

## CURSO-TENDENCIAL-INTRADIA-02

Este código implementa una estrategia de trading algorítmico tendencial basada en el cruce de medias. La principal particularidad de esta estrategia es la posibilidad de elegir los tipos de las medias que se cruzan con un input selector “**media**”.

### INPUTS DE LA ESTRATEGIA:

- **allowlong (true)**: Permite activar o desactivar la apertura de posiciones largas.
- **allowshort (true)**: Permite activar o desactivar la apertura de posiciones cortas.
- **Slow\_Avg (46)**: Número de barras usadas para calcular la media móvil lenta.
- **Fast\_Avg (2)**: Número de barras usadas para calcular la media móvil rápida.
- **media (8)**: Tipo de media a utilizar. Hay 19 combinaciones.
- **Price (Close)**: Precio utilizado para calcular las medias (precio de cierre).
- **senyalencruce (true)**: Indica si la señal se da en el cruce de medias.
- **Uso\_Donchian (false)**: Si activamos este input el precio de entrada no se produce a mercado en la siguiente barra si no con una orden stop en el canal de Donchian. La entrada larga en el canal superior al revés para el lado corto. Además habrá que poner el input anterior “senyalencruce” = False
- **Channel\_Per (11)**: Número de barras para calcular el canal de Donchian.
- **ATR\_Per (46)**: Periodo para el Average True Range (ATR).
- **ATR\_Filt\_Long (0)**: Este input multiplica al ATR y se le añade al canal de Donchian para generar una entrada a un precio ligeramente superior al canal de Donchian. Solo actúa si está Uso\_Donchian activado.
- **ATR\_Filt\_Shrt (0.3)**: Este input multiplica al ATR y se le resta al canal de Donchian para generar una entrada a un precio ligeramente inferior al canal de Donchian. Solo actúa si está Uso\_Donchian activado.
- **Start\_Equity (100000)**: Importe inicial de la cuenta. Usado para el MM.
- **MMVar\_Start (100)**: Multiplicador del capital inicial para el MM.
- **MMVar\_Profits (100)**: Multiplicador del beneficio del sistema para el MM.
- **Min\_Size (1)**: Cantidad mínima de contratos o acciones que el sistema puede abrir.
- **Max\_Size (100000)**: Cantidad máxima de contratos o acciones que el sistema puede abrir.
- **RoundTo (1)**: Unidad a la que se redondea el MM.
- **Filt\_ATR (0.2)**: Valor mínimo del ATR en valor porcentual para calcular el MM.
- **PriceOffset (1)**: Desplazamiento del precio para órdenes.
- **PlotOffset (0)**: Desplazamiento del gráfico.
- **PlotLength (1000)**: Longitud del gráfico en barras.
- **TLStyle1 (Tool\_Dotted)**: Estilo de línea 1.
- **TLStyle2 (Tool\_Solid)**: Estilo de línea 2.
- **TLCOLOR1 (Yellow)**: Color de línea 1.
- **TLCOLOR2 (LightGray)**: Color de línea 2.
- **TLCOLOR3 (Cyan)**: Color de línea 3.

- **TLCOLOR4 (Magenta):** Color de línea 4.
- **TLSIZE1 (0):** Tamaño de línea 1.
- **TLSIZE2 (1):** Tamaño de línea 2.
- **Pinta\_Medias (false):** Dibuja las medias en el gráfico.
- **Pinta\_Canales (false):** Dibuja los canales en el gráfico.
- **InicioSesion(0):** Hora inicial del periodo en el que permitimos operar. Formato "hhmm".
- **FinSesion(2300):** Hora final del periodo en el que permitimos operar. Formato "hhmm".

#### INPUTS DE LA FUNCIÓN CURSO\_FILTROS

- **Nivel\_ADX( 35 ):** Input para el filtro nº 1 y 5. Si el valor es 0 siempre es True. Los valores mayores que cero activan y desactivan el filtro comparándolo con el ADX.
- **Nivel\_ATR( 0.2 ):** Input para el filtro nº 2 y 6. Si el valor es 0 siempre es True. Los valores mayores que cero activan y desactivan el filtro comparándolo con el ATR.
- **Nivel\_Stddev( 0 ):** Input para el filtro nº 7. Si el valor es 0 siempre es True. Los valores mayores que cero activan y desactivan el filtro comparándolo con la Desviación típica.
- **Barras( 46 ):** Barras para el cálculo de ATR, ADX y Desviación típica.
- **Rango1 (2):** Número de barras para calcular el NarrowRange.
- **Rango2 (2):** Número de barras para calcular el Widespread.
- **FiltroY( 9 ):** Selector de filtros caso afirmativo. Con este input buscamos que la función devuelva True cuando se cumpla el filtro del input. Los filtros van de 1 a 15. Valor 16 devuelve siempre True. Valor mayor que 16 devuelve siempre False.
- **FiltroN( 6 ):** Selector de filtros caso negativos. Con este input buscamos que la función devuelva False, es decir, únicamente operaremos cuando NO se cumpla el filtro. Los filtros van de 1 a 15. Valor 16 devuelve siempre True. Valor mayor que 16 devuelve siempre False.

#### VARIABLES:

- **TP, Fast, Slow:** Variables para almacenar precios típicos y medias móviles.
- **Cond\_Long, Cond\_Shrt:** Condiciones para operaciones largas y cortas.
- **Channel\_Up, Channel\_Dw:** Límites superior e inferior del canal de Donchian.
- **ATR:** Valor del Average True Range.
- **ATR\_Long, ATR\_Shrt:** Cantidades que añadimos al canal de Donchian para calcular el precio de entrada tanto largo como corto.
- **lotes:** Tamaño de la posición.
- **Price\_Long, Price\_Shrt:** Precios para operaciones largas y cortas.
- **Profits:** Beneficios acumulados.
- **hayPrecio:** La variable hayPrecio sirve para actualizar el canal cuando salta el stop.
- **Ajuste:** Ajuste para la gestión de riesgo. Almacenamos el valor máximo entre Filt\_Atr y el propio ATR normalizado.
- **RiskMSA:** Riesgo calculado para obtener los lotes en el MM.
- **Largos, Cortos:** Indicadores de posición larga y corta.
- **Optimizing:** Comprobamos si el sistema está optimizando para no dibujar las medias.

## MONEY MANAGEMENT:

1. Calculamos los beneficios acumulados (incluyendo posiciones abiertas) en la variable Profits.
2. Calculamos la variable RiskMSA con la siguiente fórmula:

```
Ajuste = MaxList (AvgNormalizedTrueRange (ATR_Per), Filt_ATR);
```

```
RiskMSA = Ajuste * TP * Bigpointvalue;
```

3. Determinamos el número de contratos a operar, que se establece como un porcentaje de la cuenta inicial y otro porcentaje de los beneficios, partido por el riesgo calculado en el paso anterior:

```
lotes = ((Start_Equity + Profits) * MMVar * 0.01) / AbsValue(RiskMSA)
```

4. Hay que asegurar que el número de contratos se redondea y que esté dentro de los límites mínimos y máximos establecidos.

## ESTRATEGIA DE ENTRADA

### SELECCIÓN Y CÁLCULO DE LAS MEDIAS MÓVILES:

- 1- A través del input media seleccionamos el cruce de medias que queremos que se cumple para obtener los valores Cond\_Long y Cond\_Shrt que habilitarán la operativa tanto en el lado corto como en el largo.

### APERTURA DE POSICIONES:

1. LARGAS: Si Allow\_Long es True entonces:
  - a. Si estamos en sesión y Cond\_Long es True. Además la función Curso\_Filtros en el caso afirmativo nos da True y en el caso negativo nos da False. Continuamos.
    - i. En caso de tener una posición diferente a largo. Continuamos
      1. Si Uso\_Donchian está activo lanzamos una orden al Precio Price\_Long.
      2. En caso que Donchian no esté activo compramos a mercado en la siguiente barra.
2. CORTAS: Si Allow\_Shrt es True entonces:
  - a. Si estamos en sesión y Cond\_Shrt es True. Además, la función Curso\_Filtros en el caso afirmativo nos da True y en el caso negativo nos da False. Continuamos.
    - i. En caso de tener una posición diferente a corto. Continuamos
      1. Si Uso\_Donchian está activo lanzamos una orden al Precio Price\_Shrt.
      2. En caso que Donchian no esté activo vendemos en corto a mercado en la siguiente barra.

## GESTIÓN DE POSICIONES ABIERTAS

### SALIDA EN CRUCE CONTRARIO:

Tanto el lado largo como el corto tienen salida a mercado una vez se produce el cruce de las medias en el sentido contrario.

### CÓDIGO EASYLANGUAGE:

```
{ Variables }
inputs:
    allowlong( true ),
    allowshort( true ),
    Slow_Avg ( 46 ),
    Fast_Avg ( 2 ),
    media( 8 ),
    Price ( Close ),
    senyalencruce( true ),

        Uso_Donchian ( false ), // si activamos Donchian hay que
//poner senyalencruce=True
    Channel_Per ( 11 ),

    ATR_Per ( 46 ),
    ATR_Filt_Long ( 0 ),
    ATR_Filt_Shrt ( 0.3 ),

    Start_Equity ( 200000 ),
    MMVar ( 25 ),
    Min_Size ( 1 ),
    Max_Size ( 10000 ),
    RoundTo ( 1 ),

    Filt_ATR ( 0.2 ),

    // REPRESENTACION MEDIAS
    PriceOffset ( 1 ), { Price offset. Typically 1 used since
strategy orders usually
                    of type Buy/Sell NEXT bar at .... }
    PlotOffset ( 0 ), { distance from rightmost bar on chart that
plotting stops }
    PlotLength ( 1000 ),
    TLStyle1 ( Tool_Dotted ),
    TLStyle2 ( Tool_Solid ),
    TLCOLOR1 ( Yellow ),
    TLCOLOR2 ( LightGray ),
    TLCOLOR3 ( Cyan ),
    TLCOLOR4 ( Magenta ),
    TLSIZE1 ( 0 ),
    TLSIZE2 ( 1 ),

    Pinta_Medias ( false ),
```

```

        Pinta_Canales ( false ),
        InicioSesion( 0000 ),
        FinSesion( 2359 );

// FILTROS FUNCIÓN Curso_Filtros
Input:           Nivel_ADX( 35 ),
                Nivel_ATR( 0.2 ),
                Nivel_Stddev( 0 ),
                Barras( 46 ),
                Rango1 (2),
                Rango2 (2),
                FiltroY( 9 ),
                FiltroN( 6 );

vars:            TP (0),
                Fast (0),
                Slow (0),
                Cond_Long (false),
                Cond_Short (false),
                Channel_Up (0),
                Channel_Down (0),
                ATR (0),

                ATR_Long (0),
                ATR_Short (0),
                lotes (0),
                Price_Long (0),
                Price_Short (0),
                Profits (0),
                hayPrecio (false),

                Ajuste (0),
                RiskMSA (0),

                Largos (false),
                Cortos (false),
                Optimizing(GetAppInfo(aiOptimizing) = 1);

TP = TypicalPrice;
switch (media)
Begin
    case 1: // Medias simples
        Fast = AverageFC (Price, Slow_Avg-Fast_Avg);
        Slow = AverageFC (Price, Slow_Avg);

    case 2: // Medias exponenciales
        Fast = XAverage (Price, Slow_Avg-Fast_Avg);
        Slow = XAverage (Price, Slow_Avg);

    case 3: // Kama
        Fast = Kama (Slow_Avg-Fast_Avg);
        Slow = Kama (Slow_Avg);

```

```

        case 4: // Fama + Mama
            variables:
                int ReturnValue( 0 ),
                double oMAMA( 0 ),
                double oFAMA( 0 );
            ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
        );
        Fast = oFama;
        Slow = oMama;

        case 5: // Simple + Exponencial
            Fast = AverageFC (Price, Slow_Avg-Fast_Avg);
            Slow = XAverage (Price, Slow_Avg);

        case 6: // Simple + Kama
            Fast = AverageFC (Price, Slow_Avg-Fast_Avg);
            Slow = Kama (Slow_Avg);

        case 7: // Simple + Mama
            ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
        );
        Fast = AverageFC (Price, Slow_Avg-Fast_Avg);
        Slow = oMama;

        case 8: // Exponencial + Simple
            Fast = XAverage (Price, Slow_Avg-Fast_Avg);
            Slow = AverageFC (Price, Slow_Avg);

        case 9:
            Fast = XAverage (Price, Slow_Avg-Fast_Avg);
            Slow = Kama (Slow_Avg);

        case 10: // Exponencial + Mama
            ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
        );
        Fast = XAverage (Price, Slow_Avg-Fast_Avg);
        Slow = oMama;

        case 11: // Kama + Simple
            Fast = Kama (Slow_Avg-Fast_Avg);
            Slow = AverageFC (Price, Slow_Avg);

        case 12: // Kama + Exponencial
            Fast = Kama (Slow_Avg-Fast_Avg);
            Slow = XAverage (Price, Slow_Avg);

        case 13: // Kama + Mama
            ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
        );
        Fast = Kama (Slow_Avg-Fast_Avg);
        Slow = oMama;

        case 14: // Fama + Simple
    
```

```

        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = oFama;
        Slow = AverageFC (Price, Slow_Avg);

    case 15: // Fama + Exponencial
        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = oFama;
        Slow = XAverage (Price, Slow_Avg);

    case 16: // Fama + Kama
        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = oFama;
        Slow = Kama (Slow_Avg);

    case 17: // Simple + Fama
        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = AverageFC (Price, Slow_Avg);
        Slow = oFama;

    case 18: // Exponencial + Fama
        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = XAverage (Price, Slow_Avg);
        Slow = oFama;

    case 19: // Kama + Fama
        ReturnValue = MamaBase1( Price, 0.5, 0.05, 0.5, oMAMA, oFAMA
);
        Fast = Kama (Slow_Avg);
        Slow = oFama;
end;

if senyalencruce then // En caso de usar una señal que necesite
//confirmación hay que ponerlo en False
Begin
    Cond_Long = Fast crosses over Slow;
    Cond_Shrt = Fast crosses under Slow;
End Else
begin
    Cond_Long = Fast > Slow;
    Cond_Shrt = Fast < Slow;
end;

ATR = AvgTrueRange (ATR_Per);

ATR_Long = ATR * ATR_Filt_Long;
ATR_Shrt = ATR * ATR_Filt_Shrt;

{ Money Management }

```

```

Ajuste = MaxList (AvgNormalizedTrueRange (ATR_Per), Filt_ATR);
RiskMSA = Ajuste * TP * Bigpointvalue;

Profits = NetProfit + OpenPositionProfit;

If RiskMSA <> 0 then
    lotes = ((Start_Equity + Profits) * MMVar * 0.01) / 
AbsValue(RiskMSA)
    //lotes = ((Start_Equity + Profits) * MMVar *0.01) / C *
//BigPointValue
Else
    lotes = 1;

lotes = IntPortion(lotes / RoundTo) * RoundTo;
lotes = MaxList(lotes, Min_Size);
lotes = MinList(lotes, Max_Size);

{ Orders }
If Cond_Long then
begin
    if allowshort then
begin
        If Marketposition = -1 then
            Buycov next bar at market;
    end;
    {Calculamos los máximos y mínimos en el momento del cruce de medias}
    //La variable Hay precio sirve para actualizar el canal cuando salta
//el stop

    If Uso_Donchian and (Fast cross over Slow or not hayPrecio) then
Begin
    Channel_Up = HighestFC (High, Channel_Per);
    hayPrecio = true;
end;

    if allowlong and estamosEnSesion(InicioSesion, FinSesion) and
Curso_Filtros(+FiltroY,Nivel_ADX,Nivel_ATR, Nivel_Stddev, Barras, Rango1,
Rango2) and not Curso_Filtros(+FiltroN,Nivel_ADX,Nivel_ATR, Nivel_Stddev,
Barras, Rango1, Rango2) then
begin
    If Marketposition <> 1 then
begin
        If Uso_Donchian Then
        Begin
            Price_Long = Channel_Up + ATR_Long;
            Buy lotes contracts next bar at Price_Long
stop;
        end Else
            Buy lotes contracts next bar at market;
    end else
        hayPrecio = false;
end;
End;

```

```

If Cond_Shrt then
begin
    if allowlong then
begin
        If Marketposition = 1 then
            Sell next bar at market;
    end;
    {Calculamos los máximos y mínimos en el momento del cruce de medias}
    //La variable Hay precio sirve para actualizar el canal cuando salta
//el stop

    If Uso_Donchian and (Fast cross under Slow or not hayPrecio) then
Begin
    Channel_Dw = LowestFC (Low, Channel_Per);
    hayPrecio = true;
end;

    if allowshort and estamosEnSesion(InicioSesion, FinSesion) and
Curso_Filtros(+FiltroY,Nivel_ADX,Nivel_ATR, Nivel_Stddev, Barras, Rangol,
Rango2) and not Curso_Filtros(+FiltroN,Nivel_ADX, Nivel_Stddev, Nivel_ATR,
Barras, Rangol, Rango2) then
begin
    If Marketposition <> -1 then
Begin
        If Uso_Donchian Then
Begin
            Price_Shrt = Channel_Dw - ATR_Shrt;
            SellShort lotes contracts next bar at
Price_Shrt stop;
        end Else
            Sellshort lotes contracts next bar at market;
    end else
        hayPrecio = false;
end;
End;

{ plot only when NOT optimizing }
// Bloque para representar las medias independientemente de la que elijamos
// Este bloque en Multicharts no funciona, habría que comentarlo o
// eliminarlo

if Optimizing = false then
Begin
    If Pinta_Medias Then
Begin
        value1 = _TLPlotVec (1, Fast, PlotLength, 0, 0, TLStyle1, TLColor3,
TLSize1);
        value2 = _TLPlotVec (2, Slow, PlotLength, 0, 0, TLStyle2, TLColor4,
TLSize2);
    End;

    If Pinta_Canales Then

```

```
Begin
    once (Marketposition <> 0)
    Begin
        If Marketposition = 1 Then
            Largos = True;
        If Marketposition = -1 Then
            Cortos = True;
    End;

    If Largos and Marketposition <> 1 and Fast > Slow then
        value3 = _TLPlotVec (3, Price_Long, PlotLength,
PriceOffset, PlotOffset, TLStyle1, TLCOLOR1, TLSIZE1);

    If Cortos and Marketposition <> -1 and Fast < Slow then
        value4 = _TLPlotVec (4, Price_Shrt, PlotLength, PriceOffset,
PlotOffset, TLStyle1, TLCOLOR2, TLSIZE1);
    end;
end;
```