

FUNCIÓN CURSO_FILTROS

- Diferentes filtros neutrales para aplicar en sistema tendencial.

CÓDIGO EASYLANGUAGE (CREEMOS QUE SIRVE COMO PSEUDOCÓDIGO, PERO SI ALGUIEN NO COMPRENDE ALGO PUEDE PREGUNTARLO EN DISCORD)

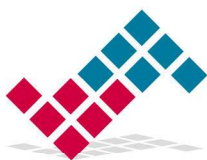
```
// Filtros neutrales, son independientes de si operamos lado largo o corto
input:      numeropattern(numeric简单), // 1 hasta 16
           Nivel_ADX(numeric简单),
           Nivel_ATR(numeric简单),
           Nivel_Stddev(numeric简单),
           Barras(numeric简单),
           Rango1(numeric简单),
           Rango2(numeric简单);

var:
           FiltroADX(False),
           Filtro_ATR(False),
           oDMIPlus(0),
           oDMIMinus(0),
           oDMI(0),
           oADX(0),
           oADXR(0),
           oVolty(0);

Value1 = DirMovement (H, L, C, Barras, oDMIPlus, oDMIMinus, oDMI, oADX,
oADXR, oVolty);

Switch(numeropattern) Begin
  case 1:
  begin
    If Nivel_ADX > 0 Then
    Begin
      FiltroADX = oDMIPlus > Nivel_ADX;
    end Else
    Begin
      FiltroADX = True;
    End;
    Curso_Filtros = FiltroADX;
  end;

  Case 2:
  begin
    If Nivel_ATR > 0 Then
      Filtro_ATR = AvgNormalizedTrueRange(Barras) < (Nivel_ATR)
    Else
      Filtro_ATR = True;
    Curso_Filtros =Filtro_ATR;
  end;
```



```
Case 3:
    Curso_Filtros = NarrowRange(RANGO1);

Case 4:
    Curso_Filtros = widespread(RANGO2);

case 5:
begin
    If Nivel_ADX > 0 Then
    Begin
        FiltroADX = oDMIPlus < Nivel_ADX;
    end Else
    Begin
        FiltroADX = True;
    End;
    Curso_Filtros = FiltroADX;
end;

Case 6:
begin
    If Nivel_ATR > 0 Then
        Filtro_ATR = AvgNormalizedTrueRange(Barras) > (Nivel_ATR)
    Else
        Filtro_ATR = True;
    Curso_Filtros = Filtro_ATR;
end;

Case 7: //Stdev
begin
    If Nivel_Stddev > 0 Then
        Curso_Filtros = Stddev(C, Barras) > Nivel_Stddev
    Else
        Curso_Filtros = True;
    end;

case 8: //GAP
Begin
    Curso_Filtros = 0 of data2 < L[1] of data2 OR 0 of data2 > H
of data2;
end;

case 9: //Inside Bar
Begin
    Curso_Filtros = H of data2 < H[1] of data2 AND L of data2 >
L[1] of data2;
end;

case 10: //Rangos Decrecientes
Begin
    Curso_Filtros = range[1] of data2 < range[2] of data2 and
range[2] of data2 < range[3] of data2;
end;
```

```
case 11: //Inside Bar Ayer
Begin
    Curso_Filtros = H[1] of data2 < H[2] of data2 AND L[1] of
data2 > L[2] of data2;
end;

case 12: //Outside Bar
Begin
    Curso_Filtros = H[0] of data2 > H[1] of data2 AND L[0] of
data2 < L[1] of data2;
end;

case 13: //Outside Bar Ayer
Begin
    Curso_Filtros = H[1] of data2 > H[2] of data2 AND L[1] of
data2 < L[2] of data2;
end;

case 14: // Rango barra anterior menor que barra previa
Begin
    Curso_Filtros = range[1] of data2 < range[2] of data2;
end;

case 15: // Rango barra anterior mayor que barra previa
Begin
    Curso_Filtros = range[1] of data2 > range[2] of data2;
end;

case 16:    Curso_Filtros = true;

case >16:    Curso_Filtros = false;

end;
```