

Função CurrentDate A função CurrentDate possui como finalidade retornar a data atual do sistema. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: CurrentDate Parâmetros: Sem parâmetros. Retorno: Integer A função CurrentTime possui como finalidade retornar a hora atual do sistema. formato: HHMM. Observação: Horas são representadas pelo tipo de dado "Integer" Sintaxe: Current Parâmetros: Sem parâmetros. Retorno: Integer A função Date possui como finalidade retornar a data do candle que está sendo analisado. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: Date Parâmetros: Sem parâmetros. Retorno: Integer Função DayOfMon: A função DayOfMonth retorna o dia do mês de uma data específica. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: DayOfMonth(Data : Integer) Parâmetros: Data: Data para obter o dia. Retorno: Integer Função DayOfWeek A função DayOfWeek retorna o dia da semana de uma data específica. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: DayOfWeek(Data : Integer) Parâmetros: Data para obter o dia da semana. Retorno: Integer: Referência de dias da semana: 0 - Domingo 1- Segunda 2-Terça 3- Quarta 4- Quinta 5 - Sexta 6 - Sábado Função DaysToExpiration DaysToExpiration é uma função a quantidade de dias úteis restantes até a terceira sexta-feira de um determinado mês e ano. Sintaxe: DaysToExpiration(Mes : Integer, Ano : Integer) Parâmetros: Mes: Determina o mês que se deseja a informação: 1-Janeiro 2- Fevereiro 3-Março 4- Abril 5-Maio 6 - Junho 7 - Julho 8 - Agosto 9 - Setembro 10 - Outubro 11 - Novembro 12- Dezembro Ano: Determina o ano desejado para análise, onde deverá estar no formato: 1AnoDesejado. Retorno: Integer No exemplo abaixo, a variável "nQtdDias" irá receber a quantidade de dias úteis até a terceira sexta-feira de dezembro/2016. Função ELDate A função ELDate possui como finalidade retorna uma data em EasyLanguage format(YYYYMMDD). Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: ELDate(Ano : Integer, Mes : Integer, Dia : Integer) Parâmetros: Ano: Ano no formato YYYY. Mes: Mes no formato MM. Dia: Dia no formato DD. Retorno: Integer Função ELDate Consol A função ELDate Consol possui como finalidade converter uma data YYYYMMDD em EasyLanguage format(YYYYMMDD). Sintaxe: ELDate Consol(Data : Integer) Parâmetros: Data: Data no formato YYYYMMDD. Retorno: Integer Função FindBar A função FindBar retorna o índice de uma barra através de uma data e hora, onde a contagem é iniciada a partir do candle atual(índice 0). Observações: Representação de datas: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Representação de horas: Horas são representadas pelo tipo de dado "Integer", no formato: HHMM. Sintaxe: FindBar(Data : Integer, Hora : Integer) Parâmetros: Data: Data do candle. Hora: Hora do candle específico. Retorno: Integer Função Friday A função Friday retorna o número 5, representando o dia da semana: sexta-feira. Parâmetros: Sem parâmetros. Retorno: Integer Função HighD A função HighD tem como finalidade retornar o valor de máxima de um número determinado de dias atrás. Sintaxe: HighD(QuantidadeDias Anteriores : Integer) Parâmetros: QuantidadeDiasAn-

teriores:: — Determina a quantidade desejada de dias anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "dMaxima" o valor de máxima de dois dias anteriores ao dia atual. Função Hig A função HighM tem como finalidade retornar o valor de máxima de um número determinado de meses atrás. Sintaxe: HighM(QuantidadeMesesAnteriores — : Integer) Parâmetros: QuantidadeMesesAnteriores: Determina a quantidade desejada de meses anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "mMaxima" o valor de máxima de cinco meses anteriores ao mês atual. Função HighW A função HighW tem como finalidade retornar o valor de máxima de um número determinado de semanas atrás. Sintaxe: HighW(QuantidadeSemanasAnteriores : Parâmetros: QuantidadeSemanasAnteriores: Determina a quantidade desejada de semanas anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "wMaxima" o valor de máxima de sete semanas anteriores à semana atual. Função HighY A função HighY tem como finalidade retornar o valor de máxima de um número determinado de anos atrás. Sintaxe: HighY(QuantidadeAnosAnteriores : Integer) Parâmetros: QuantidadeAnosAnteriores: Determina a quantidade desejada de anos anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "yMaxima" o valor de máxima de dois anos anteriores ao atual. Função LastCalcDate A função LastCalcDate retorna a data do último candle completo formado dentro do gráfico. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: LastCalcDate Parâmetros: Sem parâmetros. Retorno: Integer No exemplo abaixo, será colocado diretamente no gráfico a data dos candles de acordo com o andamento do mercado. Função LastCalcTime A função LastCalc] formato 24h(HHMM). 1e retorna a hora do último candle completo formado dentro do gráfico, no Sintaxe: LastCalcTime Parâmetros: Sem parâmetros. Retorno: Integer No exemplo abaixo, será colocado diretamente no gráfico a hora de fechamento dos candles, de acordo com o andamento do mercado. Função LastDayOfMont A função LastDayOfMoth retorna o último dia do mês de referência. Sintaxe: LastDayOfMonth(MesReferencia : Integer) Parâmetros: Determina o mês de referência, de Janeiro(1) a Dezembro(12). MesReferencia Retorno: Integer No exemplo abaixo, será atribuído à variável "nDia" o último dia(30) do mês de Setembro. Função LowD A função LowD tem como finalidade retornar o valor de mínima de um número determinado de dias atrás. Sintaxe: LowD(QuantidadeDiasAnteriores : Integer) Parâmetros: QuantidadeDiasAnteriores:: — Determina a quantidade desejada de dias anteriores Retorno: Float No exemplo a seguir, será atribuído à variável "dMinima" o valor de mínima de três dias anteriores ao dia atual. Função LowM A função LowM tem como finalidade retornar o valor de mínima de um número determinado de meses atrás. Sintaxe: LowM(QuantidadeMesesAnteriores — : Integer) Parâmetros: QuantidadeMesesAnteriores: Determina a quantidade desejada de meses anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "mMinima" o valor de mínima do mês anterior. Função LowW A função LowW tem como finalidade retornar o valor de mínima de um número determinado de semanas atrás. Sintaxe: LowW[QuantidadeSemanasAnteriores : Integer) Parâmetros: QuantidadeSemanasAnteriores: Determina a quantidade desejada de semanas

anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "wMinima" o valor de mínima de sete semanas anteriores à semana atual. Função LowY A função LowY tem como finalidade retomar o valor de mínima de um número determinado de anos atrás. Sintaxe: LowY(QuantidadeAnosAnteriores : Integer) Parâmetros: QuantidadeAnosAnteriores: — Determina a quantidade desejada de anos anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "yMinima" o valor de mínima de dois anos anteriores ao atual. Função Monday A função Monday retorna o número 1, representando o dia da semana: segunda-feira. Sintaxe: Monday Parâmetros: Sem parâmetros. Retorno: Integer A função Month retoma o mês de uma data específica. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: Month(Data : Integer) Parâmetros: Data: Data para obter o mês. Retorno: Integer Função Next3rdFriday A função Next3rdFriday retoma a quantidade de dias úteis restantes até a terceira sexta-feira de determinado mês subsequente. Sintaxe: Next3rdFriday(Mes : Integer) Parâmetros: Mes: Determina o mês que se deseja a informação, onde a contagem inicia-se a partir do mês atual(0). Retorno: Integer No exemplo abaixo, a variável "nSexta" irá receber a quantidade de dias úteis até a terceira sexta-feira do mês seguinte. Função OpenD A função OpenD tem como finalidade retornar o valor de abertura de um número determinado de dias atr Sintaxe: OpenD(QuantidadeDiasAnteriores — : Integer) Parâmetros: QuantidadeDiasAnteriores:: — Determina a quantidade desejada de dias anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "dAbertura" o valor de abertura de seis dias anteriores ao dia atual. Função OpenM Descrição A função OpenM tem como finalidade retornar o valor de abertura de um número determinado de meses atrás. Sintaxe: OpenM(QuantidadeMesesAnte Parâmetros: QuantidadeMesesAnteriores: Determina a quantidade desejada de meses anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "mAbertura" o valor de abertura do mês anterior ao atual Função OpenW A função OpenW tem como finalidade retornar o valor de abertura de um número determinado de semanas atrás. Sintaxe: "OpenW(QuantidadeSemanasAnteriores : Integer) Parâmetros: QuantidadeSemanasAnteriores: Determina a quantidade desejada de semanas anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "wAbertura"" o valor de abertura de cinco semanas anteriores à semana atual. Função OpenY A função OpenY tem como finalidade retomar o valor de abertura de um número determinado de anos atrás. Sintaxe: OpenY(QuantidadeAnosAnteriores Parâmetros: QuantidadeAnosAnteriores: — Determina a quantidade desejada de anos anteriores. Retorno: Float Sintaxe: RS BarsPerDay Parâmetros: Sem parâmetros. Retorno: Inteiro Função Saturday A função Saturday retorna o número 6, representando o dia da semana: sábado. Sintaxe: Saturday Parâmetros: Sem parâmetros. Retorno: Integer Função Sunday A função Sunday retoma o número 0, representando o dia da semana: domingo. Sintaxe: Sunday Parâmetros: Sem parâmetros. Retorno: Integer Função Thursday A função Thursday retorna o número 4, representando o dia da semana: quinta-feira. Sintaxe: Thursday Parâmetros: Sem parâmetros. Retorno: Integer A função Time possui como

finalidade retornar a hora de abertura do candle analisado. Observação: Horas são representadas pelo tipo de dado "Integer", no formato: HHMM. Sintaxe: Time Parâmetros: Sem parâmetros. Retorno: Integer Função TimeToMinutes A função TimeToMinutes possui como finalidade efetuar a conversão de um horário em minutos. Sintaxe: TimeToMinutes(Hora : Integer) Parâmetros: Hora para a conversão. Retorno: Integer Função Tuesday A função Tuesday retorna o número 2, representando o dia da semana: terça-feira. Sintaxe: Tuesday Parâmetros: Sem parâmetros. Retorno: Integer Função VolumeD A função VolumeD tem como finalidade retornar o volume financeiro de um número determinado de dias atrás. Sintaxe: VolumeD(QuantidadeDiasAnteriores Parâmetros: QuantidadeDiasAnteriores:: — Determina a quantidade desejada de dias anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "dVolume" o volume de seis dias anteriores ao dia atual. Função VolumeM A função VolumeM tem como finalidade retornar o volume financeiro de um número determinado de meses atrás. Itaxe: VolumeM(QuantidadeMesesAnteriores : Integer) Parâmetros: QuantidadeMesesAnteriores: Determina a quantidade desejada de meses anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "mAbertura o volume do mês anterior ao atual. Função VolumeW A função VolumeW tem como finalidade retornar o volume financeiro de um número determinado de semanas atrás. Sintaxe: VolumeW(QuantidadeSemanasAnteriores — : Integer) Parâmetros: QuantidadeSemanasAnteriores: Determina a quantidade desejada de semanas anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "wVolume" o valor volume de cinco semanas anteriores à semana atual. Função VolumeY A função VolumeY tem como finalidade retornar o volume financeiro de um número determinado de anos atrás. Sintaxe: VolumeY(QuantidadeAnosAnteriores : Integer) Parâmetros: QuantidadeAnosAnteriores: — Determina a quantidade desejada de anos anteriores. Retorno: Float No exemplo a seguir, será atribuído à variável "yVolume" o volume do ano anterior ao atual. Função Wednesday Descrição A função Wednesday retoma o número 3, representando o dia da semana: quarta-feira. Sintaxe: Wednesday Parâmetros: Sem parâmetros. Retorno: Integer A função Year retorna o ano de uma data específica. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: Year(Data : Integer) Parâmetros: Data: Data para obter o ano. Retorno: Integer Candlestick Função C 3WhiteSoldiers 3BlackCrows A função C 3WhiteSoldiers 3BlackCrows identifica a ocorrência de dois tipos de candles: 3 White Soldiers e 3 Black Crows Sintaxe: C 3WhiteSoldiers 3BlackCrows(Comprimento : Integer, Fator : Integer, var o3WhiteSoldiers var o3BlackCrows : Integer) Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. Fator: Determina quantas vezes a sombra do candle deve ser maior que o seu corpo. .o3WhiteSoldiers: Variável para identificação de padrão(3 White Soldiers). o3BlackCrows: Variável para identificação de padrão(3 Black Crows). Retorno: Integer: Identificação(retorno função): 0 - Algum dos padrões foi identificado. 1- Nenhuma padrão identificado. Identificação(retorno variável: o3WhiteSoldiers): 0 - Padrão 3 White Soldiers não foi identificado. 1- Padrão 3 White Soldiers identificado. Identificação(retorno variável: o3BlackCrows):

0 - Padrão 3 Black Crows não foi identificado. 1- Padrão 3 Black Crows identificado. considerando 9(Comprimento) e 2(Fator), será aplicada uma coloração(vermelha). Função C BullEng BearEn: A função C BullEng BearEng identifica a ocorrência de dois tipos de candles: Bullish Engulfing e Bearish Engulfing Sintaxe: BullEng BearEng(Comprimento oBearishEngulfing : Integer) Integer, var oBullishEngulfing: Integer, var Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. oBullishEngulfing: Variável para identificação de padrão(Bullish Engulfing). oBearishEngulfing: Variável para identificação de padrão(Bearish Engulfing). Retorno: Integer: Identificação(retorno função): 0 - Algum dos padrões foi identificado. 1- Nenhuma padrão identificado. Identificação(retorno variável: oBullishEngulfing): 0 - Padrão Bullish Engulfing não foi identificado. 1 Padrão Bullish Engulfing identificado. Identificação(retorno variável: oBearishEngulfing): 0 - Padrão Bearish Engulfing não foi identificado. 1- Padrão Bearish Engulfing identificado. aplicada uma coloração(amarela). Função C Doji Descrição: A função C. Doji identifica a ocorrência de um candle tipo Doji Sintaxe: DojiPercentual : Integer) Parâmetros: Percentual: Limiar para(Abertura - Fechamento) que seria uma percentagem do intervalo do candle. Retorno: Integer: Identificação: 0 - Padrão não identificado. 1- Padrão identificado. coloração(verde). Função C Hammer HangingMa! A função C Hammer. HangingMan identifica a ocorrência de dois tipos de candles: Hammer e Hanging Sintaxe: Hammer HangingMan(Comprimento : Integer, Fator : Integer, var oHammer oflangingMan : Integer) integer, var Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. Fator: Determina quantas vezes a sombra do candle deve ser maior que o seu corpo. Hammer: Variável para identificação de padrão(Hammer). HangingMan: Variável para identificação de padrão(Hanging Man). Retorno: Integer: Identificação(retorno função): 0 - Algum dos padrões foi identificado. 1- Nenhuma padrão identificado. Identificação(retorno variável: oHammer): 0 - Padrão Momning Hammer não foi identificado. 1- Padrão Moming Hammer identificado. Identificação(retorno variável: oHangingMan): Entrada input 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. x x * Defineos 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 valor2. 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 VARIÁVEL: TIPO Exemplo: Variável Definição sResult : Float; Cria uma varável chamada sResult do tipo ponto flutuante. nIndex:

Integer; Cria uma variável chamada nindex para armazenar números inteiros. 0 - Padrão Hanging Man não foi identificado. 1- Padrão Hanging Man identificado. 14(Comprimento) e 2(Fator), será aplicada uma coloração(amarela). Função C MornDoji EveDoj A função C MornDoji EveDoji identifica a ocorrência de dois tipos de candles: Morning Doji Star e Evening Doji Star Sintaxe: Momboji EveDojiComprimento oMlorningDojiStar : Integer, var oEveningDojiStar jer, Percentual : Float, var teger) Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. Percentual: Doj limiar para o (abrir - fechar) como uma percentagem do intervalo da barra. oMorningDojiStar: Variável para identificação de padrão(Moming Doji Star). oEveningDojiStar: Variável para identificação de padrão(Evening Doji Star). Retorno: Integer: Identificação(retorno função): 0 - Algum dos padrões foi identificado. 1- Nenhuma padrão identificado. Identificação(retorno variável: oMomingDojiStar): 0 - Padrão Morning Doji Star não foi identificado. 1- Padrão Moming Doji Star identificado. Identificação(retorno variável: oEveningDojiStar): 0 - Padrão Evening Doji Star não foi identificado. 1- Padrão Evening Doji Star identificado. considerando 9(Comprimento) e 3(Percentual), será aplicada uma coloração(branca). Função C MornStar EveStar A função C MornStar EveStar identifica a ocorrência de dois tipos de candles: Mo Star é Evening Star Sintaxe: MomsStar EveStar(Comprimento — : Integer, var MorningStar : Integer, var oEveningStar : Integer) Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. MorningStar: Variável para identificação de padrão(Morning Star). EveningStar: Variável para identificação de padrão(Evening Star). Retorno: Integer: Identificação(retorno função): 0 - Algum dos padrões foi identificado. 1 - Nenhuma padrão identificado. Identificação(retorno variável: oMorningStar): 0 - Padrão Morning Star não foi identificado. 1. Padrão Morning Star identificado. Identificação(retorno vari 0 - Padrão Evening Star não foi 1. Padrão Evening Star identificado. aplicada uma coloração(verde). Função PiercingLine DarkCloud A função PiercingLine DarkCloud identifica a ocorrência de dois tipos de candles: Piercing Line e Dark Cloud Sintaxe: PierceLine DarkCloud(Comprimento : Integer, var oPiercingLine : Ingeter, var oDarkCloud : Ingeter) Parâmetros: Comprimento: Tamanho utilizado para caleular a média do corpo do candle. PiercingLine: Variável para identificação de padrão(Piercing Line). DarkCloud: Variável para identificação de padrão(Dark Cloud). Retorno: Integer: Identificação(retomno função): 0 - Algum dos padrões foi identificado. 1- Nenhuma padrão identificado. Identificação(retorno variável: oPiercingLine): 0 - Padrão Piercing Line não foi identificado. 1- Padrão Piercing Line identificado. Identificação(retomno variável: oDarkCloud): 0 - Padrão Dark Cloud não foi ident uma coloração(amarela). Função C otingSt: A função C. ShootingStar identifica a ocorrência de candles tipo Shooting Star Sintaxe: ShootingStar(Comprimento : Integer, Fator : Integer) Parâmetros: Comprimento: Tamanho utilizado para calcular a média do corpo do candle. Fator: Determina quantas vezes a sombra do candle deve ser maior que o seu corpo. Retorno: Integer: Identificação: 0 - Padrão não identificado. 1- Padrão identificado. uma coloração(verde). Função DiMaisDiMenos O exemplo DiMaisDiMenos possui a implementação do indicador

DI-/DI- , onde foi determinado um parâmetro (input) para o período. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float Função IFR A função IFR retorna o valor(tipo clássico) do indicador IFR , de acordo com o período. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. Sintaxe: IFR(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. Função Media A função Media retoma o valor do indicador Média Móvel, tipo aritmética. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. Sintaxe: Media(Período : Integer, TipoSerie : Serie) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. TipoSerie: Série que será considerada para o cálculo. Retorno: Float Função MediaExp A função MediaExp retorna o valor do indicador Média Móvel, tipo exponencial. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. Sintaxe: MediaExp(Período : Integer, TipoSerie : Serie) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. TipoSerie: Série que será considerada para o cálculo. Retorno: Float Função PaintVar A função PaintVar possui a implementação de uma estratégia de coloração, a qual compara se o fechamento do último candle é positivo ou negativo em relação ao fechamento do(candle) anterior, e, como indicador, apenas retorna o último preço do ativo. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. Sintaxe: Paintvar Parâmetros: Sem parâmetros. ser aplicada sobre o indicador. Função WellesSum A função WellesSum retoma o valor do indicador WellesSum , de acordo com os parâmetros desejados. Observação: o exemplo com o código fonte está disponível no editor de estratégias, para visualizá-lo, acesse o menu: Abrir > exemplos”. | no editor de estratégias, para visualizá-lo, Sintaxe: WellesSum(Período, Integer, SerieReferencia) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. SerieReferencia: Série de dados. Offset: Referente ao parâmetro "Offset" do indicador. Retorno: Float série de fechamento(SerieReferencia) e 0(Offset) para o cálculo.

Gráficas A função AvgPrice retoma o valor da média entre Abertura, Fechamento, Máxima e Mínima, de determinado cande. Sintaxe: AvgPrice Parâmetros: Sem parâmetros Retorno: Float Será atribuído à variável *nAvg” o retorno da função AvgPrice. Função BarCount A função BarCount retoma a quantidade total de barras. Sintaxe: BarCount Parâmetros: Sem parâmetros. Retorno: Integer Função CurrentBar A função CurrentBar tem como finalidade retornar ao usuário o índice do candle atual. Sintaxe: CurrentBar Parâmetros: Sem parâmetros. Retorno: Integer Função GetPlotColor 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. Fluxo de Execução de uma Estratégia O código de uma estratégia é executado de

maneira sequencial. Porém, ele é executado sequencialmente candle por candle, como se houvesse um laço que percorre toda série de dados de um ativo. Sendo assim, a sequência de comandos existente na área de código será executada uma vez para cada candle existente. Para cada candle, será executado o código da estratégia em um loop implícito. À regra em pseudocódigo é, portanto:

- * Tendo uma série de dados X de tamanho N;
- * Paracada item X[i], com i variando de 0 até N-1, executa-se a área de código da estratégia;
- * A cada novo dado do ativo, o código da estratégia continua sendo executado, avaliando o último e o penúltimo candle;

Importante:

- * O mesmo candle pode ser processado diversas vezes. Isso ocorre muito em tempo real, onde cada trade pode gerar uma notificação de reprocessamento do fluxo de execução.
- * Quando uma ordem é executada, o fluxo de execução reprocessa o mesmo candle para ativar s ordens Covers.
- * Dependendo da notificação dos candles, podem ser reprocessados candles passados. Isso deve ser levado em consideração pelo programador.
- * Sempre que um candle passado é reprocessado, serão reprocessados todos candles até o atual.
- * Em tempo real, quando o código está executando ao mesmo tempo que recebe novos dados do mercado, o código da estratégia é executado diversas vezes para o último e o penúltimo candle durante a execução, podendo também executar em candles passados. Esses processamentos podem alterar valores de variáveis globais ou de série, e podem causar mudanças na coloração para candles passados ou mudança de comportamento quando o código é pausado e executado novamente. Quando estiver utilizando séries em sua estratégia, é importante lembrar que o primeiro tick do candle terá o valor [0] zerado. Como comentamos anteriormente, sua estratégia será executada não apenas no último candle e sim também ao passado, devido aos cálculos que precisam ser feitos para garantir que a NTSL funcione corretamente. É importante ressaltar que não se pode assumir que um candle no passado não será avaliado, porém, se desejar, é possível inibir o recálculo de algum valor utiizando a função LastBarOnChart() em uma condição no início do código para evitar isso. É fundamental A função GetPlotColor possui como funcionalidade retornar o valor numérico da cor de determinado Plot. Sintaxe: GetPlotColor(NúmeroPlot - : Integer) Parâmetros: NúmeroPlot: Número do Plot para obter a cor. Retorno: Integer
- Função GetPlotWidth A função GetPlotWidth possui como finalidade retomar o valor da espessura de determinado Plot. Sintaxe: GetpPlotwWidth(NúmeroPlot - : Integer) Parâmetros: NúmeroPlot: Número do Plot para obter a espessura. Retorno: Integer
- A função Graphicinterval retoma o intervalo do gráfico Sintaxe: Graphicinterval Parâmetros: Sem parâmetros. Retorno: Integer
- A função GraphicOffset retorna o offset do gráfico. Sintaxe: Graphicoffset Parâmetros: Sem parâmetros. Retorno: Integer
- Função HorizontalLine A função HorizontalLine tem como finalidade, adicionar um estudo horizontal em um indicador. Sintaxe: HorizontalLine (Y : Float; Color : Integer) Parâmetros: Y: Valor do estudo. Color: Cor da linha. Retorno: Sem retorno

No exemplo abaixo, será adicionada uma linha vermelha, no nível 5288. Função LastBarOnChart A função LastBarOnChart têm como função retomar um valor Booleano mostrando se o candle atual é o último candle do gráfico.

Sintaxe: LastBarOnChart Parâmetros: Sem parâmetros. Retorno: Boolean No exemplo a seguir, o candle atual será identificado com a coloração amarela.

Função Leader A função Leader retorna o valor de O quando o ponto médio for menor que a mínima anterior ou 1 quando o ponto médio for maior que máxima anterior. Parâmetros: Sem parâmetros Retorno: Integer No exemplo abaixo, caso a função Leader seja igual a um, o candle analisado será identificado pela cor verde.

Função MaxBarsBack A função MaxBarsBack tem como finalidade percorrer a lista da série, iniciando(índice 0) a partir do primeiro candle criado. Sintaxe: MaxBarsBack Parâmetros: Sem parâmetros. Retorno: Integer

Função MaxBarsForward A função MaxBarsForward tem como finalidade percorrer a lista da série, iniciando(índice 0) a partir do último candle criado(atual). Sintaxe: MaxBarsForward Parâmetros: Sem parâmetros. Retorno: Integer

Função MedianPrice A função MedianPrice retorna a média entre a máxima e a mínima de cada candle. Sintaxe: MedianPrice Parâmetros: Sem parâmetros. Retorno: Float No exemplo a seguir, caso o dado de MedianPrice do candle atual for maior que o do anterior, os candles serão destacados com a coloração verde.

Função NoPlot A função NoPlot tem como finalidade efetuar a remoção de determinado Plot. Sintaxe: NoPlot(NúmeroPlot : Integer) Parâmetros: NúmeroPlot: Número do Plot(1, 2, 3 e 4) para a remoção. Retorno: Void: Sem retorno. Conforme no exemplo a seguir, será removida a linha referente ao Plot, e será plotada somente a linha vinculada ao Plot2.

Função PaintBar A função PaintBar permite a aplicação de colorações, em indicadores ou candles. Sintaxe: PaintBar(Cor : Integer) Parâmetros: (Cor: Determina a coloração, podendo-se passar por parâmetro uma String ou a chamada da função RGB: clNomeCor - String para aplicação, conforme a seguinte lista: clBlack clMarrom clGreen clolive clNavy clPurple clTeal clGray clsilver clRed clLime cYellow clBlue clFuchsia claquia clwhite clMoneyGreen clSkyBlue RGB - Função para aplicação. Retorno: Float

Função Plot A função Plot realiza a ligação dos valores passados por parâmetro e cria gráficos de linhas. É possível efetuar a inserção de no máximo 99 linhas, onde deverá ser utilizada a função Plot numerada: Plot, Plot2, Plot3... Plot99. Sintaxe: Plot(Dado : Float); Plot2(Dado : Floal); Plot3(Dado : Float Plota9(Dado : Floal); Parâmetros: Dado: Pode-se utilizar variáveis, funções ou constantes para realizar o desenho do indicador. Retorno: Void: Sem retorno. Conforme no exemplo a seguir, são plotados os dados de fechamento do dia anterior e abertura do dia atual.

Função PlotN A função PlotN, assim como a Plot, desenha o indicador conforme a série de dados informada, porém o número do plot deve ser determinado por parâmetro (entre 0 e 99). Sintaxe: PlotN(Plot : Integer; Valor : Float) Parâmetros: Plot: Número do plot, entre 0 e 99. Dado: Pode-se utiizar variáveis, funções ou constantes para realizar o desenho do indicador. Retorno: Void: Sem retorno.

Função PlotText A função PlotText possui como finalidade, adicionar um texto a um indicador. Sintaxe: PlotText(Content : String Integer; FontSize : Integer; dPrice : Float) color : Integer; Positior Parâmetros: Content: Conteúdo do texto. 0 - Abaixo do candle 1- Centralizado 2- Acima do candle 3 - Posiciona no preço definido em dPrice FontSize: Tamanho da fonte do texto. dPrice: Quando utilizada Position=3, será utilizado o valor de dPrice para

posicionar o texto. Retorno: Void: Sem retorno. compreender que o [0] sempre está se referindo ao candle que está sendo avaliado, e não necessariamente será do último candle. Função Range A função Range retoma a diferença entre a máxima e a mínima do candle. Sintaxe: Range Parâmetros: Sem parâmetros. Retorno: Float Função RangeLeader A função RangeLeader verifica se a barra atual é Range Leader. Sintaxe: RangeLeader Parâmetros: Sem parâmetros Retorno: Float Função RGB A função RGB permite ao usuário customizar cores, a partir da aplicação de intensidade dos parâmetros vermelho, verde e azul. Sintaxe: RGB(Red : Integer, Green : Integer, Blue : Integer) Parâmetros: Red: Intensidade cor vermelha, variando de 0 a 255; Green: Intensidade cor verde, variando de 0 a 255; Blue: Intensidade cor azul, variando de 0 a 255. Retorno: Integer No exemplo a seguir, será aplicada a coloração azul, conforme os parâmetros de RGB. Função SetPlotColor A função SetPlotColor possui como finalidade alterar a coloração de determinado Plot(1 a 4) Sintaxe: SetPlotColor(NumeroPlot : Integer, Cor : Integer) Parâmetros: NumeroPlot: Número do Plot específico: 1-Plot 2- Plot2 3- Plot3 4- Plot4 Cor: Determina a coloração, podendo-se passar por parâmetro uma String ou a chamada da função RGB: clNomeCor - String para aplicação, conforme a seguinte lista: clBlack clMarrom clGreen clolive clNavy clPurple clTeal cleray clSiver clRed clLime clYellow clBlue clFuchsia claquia clwhite clMoneyGreen clSkyBlue RGB - Função para aplicação. Retorno: Void: Sem retorno. A partir da função SetPlotStyle, é possível alterar o estilo da linha de um plot específico. ntaxe SetPlotStyle(NumeroPlot : Integer; Estilo : Integer) etros. NumeroPiot: Número do Plot Estilo: Estio de linha 0- Contínua 1- Tracejada (Traço longo) 2- Tracejada (Traço curto) 3- Tracejada (Traço longo e curto) 4 - Tracejada (Traço longo e dois curtos) traços longos, e o segundo plot, com traços curtos. tPlotwidth Função SetPlotType A função SetPlotType possui como finalidade alterar o tipo de gráfico de determinado plot. Sintaxe: SetPlotType(Number : Integer; Type : Integer) Parâmetro: Number: Número do Plot específico Type: Tipo de visualização 0-Linha 1- Histograma Retorno: Void: Sem retorno. Função SetPlotWidth A função SetPlotWidth possui como finalidade alterar a espessura de determinado Plot(1 a 4) Sintaxe: SetPlotWidth(NumeroPlot : Integer, Espessura : Integer) Parâmetros: NumeroPiot: Número do Plot específi 1-Plot 2-Plot2 3-Plot3 4-Plot4 Espessura: Número para a nova espessura. Retorno: Void: Sem retorno. Função TrueHig A função TrueHigh retoma o maior entre o máximo da barra e o fechamento da barra anterior. Sintaxe: Truetigh Parâmetros: Sem parâmetros. Retorno: Float Função TrueLow A função TrueLow retoma o menor entre a mínima da barra e o fechamento da barra anterior. Sintaxe: TrueLow Parâmetros: Sem parâmetros. Retorno: Float Função TrueRange A função TrueRange retoma a diferença entre TrueHigh e TrueLow. Sintaxe: TrueRange Parâmetros: Sem parâmetros. Retorno: Float Função TrueRangeCustom A função TrueRangeCustom retoma o TrueRange de acordo com os dados informados pelo usuário. Sintaxe: TrueRangeCustom(Maxima : Float, Minima : Float, Fechamento : Float) Parâmetros: Maxima: Valor de máxima para o cálculo do indicador. Minima: Valor de mínima para o cálculo do indicador. Fechamento: Valor de fechamento

de referência. Retorno: Float Função TypicalPrice A função TypicalPrice retoma a média entre a máxima, mínima e fechamento do candle. Sintaxe: TypicalPrice Parâmetros: Sem parâmetros Retorno: Float Função VerticalLine A função VerticalLine adiciona um estudo vertical em um indicador. Sintaxe: VerticalLine(Y : Integer; Color : Integer) Parâmetros: Y: Data, com o formato 1A4AMDD Color: Cor da linha Retorno: Void: Sem retorno Função WeightedClose A função WeightedClose retorna a média entre o ponto médio da barra e dois fechamentos. Sintaxe: WeightedClose Parâmetros: Sem parâmetros Retorno: Float As variáveis são utilizadas para armazenar dados, conforme seu tipo específico. Float ou Real - Representa números de ponto flutuante. Integer ou Inteiro - Representa números inteiros. Boolean ou Booleano - Representação lógica: True ou Verdadeiro e False ou Falso Serie - Representa uma série de dados. As variáveis na linguagem oferecem uma flexibilidade muito maior do que na maioria dos sistemas de programação conhecidos. Todas as variáveis são globais, e deve ser dada uma atenção especial quando utilizadas. Ao programar usando variáveis globais, deve ser levado em consideração o comportamento do processamento do fluxo de execução, a cada vez que o fluxo de execução é realizado as variáveis podem ter seu valor Alterado com base no último valor que tinham. Indicadores Função AccAgressSaldo A função AccAgressSaldo retoma o valor do indicador TR - Acúmulo de Agressão - Saldo Sintaxe: AccAgressSaldo(TipoVolume : Integer) Parâmetros: Tipovolume: Tipo de volume para o cálculo: 0 - Financeiro 1- Quantidade 2- Negócios Retorno: Float Agressão - Saldo. A função AccuDistr retoma o valor do indicador Acumulação/Distribuição. Sintaxe: AccuDistr Parâmetros: Sem parâmetros. Retorno: Float Função AccuDistrW A função AccuDistrW retoma o valor do indicador Acumulação/Distribuição Williams Sintaxe: AccuDistrW Parâmetros: Sem parâmetros. Retorno: Float Função AdaptiveMovingAverage A função AdaptiveMovingAverage retoma o valor do indicador Adaptive Moving Average , de acordo “com o períodos específicos desejados. Sintaxe: AdaptiveMovingAverage(Período : Integer, FastSC : Integer, SlowSC : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. FastSC: Determina o período para o indicador FastStochastic. SlowSC: Determina o período para o indicador SlowStochastic. Retorno: Float 10 períodos para o cálculo, com 2 períodos para FastStochastic, e 30 para SlowStochastic. Função ADX A função ADX retoma o valor do indicador ADX , de acordo com os períodos desejados. Sintaxe: ADX(Período : Integer, PeríodoMedia : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. PeríodoMedia: Período utilizado no momento do cálculo da média utilizada no indicador. PeríodoMedia=9 para o cálculo. Função AgressionVolBalance A função AgressionVolBalance retorna o valor do indicador TR - Volume de Agressão - Saldo Sintaxe: AgressionVolBalance Parâmetros: Sem parâmetros. Retorno: Float Função AgressionVolBuy A função AgressionVolBuy retoma o valor do indicador TR - Volume de Agressão - Compra Sintaxe: AgressionVolBuy Parâmetros: Sem parâmetros. Retorno: Float Função AgressionVolSell A função AgressionVoiSell retorna o valor do indicador TR - Volume de Agressão - Venda Sintaxe: AgressionVolSell Parâmetros: Sem parâmetros. Retorno:

Float Função ArmsEaseOfMov A função ArmsEaseOfMov retoma o valor do indicador Arms Ease of Movement , de acordo com o período e tipo de média desejados. Sintaxe: ArmsEaseOfMov(Media : Integer, TipoMedia : Integer) Parâmetros: Media: Período da média utilizada no momento do cálculo do indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Retorno: Float 9 períodos para o tipo de média exponencial. Função AroonLin A função AroonLin retoma o valor do indicador Aroon Linha , de acordo com o período desejado. Sintaxe: AroonLin(Periodo : Integer)|Linha : Integer| Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0 - Aroon Up 1- Aroon Down Retorno: Float para o cálculo. Função Aroo A função AroonOsc retoma o valor do indicador Aroon Oscilador , de acordo com o período desejado. Sintaxe: AroonOsci(Periodo : Integer) Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. Retorno: Float períodos para o cálculo. Função AvgAgrBuySell A função AvgAgrBuySell retoma o valor do indicador TR - Agressão Média - Compra e Venda , de acordo com os parâmetros desejados. Sintaxe: AvgAgrBuySell(AlertaVariacoes : Integer, TipoVolume : Integer, TipoDesenho : Integer)|Linha : Integer| Parâmetros: AlertaVariacoes: Número de variações. TipoVolume: Determina qual tipo de volume será obtido: 0 - Financeiro 1- Quantidade TipoDesenho: Relação entre compra e venda: 0- Compra e Venda 1- CompralVenda 2- Compra-Venda Retorno: Float quantidade e tipo de desenho Função AvgAgrTotal A função AvgAgrTotal retoma o valor do indicador TR - Agressão Média - Total , de acordo com s parâmetros específicos. Sintaxe: AvgAgrTotal(AlertaVariacoes: Integer, TipoVolume : Integer, TipoDesenho : Integer)|Linha : Integer| Parâmetros: AlertaVariacoes: Número de variações. TipoVolume: Determina qual tipo de volume será obtido: 0 - Financeiro 1- Quantidade TipoDesenho: Relação entre compra e venda: 0 - Compra e Venda 1- CompralVenda 2- Compra-Venda Linha: Determina qual linha será obtida: 0 - Volume indicador 1-Avaliar Retorno: Float Função AvgaSeparation A função AvgSeparation retorna o valor do indicador Afastamento Médio , de acordo com o período e tipo de média desejados. Sintaxe: AvgSeparation(Periodo : Integer, TipoMedia : Integer) Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Retorno: Float considerando 21 períodos e tipo exponencial para o cálculo. Função AvaTrueRange A função AvgTrueRange retorna o valor do indicador True Range , de acordo com o período e tipo de média desejados. Sintaxe: AvgTrueRange(Periodo : Integer, TipoMedia : Integer) Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Welles Wikder 3- Ponderada Retorno: Float Quando uma variável for acessada utilizando colchetes (exemplo: [x]), ela será considerada uma série de dados pelo compilador. Seu comportamento é diferente: séries iniciam cada candle com seu valor Zero. Também é possível navegar entre valores atuais e passados, mediante indexação. As séries de dados são utilizadas para obter dados específicos de

cada Candle. Séries: Open ou Abertura - Retorna o valor de abertura de cada candle. Close ou Fechamento - Retorna o dado de fechamento. Low ou Mínima - Retorna o valor de mínima High ou Máxima - Retorna o dado de máxima. Quantity ou Quantidade - Retorna o valor de contratos negociados de cada candle Volume - Retorna o dado de volume financeiro de cada candle Análogo ao tipo Serie, a estrutura de um Array permite o armazenamento de dados de um tipo específico, contudo, após sua criação, seu comprimento será fixo, o que permite uma alocação menor de memória. O acesso aos elementos de um Array é efetuado mediante indexação. Estrutura da declaração: A fim de exemplificação, segue um exemplo de declaração de um Array estático de 10 elementos, o qual é preenchido com valores de 1 a 10 (primeira estrutura de repetição), e efetuado o somatório dos valores: e tipo de média ponderada para o cálculo. Função BalanceOfPower A função BalanceOfPower retorna o valor do indicador Balança do Poder, de acordo com o período desejado. Sintaxe: BalanceOfPower(Media : Integer, TipoMedia : Integer) Parâmetros: Media: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Retorno: Float períodos, e o tipo de média exponencial para o cálculo. Função BearPower A função BearPower retorna o valor do indicador Bear Power, conforme o período desejado. Sintaxe: BearPower(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float Função BollingerBands A função BollingerBands retorna o valor do indicador Bandas de Bollinger, de acordo com o período e tipo de média desejados. Sintaxe: BollingerBands(Desvio : Float, Media : Integer, TipoMedia : Integer)[Linha : Integer] Parâmetros: Desvio: Desvio utilizado no momento do cálculo do indicador. Media: Período da média utilizada no momento do cálculo do indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2-Welles Wilder 3- Ponderada Linha: Determina qual linha será obtida: 0 - Superior 1 - Inferior Retorno: Float considerando 2.0 como desvio, 20 períodos e tipo de média aritmética. Função BollingerBandW A função BollingerBandW retorna o valor do indicador Bollinger Band Width, de acordo com o período e tipo de média desejados. Sintaxe: BollingerBandW(Desvio : Float, Media : Integer, TipoMedia : Integer) Parâmetros: Desvio: Desvio utilizado no momento do cálculo do indicador. Media: Período da média utilizada no momento do cálculo do indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2-Welles Wilder 3- Ponderada Retorno: Float 2.0 como desvio, 20 períodos e tipo de média exponencial. Função BollingerBPerc A função BollingerBPerc retorna o valor do indicador Bollinger b%, de acordo com o período e tipo de média desejados. Sintaxe: BollingerBPerc(Desvio : Float, Media : Integer, TipoMedia : Integer) Parâmetros: Desvio: Desvio utilizado no momento do cálculo do indicador. Media: Período da média utilizada no momento do cálculo do indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Retorno: Float como desvio, 20 períodos e tipo de média Welles Wilder. Função BullPower A função BullPower retorna o valor do indicador

Bull Power , de acordo com o período e tipo de média desejados. Sintaxe: BullPower(Período : Integer, PeríodoMedia : Integer, TipoMedi: Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador; PeríodoMedia: Período utilizado no momento do cálculo da média utilizada no indicador. TipoMedia: Determina o tipo da média utilizada, 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Retorno: Float períodos para a média aritmética. Função CCI A função CCI retoma o valor do indicador CCI , de acordo com o período desejado. Sintaxe: CCIPeríodo : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. Função ChaikinMoneyFlow A função ChaikinMoneyFlow retorna o valor do indicador Chaikin Money Flow , de acordo com o período desejado. Sintaxe: ChaikinMoneyFlow(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 21 período para o cálculo. Função ChaikinOsc A função ChaikinOsc retoma o valor do indicador Oscilador Chaikin , de acordo com as médias desejadas. Sintaxe: ChaikinOsc(MediaLonga : Integer, MediaCurta : Integer) Parâmetros: MediaLonga: Determina o período da Média Longa para formação do cálculo. MediaCurta: Determina o período da Média Curta. Retorno: Float períodos para a média longa, e 3 para a curta. Função ChainSetup A função ChainSetup retoma o valor do indicador ChainSetup Sintaxe: ChainSetup Parâmetros: Sem parâmetros. Retorno: Float Função CohenPriceWave A função CohenPriceWave retoma o dado do indicador Cohen - Price Wave (desenvolvido por Rodrigo Cohen equipe). Sintaxe: CohenPriceWave(Ticks : Integer) Parâmetros: Ticks: Relacionado ao valor de entrada do indicador. Retorno: Float o cálculo. Função CohenWeisWave A função CohenWeisWave retoma o dado do indicador Cohen - Weis Wave (desenvolvido por Rodrigo Cohen e equipe) Sintaxe: CohenWeisWave(Ticks : Integer) Parâmetros: Ticks: Relacionado ao valor de entrada do indicador. Retorno: Float ' cálculo. Função ContadorDeCandle A função ContadorDeCandle contabiliza e sinaliza de forma numérica e organizada no gráfico o número de cada cande. Sintaxe: ContadorDeCandle Parâmetros: Sem parâmetros Retorno: Float Função DarvasBox A função DarvasBox retoma o valor do indicador Darvas Box Sintaxe: DarvasBoxLinha : Integerl Parâmetros: Linha: Determina qual dado(Compra ou Venda) será obtido: 0- Compra 1-Venda Retorno: Float Função DecisionPoints A função DecisionPoints retoma o valor do indicador Pontos de Decisão Sintaxe: DecisionPoints(Tipo : Integer, Linha : Integer) Parâmetros: Tipo: Determina o tipo: Preço, Volume, Faixas de Volume e Variação: 0 - Tipo Preço: Linha: Dado da série que será obtido 0 - Abertura 1- Máxima 2- Mínima 3 Fechamento 4- Ajuste. 1-Tipo Volume: Linha: Três maiores volumes dos períodos 0 - Dado volume 1- Dado volume 2- Dado volume É possível ainda acessar os dados, ao utilizar o recurso Asset, onde é necessário parametrizar o ativo específico, e a bolsa a qual pertence. Bolsas que podem ser parametrizadas: feedBMF feedBMFSynthetic feedBovespa feedCME feedDowJones feedEconomic feedNasdaq feedNyse Estrutura da declaração: A seguir, um exemplo de declaração, e acesso aos dados de fechamento e abertura: ; feedBMF); Além do acesso às séries de dados, pode-se retomar

o ticker (String) e bolsa (String) do Asset declarado, a partir das funções GetAsset e GetFeed: 2- Tipo Faixas de Volume: Linha: 0 - Retoma o dado específico ao tipo 3- Tipo Variação: Linha: (- Dado variação linha inferior (-2%) 1 - Dado variação linha inferior (-1%) 2- Dado variação linha superior (1%) 3 - Dado variação linha superior (2%) Retorno: Float Função DiDilIndex Descric: A função DiDilIndex retorna o valor do indicador desejados. Index , de acordo com o período e tipos de médias Sintaxe: DiDilIndex(MediaReferencia : Integer, TipoMediaReferenci Integer, Media2 : integer, TipoMedia2 : Integer)|Linha : Integer] Integer, Media1 : Integer, TipoMedia? : Parâmetros: MediaReferencia: Parâmetro para o período utilizado no cálculo da média de referência do indicador. TipoMediaReferencia: Determina qual média será considerada 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Media1: Período utilizado no cálculo da média? do indicador. TipoMedia1: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Media2: Período utilizado no cálculo da média1 do indicador. TipoMedia2: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Weltes Wikder 3- Ponderada Linha: Determina qual linha será obtida: 0-Linha 1-Linha2 Retorno: Float 3(Média 1) e 20(Média 1) períodos, aplicando o tipo de média aritmética para o cálculo. Função DiIPDIM A função DIPDIM retoma o valor do indicador DI+/DI- , de acordo com o período desejado. Sintaxe: DIPDiM(Periodo : Integer)|Linha : Integer] Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0-DE 1-DL Retorno: Float 0 cálculo. Função DonchianCH Descrição A função DonchianCh retorna o valor do indicador Canal Donchian , de acordo com o período desejado. Sintaxe: DonchianCh(Período : Integen)|Linha : Integer] Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0 - Média 1- Superior 2- Inferior Retorno: Float considerando 20 períodos para o cálculo. Função DTOScillator A função DTOScillator retoma o valor do indicador DT Oscillator , conforme os parâmetros desejados. Sintaxe: DTOScillator(PeriodoEstocastico — : Integer, PeriodoSK : Integer, TipoSK : Integer, PeríodoSD : Integer, TipoSD : Integer)|Linha : Integer] Parâmetros: PeriodoEstocastico: Período utilizado no momento do cálculo do indicador. PeriodoSK: Período referente ao parâmetro "Período SK". TipoSK: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada PeriodoSD: Período referente ao parâmetro "Período SD". TipoSD: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Determina qual linha será obtida: 0-Linha 1 1-Linha 2 Retorno: Float 12(PeriodoEstocastico), 8(PeriodoSK), 5(PeriodoSD) períodos, e tipo aritmética para o cálculo. Função Envelop: A função Envelope retorna o valor do indicador Envelope , de acordo com o período e média desejados. Sintaxe: Envelope(Percentual : Float, PeriodoMedia : Integer, TipoMedia : Integer)|Linha : Integer] Parâmetros: Percentual: Percentual utilizado no momento do cálculo do indicador. PeriodoMedia: Período utilizado para o cálculo da média. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Linha: Determina

qual linha será obtida: 0- Ponto médio 1- Superior 2- Inferior Retorno: Float considerando 5.0 como percentual, 20 períodos e tipo de média aritmética para o cálculo. Função Euroinvest A função Euroinvest retoma o valor do indicador Euroinvest , conforme os parâmetros determinados. Sintaxe: Euroinvest(Risco: Integer, ModoCalculo : Integer, Período : Integer, Desvio : Float, UsarVWAP : Boolean, UsarAtr : Boolean) Parâmetros: Risco: Determina o tipo de perfil: 0-Zero 1-um 2-Dois 3-Três ModoCalculo: Tipo de média: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Período: Período considerado para o cálculo da média. Desvio: Desvio da média. UsarVWAP: Determina se o VWAP será utilizado. UsarAtr: Determina a habilitação do StopATR. Retorno: Float média aritmética, 21 períodos, com 2 de desvio, onde o VWAP e StopAtr estão habilitados. Função FastStochastic A função FastStochastic retorna o valor do indicador Estocástico Rápido, de acordo com o período desejado. Observação: Os parâmetros PeríodoMedia e TipoMedia são opcionais, caso não sejam determinados, serão. utilizados os valores 14 e 1 respectivamente. Sintaxe: FastStochastic(Período : Integer, PeríodoMedia : Integer, TipoMedi Parâmetros: Período: Período utilizado no momento do cálculo do indicador. PeríodoMedia: Período utilizado no momento do cálculo da média. TipoMedia: Tipo da média a ser calculada pelo indicador: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float períodos para o cálculo. Função FinancialVol A função FinancialVol retorna o valor do indicador Volume Financeiro , podendo-se incluir u desconsiderar os dados: "volume projetado" e "leilão e trades diretos". Observação: O parâmetro Agressores só será considerado para o cálculo do indicador caso o seja assinado o Opcional "Plugin Tape Reading". Caso não possua o opcional o indicador será sempre calculado como Faise, independente do que for inserido no campo. Sintaxe: FinancialVol(VolumeProjetado : Boolean, Agressores : Boolean) Parâmetros: VolumeProjetado: Determina se o volume irá considerar o dado projetado. Agressores: Determina se o volume irá desconsiderar o leilão e rades diretos. Retorno: Float desconsiderando os dados de "volume projetado" e "leilão e trades diretos". Função Forcel: A função Forceindex retorna o valor do indicador Force Index , de acordo com o período e tipo de média desejados. Sintaxe: ForcelIndex(Período : Integer, TipoMedia : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float períodos e tipo de média exponencial. Função FrassonATR A função FrassonATR retomma o valor do indicador Frasson ATR , de acordo com o fator e períodos desejados. Sintaxe: FrassonATR(Fator : Float, PeríodoMaxMin : Integer, PeríodoATR : Integen)|Linha : Integer] Parâmetros: Fator: Fator de multiplicação do ATR utilizado no momento do cálculo do indi PeríodoMaxMin Determina o período de Máxima de Mínima. PeríodoATR Determina o período do cálculo do ATR. Linha: Determina qual linha será obtida: 0 - Superior 1 - Inferior Retorno: Float considerando 0,03(Fator), 15(Período Máxima/Mínima) e 50(Período ATR) para o cálculo. Função FrassonVH A função FrassonVH retorna o valor do indicador Frasson VH, de acordo com o fator e períodos desejados.

Sintaxe: FrassonVHí(Fator : Float, PeríodoMaxMin : Integer, PeríodoVH : Integer)[Linha : Integer] Parâmetros: Fator: Fator de multiplicação do ATR utilizado no momento do cálculo do indicador. PeríodoMaxMin Determina o período de Máxima de Mínima. PeríodoVH Determina o período do cálculo do VH. Linha: Determina qual linha será obtida: 0- Superior 1 -Inferior Retorno: Float considerando 0,03(Fator), 15(Período Máxima/Mínima) e 50(Período VH) para o cálculo. Função FullStochastic Descrição A função FullStochastic retorna o valor do indicador Estocástico Pleno , de acordo com o período desejado. Sintaxe: FullStochastic(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float períodos para o cálculo. Função FuraChao A função FuraChao retorna o valor do indicador Fura-Chão , de acordo com o coeficiente e deslocamento desejados. Sintaxe: FuraChao(Coeficiente : Float, Deslocamento : Integer) 4' const WDO = Asset("WDOFUT", feedBMF); begin if(MaxBarsForward PlotText("Ticker: + WDO.GetFeed, clR É possível i estratégia. juir, um exemplo de coloração, onde será aplicada a cor verc acima de sua média (20 períodos), e o ativo WIN, fechando abaixo de sua média (20 A coloração vermelha será aplicada, caso o fechamento do WDO estiver abaixo da média, e o último preço do WIN acima de sua média. A coloração cinza será identificada, caso não sejam satisfeitas as condições. const WDO = Asset("WDOFUT", feedBMF) WIN = Asset("WINFUT", feedBMF) input Período(29) var wdoMedia : Float. winMedia : Float begin wdoMedia := MediaExp(Período, WDO.close) winMedia := MediaExp(Período, WIN.close) iff(WDO. cla and (WIN.close < winMedia) then PaintBar(clLime) else if(WDO.close < wdoMe: (WIN.close > winMedia) then PaintBar(clRed) else Atenção grá iporais em conjunto com múltiplos ativos, a: fechamento de candle serão feitas sempre em função do fechamento de candle do ativo principal. Nesse momento, os ativos auxiliares momento, já que esses gráfi tempo. Parâmetros: Coeficiente: Coeficiente utilizado no momento do cálculo do indicador. Deslocamento: Determina quantos períodos anteriores serão utilizados como base no indicador. Retorno: Float 0.14(Coeficiente) e 1 período(Deslocamento) para o cálculo. Função FuraTeto A função FuraTeto retoma o valor do indicador Fura-Teto , de acordo com o coeficiente e deslocamento desejados. Sintaxe: FuraTeto(Coeficiente : Float, Deslocamento : Integer) Parâmetros: Coeficiente: Coeficiente utilizado no momento do cálculo do indicador. Deslocamento: Determina quantos períodos anteriores serão utilizados como base no indicador. Retorno: Float 0.14(Coeficiente) e 1 período(Deslocamento) para o cálculo. Função HeikinAshi A função HeikinAshi retoma o valor do indicador Heikin Ashi , de acordo com o período e tipo de média desejados. Sintaxe: HeikinAshi(Media : Integer, TipoMedia : Integer)[Dado : Integer] Parâmetros: Media: Media utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Linha: Determina qual dado será obtido: 0 - Abertura 1- Fechamento 2- Máxima 3- Mínima Retorno: Float considerando 1(Período) e O(Aritmética) para o cálculo. Função HiLoActivator A função HiloActivator retoma o valor do indicador HiLo Activator , de acordo com o período desejado. Sintaxe: HiloActivator(Período :

Integer)|Linha : Integer] Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0 - Valor Indicador 1- Tendência Retomo para identificação da tendência 0-Baixa 1-Data Retorno: Float para o cálculo. Função HistVolatility A função Histvolatility retoma o valor do indicador Volatilidade Histórica , de acordo com o período e tipo de média desejados. Sintaxe: HistVolatility(Período : Integer, TipoMedia : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: - Aritmética 1- Exponencial 2-Weltes Wikder Retorno: Float 22(Períodos) e tipo de média exponencial. Função HSI A função HSI retoma o dado do indicador IFH Índice de Força Harmônico (HS1) , conforme o período desejado. Sintaxe: const Parâmetros: Sem parâmetros. Retorno: Float Função HullMovingAverage A função HullMovingAverage retoma o valor do indicador Hull Moving Average , de acordo com o período desejado. Sintaxe: HullMovingAverage(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float períodos para o cálculo. Função IchimokuCloud A função IchimokuCloud retoma o valor do indicador Ichimoku Cloud , de acordo com os parâmetros desejados. Sintaxe: IchimokuCloud(TenkanSen : Integer, KijunSen : Integer, SenkouSpanB : Integer)|Linha : Integer] Parâmetros: TenkanSen: Utilizado no momento do cálculo do indicador. KijunSen: Utilizado no momento do cálculo do indicador. SenkouSpanB: Utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0 - Tenkan-Sen 1-Kijun-Sen 2- Chikou Span 3- Senkou Span A 4 - Senkou Span B Retorno: Float Cloud, considerando 9(Tenkan-Sen), 26(Kijun-Sen) e 52(Senkou Span B) para o cálculo. Função ImpliedVolatility A função ImpliedVolatility retoma o valor do indicador Volatilidade Implícita , de acordo com o período desejado. Sintaxe: ImpliedVolatility(ModeloTeorico : Boolean, TipoOpcao : Boolean) Parâmetros: ModeloTeorico: Determina o modelo para o cálculo: True - Black & Scholes False - Binomial TipoOpcao: Determina o tipo da opção. True - Americana False - Européia Retorno: Float modelo Black & Scholes. Função KeltnerCH A função KeltnerCH retoma o valor do indicador Keltner Channels , de acordo com o período e tipo de média desejados. Sintaxe: KeltnerCH(Desvio : Float, Período : Integer, TipoMedia : Integer)|Linha : Integer] Parâmetros: Desvio: Desvio utilizado no momento do cálculo do indicador. Período: Período utilizado para o cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Linha: Determina qual linha será obtida: 0 - Superior 1- Inferior Retorno: Float Channels, considerando 2 0(Desvio), 20(Períodos) e tipo de média exponencial. Função KVO A função KVO retoma o valor do indicador KVO , de acordo com os períodos desejados. Sintaxe: iaCurta : Integer, Sinal : Integer)|Dado : Integer] Parâmetros: MediaLonga: Determina o período da Média Longa para formação do cálculo. MediaCurta: Determina o período da Média Curta. Sinal: Determina o sinal para a formação do cálculo. Linha: Determina qual linha será obtida: 0 -Linha 1-Histograma Retorno: Float 55(Média Longa), 34(Média Curta), 13(Sinal) para o cálculo. Função LinearRegressionChannel A função LinearRegressionChannel retoma os dados do indicador Canal

de Regressão Linear. Sintaxe: LinearRegressionChannel(Período : Integer; UsarDesvioSuperior : Boolean; DesvioSuperior : Float; UsarDesvioInferior : Boolean; DesvioInferior : Float|Linha : Integer] Parâmetros: Período: Total de candles considerados. UsarDesvioSuperior: Define se o desvio superior será utilizado. DesvioSuperior: Desvio superior considerado. UsarDesvioInferior: Define se o desvio inferior será utilizado. DesvioInferior: Desvio inferior considerado. Linha: Determina qual linha será obtida: 0 - Centra 1- Superior 2- Inferior Retorno: Float Função LSVolatilityIndex A função LSVolatilityindex retoma o valor do indicador L&S Volatility Index. Sintaxe: LSVolatilityindex Parâmetros: Sem parâmetros. Retorno: Float Função MACD A função MACD retoma o valor do indicador MACD Linha e Histograma , de acordo com os períodos desejados. Sintaxe: MACD(MediaLonga : Integer, MediaCurt: Integer, Sinal : Integer)|Dado : Integer] Parâmetros: MediaLonga: Determina o período da Média Longa para formação do cálculo. MediaCurta: Determina o período da Média Curta. Longa), 12(Média Curta), 9(Sinal). Função MFI A função MFI retoma o valor do indicador Market Facilitation Index Acessando dados anteriores (O dado atual de uma variável pode ser descrito por Variavel[fo]. 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 0510/2010 Pt;ul n 04102010 Preco1] 1240 1102010 Preco2] NE 30009/2010 Preco3] s 29/09/2010 Preco4] 138 ddmm/AA Precofn) tSSA Portanto, Preco1] 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. Constantes São utilizadas para a inserção de parâmetros de determinadas funções, estes valores não podem ser alterados pelo usuário. Lado da Ordem: isBuy ou osCompra: Ordem de compra. isSell ou osVenda: Ordem de venda. mF Parâmetros: Sem parâmetros. Retorno: Float Função MIMA Descrição A função MIMA possui como característica retomar o dado do indicador PhiCube - MIMA. Sintaxe: MIMA(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float períodos para o cálculo. Função Momentum A função Momentum retorna o valor do indicador Momentum , de acordo com o período e tipo de média desejados. Sintaxe: Momentumí(Período : Integer, Media : Integer, TipoMedia : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Media: Média utilizada no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float considerando 10(Períodos), 3(Média) e tipo de média exponencial para o cálculo. Função MomentumStoc! A função MomentumStochastic retoma o valor do indicador Momento Estocástico , de acordo com o período desejado. Sintaxe: Momen- tumStochastic(Período : Integer) Parâmetros: Período: Período utilizado no

momento do cálculo do indicador. Retorno: Float considerando 14 período para o cálculo. Função MoneyFlow A função MoneyFlow retoma o valor do indicador Money Flow Sintaxe: MoneyFlow Parâmetros: Sem parâmetros. Retorno: Float Função MoneyFlowindex A função MoneyFlowindex retoma o valor do indicador Money Flow Index , de acordo com o período desejado. Sintaxe: MoneyFlowindex(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 14 período para o cálculo. Função NelogicaBottomFinder A função NelogicaBottomFinder retorna o valor do indicador Nelogica Bottom Finder , conforme o dado desejado(inha e histograma). Sintaxe: NelogicaBottomFinder[Dado : Integer] Parâmetros: Dado: Determina qual dado será obtido: 0 -Linha 1 - Histograma Função NelogicaPullBackFinder A função NelogicaPullBackFinder retoma o valor do indicador Nelogica Pullback Finder , conforme o dado desejado(inha e histoarama). Sintaxe: NelogicaPullBackFinder[Dado : Integer] Parâmetros: Dado: Determina qual dado será obtido: 0 -Linha 1 - Histograma Retorno: Float

Função NelogicaWeisWave A função NelogicaWeisWave retoma o valor do indicador Nelogica Weis Wave , de acordo com o período desejado. Sintaxe: NelogicaWeisWave(Período : Integer) Parâmetros: Período: Determina o período para o cálculo do indicador. Retorno: Float cálculo. Função OBV A função OBV retoma o valor do indicador OBV Sintaxe: oBv Parâmetros: Sem parâmetros. Retorno: Float Função OBVAvg A função OBVAvg retoma o valor do indicador OBV Ponderado Sintaxe: “OBVAvg(Período : Integer, TipoMedia : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Determina qual média será considerada: 1- Exponencial 2- Welles Wilder 3- Ponderada Retorno: Float 10(Períodos) e tipo de média ponderada(3).

Função OnBalanceTR A função OnBalanceTR retoma o valor do indicador OnBalance True Range Sintaxe: OnBalanceTR Parâmetros: Sem parâmetros. Retorno: Float Função OpenDaily A função OpenDaily retorna o dado de abertura, conforme o deslocamento especificado. Sintaxe: OpenDaily(DaysBack : Integer) Parâmetros: DaysBack: Número de dias para o deslocamento. Retorno: Float Função Openlinterest A função Openlinterest retoma o valor do indicador Contratos em Aberto Sintaxe: Openlinterest Parâmetros: Sem parâmetros. Retorno: Float Função ParabolicSAR A função ParabolicSAR retoma o valor do indicador SAR Parabólico , de acordo com os parâmetros desejados. Sintaxe: ParabolicSAR(Fator : Float, Limite : Float) Parâmetros: Fator: Determina o Fator de Aceleração para formação do cálculo. Limite: Determina o Limite de Aceleração para formação do cálculo. Retorno: Float considerando 0.02(Fator) e 0.2(Limite) para o cálculo. Função Phibo Descrç: A função Phibo retoma o valor do indicador PhiCube - Phibo Line Sintaxe: Phibo(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. optCall - Opção de compra. optPut - Opção de venda. Tipo de Ordem: atLimit - Tipo limite. atMarket - À mercado. atStopLimit - Tipo stop. Cores: clAqua ou clAzulClaro - Cor azut-claro clBlack ou clPreto - Cor preta clBlue ou clAzul - Cor azul clCream ou clCreme - Cor creme clDKGray ou

cICinzaEscuro - Cor cinza-escuro clFuschia ou clFucsia - Cor fúcsia clGray ou clCinza - Cor cinza clGreen ou clVerde - Cor verde clLime ou clVerdeLima - Cor verde-limão cILtGray ou ciCinzaClaro - Cor cinza-claro elMaroon ou clMarrom - Cor marrom ciMedGray ou clCinzaMedio - Cor Cinza médio clMoneyGreen ou clVerdeClaro - Cor verde-claro elNavy ou clAzulMarinho - Cor azul-marinho clolive ou ciVerdeOliva - Cor verde-oliva clPurple ou ciPurpura - Cor púrpura clRed ou ciVermelho - Cor vermelha clSilver ou clPrata - Cor prata clSkyBlue ou clAzulClaro - Cor azul-claro clTeal - Cor Verde-azulado elWhite ou clBranco - Cor branca clYellow ou clAmarelo - Cor amarela

Controle de Fluxo As instruções de fluxo são utilizadas para administrar a sequência de execução das instruções dentro de um programa. À NTSL apresenta três tipos desse tipo: If then else 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 função Pivot retorna o valor do indicador Pivot , de acordo com os parâmetros específicos desejados. Sintaxe: PivotíNormal : Boolean, TresLinhas : Boolean)[Linha : Integer] Parâmetros: Normat: Determina o tipo de cálculo que será efetuado para o Pivot True - (Máxima + Mínima + Fechamento) / 3 False - (Abertura + Máxima + Mínima + Fechamento) / 4 TresLinhas: Determina quantas linhas serão consideradas: True - Três Linhas False - Duas Linhas Linha: Determina qual linha será obtida: True(Três Linhas): 0 -Pivot 0-Pivot Retorno: Float cálculo: (Máxima + Mínima + Fechamento) / 3. Função PriceNery A função PriceNery retorna os dados do indicador PriceNery. Sintaxe: PriceNery Parâmetros: Sem parâmetros Retorno: Float Função PriceOsc A função PriceOsc retorna o valor do indicador Oscilador de Preços , de acordo com os períodos e tipos de médias desejados. Sintaxe: PriceOsc(Media1 : Integer, TipoMedia1 : Integer, Media2 : Integer, TipoMedia2 : Integer) Parâmetros: Media1: Período utilizado para a média 1. TipoMedia1: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Media2: Período utilizado para a média 2. TipoMedia2: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 2- Welles Wilder 3- Ponderada Retorno: Float considerando 12(Média 1) e 21(Média 2) períodos, e tipo exponencial para o cálculo. Função PriceVolumeTrend A função PriceVolumeTrend retorna o valor do indicador Tendência Preço/Volume Sintaxe: PriceVolumeTrend Parâmetros: Sem parâmetros. Retorno: Float Função PriorCote Descrição A função PriorCote retorna o valor do indicador Prior Cote , de acordo com o dado desejado. Sintaxe: PriorCote(Dado : Integer) Parâmetros: Dado: Determina o dado que será obtido: 0 - Fechamento 1- Abertura 2- Máxima 3 - Mínima 4- Ajuste Retorno: Float Exmpl: Função PTAX

A função PTAX retorna os dados do indicador TR - PTAX Sintaxe: PTAXIDado : Integer] Parâmetros: Dado: Obtém o dado desejado do indicador: 0 -Oficial 1- P4 2-P3 3-P2 4-P1 Retorno: Float Função PTAXFuturo A função PTAXFuturo retorna os dados do indicador TR - PTAX Futuro Sintaxe: PTAXFuturo[Dado : Integer] Parâmetros: Dado: Obtém o dado desejado do indicador: 0-Oficial

1-P4 2-P3 3-P2 4-P1 Retorno: Float Função QuantityVol A função QuantityVol retoma o valor do indicador Volume Quantidade , podendo-se incluir “ou desconsiderar os dados: *volume projetado” e “leilão e trades diretos”. “Observação: O parâmetro Agressores só será considerado para o cálculo do indicador caso o seja assinado o Opcional “Plugin Tape Reading”. Caso não possua o opcional o indicador será sempre calculado como False, independente do que for inserido no campo. Sintaxe: “QuantityVol(VolumeProjetado : Boolean, Agressores : Boolean) Parâmetros: VolumeProjetado: Determina se o volume irá considerar o dado projetado. Agressores: Determina se o volume irá desconsiderar o leilão e trades diretos. Retorno: Float desconsiderando os dados de “volume projetado” e “leilão e trades diretos”.

Função Rafi A função Rafi retoma o valor do indicador Rafi Sintaxe: Rafi Parâmetros: Sem parâmetros. Retorno: Float A função Ravi retorna o valor do indicador Ravi , de acordo com os períodos desejados. Sintaxe: RavilMediaLonga : Integer, MediaCurta: Integer) Parâmetros: MediaCurta: Período utilizado na média curta para o cálculo do indicador. MediaLonga: Período utilizado na média longa. Retorno: Float Curta) e 65(Média Longa) períodos para o cálculo. Função RBG A função RBG retoma os dados do indicador RBG. Sintaxe: RBG Parâmetros: Sem parâmetros Retorno: Float Função RenkoVTwo A função RenkoVTwo retorna o valor do indicador RenkoV2 , de acordo com os parâmetros desejados. Itaxe: RenkoVTwo(Periodo : Integer, Abertura : Float, Deslocamento : Integer)|Linha : Integer] Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Abertura: Referente ao parâmetro “Abertura” do indicador. Deslocamento: Relacionado ao parâmetro “Deslocamento do indicador. Linha: Determina qual linha será obtida: 0 - Linha Renkov2+ 1-Linha RenkovV2- Retorno: Float considerando 20(Período), 1.5(Abertura) e O(Deslocamento) para o cálculo Função ROC A função ROC retorna o valor do indicador ROC , de acordo com os períodos e tipo de média desejados. Sintaxe: ROC(Periodo : Integer, Media : Integer, TipoMedi: Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. Media: Período da média utilizada. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float 3(Período), 9(Média) e tipo de aritmética. Função RSI A função RSI retorna o valor do indicador IFR(RSI) , de acordo com o período e tipo desejados. Sintaxe: RSi(Período : Integer, Tipo : Integer) Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. Tipo: Tipo de cálculo aplicado: 0 - Clássico 1-Simples Retorno: Float períodos e tipo clássico para o cálculo. A expressão (condição) pode ser qualquer tipo de teste que resulte em verdadeiro ou falso, alguns 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) O comando FOR é utilizado para definir um loop controlado, ou seja, o bloco é executado Tepetidamente até que a variável de contagem saia do valor inicial e atinja o valor

final. Para a variável controladora, pode-se incrementá-la, a partir do comando To, ou decrementá-la, ao utilizar na estrutura o DownTo. Exemplo (For To Do): Observe o código da média móvel, a variável de contagem nIndex começa com o valor 0 e deve chegar ao valor de Período — 1. No exemplo, Período foi definido como parâmetro de entrada, para definir o tamanho da média. Assim, caso seja parametrizado o valor 0 (Período), 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. Função RsiStochastic A função RsiStochastic retoma o valor do indicador IFR Estocástico, de acordo com o período desejado. Sintaxe: RsiStochastic(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 2 períodos para o cálculo. Função SafeZoneDownTrend A função SafeZoneDownTrend retoma o valor do indicador Stop SafeZone DownTrend, de acordo com os parâmetros desejados Sintaxe: SafeZoneDownTrend(Multiplicador : Float, Período : Integer, Deslocamento : Integer) Parâmetros: Multiplicador: : Valor de Multiplicador utilizado no cálculo do indicador. Período: Período considerado. Deslocamento: Deslocamento de períodos. Retorno: Float considerando 2.0(Multiplicador), 10(Período) e 0(Deslocamento) para o cálculo. Função SafeZoneUpTrend A função SafeZoneUpTrend retoma o valor do indicador Stop SafeZone UpTrend, de acordo com os parâmetros desejados. Sintaxe: SafeZoneUpTrend(Multiplicador : Float, Período : Integer, Deslocamento : Integer) Parâmetros: Multiplicador: : Valor de Multiplicador utilizado no cálculo do indicador. Período: Período considerado. Deslocamento: Deslocamento de períodos Retorno: Float 2.0(Multiplicador), 10(Período) e 0(Deslocamento) para o cálculo. Santo(Período : Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Linha: Determina qual linha será obtida: 0 - Dado referente à linha Santo. 1-Dado relacionado ao Sinal. Retorno: Float à variável "vSanto" o dado do sinal, considerando 72(P« cálculo. Função SlowStochastic A função SlowsStochastic retoma o valor do indicador Estocástico Lento, de acordo com o período desejado. Observação: Os parâmetros PeríodoMedia e TipoMedia são opcionais, caso não sejam determinados, serão utilizados os valores 14 e 1 respectivamente. Sintaxe: : Integer, TipoMedia : SlowStochastic(Período : Integer, PeríodoMedi Parâmetros: Período: Período utilizado no momento do cálculo do indicador. PeríodoMedia: Período utilizado no momento do cálculo da média. TipoMedia: Tipo da média a ser calculada pelo indicador: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float períodos para o cálculo. Função StopATR A função StopATR retorna o valor do indicador Stop ATR, de acordo com os parâmetros desejados. Sintaxe: StopATR(Desvio : Float, Período : Integer, TipoMedia : Integer)[Dado : Integer] Parâmetros: Desvio: Desvio utilizado para o cálculo do indicador. Período: Período que será considerado. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Dado: Determina o dado que será obtido: 0 - Valor indicador. 2.0(Desvio), 20(Períodos) para o cálculo. Função TendencyTracker A função TendencyTracker retorna o valor do indicador Rastreador de Tendências, conforme o período desejado. Sintaxe: TendencyTracker(Dias : Integer) Parâmetros: Dias: Período considerado. Re-

torno: Float períodos para o cálculo. Função Tilson A função Tilson retoma o valor do indicador Tillson's T3 Moving Average , de acordo com os parâmetros desejados. Sintaxe: Tilson(Fator : Float, Media : Integer) Parâmetros: Fator: Determina o Fator para formação do cálculo. Media: Período utilizado no momento do cálculo do indicador. Retorno: Float Average, considerando 0.7(Fator) e 3(Media) para o cálculo. Função TimeAgrBuySell A função TimeAgrBuySell retoma o valor do indicador TR - Tempo Agressão - Compra Sintaxe: TimeAgrBuySell(AlertaVariacoes : Integer) Parâmetros: AlertaVariacoes: Quantidade de variações que serão consideradas. Retorno: Float Função TimeAgrTotal A função TimeAgrTotal retoma o valor do indicador TR - Tempo Agressão - Total Sintaxe: TimeAgrTotal(AlertaVariacoes : Integer) Parâmetros: AlertaVariacoes: Quantidade de variações que serão consideradas. Retorno: Float Função TopBottomDetector A função TopBottomDetector retorna o valor do indicador Detector de Topos e Fundos, de acordo com o período desejado. Sintaxe: TopBottomDetector(Periodo : Integer) Parâmetros: Periodo: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 2 período para o cálculo. Função Trades A função Trades retoma o valor do indicador Negócios, Parâmetros: Sem parâmetros. Retorno: Float Função TrendCloud A função TrendCloud retorna o valor do indicador TrendCloud, de acordo com os parâmetros desejados. Sintaxe: TrendCloud (DiasRetroativos : Integer; CorCompra : Integer; VWAP : Integer; CorVenda : Integer) Parâmetros: DiasRetroativos: Período utilizado no momento do cálculo do indicador. CorCompra : Determina qual média será considerada 1-Verde 2- Azul VWAP : Utilização de VWAP. 1-Ligado 2- Destigado CorVenda : Determina qual média será considerada 1-Vermelho 2- Fucsia Retorno: Float Função TrendSniper A função TrendSniper retoma o valor do indicador TrendSniper, de acordo com os parâmetros desejados. Sintaxe: TrendSniper(DiasRetroativos : Integer; VWAP : Integer; CorCompra : Integer; CorVenda : Integer; Reversao : Integer) Parâmetros: diasRetroativos: Período utilizado no momento do cálculo do indicador. VWAP : Utilização de VWAP. 1-Ligado 2- Destigado Determina qual média será considerada Corvenda : Determina qual média será considerada 1-Vermelho 2- Fucsia Reversao : Determina qual média será considerada 1-Ligado 2- Desligado Retorno: Float à variável "nTrendSniper", o retorno do indicador. A função TRIX retorna o valor do indicador TRIX , de acordo com o período e tipo de média desejados. Sintaxe: TRIX(Medi: Integer, TipoMedia : Integer) Parâmetros: Media: Período utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1- Exponencial 4' input Preco(close); Periodo(9); sResult nIndex r nIndex := & to Período -1 d begin sResult := sResult + Preco[nIndex]; Plot(sResult / Periodo); Exemplo (For To Do x For DownTo Do): A fim didático, segue um exemplo de cálculo de fatorial, utilizando as duas estruturas. For To Do: input) fatorial : Integ À : Integer begin 1, For i:=1 To n Do fatorial := fatorial Plot(fatorial) end For DownTo Do: input) 2- Welles Wider 3- Ponderada Retorno: Float períodos e tipo exponencial para o cálculo. Função TRIXM A função TRIXM retorna o valor do indicador TRIXM , de acordo com o período e tipo de média desejados. Sintaxe: TRIXM(M: Integer, TipoMedia : Integer) Parâmetros: Media: Período

utilizado no momento do cálculo do indicador. TipoMedia: Determina qual média será considerada: 0 - Aritmética 1 - Exponencial 2- Welles Wilder 3 - Ponderada Retorno: Float 9(Média) períodos e tipo exponencial para o cálculo. Função TwoMVAgression A função TwoMVAgression retoma o dado do indicador 2MV Agressão. Sintaxe: TwoMVAgression Parâmetros: Sem parâmetros Retorno: Float(Dado Obtido): Função TwoMVPower A função TwoMVPower retorna o valor do indicador 2MV Power , de acordo com os parâmetros desejados. Sintaxe: TwoMVPower(Periodo1 : Integer, Periodo2 : Integer, Periodo3 : Integer, Media : Integer) Parâmetros: Periodo1: Período em minutos utilizado no momento do cálculo do indicador. Periodo2: Período em minutos. Periodo3: Período em minutos. Media: Período determinado para a média. Retorno: Float(Dado Obtido: -1-Baixa - Neutro 1-Alta Função TwoMVStandard Descrição A função TwoMVStandard retorna o dado do indicador 2MV Padrão Sintaxe: TwoMVStandard Parâmetros: Sem parâmetros

Função TwoMvVolatility Indicador criado para auxiliar na visualização do funcionamento do modo volatilidade do robô 2MV. Sintaxe: TwoMvVolatility(Factorpo: Parâmetro: Factorpos: Parâmetro relacionado ao campo factorpos do indicador. Factomeg : Parâmetro relacionado ao campo factoreg do indicador. Aggr: Relacionado ao campo aggr do indicador. Retorno: Float Função UltimateOscillator A função UltimateOscillator retorna o valor do Ultimate Oscillator desenvolvido por Larry Williams. Sintaxe: UltimateOscillator(PeriodoCurto — : Integer, PeriodoMedio : Integer, PeriodoLongo : Integer) Parâmetros: PeriodoCurto: Período curto utilizado no momento do cálculo. PeriodoMedio: Período médio para o cálculo. PeriodoLongo: Período longo para o cálculo. Retorno: Float S(PeriodoCurto), 8(PeriodoMedio) e 12(PeriodoLongo) para o cálculo. Função Valerie A função Valerie retoma o valor do indicador Valerie, conforme os parâmetros desejados. Sintaxe: Valerie(Periodo : Integer; Offset : Integer) Parâmetros: Periodo: Período utilizado no cálculo do indicador. Offset: Offset considerado. Retorno: Float períodos, e 5 para o Offset. Função VSS A função VSS retorna o valor do indicador VSS , de acordo com os parâmetros desejados. Sintaxe: VSS(Multiplicador : Float, Media : Integer, Deslocamento : Integer) Parâmetros: icador: : Valor de Multiplicador utilizado no cálculo do indicador. Período da média utilizada. Deslocamento: Deslocamento de períodos. Retorno: Float 1.5(Multiplicador), 5(Média) períodos e O(Deslocamento) para o cálculo. Função VWAP A função VWAP retoma o valor do indicador VWAP , de acordo com a periodicidade desejada. Sintaxe: VWAP(Periodo : Integer) Parâmetros: Periodo: Período para obter o dado do indicador: 0 - Barra 2- Semanal 3 - Mensal Retorno: Float diária. Função VWAPDate A função VWAPDate retoma o preço médio ponderado pelo volume, a partir de uma data e horário específicos. Sintaxe: VWAPDate(Data : Integer; Time : Integer) Parâmetros: Date: Data de referência, para início do cálculo. Time: Horário inicial. Retorno: Float Função VWAPMonthly A função VWAPMonthiy retoma o valor do indicador VWAP Mensal Sintaxe: VWAPMonthly Parâmetros: Sem parâmetros. Retorno: Float Função VWAPWeekly Descrição A função VWAPWeekly retoma o valor do in-

indicador VWAP Semanal Sintaxe: VIWAPWeekly Parâmetros: Sem parâmetros.
 Retorno: Float Função VWMA A função VWMA retorna o valor do indicador VWMA, de acordo com o período desejado. Sintaxe: VWMA(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float 10 períodos para o cálculo. Função WAverage A função WAverage retorna o valor do indicador Média Móvel, tipo ponderada. Sintaxe: Waverage(TipoSerie : SeriePeríodo, Período : Integer) Parâmetros: TipoSerie: Série que será considerada para o cálculo. Período: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 100 períodos, e a série de fechamento(Close) para o cálculo. Função Williams A função Williams retorna o valor do indicador Williams %R, de acordo com o período desejado. Sintaxe: Williams(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float períodos para o cálculo. Função xAverage A função xAverage retorna o valor do indicador Média Móvel, tipo exponencial. 4' fatorial : Integer 4 : t begin DownTo 1 Do dal := fatorial * i Plot(fatorial) end legar 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) 120 invés de FOR (PARA) input Preco(clos: eríodo(9); esult : Float; nIndex : Integer; begin sResult nIndex : while(nIndex begi: sResult nIndex end; Plot(sResult/Período); end; Sintaxe: xAverage(TipoSerie : SeriePeríodo, Período : Integer) Parâmetros: TipoSerie: Série que será considerada para o cálculo. Período: Período utilizado no momento do cálculo do indicador. Retorno: Float considerando 100 períodos, e a série de fechamento(Close) para o cálculo. Livro Função AskPrice A função AskPrice retorna o preço da melhor oferta de venda. Sintaxe: AskPrice Parâmetros: Sem parâmetros. Retorno: Float Função AskSize A função AskSize retorna a quantidade da melhor oferta de venda. Atenção: Ao utilizar o modo de CrossOrder na automação, o AskSize irá retomar a quantidade do AskSize da serie histórica, e não no ativo de negociação. Sintaxe: AskSize Parâmetros: Sem parâmetros. Retorno: Float Função BidPrice A função BidPrice retorna o preço da melhor oferta de compra. BidPrice Parâmetros: Sem parâmetros. Retorno: Float Função BidSize A função BidSize retorna a quantidade da melhor oferta de compra. Atenção: Ao utilizar o modo de CrossOrder na automação, o BidSize irá retornar a quantidade do BidSize da serie histórica, e não no ativo de negociação. Sintaxe: BidSize Parâmetros: Sem parâmetros. Retorno: Float Função BookSpread A função BookSpread retorna a diferença entre os melhores preços de compra e venda, no topo do livro. Sintaxe: BookSpread Parâmetros: Sem parâmetros. Retorno: Float ofertas de compra e venda. Função GetAsset A função GetAsset retorna o ticker do ativo, onde a chamada pode ser efetuada realizando a passagem do parâmetro opcional, para retornar o ticker de determinado Asset, ou declarar a função sem parametrização, onde será retornado o ticker do gráfico selecionado. Para retornar o ticker (String) do Asset declarado, basta acessar de forma

análoga às séries de dados: A1.GetAsset. Sintaxe GetAsset(Asset : Ativo Variação Asset: Constante declarada a partir do recurso Asset Variação Sem parâmetros No primeiro exemplo, para o último candle, será visualizado o nome do ativo selecionado no gráfico, com os preços de bid e ask. Para o segundo exemplo, a distinção está na especificação do Asset que será exibido (WDOFUT), independentemente do ativo selecionado no gráfico. Exemplo 01 (Variação 01): PlotText(A função GetFeed retorna o código textual do Feed do ativo. A parametrização é opcional, caso não seja determinado o Asset, será retornado o ticker do gráfico selecionado. Para retomar a bolsa relacionada (String) do Asset específico, pode-se acessar de forma análoga às séries de dados: A1.GetFeed. GetFeed(Asset : Ativ Parâmetros Variação Asset: Constante declarada a partir do recurso Asset Variação Sem parâmetros No primeiro exemplo, para o último candle, será visualizado o nome do ativo selecionado no gráfico, a bolsa a qual pertence, e o spread do book. No segundo, será identificada a bolsa (F) vinculada ao Asset definido (WDO), independentemente do ativo selecionado. Exemplo 01 (Variação 01): + GetFeed(A função ISBMF retoma se o ativo pertence ao segmento BMF. Sintaxe: IsBNF Parâmetros: Sem parâmetros. Retorno: Boolean A função Lote retorna a quantidade mínima de contratos referente ao lote do ativo. Atenção: Ao utilizar o modo de CrossOrder na automação, o Lote irá retornar a quantidade do Lote da serie histórica, e não no ativo de negociação. Sintaxe: Lote Parâmetros: Sem parâmetros. Retorno: Integer Função MinPriceIncrement A função MinPriceIncrement retoma o incremento minimo do preço do ativo. Sintaxe: MinPriceIncrement Parâmetros: Sem parâmetros. Retorno: Float Matemáticas Função ABS A função ABS retorna o valor absoluto de um número ponto flutuante. Sintaxe: ABS(Valor : Float) Parâmetros: Valor: Valor ou variável para obter o módulo. Retorno: Float Função Arctangent A função Arctangent retorna o arcotangente(em graus) de determinado número. Sintaxe: Arctangent(Numero : Float) Parâmetros: Numero: Número que será convertido. Retorno: Float Ceiling(Numero : Float) Parâmetros: Numero: Número que será arredondado. Retorno: Integer No exemplo abaixo, usamos a função Ceiling para retornar o menor inteiro maior que o valor 2,3(Retomo: 3). Função Combination A função Combination calcula o número de grupos com combinação única, considerando um conjunto específico de números. Sintaxe: "Combination(Numero : Integer, QtdGrupos : Integer) Parâmetros: Este comando difere-se do For e While, no sentido de que a condição será testada no final da estrutura, e não no início, portanto, as instruções do bloco serão executadas pelo menos uma vez. Exemplo: A fim de comparação com as outras estruturas já abordadas, segue o exemplo de cálculo de fatorial input n(s Plot(fatorial) |Os operadores matemáticos são: Numero: Total de números, ou itens, a serem considerados; QtdGrupos: Número de itens únicos em cada grupo. Retorno: Integer 0 resultado será 6, conforme o cálculo: (1,2), (1,3), (1,4), (2,3), (2,4) e (3,4) Descrição A função Cos tem como objetivo retornar ao usuário o Cosseno de um valor em radianos. Sintaxe: Cos(Valor : Float) Parâmetros: Valor: Valor ou variável para obter o Cosseno. Retorno: Float No exemplo a seguir, será atribuído à variável "nCos" o Cosseno do valor 180 em

radianos(-0,60).

A função Cosine tem como objetivo retornar ao usuário o Cosseno de um valor em graus. Sintaxe: Cosine(Valor : Float) Parâmetros: Valor: Valor ou variável para obter o Cosseno. Retorno: Float No exemplo a seguir, será atribuído à variável "nCos" o Cosseno do valor 45 em graus(0,71). Função Cotangei A função Cotangent tem como objetivo retornar ao usuário a Cotangente de um valor em graus. Itaxe: CotangentíValor : Float) Parâmetros: Valor: Valor ou variável para obter a Cotangente. Retorno: Float No exemplo a seguir, será atribuído à variável *nCo" a Cotangente do valor 30 em graus. Descrição A função Cum acumula o valor de uma série de dados, desde a primeira barra até a atual. Sintaxe: Cum(SerieDeDados : Serie) Parâmetros: SerieDeDados: Série para efetuar o somatório. Retorno: Float Função Exp Descrição A função Exp tem como objetivo retornar ao usuário a enésima potência do número de Euler. Sintaxe: Expívalor : Float) Parâmetros: Valor: Valor ou uma variável para obter a enésima potência(Euler). Retorno: Float No seguinte exemplo, a função Exp recebe o valor de "2" e irá retornar o valor "7,39". Função ExpValue A função ExpValue possui como finalidade retornar o valor exponencial de um determinado número(e^x). Sintaxe: Expvalue(Valor : Float) Parâmetros: Valor: Valor ou uma variável para obter o valor exponencial(e^x). Retorno: Float No seguinte exemplo, a função ExpValue recebe o valor de "2" e irá retomar o valor "7,39". Função ExtremePriceRatio A função ExtremePriceRatio retoma o ratio das extremidades(divide o maior valor no período pelo menor valor) de um número determinado de barras. Sintaxe: ExtremePriceRatio(Length - : Integer, UseLog : Boolean) Parâmetros: Length: O número de barras que serão considerados no cálculo. UseLog: Deterna se o logaritmo de 10 do resultado da divisão será aplicado. True - Não é calculado o LOG False - É calculado o LOG Retorno: Float divisão. Função Factorial A função Factorial tem como finalidade efetuar o cálculo fatorial($n!$) de um número natural n. Sintaxe: Factorial(Valor : Float) Parâmetros: Valor: Número natural para o cálculo do factorial. Função FastD A função FastD retoma o valor de FastD do Oscilador Estocástico, de acordo com o per Sintaxe: FastD(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. Função FastK A função FastK retoma o valor de FastK do Oscilador Estocástico, de acordo com o per Sintaxe: FastK(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. Função FastKCustom A função FastKCustom retoma o valor de FastK do Oscilador Estocástico, de acordo com os preços determinados por parâmetro, e período desejado. Sintaxe: FastKCustom(PrecosH : Serie, PrecosL : Serie, PrecosC : Serie, Período : Integer) Parâmetros: PrecosH: Série de referência para a máxima. PrecosL: Série de referência para a mínima. PrecosC: Série de referência para o fechamento. Período: Período utilizado no momento do cálculo do indicador. Retorno: Float séries de máxima(PrecosH), mínima(PrecosL), fechamento(PrecosC), e 14 períodos para o cálculo. Função Floor A função Floor possui como finalidade retornar o maior valor inteiro menor que um número determinado. Sintaxe: FlooríValor : Float) Parâmetros:

Valor: Valor de referência para obter o dado específico. Retorno: Integer Nos exemplos, serão atribuídos, às variáveis *m” e ”n”, os valores -7 e 6, respectivamente. Função FracPortion A função FracPortion tem como recurso retornar a parte fracionário de determinado número. Sintaxe: FracPortion(Valor : Float) Parâmetros: Valor: Número para obter a parte fracionária. Retorno: Float Função GCD A função GCD retoma o maior denominador comum entre dois números Sintaxe: GCD(Valor1 : Float, Valor2 : Float) Parâmetros: Valort: Primeiro valor a ser analisado; Valor2: Segundo valor a ser analisado. Retorno: Float Sintaxe: HarmonicMean(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo. Período: Período utilizado no momento do cálculo. Retorno: Float Função Highest A função Highest tem como funcionalidade retornar ao usuário o maior valor da série estipulada por ele, dentro de um período determinado. Sintaxe: Highest(SerieDeDados : Serie, Período : Integer) Operadores lógicos -Os Operadores lógicos são utilizados principalmente para comparações. 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 TFALSE 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: SerieDeDados: Série de dados desejada, podendo ser a abertura, máxima, mínima, fechamento, ou até mesmo indicadores. Período: Determina o período que será considerado para a pesquisa. Retorno: Float No exemplo abaixo, usamos a função Highest para retornar a maior abertura dentro de 9 períodos. Função HighestBar A função HighestBar tem como funcionalidade retornar ao usuário o índice do maior valor da série estipulada por ele, dentro de um período determinado. Sintaxe: HighestBar(SerieDeDados - : Serie, Período : Integer) Parâmetros: SerieDeDados: Série de dados desejada, podendo ser a abertura, máxima, mínima, fechamento, ou até mesmo indicadores. Período: Determina o período que será considerado para a pesquisa. Retorno: Float No exemplo abaixo, usamos a função HighestBar para retornar o índice da maior mínima dentro de 20 períodos.

Função IntPortion A função IntPortion tem como recurso retomar a parte inteira de determinado número. Sintaxe: IntPortion(Valor : Float) Parâmetros: Valor: Número para obter a parte inteira. Retorno: Float A função Log retorna o logaritmo natural(in) de um número. Sintaxe: Log(Valor : Float) Parâmetros: Log: Número para o logaritmo natural(I). Retorno: Float Função Lowest A função Lowest tem como funcionalidade retornar ao usuário o menor valor da série estipulada por ele, dentro de um período determinado. Sintaxe: Lowest(SerieDeDados : Serie, Período : Integer) Parâmetros: SerieDeDados: Série de dados desejada, podendo ser a abertura, máxima, mínima, fechamento, ou até mesmo indicadores. Período: Determina o período que será considerado para a pesquisa. Retorno: Float No exemplo abaixo, usamos a função Lowest para retomar o menor fechamento dentro de 50 períodos. Função LowestBar

A função `LowestBar` tem como funcionalidade retomar ao usuário o índice do menor valor da série estipulada por ele, dentro de um período determinado. Sintaxe: `LowestBar(SerieDeDados — : Serie, Período : Integer)` Parâmetros: `SerieDeDados`: Série de dados desejada, podendo ser a abertura, máxima, mínima, fechamento, ou até mesmo indicadores. `Período`: Determina o período que será considerado para a pesquisa. Retorno: `Float` No exemplo abaixo, usamos a função `LowestBar` para retornar o índice da menor mínima dentro de 26 períodos. Função `MidPoint` A função `MidPoint` retorna a média entre o maior e o menor valor encontrados no período. Sintaxe: `MidPoint(SerieDados : Serie, Período : Integer)` Série de referência. `Período`: Período utilizado no momento do cálculo. Retorno: `Float` fechamento(`SerieDados`) e 15(`Períodos`) para o cálculo. Função `MinutesIntoWeek` A função `MinutesIntoWeek` retorna o número de minutos entre domingo(`Dia`: 0 - `Hora`: 0h00) até o dia e hora determinados por parâmetro. Sintaxe: `MinutesintoWeek(DiaLimite : Integer, HoraLimite : Integer)` Parâmetros: `DiaLimite`: Dia de limite para a conversão em minutos. Referência de dias da semana: 0 - Domingo 1- Segunda 2-Terça 3- Quarta 4- Quinta 5- Sexta 6 - Sábado `HoraLimite`: Hora de limite para a conversão. Retorno: `Integer` No exemplo a seguir, será atribuído à variável "tMin" o total de minutos(8.640) entre domingo(`Dia`: 0 - `Hora`: 0h00) e sábado(`Dia`: 6 - `Hora`: 0h00).

Função `MinutesToTime` A função `MinutesToTime` retorna a conversão de minutos em hora militar(contagem iniciada à meia noite). Sintaxe: `MinutesToTime(Minutos — : Integer)` Parâmetros: `Minutos`: Minutos para a conversão em horas. Retorno: `Integer` No exemplo a seguir, a variável "nHora" irá receber a conversão de 600 minutos em horas, ou seja, será retomado o valor 1000, representando 10h. Função `Mod` A função `Mod` possui como finalidade retornar o resto da divisão entre dois números inteiros. Sintaxe: `ModíDividendo : Integer, Divisor : Integer)` Parâmetros: `Dividendo`: Número referente ao Dividendo. `Divisor`: Número que será o divisor. Retorno: `Integer` Função `Neg` A função `Neg` retorna o valor negativo de um determinado número. Sintaxe: `NegíNumero : Float)` Parâmetros: `Numero`: Valor para obter seu número negativo. Retorno: `Float` Função `NumUnits` A função `NumUnits` retorna o número de contratos/ações de um certo investimento. Sintaxe: `NumuUnits(Amnt : Integer, MinLot : Integer)` Parâmetros: `Amnt`: Valor total de investimento, em reais, por trade. `MinLot`: Tamanho mínimo desejado de lote por transação. Retorno: `Integer` investir 15500 em 100 ações, você poderia comprar 200 ações(`NumUnits(15500, 160) = 200`). Função `PercentChange` A função `PercentChange` calcula a alteração percentual no preço do candle atual sobre determinado descolcamento. Sintaxe: `PercentChange(SerieDados : Serie, Período : Integer)` Parâmetros: `SerieDados`: Série base de referência. `Período`: Período anterior para a comparação com o dado da série atual. Retorno: `Float` considerando a série de fechamento(Série de dados) e 2 períodos para a comparação. Função `PercentR` A função `PercentR` retorna uma porcentagem de onde o preço atual está, relacionado com a faixa de negociação avaliada. Sintaxe: `PercentR(Comprimento : Integer)` Parâmetros: "Comprimento: Com-

primento considerado para o cálculo. Retorno: Float Será atribuído à variável "PercentR" o retorno da função PercentR, considerando 2 como comprimento.

Função Permutation A função Permutation calcula o número de permutações para um determinado número de objetos. Sintaxe: Permutation(Numero : Integer, NumeroObjetos : Integer) Parâmetros: Numero: Determina o número de candles a serem analisados. NumeroObjetos: Define o número de objetos dentro do intervalo de candles que podem ser selecionados. Retorno: Integer A variável "n" irá receber a combinação, considerando 4(Número de candle) e 2(Número de objetos). A função Pos retorna o valor absoluto de um número ponto flutuante. Sintaxe: Pos(Valor : Float) Parâmetros: Valor: Valor ou variável para obter o módulo. Retorno: Float A função Power tem como finalidade retornar a enésima potência de um valor. Sintaxe: Power(Base : Float, Expoente : Integer) Parâmetros: Base: Valor para a base da potênciação; Expoente: Valor no qual a base será elevada. Retorno: Float No exemplo a seguir, será atribuído à variável "X" a potência da operação(2^3). Função PriceOscillator A função PriceOscillator retoma o valor do indicador Price Oscillator, de acordo com os parâmetros desejados. Sintaxe: PriceOscillator(SerieDados : Serie, ComprimentoRapido : Integer, ComprimentoLento : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo do indicador. ComprimentoRapido: Referente ao parâmetro FastLength. ComprimentoLento: Referente ao parâmetro SlowLength. Retorno: Float máxima(SerieDados), 9(ComprimentoRapido) e 18(ComprimentoLento) para o cálculo. Função Random (Com a função Random, o usuário poderá gerar um número aleatório(Inteiro), dentro de um intervalo(iniciado em zero) que possui como limite o valor determinado por parâmetro. Sintaxe: Random(Limite : Integer) Parâmetros: Limite: Recebe um valor ou uma variável para determinar o limite do intervalo, para geração do número. Retorno: Integer Função RateOfChange A função RateOfChange retoma a variação percentual de uma série de dados. Sintaxe: RateOfChange(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série base de referência. Período: Índice do dado para a comparação com o último valor da série. Retorno: Float considerando a série de máxima(Série de dados), e o candle anterior para calcular a variação. Função Round A função Round possui como finalidade efetuar o arredondamento de um número ponto flutuante. Sintaxe: Round(Valor : Float) Parâmetros: Valor: Número(variável ou constante) com casas decimais. Retorno: Integer Conforme no exemplo a seguir, será atribuído à variável Aux" o valor arredondado de 2.6, ao utilizar a função Round. Função Round2Fraction A função Round2Fraction efetua o arredondamento de um número, para o valor mais próximo de um múltiplo do incremento mínimo de um ativo. Sintaxe: Round2Fraction(Valor : Float) Parâmetros: Valor: Valor desejado para o arredondamento de acordo com o incremento mínimo do ativo. Retorno: Float No exemplo abaixo(ativos Bovespa), ao aplicar a função para o valor 27.626, será atribuído o valor 27,63 para a variável 'nRound2". Função Sign A função Sign possui como definição retornar um número inteiro, baseado no sinal de um número. Sintaxe: Sign(valor : Float) Parâmetros: Valor: Número para obter seu sinal. Retorno: Float: Determina qual linha será obtida: Resultados possíveis: 1 - Número

com sinal negativo. 0 - Sem sinal (zero). 1. Número com sinal positivo. parâmetro. Sinal” o valor 1, tendo em vista o valor(-205) passado por Descrição A função Sin tem como objetivo retornar o Seno de um valor em radianos. Parâmetros: Valor: Valor para obter o Seno. Retorno: Float No exemplo a seguir, será atribuído à variável “sSeno” o Seno do valor 180 em radianos(-0,80). Descrição A função Sine tem como objetivo retornar o Seno de um valor em graus. Sintaxe: Sine(Valor : Float Parâmetros: val No exemplo a seguir, será atribuído à variável ”sSeno” o Seno do valor 45 em graus(0,71). Função SlowD Descrição A função SlowD retoma o valor do SlowD (Oscilador Estocástico), de acordo com o período desejado. Sintaxe: SlowD(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float irá receber o valor do indicador SlowD, considerando 14 per cálculo. Função SlowK A função SlowK retoma o valor do SlowkK (Oscilador Estocástico), de acordo com o período desejado. Sintaxe: SlowK(Período : Integer) Parâmetros: Período: Período utilizado no momento do cálculo do indicador. Retorno: Float cálculo. A função Sqrt tem como funcionalidade retomar ao usuário o valor da raiz quadrada de um número. Sartívalor : Float) Parâmetro: Valor: Valor para obter a raiz quadrada. Retorno: Float No exemplo a seguir, será atribuído à variável ”nRaiz”, a raiz quadrada do valor 25. Função Square A função Square tem como funcionalidade retornar ao usuário o valor de um determinado número ao quadrado. Sintaxe: Square(Valor : Float) Parâmetros: Valor: Valor para elevar ao quadrado. No exemplo a seguir, será atribuído à variável ’ o valor ao quadrado do número 5. Função StdDevs A função StdDevs retorna o desvio padrão de uma série de dados, em um determinado período. Sintaxe: StdDevs(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo. Período: Período utilizado no momento do cálculo. Retorno: Float Função Summation A função Summation efetua o somatório do valor do preço de um determinado número de barras. Sintaxe: Summation(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo. Período: Período utilizado no momento do cálculo. Retorno: Float Função Tangent A função Tangent tem como objetivo retornar ao usuário a Tangente de um valor em graus. Sintaxe: Tangent(Valor : Float) Parâmetros: Valor: Valor ou variável para obter a Tangente. Retorno: Criando Funções (Sintaxe) Function (funcao) Nome da Função ((parâmetro 1 : TIPO); (parâmetro TIPO:);(parâmetro nº TIPO)): Tipo de Retorno Begin(inicio) Comandos Float No exemplo a seguir, será atribuído à variável “nTan” a Tangente do valor 30 em graus(0,58). Função TriAverage A função TriAverage efetua a média triangular de uma série de dados, dentro de um determinado período. Sintaxe: TriAverage(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo. Período: Período utilizado no momento do cálculo. Retorno: Float máxima(Close) e 20(Períodos) para o cálculo. Função Ulcerindex A função Ulcerindex mede o nível de estresse de acordo com as condições do mercado. Sintaxe: Ulcerindex(SerieDados : Serie, Período : Integer) Parâmetros: SerieDados: Série utilizada para o cálculo. Período: Período utilizado no momento do cálculo. Retorno: Float máxima(Close) e 3(Períodos) para o cálculo. Função Volatility

A função Volatility retoma a volatilidade de determinado período. Sintaxe: Volatility(Período : Integer) Parâmetros: Período: Período para cálculo do indicador. Retorno: Float

Função VolumeOsc A função VolumeOsc retorna a diferença entre a média aritmética rápida e a média aritmética lenta da série de Volume (financeiro). Sintaxe: VolumeOsc(PeríodoMediaRapida — : Integer, PeríodoMediaLenta : Integer) Parâmetros: 9(Média Rápida) e 21(Média Lenta) períodos para o cálculo.

Função VolumeROC A função VolumeROC retoma o VolumeROC baseado em um número de barras. Sintaxe: VolumeROC(Período : Integer) Parâmetros: Período: Período utilizado para a cálculo. Retorno: Float

S(Períodos). Opções Função Delta A função Delta mede a variação do preço da opção com o preço da ação. Sintaxe: Delta(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 : Float, PutCall : Integer) Parâmetros: DaysLeft: Dias uteis até o vencimento da opção StrikePr: Preço do exercício da opção AssetPr: Preço da ação Rate100: Taxa de juros em % Volty100: Volatilidade em % PutCall: Indica se é uma put ou uma call optPut - Opção de venda optCall - Opção de compra Retorno: Float

vencimento, 16.47 como Strike, 17.25 como preço da ação, 0.1 para taxa de juros e 0.25 para volatilidade, aplicados para uma Call. A função Gamma mede a variação do delta em relação ao preço da ação. Sintaxe: Gamma(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 : Float, PutCall : Integer) Parâmetros: DaysLeft: Dias uteis até o vencimento da opção StrikePr: Preço do exercício da opção AssetPr: Preço da ação Rate100: Taxa de juros em % Volty100: Volatilidade em % PutCall: Indica se é uma put ou uma call optPut - Opção de venda optCall - Opção de compra Retorno: Float

vencimento, 11.08 como Strike, 11.94 como preço da ação, 0.2 para taxa de juros e 0.3 para volatilidade, aplicados sobre uma Call. Função Rh: A função Rho retorna a variação da opção em relação à taxa de juros. Sintaxe: Rho(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 : Float, PutCall : Integer) Parâmetros: DaysLeft: Dias uteis até o vencimento da opção Rate100: Taxa de juros em % Volty100: Volatilidade em % PutCall: Indica se é uma put ou uma call optPut - Opção de venda optCall - Opção de compra Retorno: Float

vencimento, 11.00 como Strike, 12.92 como preço da ação, 0.2 para taxa de juros e 0.3 para volatilidade, aplicados para uma Call. Função Theta A função Theta retoma a variação do preço da opção com o tempo. Sintaxe: Theta(DaysLeft : Integer, StrikePr : PutCall : Integer) Parâmetros: DaysLeft: Dias uteis até o vencimento da opção StrikePr: Preço do exercício da opção AssetPr: Preço da ação Rate100: Taxa de juros em % Volty100: Volatilidade em % PutCall: Indica se é uma put ou uma call optPut - Opção de venda optCall - Opção de compra Retorno: Float

Função Vega A função Vega retoma a variação da opção em relação à volatilidade. Sintaxe: Vega(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 : Float, PutCall : Integer) Parâmetros: DaysLeft: Dias uteis até o vencimento da opção StrikePr: Preço do exercício da opção optCall - Opção de compra Screening Função Select A função Select , apesar de estar disponível na linguagem, não possui recursos para utilização do usuário, sua implementação é específica para uso do sistema, a fim de habilitação do

Screening. Sintaxe: Select Parâmetros: Sem parâmetros. Retorno: Void: Sem retomo. - Nenhuma ordem será criada, já que existe um cancelamento de ordens. Begin BuyAtMarket; CancelPendingorders End;//Fim do fluxo 2 - Nenhuma ordem será criada, pois o ClosePosition vai cancelar a ordem BuyAtMarket. Position; End;//Fim do fluxo de execução 3 - Nenhuma ordem será criada, pois o ReversePosition vai cancelar a ordem SellShortAtMarket Begin SellShortatMarket; ReversePosition; End;//Fim do fl 4 - Nenhuma ordem será criada, pois é realizada o agrupamento de ordens AtMarket que estão dentro do mesmo fluxo de execução. Begin BuyAtMarket SellShortatMarket; End;//Fim do flux - O código abaixo não abre ordens covers. Pois ao passar pela linha BuyAtMarket não é gerada uma posição, pois não há execução da ordem. Desta maneira, a posição não é alterada e o SellToCover não será criado já que não há posição. BuyAtMarket; SelltoCoverLimit + 10 * MinPriceIncrement SellTOCoverstop(Close - 16 * MinPriceIncrement End;//Fim do fluxo de ex: Como alternativa, é aconselhável usar o código abaixo. Desta forma quando existir posição as ordens covers serão enviadas. Begin Tf not HasPosition then begin BuyAtMarket; end else begin SellToCoverLimit(Close + 10 * MinPriceIncrement). SellToCoverStop(Close - 16 * MinPriceIncrement); end; End;//Fim do fluxo de execução 6 - O código abaixo irá atualizar o valor do nCandle cada vez que processar o fluxo de execução do candle. Logo, após a posição ser alterada, será feito o reprocessamento do candie, o valor do nCandle irá receber o valor do candle anterior e como tem posição o valor do nCandle não será alterado e se manterá igual ao valor do candle anterior. Begin ncandle := ncandle[1]; if not HasPosition then begin ncandle := CurrentBar; BuyAtMarket; end; if Currentear > ncandle + 5 then ClosePosition; End;//Fim do fluxo de execução Para zerar a posição após 5 candles, use o código abaixo como exemplo. Begin if not hasposition then begin buyatmarket; ncandle[1] := & end else begin ncandle := ncandle[1] + 1; if ncandle > 5 then ClosePosition; end; End;//Fim do fluxo de execução End (fim); 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 têm 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 podem ser determinadas por Strings, ou a partir da função RGB): 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: * Listar 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”. O código de back-testing simula o mesmo comportamento da automação, as simulações de execução de ordens são processadas todas ao final do fluxo de execução do candle. Quando ocorre aumento de posição ao final do fluxo de execução, o código é reprocessado para ativar as ordens de cobertura da operação (ordens ToCover). O usuário pode observar esse comportamento quando utilizar o modo debug. Automação de Estratégias O módulo de Automação de Estratégias tem o objetivo de automatizar as estratégias de execução criadas no Editor de Estratégias. Ele está disponível no Profit Pro em modo simulação de forma gratuita e disponível para contas reais por meio de módulos opcionais. O novo módulo de automação espelha o funcionamento do backtest das estratégias para execução de ordens reais e em simulador. Para fazer o gerenciamento dessas automações pode-se acessar o menu Estratégias > Automação de Estratégias. A janela de Automações de Estratégias permite que possamos criar e gerenciar automações, acompanhar o resultado de todas as automações de maneira simplificada. Caso queira um detalhamento maior da automação pode-se clicar com o botão direito e ir em Detalhes. Nos detalhes é possível acompanhar todos os eventos que a execução gerou bem como diversos indicadores de performance da automação. Os próximos capítulos detalham desde a criação de uma estratégia de execução utilizando o novo módulo de Automação de Estratégias, como o usuário pode acompanhar essas automações, e por fim detalha regras gerais da execução das estratégias. Para criar uma automação você precisará criar uma estratégia de execução, que pode ser criada a partir do Editor de Estratégias. Se você já sabe o que é isso, pode seguir para o próximo passo. Ao abrir o Editor de Estratégias, você pode criar sua estratégia de execução utilizando funções de backtest (essas funções irão caracterizar a estratégia como sendo uma estratégia de execução e isso possibilitará a sua seleção no próximo passo para que a estratégia seja automatizada). Para criar uma estratégia de execução você deve usar funções do módulo backtest. Sugerimos que você utilize a função Nova estratégia, e selecione exemplos de estratégias de execução se você não estiver familiarizado com programação ou com o Editor de Estratégias. Abaixo é apresentado um exemplo de estratégia, utilizando o indicador IFR/RS, que pode ser utilizado no módulo de estratégias automatizadas: `if (Isought) then begin if (RSI(9, 6) < 30) then BuyToCoverAtMarket; end // Verifica se deve abrir uma posição, // com o ativo sobrecomprado ou sobrevendido else if (RSI(9, 6)[1] < 30) and (RSI(9, 6) > 30) then BuyAtMarket else if (RSI(9, 6)[1] > 70) and (RSI(9, 6) < 30) then SellShortAtMarket; end; Criando uma automação Para criar uma nova automação basta abrir a interface de controle e localizar o botão “Nova Estratégia” localizada no menu superior da janela (indicado pelo número 1`

na imagem), este botão estará sempre disponível para e visível na janela. No entanto, caso ainda não possua nenhuma automação criada, o mesmo botão “Nova Estratégia” será apresentado em destaque no centro da tela indicado. Após clicar nesse botão será aberta a janela de criação da estratégia automatizada onde será configurada os parâmetros da automação em duas etapas. A primeira etapa contém configurações essenciais e obrigatórias para a criação da automação. Na primeira etapa, nas configurações obrigatórias, é possível definir a conta, a estratégia de execução, ativo alvo, quantidade por ordem, quantidade máxima da posição e o período que a estratégia usará como base para as análises. A Estratégia de execução é o “cérebro” da automação. Aqui será selecionado aquela estratégia implementada no editor. Ao clicar no botão “Selecionar Estratégia” estarão listadas todas as estratégias de execução presentes na plataforma. Além disso, caso a estratégia selecionada possua parâmetros de entrada, os mesmos serão listados nessa janela para que possam ser editados da melhor maneira para a estratégia específica. A próxima etapa contém configurações opcionais que podem ser exploradas nos menus acima quando as Configurações Avançadas estiverem ativas (canto inferior esquerdo). Aqui podemos configurar opções de Entrada e Saída, Risco e Segurança. Essas opções estão detalhadas a seguir: Nesta seção é possível especificar parâmetros que alterem a estratégia de abertura de ordens da automatização. Em ordem de entrada você pode especificar qual vai ser o tipo de abertura de posição, ou seja, se a estratégia irá abrir novas posições apenas de compra, ou apenas de venda ou ambos os lados. Em horário de entrada é possível especificar o período de tempo em que a estratégia se manterá habilitada. Após o horário final a estratégia é pausada. Ao desabilitar esta configuração, a estratégia permanecerá executando até o usuário pausá-la ou que o software seja encerrado. É possível também escolher se você deseja confirmar cada ordem de aumento de posição. Neste caso a confirmação de ordens será mostrada para cada nova ordem da estratégia. Nesta outra etapa da configuração, é possível alterar configurações de saída de sua automação. Aqui é possível configurar uma estratégia OCO que será aberta junto com suas ordens, mas atenção, caso você já tenha ordens de saída implementadas em sua estratégia de negociação (ordens de cobertura), estas não serão mais executadas já que as ordens OCO terão preferência. Também é possível configurar nesta seção a zeragem automática por horário. A plataforma precisa estar aberta para que a zeragem aconteça, assim como para que a automação funcione. Seguindo as configurações temos as configurações de risco da automação. Na seção Pausar e Zerar ao alcançar é possível configurar parâmetros que irão pausar e encerrar as posições abertas de acordo com um objetivo de ganho, ou um limite de perda. Caso você não deseje encerrar a posição quando atingir um objetivo de ganho seleciona a opção “Não Zerar ao Atingir Objetivo de Ganho”. Aqui também disponibilizamos um atalho para que você possa configurar o risco global da carteira/conta no gerenciador de risco da plataforma. Por fim, na seção Segurança, temos algumas configurações para definir o comportamento da estratégia em caso ocorra um erro de execução no código da mesma ou eventuais mudanças do mercado, como a entrada do ativo em leilão. Uma vez criada a automação, você pode visualizá-la na janela

Automação de Estratégias: Para acompanhar a sua automação você pode visualizar um resumo rápido das principais informações na lista de automações da janela Automação de Estratégias. Nela você pode visualizar o status, conta e carteira em que a automação está vinculada, ativo, nome da estratégia, período gráfico e os resultados relacionados à execução. Você também pode visualizar nesta janela um resumo das estatísticas relacionadas a essa estratégia que contém mais detalhes da estratégia. Para ligar/pausar uma automação de maneira individual pode-se utilizar o painel, e clicar em executar: Através do menu de contexto também é possível zerar a posição de uma estratégia pausada ou Pausar + Zerar Posição de uma estratégia em execução. Podemos também editar o código da automação no Editor de Estratégias, Excluir a automação que irá excluir também a carteira atrelada a ela. Para realizar edições na sua automação, você pode, dentro do interface de controle, clicar com botão direito na estratégia desejada e ir em “Editar Automação” ou clicar diretamente na engrenagem. Lembrando que caso a sua automação esteja ligada, ao realizar alguma edição ela será pausada. Propriedades da Automação de Estratégias. Para acompanhar os detalhes do que a execução está executando, vá em Detalhes. Nesta janela você poderá acompanhar tanto as ordens e sinais que a estratégia está gerando, como também acompanhar a performance e resultados da estratégia mais detalhadamente. Na janela de Automações é possível utilizar os botões no canto superior direito da interface de controle para pausar todas as automações em execução ou para pausar e zerar todas as posições em aberto: Caso você possua alguma automação ligada será possível identificar mesmo com a janela de estratégias automatizadas fechadas, pois no menu superior do Profit será destacado em azul o menu de Estratégias. Tape: Automação de Estratégias (1) Pausar Todos Pausar + Zerar Todas Pos Editor de Estratégias Gerenciador de Estratégias Neste menu você será informado quantas estratégias você possui em execução e permitirá que você tome ações rápidas através do menu como pausar tudo ou pausar e zerar tudo. As automações de estratégias funcionam com as mesmas regras que são utilizadas no backtest das estratégias de execução. Para entender melhor o que cada função faz relacionada à execução de ordens, recomendamos primeiramente a leitura da seção Lista de Funções > BackTesting. Aqui detalhamos as regras por trás dos mecanismos de automação para facilitar o entendimento do usuário na hora de implementar uma estratégia de execução já pensando na sua automação. O código de estratégias de negociação/execução desenvolvido pelo usuário é executado ao final de cada candle, ou seja, os envios de ordens são feitos quando o candle termina e a avaliação do código é feita sobre este candle que terminou. Deste modo, a escolha de tempo gráfico é crucial para sua automação. Isso é válido para ordens enviadas pela automação que aumentem a exposição para o mercado. Estas irão ser executadas apenas ao final de cada candle. Ordens que aumentem a exposição para o mercado e foram enviadas ao final de um candle e que não forem executadas até o final do candle seguinte serão canceladas ou editadas quando o próximo candle finalizar, de acordo com a estratégia do usuário. Uma vez que, apenas sabemos que um candle foi finalizado quando o próximo iniciar em gráficos atemporais, poderá ocorrer a execução de ordens em

um candle 'subsequente caso o cancelamento chegue na bolsa após a execução da mesma. Caso houver execução de alguma ordem (aumento de posição), o código é reprocessado para a criação de eventuais ordens de cobertura da operação (ordens ToCover). A reexecução da estratégia para posicionamento de ordens de cobertura acontece quando ordens forem executadas, ou seja, podem ocorrer no meio de um candle para não deixar o usuário exposto a riscos do mercado. Mais detalhes sobre ordens de cobertura podem ser vistos a seguir. Ordens para aumento de posição (Buy e Sell) É possível enviar ordens Buy e Sell tanto quando você estiver comprado ou vendido, porém quando uma ordem Buy for enviada enquanto você estiver em uma posição vendida, ela será tratada como uma ordem de cobertura automaticamente. O mesmo acontece para uma ordem Sell, quando você estiver uma posição comprada, ela será tratada como uma ordem de cobertura. Caso contrário, elas apenas aumentaram a sua posição até o limite definido na configuração da estratégia. Ordens conflitantes são automaticamente gerenciadas pelo automatizador, ou seja, quando for enviada uma ordem BuyToCover e você estiver em uma posição comprada ou zerada, ela será ignorada, da mesma maneira se for enviada uma ordem SellToCover e você estiver em uma posição vendida ou Zerada, ela também será ignorada. Ordens de cobertura de posição (BuyToCover e SellToCover) Ordens de cobertura, ou ordens ToCover. nunca irão inverter a sua posição; elas garantem que a ordem contrária vai respeitar sempre a posição da operação. Recomendamos que o programador utilize sempre ordens de saída de posição como ordens ToCover explicitamente no código para garantir que o operacional que o usuário está programando está correto. Para isso, as ordens ToCover são sempre enviadas como ordens OCO, logo você não precisa se preocupar em gerenciar e cancelar eventuais ordens de cobertura que poderiam ficar abertas após a execução de apenas uma das pernas de saída. Ordens de cobertura são enviadas ou atualizadas a toda mudança de candle, cabe ao usuário gerenciar a quantidade de cada ordem caso você esteja posicionado em mais de um lote para cobrir corretamente a exposição ao mercado. Na finalização do candle, caso o código indique um envio de um mesmo número de ordens de cobertura, será realizada uma edição da OCO para os novos valores de preço correspondentes. Por outro lado, caso o envio aumente ou diminua o número de ordens para aquele candle, a OCO aberta será cancelada, e será enviada uma nova OCO com as saídas definidas pelo código. Caso o usuário configure uma ordem OCO pela configuração da automação, as ordens de cobertura definidas pela estratégia serão desconsideradas, já que todas ordens já vão possuir essa cobertura natural da ordem OCO. Se forem utilizados indicadores que alterem seus valores em candles antigos (candles já finalizados), o valor do indicador que será considerado será o do momento em que o candle finalizar, desse modo caso não houver a entrada da execução na virada do candle, não será feito envio de ordens para aquele candle no futuro, mesmo que algum indicador altere seu valor no passado e troque a decisão de envio de ordem. Alguns exemplos de indicadores que podem causar esse comportamento são indicadores que tem apenas um valor durante todo dia, por exemplo: Preço Ask e Bid e também indicadores que decidem apenas

depois de alguns candles, como por exemplo o Detector de topos e fundos ou a linha Chikou Span (2) do indicador IchimokuCloud, entre outros. 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; * — 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. 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" 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. 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 retomem verdadeiro irá pintar em determinada cor. Caso o usuário deseje utilizar outras informações, ele irá poder clicar no botão "Mais" que se encontra 120 lado da variável para selecionar outras condições. 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 "+", será criada uma condição para que se Condição 1 e Condição 2 sejam verdadeiras irá acionar o alarme desejado, e o pop-up de alarme será emitido de acordo com a cor estabelecida. Caso o usuário deseje utilizar outras informações, ele poderá clicar no botão "Mais" que se encontra 120 lado da variável para selecionar outras condições. Após a criação da regra de alarme, a estratégia deverá ser ativada, através do menu "Ferramentas > Gerenciador de Alarmes > Estratégias", selecione "Novo Alarme". 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 poderá clicar no botão "Mais" que se encontra 120 lado da variável para selecionar outras condições. A funcionalidade de Screening mostra, de forma visual, os ativos que se encontram na base de dados e que satisfazem as condições da estratégia. No filo, ao clicar no botão "+", será criada uma condição para que, se Condição 1 e Condição 2 Sejam verdadeiras, irá mostrar o ativo dentro da aba selecionada. 1 Estratégia de Bootar Caso o usuário deseje utilizar outras informações, ele irá poder

clicar no botão "Mais" que se encontra 120 lado da variável para selecionar outras condições. Ao clicar no botão "Aplicar" a estratégia criada é aplicada à lista e irá mostrar os ativos que satisfazem a condição. 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. m—— —RCMAIORSX |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. Cotações Anteriores A janela de Editor de estratégias é onde o usuário poderá 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. * Estatísticas: Ao executar uma estratégia de execução pelo editor, o usuário poderá visualizar a estatística do relatório de performance. 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. Propriedades do Editor de Estratégias Nas propriedades do Editor de estratégias, o usuário i no momento de criação da estratégia: * 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. *

Preenchimento: Configuração para visualizar um preenchimento entre as linhas. Ao clicar no botão de Nova Estratégia, o usuário poderá escolher entre as opções abaixo: Em branco (Texto): 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. Execução: Será criado um exemplo, com uma função para estratégia de execução. Screening: Será inserido um exemplo para a criação de um filtro para o Screening. Alarme: Será criado um exemplo, com a função Alert. A seguir de forma descritiva, segue a lista de funções presentes no Profit: AlertCor : Integer): Dispara uma notificação de alarme. BuyAtMarket: Realiza envio de ordem de compra à mercado. BuyLimit(Preco : Float; Quantidade : Integer = “): Realiza envio de uma ordem limite de compra. BuyPosition: Retoma o tamanho da posição da compra. BuyPrice: Retoma o preço de compra da posição. BuyStop(Stop : Float; Limite : Float; Quantidade : Integer = “): Realiza envio de ordem de compra stop. BuyToCoverAtMarket: Realiza o fechamento de uma operação de venda. BuyToCoverLimit(Preco : Float; Quantidade : Integer = “): Envia uma ordem de compra limite para fechar a operação. BuyToCoverStop(Stop : Float; Limite : Float; Quantidade : Integer = “): Realiza o envio de ordem de compra stop para fechar posição. CancelPendingordersrs: Cancela as ordens pendentes. ClosePosition: Envia ordens para encerrar todas as posições. HasPendingorders: — Retorna verdadeiro (True) caso existam ordens pendentes. HasPosition: Retoma verdadeiro se a posição não é zero. IsBought: Retorna verdadeiro (True) caso exista posição de compra. IsSold: Retorna verdadeiro (True) caso exista posição de venda. MyPrice: Retoma a média entre a máxima, mínima e fechamento. Position: Retoma o tamanho da posição, positivo para compra e negativo para venda. Price: Retoma o preço de compra ou venda da posição, dependendo se estiver comprado ou vendido. ReversePosition: Envia ordens para inverter a posição. SellPosition: etoma o tamanho da posição de venda. SellPrice: Retoma o preço de venda da posição. SellShortAtMarket: Envia ordem de venda à mercado para abrir posição. SellShortLimit(Preco : Float; Quantidade : Integer = “): Envia ordem de venda do tipo limite. SeliShortStop(Stop : Float, Limite : Float; Quantidade : Integer venda enviando uma ordem do tipo Stop. Abre uma posição de SellToCoverAtMarket: - Realiza uma ordem de venda à mercado caso exista uma posição de compra. SellToCoverLimit(Preco a operação. loat; Quantidade : Integer nvia uma ordem de venda limite para fechar SellToCoverStop(Stop : Float, Limite : Float; Quantidade : Integer Stop caso exista uma posição de compra. 'nvia uma ordem SendOrder(Lado : Integer, Tipo : Integer, Quantidade ordem customizada. Integer, Limite : Float, Stop : Float): Envia uma DailyResult(OpenResuit : Depuração ConsoleLogiContent : String, Color : Integer = cIBlack) Imprime uma string no terminal de console para Ajudar na depuração do código. Calendário BarAnnualization: Retoma o fator de anualização, baseado no intervalo da barra. BarDuration: Retoma, em minutos, a duração da barra atual. BarDurationF: Retorna, em

minutos, a duração da barra atual. BarType: Retoma um número referente à periodicidade setada. CalcDate(DataReferencia : Integer, DiasDeslocamento : Integer): Retoma o resultado ao efetuar a adição ou subtração de datas. CalcTime(HoraReferencia : Integer, MinutosDeslocamento : Integer): Retoma o cálculo de adição ou subtração de horas,. CloseD(QuantidadeDiasAnteriores : Integer): Retoma o valor de fechamento de determinado dia anterior. CloseM(QuantidadeMesesAnteriores : Integer): Retoma o valor de fechamento de determinado mês anterior. CloseW(QuantidadeSemanasAnteriores : Integer): Retoma o valor de fechamento de determinada semana anterior. CloseY(QuantidadeAnosAnteriores : Integer): ano anterior. retorna o valor de fechamento de determinado CurrentAssetDate: Retoma a data atual do ativo no gráfico. CurrentDate: Retorna a data atual do sistema. CurrentTime: Retoma a hora atual do sistema. Date: Retorna a data da barra que está sendo analisada. DayOfMonth(Data : Integer): Retoma o dia do mês de uma data específica. DayOfWeek(Data : Integer): Retoma o dia da semana de uma data específica. DaysToExpiration(Mes : Integer, Ano : Integer): Retoma a quantidade de dias úteis restantes até a terceira sexta-feira de um determinado mês e ano. ELDate(Ano : Integer, Mes : Integer, Disformat(YYMMDD). Integer): Retoma a data em EasyLanguage ELDate ConsolidaData : Integer): Transforma uma data YYYYMMDD em EasyLanguage format(YYYYMMDD). FindBar(Data : Integer, Hora : Integer): Localiza uma barra de uma data e hora específica. Friday: Retoma o número 5 referente ao dia da semana: sexta-feira. HighD(QuantidadeDiasAnteriores : Integer): Retoma o valor de máxima de determinado dia retroativo. HighM(QuantidadeMesesAnteriores : Integer): Retoma o valor de máxima de determinado mês retroativo. HighW(QuantidadeSemanasAnteriores : Integer): Retoma o valor de máxima de determinada semana anterior. HighY(QuantidadeAnosAnteriores : Integer): Retoma o valor de máxima de determinado ano anterior. LastCalcDate: - Retoma a data do último candle completo do gráfico. LastCalcTime: - Retoma o horário de fechamento do último candle. LastDayOfMontn(MesReferencia : Integer): Retoma o valor do último dia do mês de referência. LowD(QuantidadeDiasAnteriores : Integer): Retorna o valor de mínima de determinado dia anterior. LowM(QuantidadeMesesAnteriores : Integer): Retoma o valor de mínima de determinado mês anterior. LowW(QuantidadeSemanasAnteriores : Integer): Retoma o valor de mínima de determinada semana anterior. LowY(QuantidadeAnosAnteriores : Integer): Retoma o valor de mínima de determinado ano anterior. MinutesIntoWeek - Retoma o número de minutos até domingo 12 am. Monday: Retoma o número 1 referente ao dia da semana: segunda-feira. Month(Date : Integer): Retoma o mês de uma data específica. Next3rdFriday(Mes : Integer): Retoma quantos dias úteis faltam para a terceira sexta-feira de determinado mês. OpenD(QuantidadeDiasAnteriores : Integer): Retorna o valor de abertura de determinado dia anterior. OpenM(QuantidadeMesesAnteriores Integer): Retorna o valor de abertura de determinado mês anterior. OpenW(QuantidadeSemanasAnteriores: Integer) Retoma o valor de abertura de determinado semana anterior.

OpenY(QuantidadeAnosAnteriores : Integer): Retorna o valor de abertura de determinado ano anterior. RS BarsPerDay: Obtém o número estimado de barras de determinada periodicidade(em minutos). Saturday: Retoma o número 6 referente ao dia da semana: sábado. Sunday: Retoma o número 0 referente ao dia da semana: domingo. Thursday: Retoma o número 4 referente ao dia da semana: quinta-feira. Time: Retoma a hora de abertura da barra atual. TimeToMinutes(Hora : Integer): Converte um horário em minutos. Tuesday: Retorna o número 2 referente ao dia da semana: Terça-feira. VolumeD(QuantidadeDiasAnteriores : Integer): Retoma o volume financeiro de determinado dia retroativo. VolumeM(QuantidadeMesesAnteriores : Integer): Retoma o volume financeiro de determinado mês retroativo. VolumeW(QuantidadeSemanasAnteriores : Integer): Retoma o volume financeiro de determinada semana anterior. VolumeY(QuantidadeAnosAnteriores : Integer): Retorna o volume financeiro de determinado ano anterior. Wednesday: Retoma o número 3 referente ao dia da semana: Quarta-feira. Year(Date : Integer): Retoma o ano de uma data específica. Candlestick 3WhiteSoldiers 3BlackCrows(Comprimento : Integer, Fator : Integer, var o3WhiteSoldiers : Integer, var o3BlackCrows : Integer): Identifica a ocorrência de dois tipos de candles: 3 White Soldiers e 3 Black Crows. C BulEng BearEngiComprimento — : Integer, var oBullishEnguifing : Integer, var oBearishEnguifing : Integer): Identifica a ocorrência de dois tipos de candles: Bullish Enguifing e Bearish Engutfing. C DojitPercentual : Integer): Identifica a ocorrência de um candle: Doji. C Hammer HangingMan(Comprimento : Integer, Fator : Integer, var oHammer : Integer, var oHangingMan : Integer): Identifica a ocorrência de dois tipos de candles: Hammer e Hanging Man. MornDoji EveDojiComprimento — : Integer, Percentual : Float, var oMorningDojiStar Integer, var oEveningDojiStar : Integer): Identifica a ocorrência de dois tipos de candles: Morning Doji Star e Evening Doji Star. MomStar EveStariComprimento — : Integer, var oMomingStar : Integer, var cEveningStar : Integer): Identifica a ocorrência de dois tipos de candles: Moming Star e Evening Star. C PierceLine DKCloud(Comprimento : Integer, var oPiercingline : Integer, var oDarkCloud : Integer): Identifica a ocorrência de dois tipos de candles: Piercing Line e Dark Cloud. C ShootingStar(lComprimento : Integer, Fator : Integer): Identifica a ocorrência de candles: C Shootingstar DiMaisDiMenos(Periodo : Integer): Exemplo de implementação do indicador DI+/DI-. IFR(Periodo : Integer): Exemplo de implementação do indicador IFR. Media(Periodo : Integer, TipoSerie : Serie): Exemplo de implementação do indicador Média Móvel(Aritmética). MediaExp(Periodo : Integer, TipoSerie : Serie): Móvel(Exponencial). Exemplo de implementação do indicador Média PaintVar: Exemplo de implementação de estratégia de coloração. WellesSum(Periodo : Integer, SerieReferencia : Serie, Offset : Integer): Exemplo de implementação do indicador WellesSum. Gráficas AvgPrice: Retoma a média entre Abertura, Máxima, Mínima, Fechamento de determinado candte. BarCount: Retoma a quantidade total de barras. CurrentBar: Retorna o índice atual da barra(candle). GetPlotColor(NumeroPiot : Integer): Retoma o valor numérico da cor de determinado Plot. GetPlotwWidth(NumeroPlot : Integer): Retoma

o valor da espessura de determinado Plot. Graphicinterval: Retoma o intervalo do gráfico. Graphicoffset: Retoma o offset do gráfico. HorizontalLine (Y : Float; color : Integer): Adiciona um estudo horizontal, em um indicador. LastBarOnChart:Retoma se é a última barra do gráfico. Leader: Retorna se ponto médio é maior que mínima ou maior que máxima de candle anterior. MaxBarsBack: Percorre a lista da série, a partir do primeiro candle criado (índice 0). MaxBarsForward: Percorre a lista da série, a partir do candle atual (índice 0). MedianPrice: Retoma a média entre a máxima e a mínima de determinado candle. NoPlot: NumeroPlot : Integer): Remove determinado Plot. PaintBar(Cor : Integer): Colore os candles e indicadores do gráfico. Plot: Dado : Float): Desenha o indicador de acordo com o gráfico. PlotN(Plot : Integer; Valor : Float): Adiciona um valor em um indicador. PlotText(Content : String; Color : Integer; Position : Integer; FontSize : Integer; dPrice : Float): Adiciona um texto a um indicador. Range: Retoma o valor de Máxima menos Mínima do determinado candle. RangeLeader: Verifica se a barra atual é Range Leader. RGB(Red : Integer, Green : Integer, Blue : Integer): Retorna a cor a partir dos parâmetros RGB. SetPlotColor(NumeroPlot : Integer, Cor : Integer): Altera a coloração de determinado Plot. SetPlotStyle(NumeroPlot : Integer; Estilo : Integer): Altera o estilo da linha de um plot específico. SetPlotType(Number : Integer; Type : Integer): Altera o tipo de gráfico de determinado plot. SetPlotWidth(NumeroPlot : Integer, Espessura : Integer): Altera a espessura de determinado Plot. TrueHigh: Retorna o maior entre o máximo da barra e o fechamento da barra anterior. TrueLow: Retorna o menor entre o mínimo da barra e o fechamento da barra anterior. TrueRange: Retoma o valor do indicador TrueRange. TrueRangeCustom(Maxima : Float, Minima : Float, Fechamento : Float): Retoma o TrueRange de acordo com os dados informados pelo usuário. TypicalPrice: — Retoma o valor médio entre a máxima, mínima e fechamento de determinado candle. VerticalLine(X : Integer; Color : Integer): Adiciona um estudo vertical em um indicador. WeightedClose: Retoma a média entre o ponto médio da barra e dois fechamentos. Indicadores AccAgressaoSaldo(TipoVolume : Integer): Retoma o valor do indicador TR - Acúmulo de Agressão - Saldo. AccuDist: Retorna o valor do indicador Acumulação/Distribuição. AccuDistW: Retorna o valor do indicador Acumulação/Distribuição Williams. AdaptiveMovingAverage(Período : Integer, FastSC : Integer, SlowSC : Integer): Retoma o valor do indicador Adaptive Moving Average (AMV). ADX(Período : Integer, PeríodoMedia : Integer): Retoma o valor do indicador ADX. AgressionVolBalance: - Retorna o valor do indicador TR - Volume de Agressão - Saldo. AgressionVolBuy: Retoma o valor do indicador TR - Volume de Agressão - Compra. AgressionVolSell: Retoma o valor do indicador TR - Volume de Agressão - Venda. ArmsEaseOfMov(Media : Integer, TipoMedia : Integer): Retoma o valor do indicador Arms Ease of Movement. AroonLin(Período : Integer)|Linha Desejada: Retoma o valor do indicador Aroon Linha. AroonOsc(Período : Integer): Retoma o valor do indicador Aroon Oscilador. AvgAgrBuySell(AlertaVariacoes : Integer, TipoVolume : Integer, TipoDesenho: Integer)|Linha : Integer: Retoma o valor do indicador TR - Agressão Média - Compra e Venda. AvgAgrTotal(AlertaVariacoes — :

Integer, Tipovolume : Integer, TipoDesenho: Integer)iLinha : Integerl: Retoma o valor do indicador TR - Agressão Média - Total. AvgSeparation(Período : Integer, TipoMedia : Integer): Retoma o valor do indicador Afastamento Médio. AvgTrueRange(Período : Integer, TipoMedia : Integer): Retorna o valor do indicador True Range. BalanceOfPower(Media:Integer, TipoMedia : Integer): Retoma o valor do indicador Balança do poder. BearPower(Periods: Integer): Retoma o valor do indicador Bear Power. BollingerBands(Desvio : Float, Media : Integer, TipoMedia : Integer)|Linh: linha da Banda de Bollinger de acordo com a linha desejada. Integerl: Retoma o valor da BollingerBandWíDesvio : Float, Media Bollinger Band Width. Integer, TipoMedia : Integer): Retoma o valor do indicador BollingerBPerc(Desvio : Float, Media : Integer, TipoMedia : Integer): Retorna o valor do indicador Boolinger b%. BuliPower(Período : Integer, PeríodoMedia : Integer, TipoMedia : Integer): Retoma o valor do indicador Bull Power. CCi(Período : Integer): Retorna o valor do indicador CC!. ChaikinMoneyFlow(Período — : Integer): Retoma o valor do indicador Chaikin Money Flow. ChaikinOsc(MediaLonga: Integer, Mes Oscilador Chaikin. iCurta : Integer): Retoma o valor do indicador ChainSetup: Retoma o dado do indicador ChainSetup. CohenPriceWave(Ticks : Integer): Desenvolvido por Rodrigo Cohen e equipe, indica a diferença absoluta de preços de cada swing do ativo. CohenWeisWave(Ticks : Integer): Desenvolvido por Rodrigo Cohen e equipe, indica a soma do volume de cada swing do ativo. Serve como um indicador de exaustão. ContadorDeCandle: Contabiliza e sinaliza de forma numérica e organizada no gráfico o número de cada candee. DarvasBox|Linha : Integerl: Retoma o valor do indicador Darvas Box. DecisionPoints(Tipo : integer, Linha : Integer): Retoma o valor do indicador Pontos de Decisão. DiDiIndex(MedRef : Integer, TipoMedRef : Integer, Med1 : Integer, TipoMed1 : Integer, Med2 : Integer, TipoMed2 : Integer)|Linha : Integerl: Retomna o valor do indicador DiDi Index. DIPDiM(Período : Integen)|Linha : Integerl: Retoma o valor do indicador DI+/DI- de acordo com a linha desejada. DonchianCH(Período : Integer)|Linha : Integerl: Retoma o valor do indicador Canal Donchian de acordo com a linha desejada. DTOscillator(PeríodoEstocastico : Integer, PeríodoSK : Integer, TipoSK : Integer, PeríodoSD : Integer, TipoSD : Integer)|Linha : Integerl: Retoma o valor do indicador DT Oscillator, de acordo com a linha desejada. Envelope(Percentual : Float, PeríodoMedia : Integer, TipoMedia : Integer)|Linha : Integerl: Retoma 1 valor do indicador Envelope. Euroinvest(Risco: Integer, ModoCalculo : Integer, Período : Integer, Desvi Boolean, UsarAtr : Boolean) Retoma o valor do indicador Euroinvest. : Float, UsarVWAP : FastStochastic(Período : Integer, PeríodoMedia : Integer, TipoMedi indicador Estocástico Rápido. Integer): Retoma o valor do A FinacialVollVolumeProjetado : Boolean, Agressores : Boolean): Retoma o Valor do volume financeiro. Forceindex(Período : Integer, TipoMedia : Integer): Retoma o valor do indicador Force Index. FrassonATR(Fator : Float, PeríodoMaxMir 1 valor do indicador Frasson ATR. : Integer, PeríodoATR : Integen)Linha : Integerl: Retora FrassonVH(Fator : Float, PeríodoMaxMin : Integer, PeríodoVH : IntegenLinha : Integerl: Retoma o valor do indicador Frasson VH. FuliStochastic(Período : Integer): Retoma o valor do indicador

Estocástico Pleno. FuraChao(Coeficiente : Float, Deslocamento : Integer): Retorna o valor do indicador Fura Chão. FuraTeto(Coeficiente : Float, Deslocamento : Integer): Retoma o valor do indicador Fura Teto. HeikinAshi(Media : Integer, TipoMes Integer)|Dado : Integer!: Retoma o valor do indicador HeikinAshi. HiLoActivator(Periodo : Integer)|Linha : Integer!: Retora o valor do Hilo Activator de acordo com a linha desejada. Histvolatility(Periodo : Integer, TipoMedia : Integer): Retoma o valor do indicador Volatilidade Histórica. HSIPeriodo : Integer): Retoma o valor do indicador IFR Índice de Força Harmônico (HSI). HullMovingAverage(Periodo — : Integer): Retoma o valor do indicador Hull Moving Average. IechimokuCloud(Tenkansen — : Integer, KijunSen : Integer, SenkousSpanB Integer)|Linha : Integer!: Retoma o valor do indicador Ichimoku Cloud. IFR(Período : Integer): Retoma o valor do indicador IFR. ImpliedVolatilityModeloTeorico : Boolean, TipoOpcao : Boolean): Retoma o cálculo do indicador Volatilidade Implícita, dependendo do tipo de cálculo utilizado. KeltnerCH(Desvio : Float, Período : Integer, TipoMedia : Integer)|Linha : Integer!: Retoma o valor do indicador Keltner Channels, conforme com a linha desejada. KVO(MediaLonga : Integer, MediaCurta : Integer, Sinal : Integer)|Dado : Integer!: Retoma o dado desejado do indicador KVO Linha & Histograma. LinearRegressionChannel(Periodo : Integer; UsarDesvioSuperior : Boolean; DesvioSuperior : Float; UsarDesvioInferior : Boolean; DesvioInferior : Float): Canal de regressão linear LSVolatilityindex Retoma o dado desejado do indicador L&S Volatility Index. MACD(MediaLonga : Integer, MediaCurta : Integer, Sinal : Integer)|Dado : Integer!: Retorna o dado desejado do indicador KVO Linha & Histograma. Media(Periodo : Integer, TipoSerie : Serie): Retoma o dado do indicador Média Móvel(Aritmética). MediaExp(Período : Integer, TipoSerie : Serie): Retoma o dado do indicador Média Móvel(Exponencial). retorna o valor do indicador Market Facilitation Index. MIMA(Período : Integer): Retoma o dado do indicador PhiCube - MIMA. Momentum(Periodo : Integer, Media : Integer, TipoMedia : Integer): Retoma o valor do indicador Momentum. MomentumStochastic(Periodo : Integer): Retoma o valor do indicador Momento Estocástico. MoneyFlow: — Retoma o valor do indicador Money Flow. MoneyFlowindex(Periodo : período utilizado. Integer): Retoma o valor do indicador Money Flow Index de acordo com o NelogicaBottomFinderIDado : Integer!: Retorna o valor do indicador Nelogica Bottom Finder de acordo com a linha Desejada. NelogicaPullBackFinder|Dado acordo com a linha desejada. Integer!: Retoma o valor do indicador Nelogica PullBack Finder de NelogicaWeisWave(Periodo — : Integer): Retoma o valor do indicador Nelogica Weis Wave, conforme o período desejado. OBV: Retorna o valor do indicador OBV. OBVAvg: Retoma o valor do indicador OBV Ponderado. OnBalanceTR: Retoma o valor do indicador On Balance True Range. OpenDaily(DaysBack : Integer): Retoma o dado de abertura, conforme o deslocamento especificado. Openinterest:: - Retoma o valor do indicador Contratos em aberto. ParabolicSAR(Fator : Float, Limite float): Retoma o valor do indicador SAR Parabólico. Phibo(Período : Integer): Retoma o valor do indicador PhiCube - Phibo Line. Pivot(Norma! : Boolean, TresLinhas : Boolean)|Linha acordo

com a linha selecionada. Integerl: Retoma o valor do indicador Pivot, de PriceNery: Retoma os dados do indicador Price Nery. PriceOsc(Media1 : Integer, TipoMedia? : Integer, Media2 : Integer, TipoMedia2 : Integer) : Retoma o valor do indicador Oscilador de Preços. PriceVolumeTrend: — Retoma o valor do indicador Tendência Preço/Volume. PriorCote(Dado : Integer): Retoma o valor do indicador Prior Cote, de acordo com o dado desejado. PTAX: Retoma o valor do indicador TR - PTAX. PTAXFuturo: — Retoma o valor do indicador TR - PTAX Futuro, QuantityVol(VolumeProjetado - : Boolean, Agressores : Boolean): Retoma o valor do indicador Volume. Quantidade. Rafi: Retoma o valor do indicador Rafi RaviMediaCurta : Integer, MediaLonga : Integer): Retoma o valor do indicador Ravi, de acordo com os períodos das médias desejadas. RBG: Retoma os dados do indicador RBG. RenkoVTwo(Período : Integer, Abertura : Float, Deslocamento : Integer): Retoma o dado do indicador RenkoV2, conforme a linha desejada. ROC(Período : Integer, Media : Integer, TipoMedia : Integer): Retoma o valor do indicador ROC. RSIKPeríodo : Integer, Tipo : Integer): Retorna o valor do indicador IFR(RSI!). RsiStochastic(Período : Integer): Retoma o valor do indicador IFR Estocástico. SafeZoneDownTrend(Multiplicador — : Float, Período : Integer, Deslocamento : Integer): Retoma o valor do indicador Stop SafeZone DownTrend. SafeZoneUpTrend(Multiplicador : Float, Período : Integer, Deslocamento : Integer): Retoma o valor do indicador Stop SafeZone UpTrend. Santo(Período : Integer)|Linha : Integerl: Retoma o valor do indicador PhiCube - Santo. SlowStochastic(Período : Integer): Retoma o valor do indicador Estocástico Lento. StopATR(Desvio : Float, Período : Integer, TipoMedia : Integer)|Dado : Integerl: Retoma o valor do indicador Stop ATR, de acordo com o dado desejado. TendencyTrackerDias : Integer): Retoma o valor do indicador Rastreador de Tendências, conforme o período desejado. Tison(Fator : Float, Media : Integer): Retorna o valor do indicador Tillson's T3 Moving Average. TimeAgrBuySeli(AlertaVariacoes : Integer): Retoma o valor do indicador TR - Tempo Agressão - Compra. TimeAgrTotal(AlertaVariacoes - : Integer): Retoma o valor do indicador TR - Tempo Agressão - Total. TopBottomDetectorPeríodo — : Integer): Retoma o valor do indicador Detector de Topos e Fundos. Trades: Retoma o valor do indicador Negócios. TrendCloud (DiasRetroativos : Integer; CorCompra : Integer; VWAP : Integer; CorVenda : Integer): Retoma o valor do indicador TrendCloud, de acordo com os parâmetros desejados. TrendSniper(DiasRetroativos : Integer; VWAP : Integer; CorCompra : Integer; CorVenda : Integer; Reversao : Integer): Retorna o valor do indicador TrendSniper, de acordo com os parâmetros desejados. TRIX(Media : Integer, TipoMedia : Integer): Retorna o valor do indicador TRIX. TRIXM(Media : Integer, TipoMedia : Integer): Retoma o valor do indicador TRIXM. TwoMVAggression: — Retorna o dado do indicador 2MV Agressão. TwoMVPower(Período1 : Integer, Período2 : Integer, Período3 valor do indicador 2MV Power. Integer, Media : Integer): Retoma o TwoMVStandard:Retoma o dado do indicador 2MV Padrão. UltimateOscillator(PeríodoCurto: Integer, PeríodoMedio : Integer, PeríodoLongo : Integer): Retoma o valor do Ultimate Oscillator desenvolvido por Larry

Williams. Valerie(Período : Integer; Offset : Integer): Retoma o valor do indicador Valerie, conforme os parâmetros desejados. VSS(Multiplicador : Float, Media : Integer, Deslocamento : Integer): Retoma o valor do indicador VSS. VWAP(Período : Integer): Retoma o valor do indicador VWAP, conforme a periodicidade desejada. VWAPDate(Date : Integer; Time : Integer)(Período : Integer): Retoma o preço médio ponderado pelo volume, a partir de uma data e horário específicos. VWAPMonthiy: Retoma o dado do indicador VWMA Mensal. VWAPWeekly: Retoma o dado do indicador VWMA Semanal. VWMA(Período : Integer): Retoma o dado do indicador VWMA, conforme o período desejado. Waverage(TipoSerie : SeriePeríodo, Período : Integer): Retoma o dado do indicador Média Móvel(Ponderada). WellesSum(Período WellesSum. Integer, SerieReferencia : Serie, Offset : Integer): Retorna o dado do indicador Williams(Período : Integer): Retorna o valor do indicador Wiliams %R, de acordo com o período desejado. xAverage(TipoSerie : SeriePeríodo, Período : Integer): Retoma o dado do indicador Média Móvel(Exponencial). Livro AskPrice: Retoma o preço da melhor oferta de venda. AskSize: Refoma a quantidade da melhor oferta de venda. BidPrice: Retoma o preço da melhor oferta de compra. BidAsk: Retoma a quantidade da melhor oferta de compra. BookSpread: — Retoma a diferença entre o topo do livro. GetAsset(Asset : Ativo = ”): Retorna o ticker do ativo GetFeed(Asset : Ativo = “): Retorna o código textual do Feed do ativo. ISBMF: Verifica se o ativo pertence ao segmento BMF. Lote: Retorna a quantidade de contratos referente ao lote. MinPricelncement:: — Retoma o incremento. Matemáticas ABS(Valor : Float): Retoma o valor absoluto(sem sinal). Arctangent(Número : Float): Retoma o arcotangente(em graus) de um número. Ceiting(Número : Float): Retorna o menor inteiro maior que um número específico. Combination(Número : Integer, QtdGrupos : Integer): Retoma a quantidade de combinações, considerando um conjunto específico de números. Cos(Valor : Float): Retorna o valor de um Cosseno em radianos. Cosine(Valor : Float): Retoma o valor de um Cosseno em graus. Cotangentí(Valor : Float): Retoma o valor de uma Cotangente em graus. CumíSerieDeDados : Serie): Acumula o valor de uma série de dados. ExpíValor : Float): Retoma a enésima potência do número de Euler. Expvalue(Valor : Float): Retoma o valor exponencial de um determinado número(e^x). ExtremePriceRatio:: = Obtém o ratio das extremidades de um número determinado de barras. Factorial(Valor : Float): Retoma o fatorial do valor estabelecido ($1^{\circ}2^{\circ} \dots n$). FastD(Período : Integer): Retoma o valor de FastD do Oscilador Estocástico. FastK(Período : Integer): Retoma o valor de FastK do Oscilador Estocástico. FastKCustom(PrecosH : Serie, PrecosL : Serie, PrecosC : Serie, Período : Integer): Retoma o valor de FastK a partir de preços específicos. Floor(Valor : Float): Retoma o maior inteiro, menor que um número específico. FracPortion(Valor : Float): Retoma a parte fracionário de um número. GCD(Valor1 : Float, Valor2 : Float): Retoma o maior divisor comum entre dois números. HarmonicMean(SerieDados — : Serie, Período : Integer): Retoma a média harmônica de uma série de dados baseada em um período. Highest(SerieS Dados : Serie, Período : Integer): Retoma o maior valor da série dentro do período. HighestBar(SerieDeDados

: Serie, Período : Integer): Retoma o índice do maior valor da série no período. IntPortion(Valor : Float): Retoma a parte inteira de um número. LogValor : Float): Retoma o logaritmo natural(in) de um número de um número. Lowest(SerieDeDados : Serie, Período : Integer): Retoma o menor valor da série dentro no período. LowestBar(SerieDeDados : Serie, Período : integer): Retoma o índice da barra com o menor Valor da série no período. MidPoint(SerieDados : Serie, Período : Integer): Retoma a média entre o maior e o menor valor encontrados no período. MinutesintoWeek(DiaLimite — : Integer, HoraLimite : integer): Retoma o valor de minutos desde a meia noite em horas. MinutesToTime(Minutos : Integer): Retoma o valor de minutos desde a meia noite em horas. ModDividendo : Integer, Divisor : Integer): Retoma o resto da divisão entre dois números. Neg(Numero : Float): Retoma o valor negativo de um determinado número. NumUunits(Amnt : Integer, MinLot : Integer): Retoma o número de contratos/ações de um certo investimento. PercentChange(SerieDados : Serie, Período : Integer): Retoma a alteração percentual no preço do candie atual. PercentR(Comprimento : Integer): Retoma uma porcentagem de onde o preço atual está, relacionado com 2 faixa de negociação avaliada. Permutation(Numero : Integer, NumeroObjetos : Integer): Calcula um número de permutações. Pos(Valor : Float): Retoma o valor absoluto(sem sinal). Power(Base : Float, Expoente : Integer): Eleva valores nas determinadas potências. PriceOscillator(SerieDados : Serie, ComprimentoRapido : Integer, ComprimentoLento : Integer): Retoma o valor do indicador Price Osciliator. Randomí(Limite : Integer): Retoma um valor aleatório de 0 até o limite. RateOfChange(SerieDados : Serie, Período : Integer): Retoma a variação percentual de uma série. Roundí(Valor : Float): Arredondamento de número. Round2Fraction(Valor : Float): Arredonda o número para o valor mais próximo de um múltiplo do incremento mínimo de um ativo. Sign(Valor : Float): Retoma um número inteiro, baseado no sinal de um número. Sin(Valor : Float): Retoma o valor de Seno em radianos. Sine(Valor : Float): Retoma o valor de Seno em graus SlowD(Período : Integer): Retoma o valor SlowD do Oscilador Estocástico. SlowK(Período : Integer): Retoma o valor SlowkK do Oscilador Estocástico. Sqrtí(Valor : Float): Retoma raiz quadrada dos valores. Square(Valor : Float): Retorna o valor de um número ao quadrado. StdDevs(SerieDados : Serie, Período : Integer): Calcula o desvio padrão de uma série de dados em um determinado período. Summation(SerieDados : Serie, Período : Integer): Acumula o valor do preço de um determinado número de barras. TangentValor : Float): Retoma a tangente de um número em graus. Triaverage(SerieDados : Serie, Período : Integer): Calcula a média triangular de uma série de dados, dentro de um certo período. Ulcerindex(SerieDados : Serie, Período : Integer): Retoma o valor do indicador Ulcerindex. Volatility(Período : Integer): Retorna a volatilidade de determinado período. VolumeOsc(PeriodoMediaRapida — : Integer, PeriodoMediaLenta : Integer): Retorna a diferença entre a média aritmética rápida e a média aritmética lenta da série Volume. VolumeROC(Periodo : Integer): Retoma VolumeROC baseado em um número de barras. Opções Delta(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100

Integer): Retoma o valor de Deita. Gamma(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 Integer): Retoma o valor de Gamma. Rho(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 Integer): Retoma o valor de Gamma. Theta(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 Integer): Retoma o valor de Theta. Vega(DaysLeft : Integer, StrikePr : Float, AssetPr : Float, Rate100 : Float, Volty100 Integer): Retoma o valor de Vega. Screening Select: Seleciona um ativo para mostrar no Screening. : Float, PutCall : : Float, PutCall : : Float, PutCall : : Float, PutCall Alarme Função Alert A função Alert tem como finalidade gerar um alarme ao usuário. Sintaxe: AlertCor : Integer) Parâmetros: Cor: Determina a cor desejada para o popup de notificação, no momento de execução do alarme. Observação: Uma cor pode ser determinada a partir de uma função RGB, ou através de uma String com o nome da cor. Retorno: Void: Sem retorno. Back-Testing Função BuyAtMarket A função BuyAtMarket tem como funcionalidade realizar uma ordem de compra à mercado. Observação: A parametrização é opcional, caso não seja determinada, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyAtMarket(Quantidade : Integer Parâmetros: Sem parâmetros. Retorno: Sem retorno. Função BuyLimit Descrição A função BuyLimit possui como finalidade enviar uma ordem de compra do tipo limite. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyLimit(Preco : Float, Quantidade : Integer Parâmetros: Preço: Preço para a inserção da ordem. Quantidade: Quantidade da ordem. Retorno: Sem retorno. Função BuyPosition Descrição: A função BuyPosition retorna o tamanho da posição da compra. Sintaxe: BuyPosition Parâmetros: Sem parâmetros. Retorno: Integer Função BuyPrice A função BuyPrice retorna o preço de compra da posição. Sintaxe: BuyPrice Parâmetros: Sem parâmetros. Retorno: Float Função BuyStop A partir da função BuyStop, é possível criar uma ordem do tipo Stop, onde o preço stop determina o preço de gatilho, e o limite especifica até qual preço a ordem poderá ser executada. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyStop(Stop : Float, Limite : Float, Quantidade : Integer Parâmetros: Stop: Valor do tipo Float que será o gatilho da ordem; Limite: Valor do tipo Float que será o limite de preço aceito para execução. Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função BuyToCoverAtMarket A função BuyToCoverAtMarket realiza o envio de uma ordem de compra à mercado, caso exista uma posição de venda em aberto. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyToCoverAtMarket(Quantidade : Integer = ") Parâmetros: Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função BuyToCoverLimit A função BuyToCoverLimit possui como finalidade enviar uma ordem de compra, do tipo limite, para finalizar a operação. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyToCoverLimit(Preco

: Float, Quantidade : Integer Parâmetros: Preço: Preço para a inserção da ordem. Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno.

Função BuyToCoverStop A função BuyToCoverStop tem como funcionalidade enviar uma ordem do tipo Stop de compra, caso exista uma posição de venda no determinado ativo. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: BuyToCoverStop(Stop : Float, Limite : Float, Quantidade : Integer Parâmetros: Stop: Valor que será o gatilho da ordem. Limite: Valor que será o limite do preço aceito para execução. Quantidade: Quantidade da ordem. Retorno: Sem retorno.

Função CancelPendingordersrs A função CancelPendingordersrs possui como recurso efetuar o cancelamento de todas ordens enviadas até o momento de sua chamada, no fluxo atual de execução, e as ordens abertas. CancelPendingordersrs Sem retorno.

Função ClosePosition A função ClosePosition envia ordens para encerrar a posição. Esta função cancela todas ordens enviadas até o momento de sua chamada, no fluxo atual de execução, e as ordens abertas. Após confirmação do cancelamento das ordens, será enviada uma ordem para fechar a posição. Sintaxe: ClosePosition Parâmetros: Sem parâmetros. Retorno: Void: Sem retorno.

A função HasPendingOrders retoma se há ordens pendentes. Sintaxe: HasPendingOrders Parâmetros: Sem parâmetros. Retorno: Boolean

A função HasPosition retoma verdadeiro se a posição não é zero. Sintaxe: HasPosition Parâmetros: Sem parâmetros. Retorno: Boolean

Função IsBought: Descrição A função IsBought tem como funcionalidade determinar se há uma posição de compra em aberto. Sintaxe: IsBought Parâmetros: Sem parâmetros. Retorno: Boolean: False - Não há posição em aberto. True - Há posição em aberto.

Função IsSold A função IsSold tem como funcionalidade determinar se há uma posição de venda em aberto. Sintaxe: IsSold Parâmetros: Sem parâmetros. Retorno: Boolean: False - Não há posição em aberto. True - Há posição em aberto.

Função MyPrice A função MyPrice retorna a média entre a máxima, mínima e fechamento. Sintaxe: MyPrice Parâmetros: Sem parâmetros. Retorno: Float

A função Position retorna o tamanho da posição, positivo para compra e negativo para venda. Parâmetros: Sem parâmetros. Retorno: Integer

A função Price retorna o preço de compra ou venda da posição, dependendo se estiver comprado ou vendido. Parâmetros: Sem parâmetros. Retorno: Float

Função ReversePosition A função ReversePosition tem como funcionalidade realizar o envio de ordens, a fim de inversão da posição. Esta função cancela todas ordens enviadas até o momento de sua chamada, no fluxo atual de execução, e as ordens abertas. Após confirmação do cancelamento das ordens, será enviada uma ordem para reverter a posição. Sintaxe: ReversePosition Parâmetros: Sem parâmetros. Retorno: Void: Sem retorno.

Função SellPos: A função SellPosition retorna o tamanho da posição de venda. Sintaxe: SellPosition Parâmetros: Sem parâmetros. Retorno: Integer

Função SellPrice A função SellPrice retorna o preço de venda da posição. Sintaxe: SellPrice Parâmetros: Sem parâmetros. Retorno: Float

Função SellShortAtMarket A função SellShortAtMarket tem como funcionalidade o envio de uma ordem à mercado de venda. Observação: O parâmetro de quantidade é opcional,

caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellShortAtMarket(Quantidade) Parâmetros: Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função SellShortLimit A função SellShortLimit possui como finalidade enviar uma ordem de venda do tipo limite. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellShortLimit(Preco : Float, Quantidade : Integer = ") Parâmetros: Preco: Preço para a inserção da ordem. Quantidade: Quantidade da ordem. Retorno: Sem retorno. Função SellShortStop A função SellShortStop tem como finalidade enviar uma ordem de venda do tipo stop. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellShortStop(Stop : Float, Limite : Float = Market, Quantidade : Integer Parâmetros: Stop: Valor do tipo Float que será o gatilho da ordem; Limite: Valor do tipo Float que será o limite de preço aceito para execução. Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função SellToCoverAtMarket A função SellToCoverAtMarket tem como funcionalidade realizar o envio de uma ordem de venda à mercado, caso exista uma posição de compra. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellToCoverAtMarket Parâmetros: Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função SellToCoverLimit A função SellToCoverLimit possui como finalidade enviar uma ordem de venda, do tipo limite, para finalizar a operação. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellToCoverLimit(Preco : Float, Quantidade : Integer = ") Parâmetros: Preco: Preço para a inserção da ordem. Quantidade: Quantidade da ordem. Retorno: Sem retorno. A função SellToCoverStop tem como funcionalidade enviar uma ordem do tipo stop de venda, caso exista uma posição de compra no ativo. Observação: O parâmetro de quantidade é opcional, caso não seja especificado, a ordem será inserida considerando o lote mínimo do ativo. Sintaxe: SellToCoverStop(Stop : Float, Limite : Float, Quantidade : Integer = ") Parâmetros: Stop: Valor que será o gatilho da ordem. Limite: Valor que será o limite do preço aceito para execução. Quantidade: Quantidade da ordem. Retorno: Void: Sem retorno. Função SendOrder A partir da função SendOrder, pode-se enviar ordens customizadas, determinando o lado, tipo e quantidade. Sintaxe: SendOrder(Lado : Integer, Tipo : Integer, Quantidade : Integer, Limite : Float, Stop : Float) Parâmetros: Lado: Determina o lado da ordem: isBuy - Compra isSell - Venda Tipo: Tipo da ordem: atMarket - À mercado atLimit - Limite atStopLimit - Stop Quantidade: Quantidade de contratos. Limite: Limite do preço aceito para execução. Stop: Valor que será o gatilho da ordem. Retorno: Void: Sem retorno. Função DailyResult Descrição A partir da função DailyResult, pode-se verificar o resultado diário das operações. Observação: O parâmetro OpenResult é opcional, o valor default é verdadeiro. Sintaxe: DailyResult(OpenResult : Boolean = True) Parâmetros: OpenResult : Se deve retornar o resultado aberto: Verdadeiro: Retorna o resultado

fechado + aberto. Falso: Retoma apenas o resultado fechado. Retorno: Float
 Depuração Função ConsoleLog Descrição: A função ConsoleLog imprime uma
 string no terminal de console para ajudar na depuração do código. Sintaxe:
 ConsoleLog(Content : String, Color : Integer = cIBlack) Parâmetros: Content:
 String que irá ser impressa no console de depuração. Color: Cor da string
 impressa. Retorno: Void. Sem retorno. Calendário BarAnnualization A
 função BarAnnualization retoma o fator de anualização(raiz quadrada) baseado
 no intervalo da barra(diário = 365, semanal = 52, mensal = 12). Sintaxe:
 BarAnnualization Parâmetros: Sem parâmetros. Retorno: Float Função Bar-
 Duration A função BarDuration retorna, em minutos, a duração da barra atual.
 Sintaxe: BarDuration Parâmetros: Sem parâmetros. Função BarDurationF A
 função BarDurationF retorna, em minutos, a duração da barra atual. Sintaxe:
 BarDurationF Parâmetros: Sem parâmetros. Retorno: Float Função Bartype
 A função Bartype retorna um código numérico referente ao período utilizado.
 Parâmetros: Sem parâmetros. Retorno: Inteiro: 0- outros 1-Intraday 2- Diário
 3- Semanal 4 - Mensal Função CalcDate A função CalcDate retoma um valor
 o qual representa uma data deslocada, obtida ao adicionar ou subtrair dias
 de uma data de referência. Observação: Datas são representadas pelo tipo de
 dado "Integer", no formato: 1AnoMêsDia. Sintaxe: CalcDate(DataReferencia :
 Integer, DiasDeslocamento : Integer) Parâmetros: DataReferencia: Determina
 a data que será utilizada como base para o deslocamento. DiasDeslocamento:
 Determina quantos dias serão adicionados ou subtraídos da data de refer-
 ência. Retorno: Integer Função CalcTime A função CalcTime retorna um
 valor o qual representa uma hora deslocada, obtida ao adicionar ou subtrair
 minutos de uma hora de referência. Observação: Horas são representadas
 pelo tipo de dado HorasMinutos. Integer", no formato(24 horas): Sintaxe:
 CalcTime(HoraReferencia : Integer, MinutosDeslocamento : Integer) Parâmet-
 ros: HoraReferencia: Determina a hora que será utilizada como base para o
 deslocamento; MinutosDeslocamento: Determina quantos minutos serão adi-
 cionados ou subtraídos da hora de referência. Retorno: Integer Função CloseD
 A função CloseD tem como finalidade retornar o valor de fechamento de um
 número determinado de dias atrás. Sintaxe: CloseD(QuantidadeDiasAnteriores
 : Integer) Parâmetros: QuantidadeDiasAnteriores: — Determina a quantidade
 desejada de dias anteriores. Retorno: Float Função CloseM A função CloseM
 tem como finalidade retornar o valor de fechamento de um número determinado
 de meses atrás. Sintaxe: CloseM(QuantidadeMesesAnteriores : Integer)
 Parâmetros: QuantidadeMesesAnteriores: Determina a quantidade desejada
 de meses anteriores. Retorno: Float Função CloseW A função CloseW tem
 como finalidade retornar o valor de fechamento de um número determinado
 de semanas atrás. Sintaxe: CloseW(QuantidadeSemanasAnteriores : Integer)
 Parâmetros: QuantidadeSemanasAnteriores: Determina a quantidade desejada
 de semanas anteriores. Retorno: Float Função CloseY A função CloseY tem
 como finalidade retornar o valor de fechamento de um número determinado
 de anos atrás. Sintaxe: CloseY(QuantidadeAnosAnteriores — : Integer)
 Parâmetros: QuantidadeAnosAnteriores: — Determina a quantidade desejada
 de anos anteriores. Retorno: Float Função CurrentAssetDate A função

CurrentAssetDate retorna a data atual do ativo no gráfico. Observação: Datas são representadas pelo tipo de dado "Integer", no formato: 1AnoMêsDia. Sintaxe: CurrentAssetDate Parâmetros: Sem parâmetros. Retorno: Integer