SERSAN SISTEMAS
ALGORITHMIC TRADING

BUSCADOR DE ENTRADAS Y SALIDAS

- Este código sirve para probar diferentes combinaciones de entradas y salidas.
- Hay un total de 23 entradas y 35 salidas, aunque son solo algunos ejemplos para que podáis extenderlo o modificarlo con vuestras propias ideas.

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

```
inputs:
  Entrada (0), //elegir de 0 a 23
  Per_E (15),

  Salida (2), //elegir de 0 a 35
  Per_S (15);

vars:
  MP (0);

MP = MarketPosition;

switch (Entrada)
Begin
  case 0: //No entries
  Begin

  End;

  case 1: //Momentum
  Begin
    If Close > Close[Per_E] then
      Buy ("E01_LE") next bar at market;

    If Close < Close[Per_E] then
      SellShort ("E01_SE") next bar at market;
  End;

  case 2: //Breakout
  Begin
    input:
      E02_Precio (High);

    var:
      Precio_Corto (0);

    IfE02_Precio = High Then
      Precio_Corto = Low
```

```
  Else
    Precio_Corto = Close;

  If E02_Precio > Highest(E02_Precio, Per_E)[1] then
    Buy ("E02_LE") next bar at market;

  If Precio_Corto < Lowest(Precio_Corto, Per_E)[1] then
    SellShort ("E02_SE") next bar at market;
end;

case 3:  //Cruce de medias simples
Begin
  If Close crosses above Average(Close, Per_E) then
    Buy ("E03_LE") next bar at market;

  If Close crosses below Average(Close, Per_E) then
    SellShort ("E03_SE") next bar at market;
end;

case 4: //Bollinger Band MR
Begin
  input:
    E04_Desv (2);

  If Close crosses above BollingerBand(Close, Per_E, -E04_Desv)
  then //largos cuando recupera banda inferior
    Buy ("E04_LE") next bar at market;

  If Close crosses below BollingerBand(Close, Per_E, E04_Desv)
  then //cortos cuando pierde banda superior
    SellShort ("E04_SE") next bar at market;
end;

case 5: //Volatilidad
Begin
  input:
    E05_ATRs (1.5);

  If Close > (Close[1] + AvgTrueRange(Per_E) * E05_ATRs) then
    Buy ("E05_LE") next bar at market;

  If Close < (Close[1] - AvgTrueRange(Per_E) * E05_ATRs) then
    SellShort ("E05_SE") next bar at market;
end;

case 6: //Bollinger Band trend
Begin
  input:
    E06_Desv (2);
```

```
  If Close crosses above BollingerBand(Close, Per_E, E06_Desv)
  then  //largos cuando recupera banda superior
    Buy ("E06_LE") next bar at market;

  If Close crosses below BollingerBand(Close, Per_E, -E06_Desv)
  then  //cortos cuando pierde banda inferior
    SellShort ("E06_SE") next bar at market;
End;

case 7: //Donchian clásico
Begin
  Buy ("E07_LE") next bar at Highest(High, Per_E) stop;
  Sellshort ("E07_SE") next bar at Lowest(Low, Per_E) stop;
End;

case 8: //Key Reversal
Begin
  If Low < Lowest(Low, Per_E)[1] and Close > Close[1] then
    Buy ("E08_LE") next bar at market;

  If High > Highest(High, Per_E)[1] and Close < Close[1] then
    Sell Short ("E08_SE") next bar at market;
End;

case 9: //Nos incorporamos a la tendencia primaria tras un
pullback
Begin
  Input:
    E09_n (3); //multiplicador para el periodo de largo plazo

  If Close > Close[Per_E * E09_n] and Close < Close[Per_E] then
    Buy ("E09_LE") next bar at market;

  If Close < Close[Per_E * E09_n] and Close > Close[Per_E] then
    Sellshort ("E09_SE") next bar at market;
End;

case 10: //Entramos en el pullback contra la tendencia primaria
Begin
  Input:
    E10_n (3); //multiplicador para el periodo de largo plazo

  If Close < Close[Per_E * E10_n] and Close > Close[Per_E] then
    Buy ("E10_LE") next bar at market;

  If Close > Close[Per_E * E10_n] and Close < Close[Per_E] then
    Sellshort ("E10_SE") next bar at market;
End;
```

```
case 11: //Breakout + momentum
Begin
  Input:
    E11_ATRs (1.5),
    E11_Per (10);

  var: canal (0);

  canal =(highest(high, E11_Per) + lowest(low, E11_Per)) / 2;

  If close > close[Per_E] then
    Buy ("E11_LE") next bar at canal + E11_ATRs *
    AvgTrueRange(E11_Per) stop;

  If close < close[Per_E] then
    SellShort ("E11_SE") next bar at canal - E11_ATRs *
    AvgTrueRange(E11_Per) stop;
End;

case 12: //RSI entrada saliendo de sobrecompra/sobreventa MR
Begin
  input:
    E12_minRSI (30),
    E12_maxRSI (70);

  var:
    RSI_value (0),
    Cond_Long (false),
    Cond_Short (false);

  RSI_value = RSI(Close, Per_E);

  Cond_Long = RSI_value crosses over E12_minRSI; //largos en cruce
  arriba de banda inferior
  Cond_Short = RSI_value crosses under E12_maxRSI; //cortos en
  cruce abajo de banda superior

  If MarketPosition <> 1 and Cond_Long then
    Buy ("E12_LE") next bar at market;

  If MarketPosition <> -1 and Cond_Short then
    SellShort ("E12_SE") next bar at market;
End;

case 13: //RSI entrada en tendencia
Begin
  input:
    E13_minRSI (30),
```

```
    E13_maxRSI (70);

  RSI_value = RSI(Close, Per_E);

  Cond_Long = RSI_value crosses over E13_maxRSI; //largos en cruce
  arriba de banda superior
  Cond_Short = RSI_value crosses under E13_minRSI; //cortos en
  cruce abajo de banda inferior

  If MarketPosition <> 1 and Cond_Long then
    Buy ("E13_LE") next bar at market;

  If MarketPosition <> -1 and Cond_Short then
    SellShort ("E13_SE") next bar at market;
End;

case 14: //RSI entrada saliendo de sobrecompra/sobreventa MR
Begin
  input:
    E14_minRSI (30),
    E14_maxRSI (70);

  RSI_value = RSI(Close, Per_E);

  Cond_Long = RSI_value crosses under E14_minRSI; //largos en
  cruce abajo de banda inferior
  Cond_Short = RSI_value crosses over E14_maxRSI; //cortos en
  cruce arriba de banda superior

  If MarketPosition <> 1 and Cond_Long then
    Buy ("E14_LE") next bar at market;

  If MarketPosition <> -1 and Cond_Short then
    SellShort ("E14_SE") next bar at market;
End;

case 15: //Nos incorporamos a la tendencia primaria tras un
pullback (otra forma)
Begin
  inputs:
    E15_Per (15);

    Condition1 = close > average(close, Per_E);
    Condition2 = close < Percentile(.10, Close, E15_Per);
    Condition3 = close < close[1] and close[1] < close[2] and
    close[2] < close[3];

    If condition1 and (condition2 or condition3) then
      Buy ("E15_LE") next bar at market;
```

```
    Condition4 = close < average(close, Per_E);
    Condition5 = close > Percentile(.90, Close, E15_Per);
    Condition6 = close > close[1] and close[1] > close[2] and
    close[2] > close[3];

    If condition4 and (condition5 or condition6) then
      SellShort ("E15_SE") next bar at market;
End;

case 16: //Entrada en velas muy expansivas
Begin
  Inputs:
    E16_Desv (2);

  value1 = (E16_Desv * stddev(range, Per_E)) + (average(range,
  Per_E));

  if range > value1 and close > close[Per_E] then
    Buy ("E16_LE") next bar at market;

  if range > value1 and close < close[Per_E] then
    SellShort ("E16_SE") next bar at market;
End;

case 17: //entrada en una hora buscando expansión con un dato
Begin
  Input:
    E17_Time (828); //hora previa que colocarías las órdenes

  Var:
    Price_Long (0),
    Price_Short (0);

  If BarType = 1 then //esto es true en un chart intradía
  Begin
    If time = E17_Time then //en un chart en 1 minuto en el minuto
    siguiente se colocan las órdenes
    Begin
      Price_Long = high + (3 * tick);
      Price_Short = low - (3 * tick);

      Buy ("E17_LE") next bar at Price_Long stop;
      SellShort ("E17_SE") next bar at Price_Short stop;
    End;
  end else

    Raiseruntimeerror("Esta entrada solo funciona en intradía, se
    recomienda 1 min");
```

```
End;

case 18: //entrada en bajo volumen en reversión (o en tendencia
comentado en código)
Begin

  If V < average(V, Per_E) then
  Begin
    //If close = Highest(close, Per_E) then
    If close = Lowest(close, Per_E) then
      Buy ("E18_LE") next bar at market;

    //If close = Lowest(close, Per_E) then
    If close = Highest(close, Per_E) then
      SellShort ("E18_SE") next bar at market;
  end;
End;

case 19: //variación de key reversal
Begin
  If low[3] > low[2] and low[2] > low[1] and close > high[1] then
    Buy ("E19_LE") next bar at market;

  If high[3] < high[2] and high[2] < high[1] and close < low[1]
  then
    SellShort ("E19_SE") next bar at market;
End;

case 20: //candlestick con martillo/hombre colgado (ejemplo de uso
de funciones candle)
Begin
  inputs:
    ES20_Per (15), //periodo para calcular el cuerpo medio
    ES20_n (2); //factor que dice cuántas veces debe ser mayor la
    sombra que el cuerpo

  variables:
    oHammer(0),
    oHangingMan(0);

  Value1 = C_Hammer_HangingMan(ES20_Per, ES20_n, oHammer,
  oHangingMan);

  if close < close[Per_E] and L < L[1] and oHammer = 1 then
    Buy ("ES20_LE") next bar at H stop;

  if close > close[Per_E] and H > H[1] and oHangingMan = 1 then
    SellShort ("ES20_SE") next bar at L stop;
End;
```

```
  case 21: //Otro Breakout + Momentum
  Begin
    inputs:
      E21_Per (10);

    if close > close[Per_E] and close = highest(close, E21_Per) then
      Buy ("ES21_LE") next bar at market;

    if close < close[Per_E] and close = lowest(close, E21_Per) then
      SellShort ("ES21_SE") next bar at market;
  End;

  case 22: //Momentum corto plazo + Momentum largo plazo
  Begin
    If Close > Close[Per_E * E09_n] and Close > Close[Per_E] then
      Buy ("ES22_LE") next bar at market;

    If Close < Close[Per_E * E09_n] and Close < Close[Per_E] then
      SellShort ("ES22_SE") next bar at market;
  End;

  case 23: //candlestick con martillo/hombre colgado invertido
  Begin
    inputs:
      ES23_Per (15), //periodo para calcular el cuerpo medio
      ES23_n (2); //factor que dice cuántas veces debe ser mayor la
      sombra que el cuerpo

    Value1 = C_Hammer_HangingMan(ES23_Per, ES23_n, oHammer,
    oHangingMan);

    if close < close[Per_E] and oHammer = 1 then
      Sellshort ("ES23_SE") next bar at market;

    if close > close[Per_E] and oHangingMan = 1 then
      Buy ("ES23_LE") next bar at market;
  End;
End;


switch (salida)
Begin
  case 0: //no exits
  Begin

  End;

  case 1: //Stop $;
```

```
Begin
  Input:
    S01_Stop_$ (4000);

  SetStopContract;
  SetStopLoss(S01_Stop_$);
end;

case 2: //Stop %;
Begin
  Input:
    S02_Stop_Pct (2.0); //en tanto por 100

  SetStopShare;

  If MP <> 0 then
    SetStopLoss(EntryPrice * S02_Stop_Pct / 100 * BigPointValue)
  else
    SetStopLoss(Close * S02_Stop_Pct / 100 * BigPointValue);
end;

case 3: //Stop ATR
Begin
  Input:
    S03_ATRs (5.0),

    S03_ATR_suelo (0.0), //límite inferior del stop ATR en tanto
    por 100
    S03_ATR_techo (20.0); // límite superior del stop ATR en tanto
    por 100

  Value1 = S03_ATRs * AvgTrueRange(Per_S) * BigPointValue;
  Value2 = S03_ATR_suelo / 100 * C * BigPointValue;
  Value3 = S03_ATR_techo / 100 * C * BigPointValue;

  Value1 = MaxList(Value1, Value2);
  Value1 = MinList(Value1, Value3);

  SetStopContract;
  SetStopLoss(Value1);
end;

case 4: //Profit $
Begin
  Input:
    S04_Profit_$ (7500);

  SetStopContract;
  SetProfitTarget(S04_Profit_$);
```

```
end;

case 5: //Profit %
Begin
  Input:
    S05_Profit_Pct (1.5);

  SetStopShare;

  If MP <> 0 then
    SetProfitTarget(EntryPrice * S05_Profit_Pct / 100 *
    BigPointValue)
  else
    SetProfitTarget(Close * S05_Profit_Pct / 100 * BigPointValue);
end;

case 6: //Profit ATR
Begin
  Input:
    S06_ATRs (10.0),
    S06_ATR_suelo (0.0),
    S06_ATR_techo (20.0);

  Value1 = S06_ATRs * AvgTrueRange(Per_S) * BigPointValue;
  Value2 = S06_ATR_suelo / 100 * C * BigPointValue;
  Value3 = S06_ATR_techo / 100 * C * BigPointValue;

  Value1 = MaxList(Value1, Value2);
  Value1 = MinList(Value1, Value3);

  SetStopContract;
  SetProfitTarget(Value1);
end;

case 7: //Breakeven $
Begin
  Input:
    S07_BreakEven_$ (3000);

  SetBreakEven(S07_BreakEven_$);
end;

case 8: //Trailing ATR
Begin
  Input:
    S08_ATRs (9.0);

  variables:
    ATRCalc(0),
```

```
    TT(0),
    PosHigh(0),
    PosLow(0);

  ATRCalc = AvgTrueRange(Per_S) * S08_ATRs;
  TT = TotalTrades;

  If MP = 1 then
  Begin
    if TT <> TT[1] or MP[1] <> 1 or High > PosHigh then
      PosHigh = High;

    Sell ("S08_LX1") next bar at PosHigh - ATRCalc stop;

  end else

    Sell ("S08_LX2") next bar at High - ATRCalc stop;

  if MP = -1 then
  Begin
    if TT <> TT[1] or MP[1] <> -1 or Low < PosLow then
      PosLow = Low;

    BuyToCover ("S08_SX1") next bar at PosLow + ATRCalc stop;

  end else

    BuyToCover ("S08_SX2") next bar at Low + ATRCalc stop;
end;

case 9: //Salida por Tiempo
Begin
  Input:
    S09_NBars (30);

  If BarsSinceEntry >= S09_NBars then
  Begin
      Sell ("S09_LX") next bar at market;
      BuyToCover ("S09_SX") next bar at market;
  end;
end;

case 10: //Salida al cierre (necesita modificar la sesión para que
acabe n minutos antes del cierre real)
Begin
  If BarType = 1 then //esto es true en un chart intradía
    SetExitOnClose
  else
```

```
    Raiseruntimeerror("La salida a fin de día solo funciona en
    intradía");
End;

case 11: //Trailing ATR + Profit ATR
Begin
  Input:
    S11_ATRs_01 (7.75),
    S11_ATRs_02 (7.25);

  ATRCalc = AvgTrueRange(Per_S) * S11_ATRs_01;
  TT = TotalTrades;

  If MP = 1 then
  Begin
    if TT <> TT[1] or MP[1] <> 1 or High > PosHigh then
      PosHigh = High;

    Sell ("S11_LX1") next bar at PosHigh - ATRCalc stop;

  end else

    Sell ("S11_LX2") next bar at High - ATRCalc stop;

  If MP = -1 then
  Begin
    if TT <> TT[1] or MP[1] <> -1 or Low < PosLow then
      PosLow = Low;

    BuyToCover ("S11_SX1") next bar at PosLow + ATRCalc stop;

  end else

    BuyToCover ("S11_SX2") next bar at Low + ATRCalc stop;

  Value1 = S11_ATRs_02 * AvgTrueRange(Per_S) * BigPointValue;
  Value2 = S03_ATR_suelo / 100 * C * BigPointValue;
  Value3 = S03_ATR_techo / 100 * C * BigPointValue;

  Value1 = MaxList(Value1, Value2);
  Value1 = MinList(Value1, Value3);

  SetStopContract;
  SetProfitTarget(Value1);
end;

case 12: //Chandelier
Begin
  Input:
```

```
    S12_ATRs (9.75);

  vars:
    BandaSuperior (0),
    BandaInferior (0);

  ATRCalc = AvgTrueRange(Per_S) * S12_ATRs;
  BandaSuperior = Highest (High, Per_S) - ATRCalc;
  BandaInferior = Lowest (Low, Per_S) + ATRCalc;

  Sell ("S12_LX") next bar at BandaSuperior stop;
  BuyToCover ("S12_SE") next bar at BandaInferior stop;
end;

case 13: //Bollinger banda contraria
Begin
  Inputs:
    S13_Desv (2);

  vars:
    HiBand(0),
    LoBand(0);

  HiBand = BollingerBand(Close, Per_S, S13_Desv);
  LoBand = BollingerBand(Close, Per_S, -S13_Desv);

  If MP <> -1 then
    Sell ("S13_LX") next bar at HiBand limit;

  If MP <> 1 then
    BuyToCover ("S13_SX") next bar at LoBand limit;
end;

case 14: //Salida Trailling %
Begin
  Inputs:
    S14_Prc_Trail (2.0); //en tanto por 100

  var:
    Trailing_Long(0),
    Trailing_Shrt(99999);

  If S14_Prc_Trail > 0 Then
  Begin
    If MP <> 1 Then
      Trailing_Long = 0;

    If MP <> -1 Then
      Trailing_Shrt = 99999;
```

```
      If MP = 1 then
      Begin // Para posiciones largas
        Trailing_Long = maxList(Trailing_Long, High - (High *
        S14_Prc_Trail / 100));
        Sell ("S14_LX") next bar at Trailing_Long stop;
      end;

      If MP = -1 then
      Begin // Para posiciones cortas
        Trailing_Shrt = minList(Trailing_Shrt, Low + (Low *
        S14_Prc_Trail / 100));
        BuyToCover ("S14_SX") next bar at Trailing_Shrt stop;
      end;
    End;
end;

case 15: //Stop $ + Profit $
Begin
    Input:
      S15_Stop_$ (4000),
      S15_Profit_NxStop (2.0);

    SetStopContract;
    SetStopLoss(S15_Stop_$);
    SetProfitTarget(S15_Stop_$ * S15_Profit_NxStop);
end;

case 16: //Stop% + Profit%
Begin
    Input:
      S16_Stop_Pct (2.25), //en tanto por 100
      S16_Profit_NxStopPct (0.7);

    SetStopShare;

    If MP <> 0 then
    Begin
      SetStopLoss(EntryPrice * S16_Stop_Pct / 100 * BigPointValue);
      SetProfitTarget(EntryPrice * (S16_Stop_Pct *
      S16_Profit_NxStopPct) / 100 * BigPointValue);
    end else
    Begin
      SetStopLoss(Close * S16_Stop_Pct / 100 * BigPointValue);
      SetProfitTarget(Close * (S16_Stop_Pct * S16_Profit_NxStopPct)
      / 100 * BigPointValue);
    end;
end;
```

```
case 17: //Stop ATR + Profit ATR
Begin
  Input:
    S17_NumATRs (5.0),
    S17_Profit_NxStopATR (2.5),
    S17_ATR_suelo (0.0),
    S17_ATR_techo (20.0);

    Value1 = S17_NumATRs * AvgTrueRange(Per_S) * BigPointValue;

    Value2 = S17_ATR_suelo / 100 * C * BigPointValue;
    Value3 = S17_ATR_techo / 100 * C * BigPointValue;

    Value1 = MaxList(Value1, Value2);
    Value1 = MinList(Value1, Value3);

    Value4 = S17_NumATRs * S17_Profit_NxStopATR *
    AvgTrueRange(Per_S) * BigPointValue;
    Value4 = MaxList(Value4, Value2);
    Value4 = MinList(Value4, Value3);

    SetStopContract;
    SetStopLoss(Value1);
    SetProfitTarget(Value4);
end;

case 18: //Chandelier + Profit%
Begin
  Input:
    S18_ATRs (6.5),
    S18_Profit_Pct (1.5);

  ATRCalc = AvgTrueRange(Per_S) * S18_ATRs;
  BandaSuperior = Highest(High, Per_S) - ATRCalc;
  BandaInferior = Lowest(Low, Per_S) + ATRCalc;

  Sell ("S18_LX") next bar at BandaSuperior stop;
  BuyToCover ("S18_SX") next bar at BandaInferior stop;

  SetStopShare;

  If MP <> 0 then
    SetProfitTarget(EntryPrice * S18_Profit_Pct / 100 *
    BigPointValue)
  else
    SetProfitTarget(Close * S18_Profit_Pct / 100 * BigPointValue);
end;

case 19: //Stop% + temporal
```

```
Begin
  Input:
    S19_Stop_Pct (4.5),
    S19_NBars (30);

  SetStopShare;

  If MP <> 0 then
    SetStopLoss(EntryPrice * S19_Stop_Pct / 100 * BigPointValue)
  else
    SetStopLoss(Close * S19_Stop_Pct / 100 * BigPointValue);

  If BarsSinceEntry >= S19_NBars then
  Begin
      Sell ("S19_LX") next bar at market;
      BuyToCover ("S19_SX") next bar at market;
  end;
end;

case 20: //Chandelier + temporal
Begin
  Input:
    S20_ATRs (5.0),
    S20_NBars (59);

  ATRCalc = AvgTrueRange(Per_S) * S20_ATRs;
  BandaSuperior = Highest(High, Per_S) - ATRCalc;
  BandaInferior = Lowest(Low, Per_S) + ATRCalc;

  Sell ("S20_LX1") next bar at BandaSuperior stop;
  BuyToCover ("S20_SX1") next bar at BandaInferior stop;

  If BarsSinceEntry >= S20_NBars then
  Begin
      Sell ("S20_LX2") next bar at market;
      BuyToCover ("S20_SX2") next bar at market;
  end;
end;

case 21: //Bollinger banda contraria + temporal
Begin
  Inputs:
    S21_Desv (2.0),
    S21_NBars (10);

  If BarsSinceEntry >= S21_NBars then
  Begin
      Sell ("S21_LX1") next bar at market;
      BuyToCover ("S21_SX1") next bar at market;
```

```
    end;

    HiBand = BollingerBand(Close, Per_S, S21_Desv);
    LoBand = BollingerBand(Close,Per_S,-S21_Desv);

    If MP <> -1 then
      Sell ("S21_LX2") next bar at HiBand limit;

    If MP <> 1 then
      BuyToCover ("S21_SX2") next bar at LoBand limit;
  end;

  case 22: //Bollinger banda contraria + Stop%
  Begin
    Inputs:
      S22_Desv (2.0),
      S22_Stop_Pct (2.0);

    HiBand = BollingerBand(Close, Per_S, S22_Desv);
    LoBand = BollingerBand(Close, Per_S, -S22_Desv);

    If MP <> -1 then
      Sell ("S22_LX") next bar at HiBand limit;

    If MP <> 1 then
      BuyToCover ("S22_SX") next bar at LoBand limit;

    SetStopShare;

    If MP <> 0 then
      SetStopLoss(EntryPrice * S22_Stop_Pct/100 * BigPointValue)
    else
      SetStopLoss(Close * S22_Stop_Pct/100 * BigPointValue);
  end;

  case 23: //ParabolicSAR Exit
  Begin
    Inputs:
      S23_AfStep (0.02),
      S23_AfLimit (0.2);

    Variables:
      ReturnValue( 0 ),
      oParCl( 0 ),
      oParOp( 0 ),
      oPosition( 0 ),
      oTransition( 0 );
```

```
    ReturnValue = ParabolicSAR(S23_AfStep, S23_AfLimit, oParCl,
    oParOp, oPosition, oTransition);

    if oPosition = -1 then
      BuyToCover ("S23_SX") next bar at oParOp stop;

    If oPosition = 1 Then
      Sell ("S23_LX") next bar at oParOp stop;
End;

case 24: //Profit% + BreakEven$
Begin
    Input:
      S24_Profit_Pct (2.75),
      S24_BreakEven_$ (3000);

    SetStopShare;

    If MP <> 0 then
      SetProfitTarget(EntryPrice * S24_Profit_Pct / 100 *
      BigPointValue)
    else
      SetProfitTarget(Close * S24_Profit_Pct / 100 * BigPointValue);


    SetBreakEven (S24_BreakEven_$);
end;

case 25: //Stop% + BreakEven$
Begin
    Input:
      S25_Stop_Pct (1.5),
      S25_BreakEven_$ (3000);

    SetStopShare;

    If MP <> 0 then
      SetStopLoss(EntryPrice * S25_Stop_Pct / 100 * BigPointValue)
    else
      SetStopLoss(Close * S25_Stop_Pct / 100 * BigPointValue);

    SetBreakEven(S25_BreakEven_$);
end;

case 26: //Profit% + BreakEven%
Begin
    Input:
      S26_Profit_Pct (2.2),
      S26_BreakEven_Pct (1.6); //en tanto por 100
```

```
    SetStopShare;

    If MP <> 0 then
    Begin
      SetProfitTarget(EntryPrice * S26_Profit_Pct / 100 *
      BigPointValue);
      SetBreakEven(EntryPrice * S26_BreakEven_Pct / 100 *
      BigPointValue);
    end else
    Begin
      SetProfitTarget(Close * S26_Profit_Pct / 100 * BigPointValue);
      SetBreakEven(Close * S26_BreakEven_Pct / 100 * BigPointValue);
    End;
  end;

  case 27: //Stop% + BreakEven%
  Begin
    Input:
      S27_Stop_Pct (6.75),
      S27_BreakEven_Pct (1.5);

    SetStopShare;

    If MP <> 0 then
    Begin
      SetStopLoss(EntryPrice * S27_Stop_Pct / 100 * BigPointValue);
      SetBreakEven(EntryPrice * S27_BreakEven_Pct / 100 *
      BigPointValue);
    end else
    Begin
      SetStopLoss(Close * S27_Stop_Pct / 100 * BigPointValue);
      SetBreakEven(Close * S27_BreakEven_Pct / 100 * BigPointValue);
    End;
  end;

  case 28: //BreakEven%
  Begin
    Input:
      S28_BreakEven_Pct (2.2);

    SetStopShare;

    If MP <> 0 then
    Begin
      SetBreakEven(EntryPrice * S28_BreakEven_Pct / 100 *
      BigPointValue);
    end else
    Begin
```

```
      SetBreakEven(Close * S28_BreakEven_Pct / 100 * BigPointValue);
    End;
end;

case 29: //Stop% + Profit% + BreakEven%
Begin
    Input:
      S29_Stop_Pct (3.5),
      S29_Profit_Pct (2.0),
      S29_BreakEven_Pct (1.75);

    SetStopShare;

    If MP <> 0 then
    Begin
      SetStopLoss(EntryPrice * S29_Stop_Pct / 100 * BigPointValue);
      SetProfitTarget (EntryPrice * S29_Profit_Pct / 100 *
      BigPointValue);
      SetBreakEven(EntryPrice * S29_BreakEven_Pct / 100 *
      BigPointValue);
    end else
    Begin
      SetStopLoss(Close * S29_Stop_Pct / 100 * BigPointValue);
      SetProfitTarget (Close * S29_Profit_Pct / 100 *
      BigPointValue);
      SetBreakEven(Close * S29_BreakEven_Pct / 100 * BigPointValue);
    End;
end;

case 30: //Stop $ + Profit $ + BreakEven $
Begin
    Input:
      S30_Stop_$ (4000),
      S30_Profit_$ (8000),
      S30_BreakEven_$ (3000);

    SetStopShare;

      SetStopLoss(S30_Stop_$);
      SetProfitTarget(S30_Profit_$);
      SetBreakEven(S30_BreakEven_$);
end;

case 31: //Stop ATR + Profit ATR + BreakEven%
Begin
    Input:
      S31_ATRs (3.0),
      S31_Profit_NxStopATR (2.5),
      S31_BreakEven_Pct (2.5),
```

```
      S31_ATR_suelo (0.0),
      S31_ATR_techo (20.0);

      Value1 = S31_ATRs * AvgTrueRange(Per_S) * BigPointValue;

      Value2 = S31_ATR_suelo / 100 * C * BigPointValue;
      Value3 = S31_ATR_techo / 100 * C * BigPointValue;

      Value1 = MaxList(Value1, Value2);
      Value1 = MinList(Value1, Value3);

      Value4 = S31_ATRs * S31_Profit_NxStopATR * AvgTrueRange(Per_S)
      * BigPointValue;
      Value4 = MaxList(Value4, Value2);
      Value4 = MinList(Value4, Value3);

      SetStopContract;
      SetStopLoss(Value1);
      SetProfitTarget(Value4);

      If MP <> 0 then
      Begin
        SetBreakEven(EntryPrice * S31_BreakEven_Pct / 100 *
        BigPointValue);
      end else
      Begin
        SetBreakEven(Close * S31_BreakEven_Pct / 100 *
        BigPointValue);
      End;
end;

case 32: //Cierre Media
Begin
  If Close crosses above Average(Close, Per_S) then
    BuyToCover ("S32_SX") next bar at market;

  If Close crosses below Average(Close, Per_S) then
    Sell ("S32_LX") next bar at market;
end;

case 33: //Key Reversal (si vemos que funciona la combinamos con
otras salidas)
Begin
  If High > Highest(High, Per_S)[1] and Close < Close[1] then
    Sell ("S33_LX") next bar at market;

  If Low < Lowest(Low, Per_S)[1] and Close > Close[1] then
    Buy To Cover ("S33_SX") next bar at market;
End;
```

```
    case 34: //cierre por momentum
    Begin
      If close > close[Per_S] then
        BuyToCover ("S34_SX") next bar at market;

      If close < close[Per_S] then
        Sell ("S34_LX") next bar at market;
    End;

    case 35: //cierre por donchian
    Begin
      If Close > Highest(High, Per_S)[1] then
        BuyToCover ("S35_SX") next bar at market;

      If Close < Lowest(Low, Per_S)[1] then
        Sell ("S35_LX") next bar at market;
    End;
End;
```