estratégia ou apenas o arquivo executável.



Na importação, o usuário tem a funcionalidade de escolher quais estratégias serão carregadas e adicionadas junto ao ProfitChart.



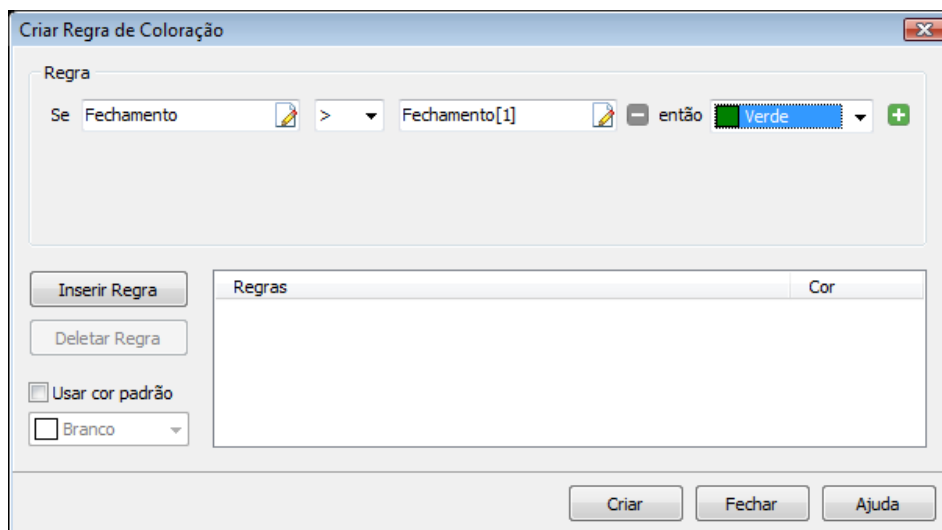
[Home](#) > [Módulo de Estratégias](#) > Criar Regra de Coloração

Criar Regra de Coloração

A funcionalidade de criar regra de coloração, mostra de forma visual, como criar uma regra de coloração com as condições desejadas.

A cada vez que for clicado no botão "+" irá criar uma condição para que se Condição 1 e Condição 2 sejam verdadeiras irá colorir de acordo com a cor desejada, na cor padrão será se caso as condições não retornem verdadeiro irá pintar em determinada cor.

Caso o usuário deseje utilizar outras informações, ele irá poder clicar no botão "Mais" que se encontra ao lado da variável para selecionar outras condições.



[Home](#) > [Módulo de Estratégias](#) > Criar Regra de Alarme

Criar Regra de Alarme

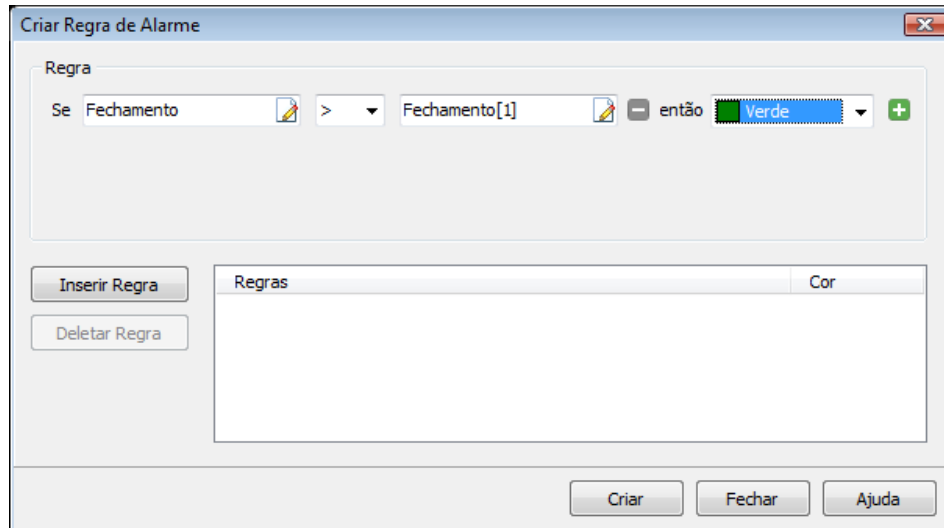
A funcionalidade do Criar Regra de Alarme mostra de forma visual, como criar uma regra de alarme de acordo com as condições desejadas.

A cada vez que for clicado no botão "+" irá criar uma condição para que se Condição 1 e Condição 2 sejam verdadeiras irá acionar o alarme desejado e irá colorir o pop-up de alarme de acordo com a cor estabelecida.

Caso o usuário deseje utilizar outras informações, ele irá poder clicar no botão "Mais" que se encontra ao lado da variável para selecionar outras condições.

Após de criada a regra de alarme, para ser acionado se deve ir no menu Ferramentas>Alarmes>Alarmes>Alarmes de estratégias, que

pode ser visualizado o procedimento clicando [aqui](#).



[Home](#) > [Módulo de Estratégias](#) > Criar Regra de Execução

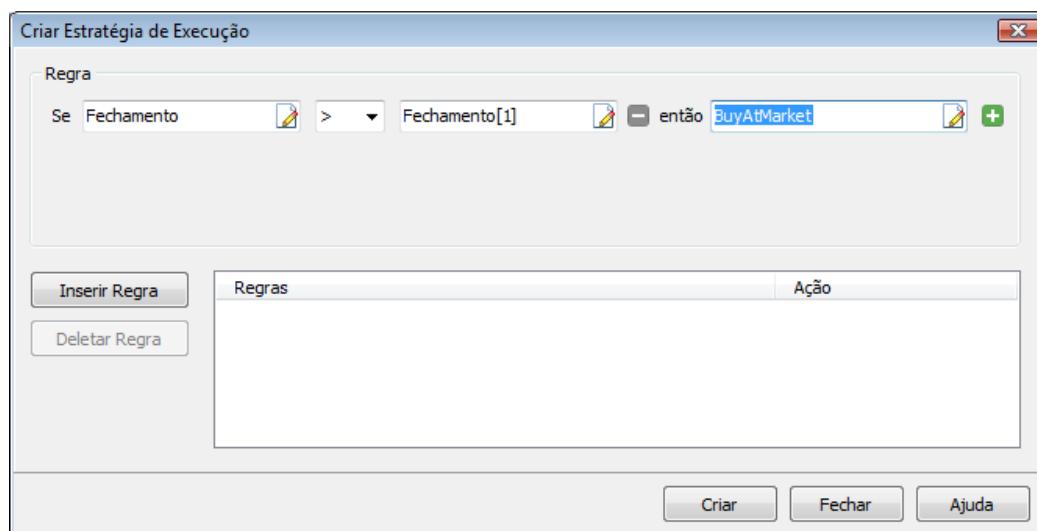
Criar Regra de Execução

A funcionalidade do Criar Regra de Execução mostra de forma visual, como criar uma regra de execução de acordo com as condições desejadas.

A cada vez que for clicado no botão "+" irá criar uma condição para que se Condição 1 e Condição 2 sejam verdadeiras irá acionar o alarme desejado e irá colorir o pop-up de alarme de acordo com a cor estabelecida.

Caso o usuário deseje utilizar outras informações, ele irá poder clicar no botão "Mais" que se encontra ao lado da variável para selecionar outras condições.

Para saber qual tratamento deve ser dado quando a condição for satisfeita, as funções de execução podem ser vistas [aqui](#).



[Home](#) > [Módulo de Estratégias](#) > Screening

Screening

A funcionalidade de Screening mostra de forma visual, os ativos que se encontram na base de dados e que satisfazem as condições da estratégia.

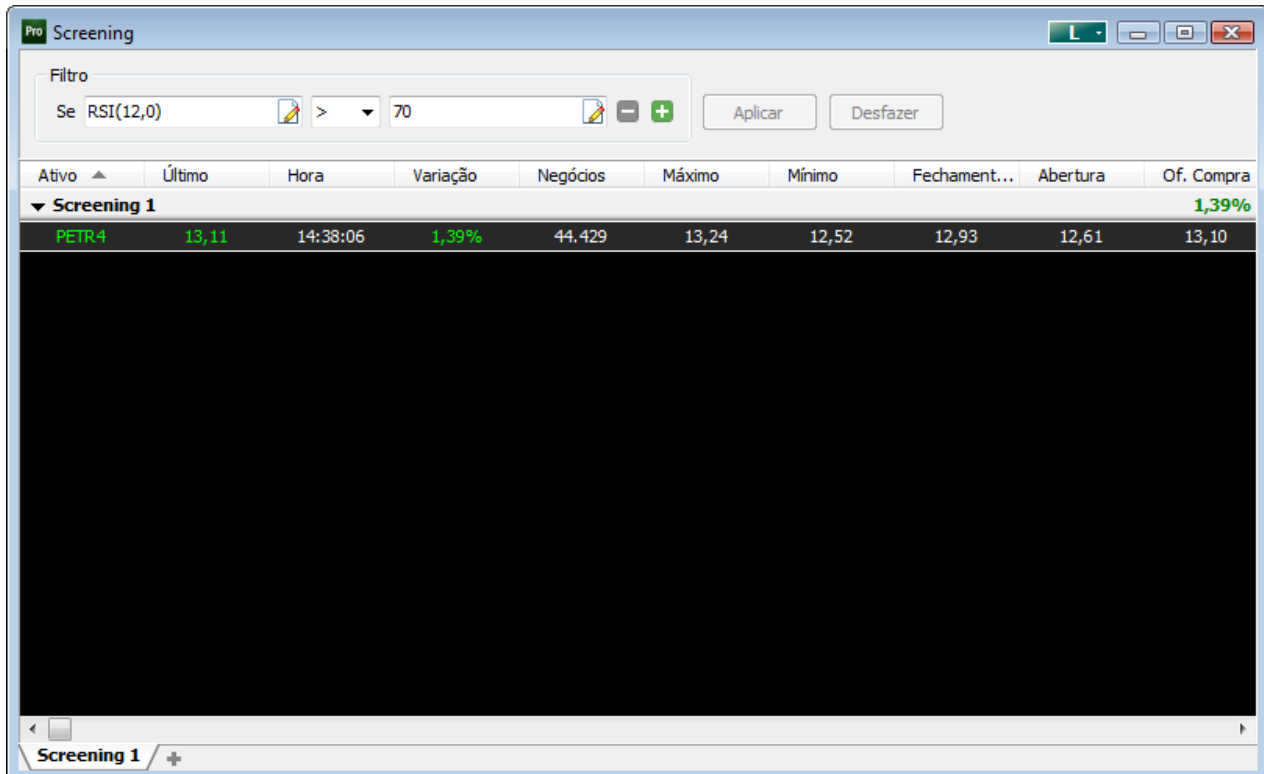
A cada vez que for clicado no botão "+" irá criar uma condição para que se Condição 1 e Condição 2 sejam verdadeiras irá mostrar o ativo dentro da aba selecionada.

Caso o usuário deseje utilizar outras informações, ele irá poder clicar no botão "Mais" que se encontra ao lado da variável para selecionar outras condições.

Ao clicar no botão "Aplicar" a estratégia criada é aplicada a grade e irá mostrar os ativos que satisfazem a condição.

Ao clicar no botão "Desfazer" a estratégia irá retornar para a última estratégia aplicada a grade.

Para maiores informações sobre o funcionamento de uma grade de cotações, clique [aqui](#).

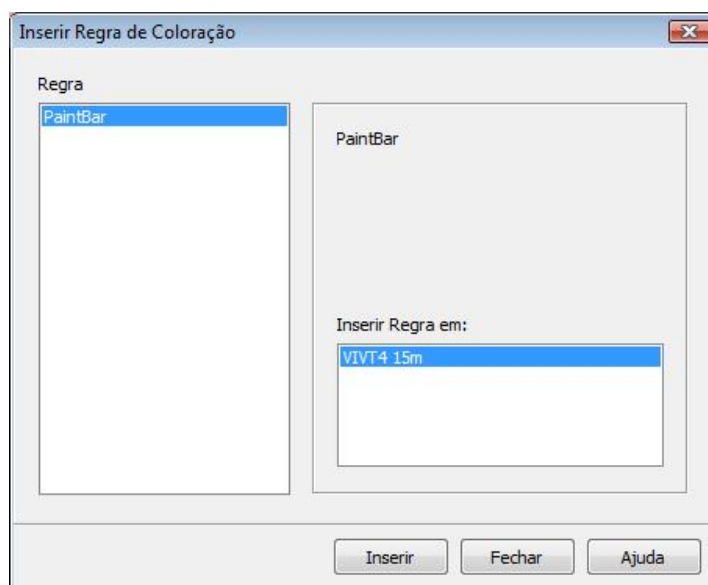


[Home](#) > [Módulo de Estratégias](#) > Inserir Regra de Coloração

Inserir Regra de Coloração

A funcionalidade de inserir regra de coloração permite ao usuário colorir o gráfico de acordo como ele determinou os parâmetros nos gráficos dos ativos.

As regras de coloração serão feitas seguindo a ideologia de um indicador ou seja, respeitando o layout atual da janela.

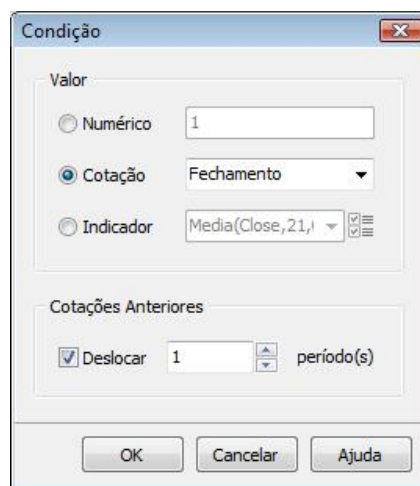


[Home](#) > [Módulo de Estratégias](#) > Condições de Coloração

Condições de Coloração

As condições de coloração permitem ao usuário, quais valores ele irá utilizar no momento em que irá criar uma nova regra de coloração, elas podem ser:

- Numérico: O usuário poderá utilizar números inteiros ou reais;
- Cotação: O usuário poderá utilizar os valores presente nas cotações, sendo elas: Abertura, Máxima, Mínima, Fechamento, Quantidade;
- Indicador: O usuário poderá utilizar os valores presentes nos indicadores criados por ele, e alterar os parâmetros presentes para de acordo com a vontade para coloração;
- Cotações Anteriores: Permite ao usuário utilizar os valores presentes nas cotações anteriores, conforme mostra na guia [Variáveis e séries de dados](#).



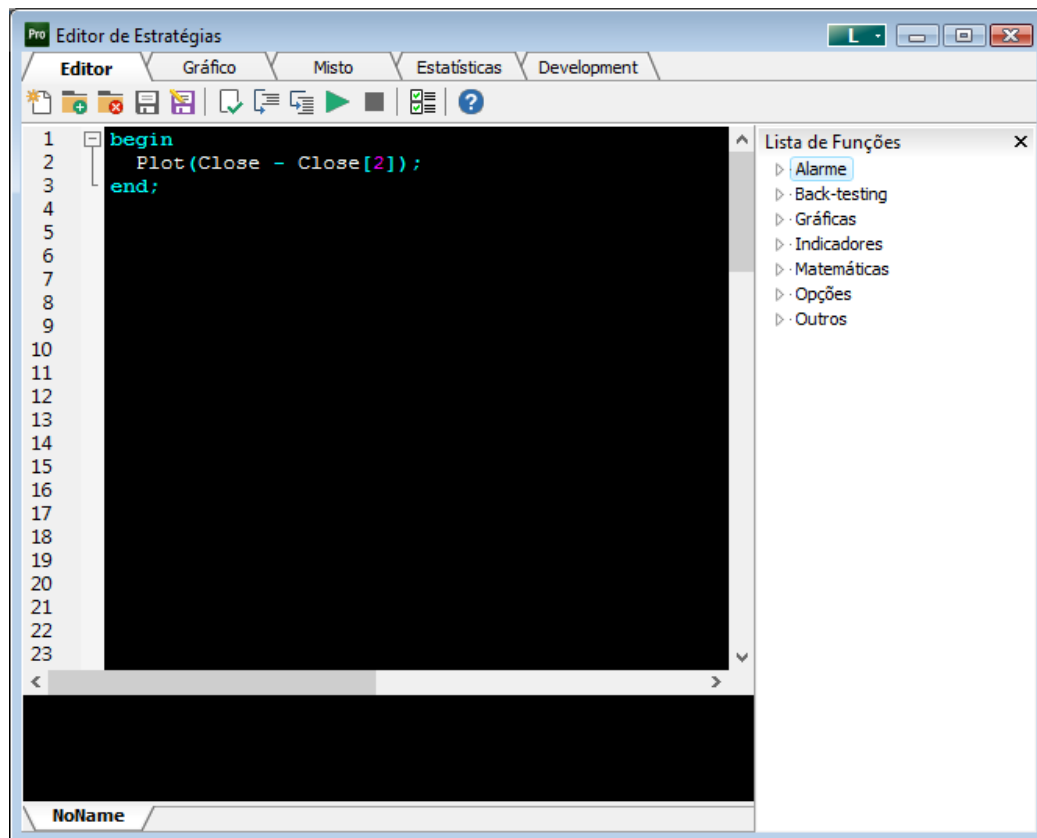
[Home](#) > [Módulo de Estratégias](#) > Editor de Estratégias

Editor de Estratégias











A janela de Editor de estratégias é onde o usuário irá poder criar suas próprias estratégias juntamente com a funcionalidade de visualizar diretamente no gráfico a estratégia criada.


O Editor de Estratégias possui três abas:

- Editor: Onde o usuário escreve a estratégia seguindo as instruções da linguagem NTSL.
-
- Gráfico: Onde o usuário visualiza a estratégia após executada diretamente no gráfico;
-
- Misto: É onde o usuário visualiza as informações da aba Editor e a aba Gráfico em uma só, onde a cada vez que ele execute o código, já irá aparecer diretamente no gráfico.



Dentro do Editor de estratégias o usuário irá possuir as seguintes opções:

-  Nova Estratégia: Onde o usuário irá poder criar uma nova estratégia;
-  Abrir Estratégia: Permite ao usuário abrir estratégias já criadas e edita-las
-  Fechar: Fecha a aba da estratégia atual;
-  Salvar: Salva a estratégia atual;
-  Salvar Como: Salva a estratégia atual, podendo ser adicionado uma descrição da mesma;
-  Verificar Sintaxe: Realiza a leitura do código verificando se há erros e a transforma em uma estratégia executável;
-  Trace: Mostra passo-a-passo o que o código da estratégia está realizando e mostrando os valores naquele momento;
-  Trace Into: Semelhante ao Trace, mostra passo a passo o que a estratégia está realizando no momento de criação, porém, quando há funções no código ele irá abri a função para mostrar ao usuário que a função está executando;
-  Executar: Após apertar o botão "Compilar", o botão executar irá executar a estratégia criada e a mostra no gráfico;
-  Parar: Tem como funcionalidade parar a estratégia para que o usuário pare a execução da estratégia criada;

-  **Propriedades:** Tem como funcionalidade mostrar as propriedades que irão constituir a estratégia, como desenho no gráfico, linhas guias e escala;
- **Lista de Funções:** Mostra a lista de funções disponíveis para a construção de estratégias, sendo possível copiar o nome da função clicando com o botão direito do mouse sobre a função desejada e clicar no botão "Copiar".

Estatísticas

Nas guias dentro de Estatísticas o usuário irá visualizar as informações resultantes da estratégia que se encontra em execução no determinado ativo.

- **Resumo:** Mostra o resumo das operações resultantes das estratégias, utilizando a série de dados;
- **Operações:** Mostra o total de operações fechadas foram realizadas utilizando a estratégia;
- **Gráfico de operações:** Mostra as operações de forma gráfica demarcando os prejuízos e lucros juntamente com o montante da operação;
- **Patrimônio:** Mostra o patrimônio total adquirido ao longo da escala de tempo das operações.

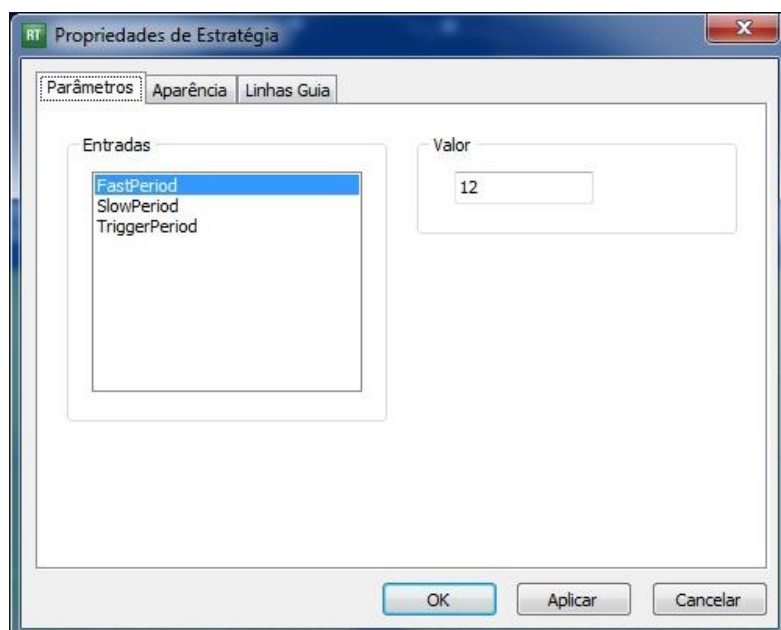
Para maiores informações sobre os dados presentes, você pode visualizar através do [Relatório de performance](#).

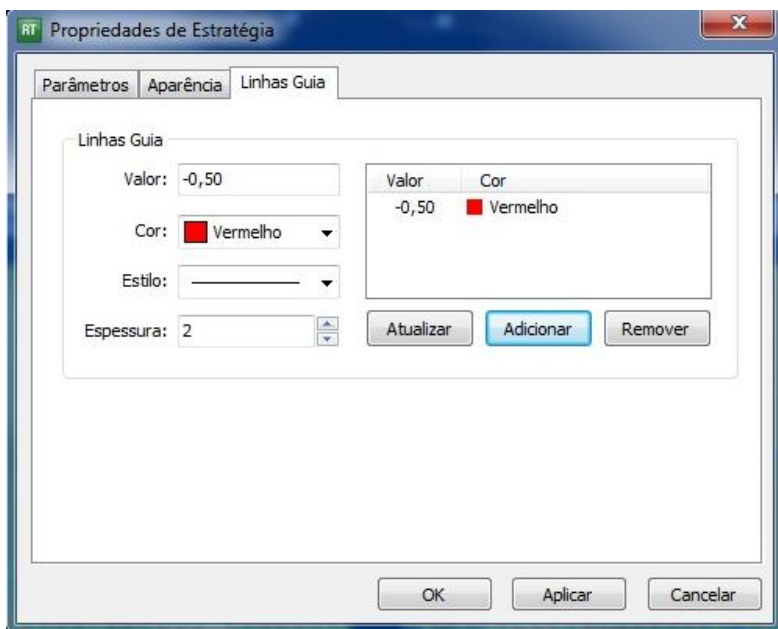
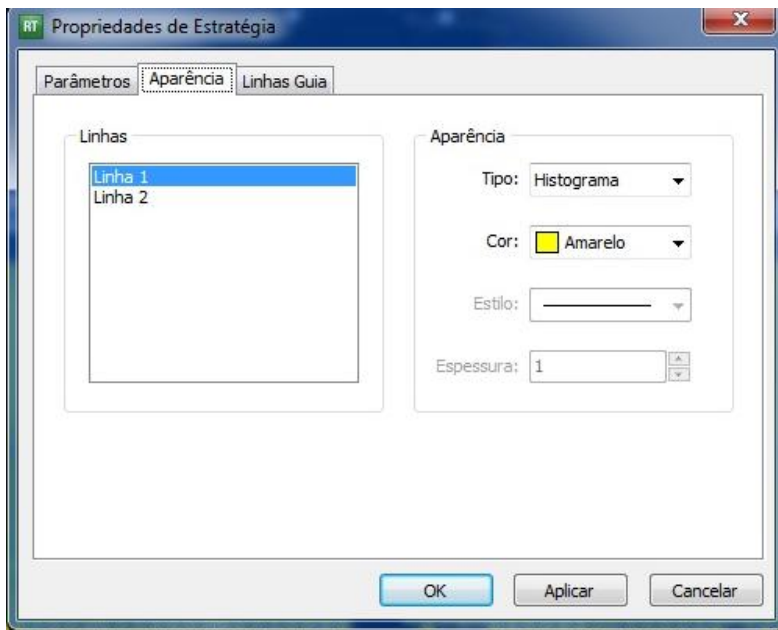
[Home](#) > [Módulo de Estratégias](#) > Propriedades do Editor de Estratégias

Propriedades do Editor de Estratégias

Nas propriedades do Editor de estratégia o usuário irá poder utilizar valores e informações adicionais no momento da criação da estratégia, elas são:

- **Parâmetros:** Permite ao usuário utilizar estratégias já criadas como parâmetros para uma nova estratégia junto com o valor desejado para a mesma;
- **Aparência:** Permite ao usuário determinar se deseja que a estratégia seja mostrada em linha ou em histograma;
- **Linhas Guia:** Permite ao usuário criar linhas para se basear como exemplos de linha de suporte e resistência.





[Home](#) > [Módulo de Estratégias](#) > Nova Estratégia

Nova Estratégia

Ao clicar no botão de Nova Estratégia, o usuário poderá escolher entre uma das três opções abaixo:

- Em branco: Ao escolher desta maneira, o usuário irá criar uma estratégia em branco;
- Indicador: Ao abrir a estratégia, irá carregar um exemplo de estratégia de indicador para o usuário;
- Coloração: Ao abrir a estratégia, irá carregar um exemplo de regra de coloração para o usuário.

