

This is Jose Ten brief about my experience making experts for FINANCIAL TRADING

I was developing the complete 2021 year expert for Metatrader4, using 4 hours daily for this task.

My idea was dedicate my money and myself to trading professionally, learn the basic principles and take advantage from my software develop experience making EA (experts) which automated invest way.

I contracted with FOREX-TB broker located in Malta (like almost all those in Europe) and I opened a small account to trade.

I learned all the quotes paraphernalia: charts, candles, highs, lows, spreads, ticks, buy, sell, long and short operations..... And a very long etc. ... It is a world rich in technical words.

I was study the main leading indicators (Fibonacci, Gann, Williams...) . I was learn to use decision channels such as Gann fans, Fibonacci, Alligator, Ichimoku, Bollinger Bands, Averages..... .

I used virtually close to all indicators that the historical prices allowed to analyze (trading is based on making decisions by analyzing the history and evolution of the prices and his trends).

I was using these experts in pairs cryptocurrency for a months and I got multiplied my small trade capital by 5. But one night ( in the continuous market) i was suffer a STOP OUT that left my account at 0 Euros and my morale at rock bottom.

At the bottom is a basic EA that I developed for the YCC/USD pair (Crypto Chinese Yuan versus US Dollar). It is based on candlestick patterns with bullish or bearish trends.

I used MQL4, language supported by metatrader 4 , is basically a C++ with predefined functions and attributes for trading.

Developing this type of software in Python would be easier and powerful, in addition will add the possibility to use Django to create SQL modeling

database for market ticks, and after even we can use some IA software to learn about and take decisions bullish or bearish with high success percentage.

Python provide 3 very powerful libraries to develop trading: TA-Lib, FinTA and Pandas-TA. I advise to use FinTa because is compatible with Pandas, and Pandas is compatible too with TA-Lib (necessary if we need take decisions with candles stick patterns ).

For this project I think shall be necessary to use the next Python libraries environment:

**Anaconda** Python distribution, **Jupyter** notebook as editor, **Pandas** to work with series and dataframes, **NumPy** to work with multidimensional big arrays, **SciPy** to optimizing big data, **Matplotlib and Seaborn** for graphic statistics, **Bokke** to generate interactive Web graphics like candles and **keras /spaCy** for learning machine, **Numba** to compile code and get speeds close to C or fortran language.

**Django** as framework to develop Web Interface and the Database can be the one used by the company.

Python is a big universe and have close infinite libraries and functions to find good solutions. All this last advisements can be change when we make the actual study for the project.

It would be for me a really pleasure have the opportunity to develop the software you need for the wholesale electricity market. For me will be a nice challenge job.

Thanks

```

/+-----+
//|          EA V System 1.mq4 |
//|      Copyright 2021, MetaQuotes Software Corp. |
//|          https://www.mql5.com |
//+-----+
//PARAMETROS

#property copyright "Copyright 2021, MetaQuotes Software Corp."
#property link      "https://www.mql5.com"
#property version   "1.00"
#property strict

//PARAMETROS

input double MaxMargin=600;           // Max. Operating Capital:
input double Apalancamiento=0.5;      // Leverage in decimals:
input bool   Stop=true;               // Stop Operations:
input bool   WednesdayOperative=false; // Wednesday Operative:

// CONSTANTES BASICAS DE CALCULO

////////////////////////////////////

double FactorStopLoss=0.3;

int   Periodo_Chart=PERIOD_M15;

int   Periodo_P1=Periodo_Chart;
int   Barras_P1=144;

int   Periodo_P2=Periodo_Chart;
int   Barras_P2=55;

int   Periodo_P3=Periodo_Chart;
int   Barras_P3=13;

int   MaxOrd=99;

int   TempsR1[60]={0,2,4,6,8,10,12,14,16,18,20,22};
int   TempsR2[60]={2,27,47};
int   TempsR3[60]={2,7,12,17,22,27,32,37,42,47,52,57};

string DatosOrdenes[2000];
int    Tickets;
datetime XXX[6];

////////////////////////////////////

string FechaExpiracion="2021.10.01";
string Simbolos[10]={"YCCUSD", "YCCUSD!"};

// VARIABLES OPERATIVAS

double MAGICNUMBER;
string ProgramaYVersion ="JMTen EA Pro" ;
double Spread;
double BeneficioMinimoEsperado;
double MediaP1;
double MediaP2;
double MediaP3;
double GapBUY;
double RSDisparo;
double RS2Disparo;
double CurrentGap;
double CurrentGapP2P3;
double StopTimeOperaciones;
double MediaDeCorte;
double RS;
double RS2;
int    Minimos=0;
double MinimoAnterior;
double Minuto;
double Momento[100];
double CoeficienteDeVertigoDeCompra;
double TimeLastTouch;
double CaidaP1;
double P3BajistaEsperado;
double P3Alcista;
double P3Bajista;
datetime NuevoBar;

string TipoDeVela[6];
double Apertura[6];
double Cierre[6];
double Maximo[6];
double Minimo[6];
double HL[6];
double Cuerpo[6];
double MechaSuperior[6];
double MechaInferior[6];
string Signo[6];
double Dim[6];
string TamVela[6];
double P3B[6];
double P3A[6];
string PatronDeCompraActivado;

string Comentario;
int    OrdenesAbiertasBUY=0;
int    OrdenesAbiertasSELL=0;
double SaldoBUY=0;
double SaldoSELL=0;
int    OperacionesCerradasSemana=0;
double BeneficiosSemana=0;
double MaximoRiesgo=0;
double MaximoInvertido=0;
double TotalInvertido=0;
string Texto;

```

```

double UltimoMaximo=0;
double UltimoMinimo=99999;
bool SubePrecio=true;
double RangoMaximo=0;
double RangoMaximo_M=0;
double RangoMaximo_m=0;
double Rango=0;
double Rango_M=0;
double Rango_m=0;
double Vib;
double g;
double VibAcu;
double SaldoMaximo;
double DownB;

double P2=0;
double R2=0;
double P3=0;
double P1=0;
double R1=0;
double R3=0;
double Pot3=0;
double Max3=0;
double Max2=0;
double Max1=0;
double CaidaDeDisparo=0;
double CorreccionDeCaida;
double MinimosNecesarios;
double Caida60;
double CaidaRS2;
double Caida24Horas;
double Subida24Horas;
double CaidaDeVertigoSemana;
double CaidaDeVertigoMes;

double BolingerBajo;
double BolingerAlto;
double BolingerMedio;

double Min3=0;
double Min2=0;
double Min1=0;

double P3X=0;
double P2A=0;
double P1A=0;
double P2PAc=0;

double P1PAcP=0;
double P1PAcN=0;

double Holgura;
double P2PM;
int nZ=0;

bool P1P2;
bool P2P3;
bool BidBB;

bool SubeMedia=false;
double UltimaMediaAlta=0;
double UltimaMediaBaja=9999;
double PrecioDeCompra=0;
double FechaDeCompra=0;
double TiempoPasado=0;
double MaximoAlcanzado=0;
double MinimoAlcanzado=9999;
bool HechoR2=false;
bool HechoR3=false;
bool HechoR1=false;

bool OKMediaAlta=false;
bool OKMediaBaja=false;

/////

//+-----+
//|               |
//+-----+

int OnInit()
{
    EventSetTimer(60*60);

    MAGICNUMBER=MAGICNUMBER(Symbol());

    for(int x=0; x<9; x++)
        if(Simbolos[x]==Symbol())
            return(INIT_SUCCEEDED);

    Comment(MAGICNUMBER+" THIS EA IS NOT COMPATIBLE WITH THIS SYMBOL. PLEASE CONTACT WITH ( service@eapro.info )"); //
    return(INIT_FAILED);

}

//+-----+
//|               |
//+-----+

void OnDeinit(const int reason)
{
    EventKillTimer();
}

////////////////////////////////////

//+-----+
//|               |
//+-----+

void OnTick()

```

```
{

if (Control()!=false) return;

A_ParametrosTecnicos();

CerrarOperaciones();

AbrirOperaciones();

TextosDeCabecera();

}
//+-----+

//+-----+
//|               |
//+-----+
void A_ParametrosTecnicos()
{

    Spread=Ask-Bid;
    Spread=(Bid*0.00012)/0.01400;

// 0, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, ...

    MediaP3=MediaActual(Periodo_P3,Barras_P3);
    MediaP2=MediaActual(Periodo_P2,Barras_P2);
    MediaP1=MediaActual(Periodo_P1,Barras_P1);

    Holgura=Pips(MaximaCotizacion(PERIOD_H1,24)-MinimaCotizacion(PERIOD_H1,24));

    Rango=Bid-MediaP1;

    if (Rango_M==0) Rango_M=Rango;
    if (Rango_m==0) Rango_m=Rango;

    if (Rango_M<Rango) Rango_M=Rango;
    if (Rango_m>Rango) Rango_m=Rango;

    RangoMaximo=MaximaCotizacion(Periodo_P1,7)-MediaP1;

    if (RangoMaximo_M==0) RangoMaximo_M=RangoMaximo;
    if (RangoMaximo_m==0) RangoMaximo_m=RangoMaximo;
    if (RangoMaximo<RangoMaximo_M) RangoMaximo_M=RangoMaximo;
    if (RangoMaximo>RangoMaximo_m) RangoMaximo_m=RangoMaximo;

    P1=Pendiente(Periodo_P1,Barras_P1,24);
    P2=Pendiente(Periodo_P2,Barras_P2,6);
    P3=Pendiente(Periodo_P3,Barras_P3,6);

    P3Bajista=PendienteBajista(Periodo_P3,Barras_P3,5);
    P3Alcista=PendienteAlcista(Periodo_P3,Barras_P3,5);

    PendientesMaxMin();

    for(int i=0; i<59; i++)
    {
        if ((TempsR1[i]==Hour()) && (HechoR1==false))

        {
            R1=P1-P1A;
            P1A=P1;
            if (R1>0) P1PacP=P1PacP+R1;
            if (R1<0) P1PacN=P1PacN+R1;
            HechoR1=true;

        }

        if ((TempsR3[i]==Minute()) && (HechoR3==false))
        {
            R3=P3Bajista-P3X;
            P3X=P3;
            HechoR3=true;
//      DibujaMarcador3(clrGreen);

        }

        if ((TempsR2[i]==Minute()) && (HechoR2==false))
        {
            R2=P2-P2A;
            P2A=P2;
            n2++;
//      if (SubeMedia==false) P2Pac=P2Pac+P2;
            P2Pac=P2Pac+P2;
            P2PM=0;
            if (n2!=0) P2PM=P2Pac/n2;
            HechoR2=true;

        }

    }

}

for(int i=0; i<59; i++)
{
    if (TempsR1[i]+1==Hour()) HechoR1=false;
    if (TempsR3[i]+1==Minute()) HechoR3=false;
    if (TempsR2[i]+1==Minute()) HechoR2=false;
//  DibujaMarcador(clrGreen);
}
```

OKMediaAlta=NuevaMediaAlta(Periodo\_P3,Barras\_P3,2\*Spread);  
 OKMediaBaja=NuevaMediaBaja(Periodo\_P3,Barras\_P3,2\*Spread);

```

if (CruceP2P3(P2P3)!=="") {P2PAc=0 ; n2=0;}

if ((P1<1) && (P1>1)) {P1PAcN=0; P1PAcP=0;}

TiempoPasado=(TimeCurrent()-FechaDeCompra)/60;

BeneficioMinimoEsperado=1.25*Spread;

// if (CoeficienteDeVertigo(7)>0.5) BeneficioMinimoEsperado=5*Spread;
// if (CoeficienteDeVertigo(7)>0.8) BeneficioMinimoEsperado=10*Spread;

CurrentGap=MediaP1-Bid;
RS=CurrentGap/Spread;

CurrentGapP2P3=MediaP2-MediaP3;
RS2=CurrentGapP2P3/Spread;

GapBUY=2*Spread;

RSDisparo=2;
RS2Disparo=2;

if(CruceP2P3(P2P3)=="DeVenta") UltimoMinimo=99999;

DetectaMinimo(Spread);

FTipoDeVela{Periodo_Chart};
}

void CerrarOperaciones()
{
    StopLossForMoney(FactorStopLoss);

    // VendeViejas(24,-9999);

    // double CaidaDeCierre=2;

    //
    // if (Caida(120)>5) Vende(BeneficioMinimoEsperado);

    // if (MediaP2>MediaP3) Vende(-10*BeneficioMinimoEsperado);
    // if (P3>0) return;
    // if (P2>0) return;

    // if (Bid>PrecioDeCompra))

    // if (CerrarYa()==true) Vende(BeneficioMinimoEsperado);
    // if (MediaP3>MediaP2) return;
    // if (CerrarYa()==true) Vende(BeneficioMinimoEsperado);

    // if (CoeficienteDeVertigoDeCompra>0.95) BeneficioMinimoEsperado=5;
    // if (P2>0) CaidaDeCierre=(Bid-PrecioDeCompra)/Spread; // NUEVO
    // if (CaidaDeCierre>4) CaidaDeCierre=4; // NUEVO
    // if (CaidaDeCierre<2) CaidaDeCierre=2; // NUEVO
    // FactorStopLoss=0.2*CoeficienteDeVertigoDeCompra+0.1;

    // if ((PrecioDeCompra-Bid)/Spread>5) CaidaDeCierre=3;
    // if ((PrecioDeCompra-Bid)/Spread>8) CaidaDeCierre=4;

    // if (Caida(10)<CaidaDeCierre) return;

    Vende(BeneficioMinimoEsperado);
}

//+-----+
//|               |
//+-----+

void AbrirOperaciones()
{
    // if (Bid<Cierre[0]) return;

    // if (CoeficienteDeVertigo(7)<0.2) return;
    // if (CoeficienteDeVertigo(1)<0.2) return;

    if (P1<0) return;

    PatronDeCompraActivado="";

    // P3BajistaEsperado=10;

    if (PatronMartilloAlcista()==true) PatronDeCompraActivado="PatronMartilloAlcista";
    if (PatronMartilloInvertidoAlcista()==true) PatronDeCompraActivado="PatronMartilloInvertidoAlcista";
    if (PatronBeltHoldAlcista()==true) PatronDeCompraActivado="PatronBeltHoldAlcista";
    if (PatronEnvolventeAlcista()==true) PatronDeCompraActivado="PatronEnvolventeAlcista";
    if (PatronHaramiCrossAlcista()==true) PatronDeCompraActivado="PatronHaramiCrossAlcista";
    // if (PatronHaramiAlcista()==true) PatronDeCompraActivado="PatronHaramiAlcista";
    if (PatronEstrellaDojiAlcista()==true) PatronDeCompraActivado="PatronEstrellaDojiAlcista";
    if (PatronPenetranteAlcista()==true) PatronDeCompraActivado="PatronPenetranteAlcista";

```

```

if (PatronLineasEncontradasAlcista() ==true) PatronDeCompraActivado="PatronLineasEncontradasAlcista";
if (PatronMinimosCoincidentesAlcista())PatronDeCompraActivado="PatronMinimosCoincidentesAlcista";
if (PatronBebeAbandonadoAlcista())PatronDeCompraActivado="PatronBebeAbandonadoAlcista";
if (PatronMorningStarAlcista())PatronDeCompraActivado="PatronMorningStarAlcista";
if (PatronMorningStarDojiAlcista())PatronDeCompraActivado="PatronMorningStarDojiAlcista";
if (PatronTresEstrellasEnElSurAlcista())PatronDeCompraActivado="PatronTresEstrellasEnElSurAlcista";
if (PatronTresSoldadosBlancosAlcista()) PatronDeCompraActivado="PatronTresSoldadosBlancosAlcista";
if (PatronTresVelasExterioresAlcista()) PatronDeCompraActivado="PatronTresVelasExterioresAlcista";
if (PatronTresVelasInterioresAlcista()) PatronDeCompraActivado="PatronTresVelasInterioresAlcista";
if (PatronTresEstrellasAlcista()) PatronDeCompraActivado="PatronTresEstrellasAlcista";
if (PatronSueloDeTresRiosAlcista()) PatronDeCompraActivado="PatronSueloDeTresRiosAlcista";
if (PatronPequenaGolondrinaEscondidaAlcista()) PatronDeCompraActivado="PatronPequenaGolondrinaEscondidaAlcista";
if (PatronEscapeAlcista()) PatronDeCompraActivado="PatronEscapeAlcista";
if (PatronPatadaOCozAlcista()) PatronDeCompraActivado="PatronPatadaOCozAlcista";
if (PatronHuecoAlcistaDeTripleFormacionAlcista()) PatronDeCompraActivado="PatronHuecoAlcistaDeTripleFormacionAlcista";
if (PatronHuecoAlcistaTasukiAlcista()) PatronDeCompraActivado="PatronHuecoAlcistaTasukiAlcista";
if (PatronTripleGolpeAlcista()) PatronDeCompraActivado="PatronTripleGolpeAlcista";

if (RS<2) return;
if (MediaP3<MediaP2) return;
// if (MathAbs(RS2)<0.1) return;
if (P3<0) return;
AbreNuevaBUY(GapBUY);

// if (PatronDeCompraActivado!="") AbreNuevaBUY(GapBUY);

// if (VelaAlcista(Periodo_Chart)==false) return;

// if (Caida(120)>CaidaDeDisparo) DibujaMarcadorCaidas(cirAquamarine);

// if (Caida(120)<CaidaDeDisparo()) return;

// DibujaMarcador(cirGray);

}

//+-----+
//|                                     |
//+-----+
void OnTimer()
{

}

// FUNCIONES BASICAS

// 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 377, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 75025, 121393, 196418, 317811, ...

//+-----+
//|                                     |
//+-----+

int MAGICNUMBER(string Simbolo)
{
int x=0;
int z=StringLen(Simbolo);

for(int i=0; i<z; i++)
x=x+StringGetChar(Simbolo,i);

string y=IntegerToString(x);
for(int i=0; i<3; i++)
y=y+StringGetChar(Simbolo,z-i);

return(y);
}

//+-----+
//|                                     |
//+-----+

string TextosDeCabecera()
{
OperacionesAbiertas();

Comentario=
MAGICNUMBER+" "+
Symbol()+" "+
DiaDeLaSemana()+" "+
" SPREAD: "+Pips(Spread)+
// " " +CosteUnPunto()+
// " " +Exchange()+

" PROFIT MARGIN: "+Pips(BeneficioMinimoEsperado)+
// " MAX.TICK: "+NormalizeDouble(MaximaCotizacion(HorasCV*60),Digits)+
// " MIN.TICK: "+NormalizeDouble(MinimaCotizacion(7 *HorasCV*60),Digits)+
"\n"+
" CAPITAL : "+MaxMargin+" € WORKING: "+NormalizeDouble(TotalInvertido,0)+" (" +NormalizeDouble(100*TotalInvertido/MaxMargin,2)+"%) (" +NormalizeDouble(MaximoRiesgo,0)+" € )" +
" MAX.USED : "+NormalizeDouble(MaximoInvertido,0)+" € "+
" CV.BUY(7): " +NormalizeDouble(100*CoeficienteDeVertigo(7),2)+" % "+
" CV.BUY(31): " +NormalizeDouble(100*CoeficienteDeVertigo(31),2)+" % "+

" Max: " + NormalizeDouble(MaximaCotizacion(PERIOD_D1,7),5)+
" Min: " + NormalizeDouble(MinimaCotizacion(PERIOD_D1,7),5)
+"\n"+
" FROM ONE MONTH AGO: "+OperacionesCerradas(30)
+"\n"+
" FROM ONE WEEK AGO: "+OperacionesCerradas(7)+
"\n"+
" OPENED BUY: "+OrdenesAbiertasBUY+ " PROFIT: "+NormalizeDouble(SaldoBUY,2)+" € "
+"\n"+
" AMOUNT TO BUY: "+NormalizeDouble(CosteOperacion(Ask,Lote()),0)+" € "
+"\n"+

" YEPS P1: "+P1+" /"+R1+

```

```

" P2: "+P2+"/"+"R2+
" P3: "+P3+"/"+"R3+
"\n"+
" RS: "+NormalizeDouble(RS,2)+
"\n"+

" P2PAc(n2) "+P2PAc+"("+"n2+") <"+NormalizeDouble(P2PM,2)+">"+
"\n"+

" mEPS  "+Min1+"/"+"Min2+"/"+"Min3+
" MEPS  "+Max1+"/"+"Max2+"/"+"Max3+
"\n"+
" Rg: "+Pips(Rango)+ " ("+"Pips(RangoMaximo)+")"+
"\n"+
" TopRg: "+Pips(Rango_M)+ " ("+"Pips(RangoMaximo_M)+")"+
"\n"+
" LowRg: "+Pips(Rango_m)+ " ("+"Pips(RangoMaximo_m)+")"+
"\n"+
" Holg: "+Holgura
;

Comment(Comentario);
return(Comentario);
}

//+-----+
//|                                     |
//+-----+
void OperacionesAbiertas()
{

OrdenesAbiertasBUY=0;
OrdenesAbiertasSELL=0;

SaldoBUY=0;
SaldoSELL=0;

TotalInvertido=0;
PrecioDeCompra=0;

for(int i=0; i<OrdersTotal(); i++)

{
if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True) && (OrderSymbol()==Symbol()) && (OrderType()==OP_BUY))
{
OrdenesAbiertasBUY++;
SaldoBUY=SaldoBUY+OrderProfit()+OrderSwap()+OrderCommission();
TotalInvertido=TotalInvertido+CosteOperacion(OrderOpenPrice(),OrderLots());
if(PrecioDeCompra==0) { PrecioDeCompra=OrderOpenPrice(); FechaDeCompra=OrderOpenTime();}
}
if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True) && (OrderSymbol()==Symbol()) && (OrderType()==OP_SELL))
{
OrdenesAbiertasSELL++;
SaldoSELL=SaldoSELL+OrderProfit()+OrderSwap()+OrderCommission();
TotalInvertido=TotalInvertido+CosteOperacion(OrderOpenPrice(),OrderLots());
}

if(TotalInvertido>MaximoInvertido)
MaximoInvertido=TotalInvertido;
if(SaldoBUY+SaldoSELL<MaximoRiesgo)
{
MaximoRiesgo=SaldoBUY+SaldoSELL;
}
if (SaldoBUY>SaldoMaximo) SaldoMaximo=SaldoBUY;
}
}

string OperacionesCerradas(int Dias)

{
int OperacionesCerradas=0;
double Beneficios=0;

for(int i=0; i<OrdersHistoryTotal(); i++)

if((OrderSelect(i,SELECT_BY_POS,MODE_HISTORY)==True)
&& (OrderSymbol()==Symbol()) && ((OrderOpenTime()+Dias*24*60*60)>TimeCurrent()))
{
OperacionesCerradas++;
Beneficios=Beneficios+OrderProfit()+OrderSwap();
}

return
(
IntegerToString(OperacionesCerradas)+
" CLOSED ORDERS WITH PROFIT: "
+NormalizeDouble(Beneficios,0)+" € ("
+NormalizeDouble(100*Beneficios/MaxMargin,2)+" %)"
);
}

//+-----+
//|                                     |
//+-----+
string DiaDeLaSemana()
{
if(DayOfWeek()==1)
return (" MONDAY ");
if(DayOfWeek()==2)
return (" TUESDAY ");
if(DayOfWeek()==3)
return (" WEDNESDAY ");
if(DayOfWeek()==4)
return (" THURSDAY ");
if(DayOfWeek()==5)
return (" FRIDAY ");
if(DayOfWeek()==6)

```



```

return (" SATURDAY ");
if(DayOfWeek()==0)
return (" SUNDAY ");
return("");
}

//+-----+
//|                                     |
//+-----+
void Vende(double Beneficio)
{
    int Descuento=0;

// Descuentos por horario para Cierre Del Dia

// if (Hour()>18) Descuento=0.1;
// if (Hour()>19) Descuento=0.2;
// if (Hour()>20) Descuento=0.3;
// if (Hour()>21) Descuento=0.4;
// if (Hour()>22) Descuento=0.5;
// if (Hour()>23) Descuento=0.6;

// if ((DayOfWeek()==3) && (Hour()>23)) Descuento=-1;

// if((DayOfWeek()==0) || (DayOfWeek()==6)) Descuento=0;

Beneficio=Beneficio*(1-Descuento);

for(int i=0; i<OrdersTotal(); i++)
{

    if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True)
    && (OrderType()==OP_BUY)
    && (OrderMagicNumber()==MAGICNUMBER)
    && (OrderTakeProfit()==0)
    && (Bid>OrderOpenPrice()+Beneficio))
    {
        int Ticket=OrderTicket();
        OrderClose(OrderTicket(),OrderLots(),Bid,3,Green);
        ImprimeDatosOrden(Ticket);
        crearcsv();
        DibujaMarcador clrGreen;
        UltimoMinimo=Bid-4*Spread;
// P2PAc=0; n2=0;
    }

}

}

//+-----+
//|                                     |
//+-----+
void StopLossForMoney(double Factor)
{
    for(int i=0; i<OrdersTotal(); i++)

    {
        if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True)
        && (OrderType()==OP_BUY)
        && (OrderMagicNumber()==MAGICNUMBER)
        && (OrderProfit()+OrderSwap()+OrderCommission)<Factor*CosteOperacion(OrderOpenPrice(),OrderLots()))
        {
            int Ticket=OrderTicket();
            OrderClose(OrderTicket(),OrderLots(),Bid,3,Green);
            ImprimeDatosOrden(Ticket);
// DibujaMarcador clrViolet;
// StopTimeOperaciones=TimeCurrent();
        }

    }

}

//+-----+
//|                                     |
//+-----+

void AbreNuevaBUY(double Gap)

{

    if(StopOperaciones()==true)return;
    double StopLoss=Bid*0.8;

    if(EntraNuevoPrecioBUY(Gap)==true)
    {

        if(Lote()==0)return;

        OrderSend(Symbol(),OP_BUY,Lote(),Ask,3,0,0,ParametrosApertura(),MAGICNUMBER,0,Blue);

        DibujaMarcador clrYellow;
// DibujaMarcador2 clrYellow;
// DibujaMarcadorCaidas clrYellow;

        CoeficienteDeVertigoDeCompra=CoeficienteDeVertigo(31);
        PrecioDeCompra=Bid;
        DownB=0;
        Tickets++;
        DatosOrdenes[Tickets]=ParametrosApertura();
// P2PAc=0; n2=0;g=0;VibAcu=0;
    }

}

//+-----+
//|                                     |
//+-----+
double Lote()
{
    double CosteOperacion=0;

```

```

double LibreParalInvertir=0;
double Lote;

return(100);

LibreParalInvertir=0.8*(MaxMargin-TotalInvertido);

if(LibreParalInvertir<10) return(0);

if(LibreParalInvertir>0.25*MaxMargin) LibreParalInvertir=0.25*MaxMargin;

if(LibreParalInvertir+AccountMargin()>0.75*AccountEquity()) return(0);

Lote=CoficienteDeVertigo(7)*NormalizeDouble(LibreParalInvertir/CosteUnPunto(),2);

if(Lote>MarketInfo(Symbol(),MODE_MAXLOT)) Lote=MarketInfo(Symbol(),MODE_MAXLOT);
if(Lote<MarketInfo(Symbol(),MODE_MINLOT)) return(0);

Lote=0.01;
return(Lote);

}

//+-----+
//|               |
//+-----+
double Media(int PERIODO,int Barras,int DesdeBarra)
{
    return (iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_OPEN,DesdeBarra));
}

double Pendiente(int PERIODO,int Barras,int DesdeBarra)
{
    double x=1;
    double y=1;
    int z=1;
    x=iMA(Symbol(),PERIODO,Barras,0,MODE_SMMA,PRICE_OPEN,DesdeBarra);
    y=iMA(Symbol(),PERIODO,Barras,0,MODE_SMMA,PRICE_OPEN,1);

    if(x==0)
        return(-1000);

    z=10000*((y/x)-1);

    string Nombre=(PERIODO+" "+Barras+" "+DesdeBarra);
    ObjectDelete(Nombre);

    ObjectCreate(Nombre,OBJ_TREND,0,iTime(Symbol(),PERIODO,DesdeBarra),x,TimeCurrent(),y);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrWhite);
    if(StopOperaciones()==true)
        ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrRed);

    return(z);
}

double PendienteBajistaX(int PERIODO,int Barras,int DesdeBarra)
{
    double x=1;
    double y=1;
    int z=1;
    x=iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_OPEN,DesdeBarra);
    y=iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_OPEN,1);

    if(x==0)
        return(-1000);

    z=10000*((y/x)-1);

    string Nombre=(PERIODO+" "+Barras+" "+DesdeBarra);
    ObjectDelete(Nombre);

    ObjectCreate(Nombre,OBJ_TREND,0,iTime(Symbol(),PERIODO,DesdeBarra),x,TimeCurrent(),y);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrWhite);
    if(StopOperaciones()==true)
        ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrRed);

    return(z);
}

double PendienteBajista(int PERIODO,int Barras,int DesdeBarra)
{
    double x=0;
    double y=0;
    int z=1;

    for(int i=1; i<DesdeBarra;i++) x=x+Open(Symbol(),PERIODO,i)-iOpen(Symbol(),PERIODO,i+1);

    string Nombre=(PERIODO+" "+Barras+" "+DesdeBarra);
    ObjectDelete(Nombre);

    ObjectCreate(Nombre,OBJ_TREND,0,iTime(Symbol(),PERIODO,DesdeBarra),x,TimeCurrent(),y);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrWhite);
    if(StopOperaciones()==true)
        ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrRed);

    return(NormalizeDouble(x,0));
}

double PendienteAlcista(int PERIODO,int Barras,int DesdeBarra)

```

```

{
double x=0;
double y=0;
int z=1;

for(int i=1; i<DesdeBarra;i++) x=x+Close(Symbol(),PERIODO,i)-iClose(Symbol(),PERIODO,i+1);

string Nombre=(PERIODO+" "+Barras+" "+DesdeBarra);
ObjectDelete(Nombre);

ObjectCreate(Nombre,OBJ_TREND,0,iTime(Symbol(),PERIODO,DesdeBarra),x,TimeCurrent(),y);
ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrWhite);
if(StopOperaciones()!=true)
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrRed);

return(NormalizeDouble(x,0));
}

double PendienteAlcistaX(int PERIODO,int Barras,int DesdeBarra)
{
double x=1;
double y=1;
int z=1;
x=iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_CLOSE,DesdeBarra);
y=iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_CLOSE,1);

if(x==0)
    return(-1000);

z=100000*((y/x)-1);

string Nombre=(PERIODO+" "+Barras+" "+DesdeBarra);
ObjectDelete(Nombre);

ObjectCreate(Nombre,OBJ_TREND,0,iTime(Symbol(),PERIODO,DesdeBarra),x,TimeCurrent(),y);
ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrWhite);
if(StopOperaciones()!=true)
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,clrRed);

return(z);
}

//+-----+
//|               |
//+-----+
bool EntraNuevoPrecioBUY(double Gap)
{
if(Gap==0) return (true);

double NuevoPrecioAceptado=9999;

int Ordenes=0;

for(int i=0; i<OrdersTotal(); i++)
{
    if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True)
        && (OrderSymbol()==Symbol())
        && (OrderMagicNumber()==MAGICNUMBER)
        && (OrderType()==OP_BUY))

        NuevoPrecioAceptado=OrderOpenPrice()-Gap; //BUY
        Ordenes++;
}

if (Ordenes>=MaxOrd) return(false);

if(Ordenes==0) return(true);

if(Ask<NuevoPrecioAceptado) return(true);
return(false);

}

//+-----+
//|               |
//+-----+
double MediaActual(int PERIODO,int Barras)
{
    return (iMA(Symbol(),PERIODO,Barras,0,MODE_SMA,PRICE_OPEN,0));
}
double MediaActualFuerte(int PERIODO,int Barras)
{
    return (iMA(Symbol(),PERIODO,Barras,0,MODE_LWMA,PRICE_OPEN,0));
}

//+-----+
//|               |
//+-----+

double Exchange()
{
    double Cambio=1;

    Cambio=MarketInfo("EURUSD",MODE_ASK);

    if(Cambio>0) return (1/Cambio);

```

```

    return (0.84);
}

bool StopOperaciones()
{
    if(Stop==true) return(true);

    // if (Caida(1440,32*Spread)==true) StopTimeOperaciones=TimeCurrent();

    // if (P2<-250) return (true);

    // if (P1<-2000) return (true);
    // if (P1> 2000) return (true);

    // if (P3<-30) return (true);

    if((DayOfWeek()==3) && (WednesdayOperative==false)) return(true);

    if(StopTimeOperaciones(1)==true) return (true);

    return(false);

}

//+-----+
//|               |
//+-----+

void VendeViejas(int Horas,double Margen)
{
    for(int i=0; i<OrdersTotal(); i++)
    {
        if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True)
            && (OrderSymbol()==Symbol())
            && (OrderMagicNumber()==MAGICNUMBER))
        {
            if(OrderOpenTime()+Horas*60*60>TimeCurrent())
                return;
            if(OrderOpenPrice()<Bid+Margen)
                return;
            int Ticket=OrderTicket();
            OrderClose(OrderTicket(),OrderLots(),Bid,3,clrBlack);
            StopTimeOperaciones==true;
            creatcsv();
            ImprimeDatosOrden(Ticket);
            StopTimeOperaciones=TimeCurrent();
            // DibujarMarcador(clrBeige);
        }
    }
}

//+-----+
//|               |
//+-----+

double MaximaCotizacion(int PER, int barras)
{
    int barra=iHighest(Symbol(),PER,MODE_HIGH,barras,0);
    double x=iHigh(Symbol(),PER,barra);
    if(x==0)
        x=1.2*Ask;
    return(x);
}

//+-----+
//|               |
//+-----+

double MinimaCotizacion(int PER,int barras)
{
    int barra=iLowest(Symbol(),PER,MODE_LOW,barras,0);
    double x=iLow(Symbol(),PER,barra);

    if(x==0)
        x=Bid;
    return(x);
}

//+-----+
//|               |
//+-----+

double CoeficienteDeVertigo(int Dias)
{
    // double x=RangoMaximo;
    // double y=Rango;

    double x=MaximaCotizacion(PERIOD_D1,Dias);
    double y=MinimaCotizacion(PERIOD_D1,Dias);

    double z=x-y;

    if(x==0) return(0);
    if(y==0) return(0);
    if(z==0) return(0);

    double g=(x-Bid)/z;

    if(g<0) g=0; // POSTERIOR
    if(g>1) g=1;

    return(g);
}

double CoeficienteDeVertigoHoras(int Horas)
{

```

```
// double x=RangoMaximo;
// double y=Rango;

double x=MaximaCotizacion(PERIOD_H1,Horas);
double y=MinimaCotizacion(PERIOD_H1,Horas);

double z=x-y;

if(x==0) return(0);
if(y==0) return(0);
if(z==0) return(0);

double g=(x-Bid)/z;

if(g<0) g=0; // POSTERIOR
if(g>1) g=1;

return(g);
}

//+-----+
//|               |
//+-----+
int crearcsv()
{
    string Usuario=AccountName();
    string NombreDelFichero="JMTen"+AccountName();
    int handle = FileOpen(NombreDelFichero, FILE_CSV|FILE_WRITE, ' ');

    if(handle < 1)
    {
        return(handle);
    }
    else
    {
        FileWrite(handle, OperacionesLiquidacion());
    }

    FileClose(handle);
    return(0);
}

string OperacionesLiquidacion()
{
    double x=0;
    double y=0;

    string registro="";
    for(int i=0; i<OrdersHistoryTotal(); i++)

        if((OrderSelect(i,SELECT_BY_POS,MODE_HISTORY)==True) && (StringSubstr(OrderComment(),0,12)==ProgramaYVersion))
        {
            registro=registro+
                OrderSymbol()+" "+
                OrderTicket()+" "+
                OrderOpenTime()+" "+
                OrderCloseTime()+" "+
                NormalizeDouble(CosteOperacion(OrderOpenPrice(),OrderLots()),2)+" "+
                NormalizeDouble(OrderProfit(),2)+"\n";

        }

    return(registro);
}

//+-----+
//|               |
//+-----+
void DibujaMarcador(int Color)
{
    string Nombre=
    // PatronDeCompraActivado+
    // Signo[0]+TipoDeVela[0]+TamVela[0]+
    // Signo[1]+TipoDeVela[1]+TamVela[1]+
    // Signo[2]+TipoDeVela[2]+TamVela[2]+
    // Signo[3]+TipoDeVela[3]+TamVela[3]+
    // Signo[4]+TipoDeVela[4]+TamVela[4]+
    " P3Baj:"+"P3+ "/"+"R3+
    TipoDeVela[0]+
    // PatronDeCompraActivado+
    // " P3Alc:"+"P3Alcista+
    // " P1:"+"P1+ "/"+"R2+ " "+PatronDeCompraActivado+
    Spread+
    // " P2:"+"P2+ "/"+"R2+
    // " P3:"+"P3+ "/"+"R3+
    // " PAc1:"+"P1PacN+ // "("+"P1PacP+")"+
    // " PAc2:"+"P2PacN+ "("+"2+")"+
    // " P2PM:"+"NormalizeDouble(P2PM,2)+
    // " RS:" + NormalizeDouble(RS,2)+
    // " RS2:" + NormalizeDouble(RS2,2)+
    " CV:"+"DoubleToStr(CoeficienteDeVertigoHoras(24),2)+
    // " Rg: "+"Pips(Rango)+ "("+"Pips(RangoMaximo)+")"+
    // " CP:"+"DoubleToStr(CaidaDeDisparo,2) + "/"+"DoubleToStr(Caida(45),2)+

    "",

    ObjectCreate(Nombre,OBJ_ARROW_CHECK,0,TimeCurrent(),Ask);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,Color);
}

void DibujaMarcador2(Int Color)
```

```

{
    string Nombre=
        Minimos+
    // " P1:" + P1 + "/" + R2 +
    // " P2:" + P2 + "/" + R2 +
    // " P3:" + P3 + "/" + R3 +
    // " Holg:" + Holgura +
    // " PAC1:" + P1PacN + // "(" + P1PacP + ")" +
    // " PAC2:" + P2Pac + "(" + n2 + ")" +
    // " P2PM:" + NormalizeDouble(P2PM,2)+
    // " RS:" + NormalizeDouble(RS,2)+
    // " RS2:" + NormalizeDouble(RS2,2)+
    // " CV:" + DoubleToStr(CoeficienteDeVertigo(7),2)+
    // " Rg:" + Pips(Rango) + "(" + Pips(RangoMaximo) + ")" +
    // " CP:" + DoubleToStr(CaidaDeDisparo(),2) + "/" + DoubleToStr(Caida(120),2)+

    "",

    ObjectCreate(Nombre,OBJ_ARROW_CHECK,0,TimeCurrent(),Bid);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,Color);

}

void DibujaMarcadorCaidas(int Color)
{
    string Nombre=

    "P1:" + DoubleToStr(Caida60,1)+
    " 7:" + DoubleToStr(2 * (1 - CoeficienteDeVertigo(7)),1)+
    " 31:" + DoubleToStr(2 * (1 - CoeficienteDeVertigo(31)),1)+
    " RS2:" + DoubleToStr(CaidaRS2,1)+
    " Cae:" + DoubleToStr(Caida(1440)/6,1)+
    " Sube:" + DoubleToStr(-Subida(1440)/6,1)+
    " CP:" + DoubleToStr(CaidaDeDisparo,1) + "/" + DoubleToStr(Caida(180),1)+

    // " RS2:" + NormalizeDouble(RS2,2)+

    "",

    ObjectCreate(Nombre,OBJ_ARROW_CHECK,0,TimeCurrent(),Bid-1.5*Spread);
    ObjectSetInteger(0,Nombre,OBJPROP_COLOR,Color);

}

//+-----+
//|               |
//+-----+

double Caida(int Minutos)
{
    double x=MaximaCotizacion(PERIOD_M1,Minutos);
    return((x-Bid)/Spread);
}

double Subida(int Minutos)
{
    double x=MinimaCotizacion(PERIOD_M1,Minutos);
    return((Bid-x)/Spread);
}

double Pips(double Valor)
{
    {
        for(int i=0; i<Digits; i++)
            Valor=Valor*10;

        return (NormalizeDouble(Valor,0));
    }

    //+-----+
    //|               |
    //+-----+

double PipsToPrice(double Valor)
{
    {
        for(int i=0; i<Digits; i++)
            Valor=Valor/10;
        return (Valor);
    }

    //+-----+
    //|               |
    //+-----+

bool StopTimeOperaciones(int Horas)
{
    if(StopTimeOperaciones+Horas*60>TimeCurrent())
        return (true);
    StopTimeOperaciones=0;
    return(false);
}

//+-----+
//|               |
//+-----+

double ValorMasBajo(int PERIODO)
{
    {
        int barra=iLowest(Symbol(),PERIODO,MODE_LOW,1,0);
        double x=iLow(Symbol(),PERIODO,1);
        return(x);
    }

    //+-----+
    //|               |
    //+-----+

bool UltimoMaximo(double Gap)
{
    {
        if (SubePrecio==false) return(false);
        double x=Ask;
    }
}

```

```

    if(x>UltimoMaximo)
    {
        UltimoMaximo=x;
        return (false);
    }

    if(x<=UltimoMaximo-Gap)
    {
        SubePrecio=false;
        if(x>MaximoAlcanzado) MaximoAlcanzado=x;
        return (true);
    }

    return(false);
}

//+-----+
//|               |
//+-----+
void DetectaMinimo(double Gap)
{
    if(Bid<UltimoMinimo) UltimoMinimo=Bid;
}

//+-----+
//|               |
//+-----+
void ImprimeDatosOrden(int i)
{
    if((OrderSelect(i,SELECT_BY_TICKET)==false))
        return;

    DatosOrdenes[i]=
        DatosOrdenes[i]+","+
        OrderSymbol(i)+","+
        OrderTicket(i)+","+
        OrderOpenTime(i)+","+
        OrderCloseTime(i)+","+
        NormalizeDouble(CosteOperacion(OrderOpenPrice(),OrderLots(i),2)+","+
        NormalizeDouble(OrderProfit(),2)+","+
        NormalizeDouble(OrderSwap(),2);
}

//+-----+
//|               |
//+-----+
double PorcentajeBajadaCotizacion(int Minutos)
{
    double x;
    double Maxima=MaximaCotizacion(PERIOD_M1,Minutos);
    if(Maxima==0)
        return (0);
    x=100*(Maxima-Bid)/Maxima;
    return(NormalizeDouble(x,2));
}

//+-----+
//|               |
//+-----+
double PorcentajeSubidaCotizacion(int Minutos)
{
    double x;
    double Minimo=MinimaCotizacion(PERIOD_M1,Minutos);
    if(Minimo==0)
        return (0);
    x=100*(Bid-Minimo)/Minimo;
    return(NormalizeDouble(x,2));
}

//+-----+
//|               |
//+-----+
void DibujaLineaH(double x,int col)
{
    string name="Linea"+col;
    ObjectDelete(name);
    ObjectCreate(name,OBJ_HLINE,0,TimeCurrent(),x);
    ObjectSetInteger(0,name,OBJPROP_COLOR,col);
}

//+-----+

void DibujaLineaV(double x,int col)
{
    string name=DoubleToStr(TimeCurrent());
    // ObjectDelete(name);
    ObjectCreate(name,OBJ_VLINE,0,TimeCurrent(),0);
    ObjectSetInteger(0,name,OBJPROP_COLOR,col);
}

void PendientesMaxMin()
{
    if(P3>Max3)
        Max3=P3;
    if(P3<Min3)
        Min3=P3;

    if(P2>Max2)
        Max2=P2;
    if(P2<Min2)
        Min2=P2;

    if(P1>Max1)
        Max1=P1;
    if(P1<Min1)
        Min1=P1;
}

```

```

    }

    //+-----+
    //|               |
    //+-----+
    bool NuevaMediaAlta(int PERIODO,int Barras,double Gap)
    {

        if (SubeMedia==false) return(false);
        double x=MediaActual(PERIODO,Barras);

        if(x>UltimaMediaAlta)
        {
            UltimaMediaAlta=x;
            return (false);
        }

        if(x<=UltimaMediaAlta-Gap)
        {
            UltimaMediaBaja=x;
            SubeMedia=false;
            // P2PAc=0; n2=0;
            return (true);
        }

        return(false);
    }

    //+-----+
    //|               |
    //+-----+
    bool NuevaMediaBaja(int PERIODO,int Barras,double Gap)
    {

        if (SubeMedia==true) return(false);
        double x=MediaActual(PERIODO,Barras);

        if(x<UltimaMediaBaja)
        {
            UltimaMediaBaja=x;
            return (false);
        }

        if(x>=UltimaMediaBaja+Gap)
        {
            UltimaMediaAlta=x;
            SubeMedia=true;
            // P2PAc=0; n2=0;
            return (true);
        }

        return(false);
    }

    string ParametrosApertura()
    {
        return (ProgramaYVersion+" "+
            SubeMedia+" "+
            P1+" "+
            P2+" "+
            P3+" "+
            R1+" "+
            R2+" "+
            R3+" "+
            P2PAc+" "+
            n2+" "+
            NormalizeDouble(CoeficienteDeVertigo(7),3)+" "+
            Pips(Rango)+" "+
            Pips(RangoMaximo)+" "+
            Pips(MediaP3-Bid)+" "+
            NormalizeDouble(Vib,2)+" "+
            Holgura+" "+
            iVolume(Symbol(),Periodo_Chart,1));
    }

    double CosteUnPunto()
    {
        return(Ask*MarketInfo(Symbol(),MODE_LOTSIZE)*Exchange());
    }

    double CosteOperacion(double Precio,double Lote)
    {
        return(Lote*CosteUnPunto()*Apalancamiento);
    }

    void StopLossCaida()
    {
        VendeViejas(4,10*Spread);
        VendeViejas(8,15*Spread);
        VendeViejas(12,20*Spread);
    }

    //+-----+
    //|               |
    //+-----+
    void VendePrimera()
    {

```



```

for(int i=0; i<OrdersTotal(); i++)
{

    if((OrderSelect(i,SELECT_BY_POS,MODE_TRADES)==True)
        && (OrderSymbol()==Symbol())
        && (OrderMagicNumber()==MAGICNUMBER))
    {
        int Ticket=OrderTicket();
        OrderClose(OrderTicket(),OrderLots(),Bid,3,clrBlack);
        crearcsv();
        ImprimeDatosOrden(Ticket);
        StopTimeOperaciones=TimeCurrent();
        return;
    }
}

}

void FTipoDeVela(int PER)
{

    if (NuevoBar==Time(Symbol(),PER,0)) return;
    NuevoBar=Time(Symbol(),PER,0);

    for(int i=4; i>-1; i--)
    {
        P3B[i+1]=P3B[i];
        P3A[i+1]=P3A[i];
        TipoDeVela[i+1]=TipoDeVela[i];
        Apertura[i+1]=Apertura[i];
        Cierre[i+1]=Cierre[i];
        Maximo[i+1]=Maximo[i];
        Minimo[i+1]=Minimo[i];
        HL[i+1]=HL[i];
        Cuerpo[i+1]=Cuerpo[i];
        MechaSuperior[i+1]=MechaSuperior[i];
        MechaInferior[i+1]=MechaInferior[i];
        Signo[i+1]=Signo[i];
        TamVela[i+1]=TamVela[i];
    }

    TipoDeVela[0]="";
    Apertura[0]=Open(Symbol(),PER,4);
    Cierre[0]=Close(Symbol(),PER,4);
    Maximo[0]=High(Symbol(),PER,4);
    Minimo[0]=Low(Symbol(),PER,4);
    HL[0]=Maximo[0]-Minimo[0];
    Cuerpo[0]=Cierre[0]-Apertura[0];
    MechaSuperior[0]=Maximo[0]-Cierre[0];
    MechaInferior[0]=Apertura[0]-Minimo[0];
    Signo[0]="+";
    P3B[0]=P3Bajista;
    P3A[0]=P3Alcista;

    if (Cuerpo[0]<0) // BAJISTA
    {
        MechaSuperior[0]=Maximo[0]-Apertura[0];
        MechaInferior[0]=Cierre[0]-Minimo[0];
        Signo[0]="-";
    }

    Cuerpo[0]=MathAbs(Cuerpo[0]);

    // VELA CORTA,LARGA O NORMAL

    Dim[0]=Dimension(Periodo_Chart,30);

    TamVela[0]="NORMAL" ;
    if(Cuerpo[0]>Dim[0]*1.3) TamVela[0]="LARGA" ;
    if(Cuerpo[0]<Dim[0]*0.5) TamVela[0]="CORTA" ;
    if(Cuerpo[0]>Dim[0]*2) TamVela[0]="EXTRA";

    // HAMMER

    if ((Cuerpo[0]>0)
        && (MechaInferior[0]>Cuerpo[0]*2)
        && (MechaSuperior[0]<Cuerpo[0]*0.1))

        TipoDeVela[0]="HAMMER";

    // HAMMER INVERTIDO

    if ((Cuerpo[0]>0)
        && (MechaSuperior[0]>Cuerpo[0]*2)
        && (MechaInferior[0]<Cuerpo[0]*0.1))

        TipoDeVela[0]="HAMMER INVERTIDO";

    // SPIN

    if ((TamVela[0]=="CORTA")
        && (MechaInferior[0]>Cuerpo[0]*1.1))

        TipoDeVela[0]="SPIN";

    // SPIN

    if ((TamVela[0]=="CORTA")
        && (MechaSuperior[0]>Cuerpo[0]*1.1))

        TipoDeVela[0]="SPIN";

    // DOJI

    if(Cuerpo[0]<=Dim[0]*0.01) TipoDeVela[0]="DOJI";

    // MARUBOZU

    if((MechaInferior[0]<Cuerpo[0]*0.01)
        && (MechaSuperior[0]<Cuerpo[0]*0.01)

```

```

    && (Cuerpo[0]>0))

    TipoDeVela[0]="MARUBOZU";

// RESTO DE VELAS

    if(TipoDeVela[0]=="") TipoDeVela[0]="NORMAL";
    // if(TipoDeVela[0]=="DOJI") Print(Dim[0],NuevoBar,"

// DibujaMarcador clrAqua);
// if(TipoDeVela[0]=="DOJI") Print(Dim[0],NuevoBar," ",TipoDeVela[0],Apertura[0],Cierre[0],Maximo[0],Minimo[0],HL[0],Cuerpo[0],MechaSuperior[0],MechaInferior[0],Signo[0],TamVela[0]);
// for(int i=0; i<5; i++) { Print(NuevoBar," ",TipoDeVela[i],Apertura[i],Cierre[i],Maximo[i],Minimo[i],HL[i],Cuerpo[i],MechaSuperior[i],MechaInferior[i],Signo[i],TamVela[i]);}

}

bool VelaBajista(int Periodo)

{

// if (iOpen(Symbol(),Periodo,1)>iClose(Symbol(),Periodo,1)) return(true);
if (Bid<iOpen(Symbol(),Periodo,1)) return(true);

return(false);

}

double Dimension(int PER,int Barras)
{
double x=0;
for(int i=1; i<Barras; i++) x=x+MathAbs(iClose(Symbol(),PER,i)-iOpen(Symbol(),PER,i));
return(x/Barras);

}

//+-----+
//|               |
//+-----+

double Rentabilidad (double Beneficio,double tiempo,double Invertido)
{

double FactorT=365*24*60*60/tiempo;
double Rent=Beneficio/Invertido;

return(FactorT*Rent);
}

string CruceP1P2(bool P1P2i)
{

////////////////////////////////////
if (P1P2i)!=P1P2i) return("");
P1P2=P1P2i;
////////////////////////////////////

if (P1P2i==1)
{
return ("Up");}

if (P1P2==0)
{
return ("Down");
}

return("");
}

string CruceP2P3(bool P2P3i)
{

////////////////////////////////////
if (P2P3i)!=P2P3i) return("");
P2P3=P2P3i;
////////////////////////////////////

if (P2P3i==1)
{

// DibujaLineaV(TimeCurrent(),clrWhite);
return ("DeCompra");}

if (P2P3==0)
{
// DibujaLineaV(TimeCurrent(),clrRed);
return ("DeVenta");
}

return("");
}

bool P2P3()
{
if (MediaP3>MediaP2) return (true);
return (false);
}

bool P1P2()
{
if (MediaP2>MediaP1) return (true);
return (false);
}

void DibujaTexto(string numero)

{

```

```

string Nombre= TimeCurrent();

ObjectCreate(0,Nombre,OBJ_TEXT,0,TimeCurrent(),Bid);
// ObjectSetText(Nombre,numero,8,"Arial",clrWhite);
// ObjectSetDouble(0,Nombre,OBJPROP_ANGLE,ALIGN_LEFT);

// DibujaMarcador (clrYellow);
// DibujaMarcador2 (clrYellow);
// DibujaMarcadorCaidas (clrYellow);

}

double CaidaDeDisparo()
{
    CaidaP1=0.1*(480-P1);

    CaidaRS2=2-RS2;

    Caida24Horas=Caida(1440)/6;
    Subida24Horas=Subida(1440)/6;

    if (Caida24Horas<0) Caida24Horas=0;

    CaidaDeDisparo=(Caida60+CaidaRS2-Caida24Horas+Subida24Horas);

    if (CaidaDeDisparo>6) CaidaDeDisparo=6;
    if (CaidaDeDisparo<3) CaidaDeDisparo=3;

    CaidaDeVertigoSemana=3*(1-CoeficienteDeVertigo(7));
    CaidaDeVertigoMes=3*(1-CoeficienteDeVertigo(31));

    CaidaDeDisparo=CaidaDeDisparo+CaidaDeVertigoSemana+CaidaDeVertigoMes;

    return(CaidaDeDisparo);
}
double CaidaDeDisparoOld()
{
    Caida60=0.001*(5000-P1);

    CaidaRS2=4-RS2;

    Caida24Horas=Caida(1440)/6;
    Subida24Horas=Subida(1440)/6;

    if (Caida24Horas<0) Caida24Horas=0;

    CaidaDeDisparo=(Caida60+CaidaRS2-Caida24Horas+Subida24Horas);

    if (CaidaDeDisparo>8) CaidaDeDisparo=8;
    if (CaidaDeDisparo<4) CaidaDeDisparo=4;

    CaidaDeVertigoSemana=3*(1-CoeficienteDeVertigo(7));
    CaidaDeVertigoMes=3*(1-CoeficienteDeVertigo(31));

    CaidaDeDisparo=CaidaDeDisparo+CaidaDeVertigoSemana+CaidaDeVertigoMes;

    return(CaidaDeDisparo);
}

bool Control()
{
    if(TimeCurrent()>StringToTime(FechaExpiracion)==true)
    {
        Comment("EXPERT ADVISOR EXPIRED, NECESSARY TO RENEW THE LICENCE. PLEASE CONTACT WITH (service@eapro.info) "); //
        return(false);
    }

    if(IsTradeAllowed(Symbol(),TimeCurrent())==false)
    {
        Comment("CLOSED MARKET. WAITING FOR OPENING");
        return(false);
    }

    return (true);
}

bool CerrarYa()
{
    if (SaldoBUY<=0) return (false);

    double x=SaldoMaximo-SaldoBUY;

    if (Bid-PrecioDeCompra<8*Spread) return (false);

    if (x/SaldoMaximo>0.2) return(true);

    return(false);
}

//////////////////////////////// PATRONES DE COMPRA //////////////////////////////////

bool PatronMartilloInvertidoAlcista()
{
    if (P3B[0]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]=="HAMMER INVERTIDO") return(false);
}

```

```

for(int i=1; i<4; i++) if(Signo[i]=="+") return(false);

return (true);
}

bool PatronMartilloAlcista()
{
    if (P3B[0]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="HAMMER") return(false);

    for(int i=1; i<4; i++) if(Signo[i]=="+") return(false);

    return (true);
}

bool PatronBeltHoldAlcista() // BELT HOLD
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="MARUBOZU") return (false);

    if (Signo[0]=="-") return (false);
    if (Signo[1]=="+") return (false);

    if (TamVela[0]!="LARGA") return (false);

    if (Cuerpo[0]<Cuerpo[1]*1.3) return (false);

    if (Apertura[0]<Cierre[1]) return (false);

    return (true);
}

bool PatronEnvolventeAlcista() // ENVOLVENTE
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[0]=="-") return (false);
    if (Signo[1]=="+") return (false);

    if (TamVela[0]!="CORTA") return (false);
    if (TamVela[1]!="CORTA") return (false);
    if (TamVela[0]!="EXTRA") return (false);
    if (TamVela[1]!="EXTRA") return (false);

    if (Cuerpo[1]*1.2>Cuerpo[0]) return (false);

    if (Cierre[1]<Apertura[0]) return (false);
    if (Apertura[1]>Cierre[0]) return (false);

    Print(P3B[1]);

    return (true);
}

bool PatronHaramiCrossAlcista() // HARAMI CROSS
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="DOJI") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[1]!="-") return (false);

    if (TamVela[1]!="LARGA") return (false);
    if (TamVela[1]!="CORTA") return (false);

    if (Maximo[1]<Maximo[0]) return (false);

    if (Minimo[1]>Minimo[0]) return (false);

    return (true);
}

bool PatronHaramiAlcista() // HARAMI CROSS
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[1]!="-") return (false);
    if (Signo[0]!="+") return (false);

    if (TamVela[0]!="CORTA") return (false);

    if (Apertura[1]-Dim[0]*0.2<Cierre[0]) return (false);
    if (Cierre[1]>Apertura[0]) return(false);

    return (true);
}

```

```

}

bool PatronEstrellaDojiAlcista()
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="DOJI") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[1]=="+") return (false);
    if (TamVela[1]=="CORTA") return (false);

    if (MechaInferior[0]>Dim[0]*0.4) return (false);
    if (MechaSuperior[0]>Dim[0]*0.4) return (false);

    if (Apertura[1]<Apertura[0]) return(false);

    return (true);
}

bool PatronPenetranteAlcista()
{
    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[0]=="-") return (false);
    if (Signo[1]=="+") return (false);

    if (TamVela[0]!="EXTRA") return (false);
    if (TamVela[0]!="CORTA") return (false);
    if (TamVela[1]!="LARGA") return (false);

    if (Minimo[1]<Apertura[0]) return (false);
    if (Apertura[1]<Cierre[0]) return (false);
    if (Cierre[0]<Cierre[1]+Cuerpo[1]*.6) return (false);

    return (true);
}

bool PatronLineasEncontradasAlcista()
{
    double Gap=Dim[0]*.02;

    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[0]!="+") return (false);
    if (Signo[1]!="-") return (false);

    if (TamVela[1]!="CORTA") return (false);

    if(Apertura[0]>Minimo[1]) return(false);

    if (Cuerpo[0]<Cuerpo[1]) return (false);

    if (Cierre[0]>Cierre[1]+Gap) return(false);

    if (Cierre[0]<Cierre[1]-Gap) return(false);

    return (true);
}

bool PatronMinimosCoincidentesAlcista()
{
    double Gap=Dim[0]*.05;

    if (P3B[1]>P3BajistaEsperado) return (false);

    if (TipoDeVela[0]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if (Signo[0]!="-") return (false);
    if (Signo[1]!="-") return (false);

    if (TamVela[1]!="CORTA") return (false);

    if (Apertura[0]>Apertura[1]-Cuerpo[1]*.1) return(false);

    if (MechaInferior[0]>Gap) return(false);
    if (MechaInferior[1]>Gap) return(false);

    if (Cierre[0]>Cierre[1]+Gap) return(false);
    if (Cierre[0]<Cierre[1]-Gap) return(false);

    return (true);
}

bool PatronBebeAbandonadoAlcista()
{
    double Gap=Dim[0]*.01;

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (TipoDeVela[2]!="NORMAL") return (false);
    if (TipoDeVela[1]!="DOJI") return (false);
    if (TipoDeVela[0]!="NORMAL") return (false);

```

```

if (Signo[2]!=".") return (false);
if (Signo[0]!="+*") return (false);

if (TamVela[2]!="LARGA") return (false);
if (TamVela[0]!="LARGA") return (false);

if (Minimo[2]<Cierre[1]-Gap) return (false);
if (Minimo[0]<Cierre[1]-Gap) return (false);

Print("4");

if (Cierre[0]>Apertura[2]) return (false);
Print("5");

return (true);
}

bool PatronMorningStarAlcista()
{
    double Gap=Dim[0]*.05;

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (TipoDeVela[2]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);
    if (TipoDeVela[0]!="NORMAL") return (false);

    if (Signo[2]!=".") return (false);
    if (Signo[0]!="+*") return (false);

    if (TamVela[2]!="LARGA") return (false);
    if (TamVela[1]!="CORTA") return (false);
    if (TamVela[0]!="LARGA") return (false);

    if (Apertura[1]>Cierre[2]) return (false);
    if (Apertura[0]<Cierre[1]) return (false);
    if (Cierre[0]>Apertura[2]) return(false);

    return (true);
}

bool PatronMorningStarDojiAlcista()
{
    double Gap=Dim[0]*.05;

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (TipoDeVela[2]!="NORMAL") return (false);
    if (TipoDeVela[1]!="DOJI") return (false);
    if (TipoDeVela[0]!="NORMAL") return (false);

    if (Signo[2]!=".") return (false);
    if (Signo[0]!="+*") return (false);

    if (TamVela[2]!="LARGA") return (false);
    if (TamVela[1]!="CORTA") return (false);
    if (TamVela[0]!="LARGA") return (false);

    if (Apertura[1]>Cierre[2]) return (false);
    if (Apertura[0]<Cierre[1]) return (false);
    if (Cierre[0]>Apertura[2]) return(false);

    return (true);
}

bool PatronTresEstrellasEnElSurAlcista()
{
    double Gap=Dim[0]*.05;
    bool Seguimos=false;

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (TipoDeVela[2]!="NORMAL") return (false);
    if (TipoDeVela[1]!="NORMAL") return (false);

    if ((TipoDeVela[0]=="DOJI") || (TipoDeVela[0]=="MARUBOZU")) Seguimos=true;
    if (Seguimos==false) return (false);

    if ((TipoDeVela[0]=="MARUBOZU") && Signo[1]=="+") return(false);

    if (Signo[2]=="+") return (false);
    if (Signo[1]=="+") return (false);

    if (TamVela[2]=="CORTA") return (false);

    if (MechaInferior[2]<Cuerpo[2]*0.2) return (false);

    if (Minimo[2]>Minimo[1]) return (false);
    if (Minimo[1]>Minimo[0]) return (false);

    if (HL[2]<HL[1]) return (false);
    if (HL[1]<HL[0]) return (false);

    if (Apertura[2]<Apertura[1]) return (false);
    if (Apertura[1]<Apertura[0]) return (false);

```

```
return (true);
}

bool PatronTresSoldadosBlancosAlcista()
{

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (Signo[2]!="+") return (false);
    if (Signo[1]!="+") return (false);
    if (Signo[0]!="+") return (false);

    for(int i=0; i<3; i++)
    {
        if (TamVela[i]=="EXTRA") return(false);
        if (TipoDeVela[i]=="DOJI") return(false);
        if (TipoDeVela[i]=="HAMMER") return(false);
        if (TipoDeVela[i]=="HAMMER INVERTIDO") return(false);
    }

    if (Apertura[2]>Apertura[1]) return (false);
    if (Apertura[1]>Apertura[0]) return (false);

    if (Cierre[2]<Apertura[1]) return (false);
    if (Cierre[1]<Apertura[0]) return (false);

    // if (Cuerpo[1] < Cuerpo[0]*.5) return (false);
    // if (Cuerpo[2] < Cuerpo[1]*.5) return (false);

    // if (MechaSuperior[2]>Cuerpo[2]*.2) return (false);
    // if (MechaSuperior[1]>Cuerpo[1]*.2) return (false);
    // if (MechaSuperior[0]>Cuerpo[0]*.2) return (false);

    return (true);
}

bool PatronTresVelasExterioresAlcista()
{

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (Signo[2]!=".") return (false);
    if (Signo[1]!="+") return (false);
    if (Signo[0]!="+") return (false);

    for(int i=0; i<3; i++)
    {
        if (TipoDeVela[i]=="SPIN") return(false);
        if (TipoDeVela[i]=="DOJI") return(false);
        if (TipoDeVela[i]=="HAMMER") return(false);
        if (TipoDeVela[i]=="HAMMER INVERTIDO") return(false);
    }

    for(int i=0; i<3; i++)
    {
        if (TamVela[i]=="CORTA") return(false);
        if (TamVela[i]=="EXTRA") return(false);
    }

    if (Apertura[2]>Cierre[1]) return (false);
    if (Cierre[2]<Apertura[1]) return (false);

    if (Cierre[0]<Cierre[1]) return (false);

    return (true);
}

bool PatronTresVelasInterioresAlcista()
{

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (Signo[2]!=".") return (false);
    if (Signo[1]!="+") return (false);
    if (Signo[0]!="+") return (false);

    for(int i=0; i<3; i++)
    {
        if (TipoDeVela[i]=="DOJI") return(false);
        if (TipoDeVela[i]=="HAMMER") return(false);
        if (TipoDeVela[i]=="HAMMER INVERTIDO") return(false);
    }

    if (TamVela[2]=="CORTA") return(false);
    if (TamVela[1]=="CORTA") return(false);
    if (TamVela[0]=="LARGA") return(false);

    if (Apertura[2]<Cierre[1]) return (false);
    if (Cierre[2]>Apertura[1]) return (false);

    if (Apertura[0]>Cierre[1]) return (false);

    if (Cierre[0]<Apertura[2]) return (false);

    return (true);
}

bool PatronTresEstrellasAlcista()
{

    if (P3B[2]>P3BajistaEsperado) return (false);

    for(int i=0; i<3; i++)
    {
        if (TipoDeVela[i]=="DOJI") return(false);
    }

    if (Apertura[1]>Apertura[2]) return(false);

    return (true);
}
```

```

}

bool PatronSueloDeTresRiosAlcista()
{

    if (P3B[2]>P3BajistaEsperado) return (false);

    if (Signo[2]!= "-" ) return (false);
    if (Signo[1]!= "-" ) return (false);
    if (Signo[0]!= "-" + "-" ) return (false);

    Print("1");

    if (TipoDeVela[2]!= "NORMAL") return(false);
    if (TipoDeVela[1]!= "HAMMER") return(false);
    if (TipoDeVela[0]!= "NORMAL") return(false);

    Print("2");


    if (TamVela[2]== "LARGA") return(false);
    if (TamVela[1]!= "CORTA") return(false);
    if (TamVela[0]!= "CORTA") return(false);

    if (Apertura[2]<Cierre[1]) return (false);
    if (Cierre[2]>Minimo[1]) return (false);
    if (Minimo[2]>Minimo[1]) return (false);

    if (Cierre[1]<Cierre[0]) return (false);

    return (true);
}

bool PatronPequenaGolondrinaEscondidaAlcista()
{

    bool Seguimos=false;

    if (P3B[3]>P3BajistaEsperado) return (false);

    if (Signo[3]!= "-" ) return (false);
    if (Signo[2]!= "-" ) return (false);
    if (Signo[1]!= "-" ) return (false);
    if (Signo[0]!= "-" ) return (false);

    Print("1");

    if (TipoDeVela[3]!= "MARUBOZU") return(false);
    if (TipoDeVela[2]!= "MARUBOZU") return(false);

    Print("0");
    if (TipoDeVela[1]!= "HAMMER INVERTIDO") return(false);

    if (TipoDeVela[0]== "NORMAL") Seguimos=true;
    if (Seguimos==false) return (false);

    if (TipoDeVela[0]== "MARUBOZU") Seguimos=true;
    if (Seguimos==false) return (false);

    Print("2");


    if (TamVela[3]== "LARGA") return(false);
    if (TamVela[2]== "LARGA") return(false);
    if (TamVela[1]!= "CORTA") return(false);
    if (TamVela[0]!= "LARGA") return(false);

    if (Apertura[3]<Apertura[2]) return (false);
    if (Apertura[2]<Apertura[1]) return (false);
    if (Cierre[3]<Cierre[2]) return(false);
    if (Cierre[2]<Cierre[1]) return(false);

    if (Minimo[1]<Minimo[0]) return (false);

    if (Maximo[1]>Maximo[0]) return (false);

    return (true);
}

bool PatronEscapeAlcista()
{

    bool Seguimos=false;

    if (P3B[4]>P3BajistaEsperado) return (false);

    if (Signo[4]!= "-" ) return (false);
    if (Signo[3]!= "-" ) return (false);
    if (Signo[1]!= "-" ) return (false);
    if (Signo[0]!= "-" + "-" ) return (false);

    if (TipoDeVela[4]!= "NORMAL") return(false);

    if (TipoDeVela[3]== "NORMAL") Seguimos=true;
    if (Seguimos==false) return (false);
    if (TipoDeVela[3]== "SPIN") Seguimos=true;
    if (Seguimos==false) return (false);

    if (TipoDeVela[2]== "NORMAL") Seguimos=true;
    if (Seguimos==false) return (false);
    if (TipoDeVela[2]== "SPIN") Seguimos=true;
    if (Seguimos==false) return (false);

    if (TipoDeVela[1]== "NORMAL") Seguimos=true;
    if (Seguimos==false) return (false);
    if (TipoDeVela[1]== "SPIN") Seguimos=true;
    if (Seguimos==false) return (false);

    if (TipoDeVela[0]!= "NORMAL") return(false);

    Print("2");

```



```
if (TamVela[4])!="LARGA") return(false);
if (TamVela[3])=="CORTA") return(false);
if (TamVela[2])=="CORTA") return(false);
if (TamVela[1])=="CORTA") return(false);
if (TamVela[0])!="LARGA") return(false);

if (Cierre[4]-0.5*Dim[0]<Cierre[3]) return(false);
if (Cierre[3]<Cierre[2]) return(false);
if (Cierre[2]<Cierre[1]) return(false);
if (Cierre[0]>Cierre[4]) return(false);
if (Cierre[0]<Apertura[1]) return(false);

return(true);
}

bool PatronPatadaOCozAlcista()
{
    if (Signo[1])!=".") return(false);
    if (Signo[0])!="+.") return(false);

    if (TamVela[0])=="EXTRA") return(false);

    if (TipoDeVela[1])!="MARUBOZU") return(false);
    if (TipoDeVela[0])!="MARUBOZU") return(false);

    Print("2");

    if (Apertura[0]<0.2*Dim[0]+Apertura[1]) return(false);

    return(true);
}

bool PatronHuecoAlcistaDeTripleFormacionAlcista()
{
    if (P3A[2]<0) return(false);

    if (Signo[2])!="+.") return(false);
    if (Signo[1])!="+.") return(false);
    if (Signo[0])!=".") return(false);

    if (TamVela[2])!="LARGA") return(false);
    if (TamVela[1])!="LARGA") return(false);
    if (TamVela[2])=="EXTRA") return(false);
    if (TamVela[1])=="EXTRA") return(false);

    if (TipoDeVela[2])!="NORMAL") return(false);
    if (TipoDeVela[1])!="NORMAL") return(false);

    Print("2");

    if (Cierre[2]+0.2*Dim[0]<Apertura[1]) return(false);
    if (Apertura[0]<Apertura[1]) return(false);
    if (Cierre[0]>Cierre[2]) return(false);

    return(true);
}

bool PatronHuecoAlcistaTasukiAlcista()
{
    if (P3A[2]<0) return(false);

    if (Signo[2])!="+.") return(false);
    if (Signo[1])!="+.") return(false);
    if (Signo[0])!=".") return(false);

    if (TamVela[2])=="EXTRA") return(false);
    if (TamVela[1])=="EXTRA") return(false);

    if (TipoDeVela[2])!="NORMAL") return(false);
    if (TipoDeVela[1])!="NORMAL") return(false);

    if (Cierre[2]+Dim[0]*.4>Apertura[1]) return(false);
    if (Apertura[0]<Apertura[1]) return(false);
    if (Apertura[0]<Cierre[2]) return(false);
    if (Cierre[0]<Cierre[2]) return(false);
    if (Apertura[0]>Apertura[1]) return(false);

    return(true);
}

bool PatronTripleGolpeAlcista()
{
    if (P3A[3]<0) return(false);

    if (Signo[3])!="+.") return(false);
    if (Signo[2])!="+.") return(false);
    if (Signo[1])!="+.") return(false);
    if (Signo[0])!=".") return(false);

    for(int i=1; i<4; i++)
    {
        if (TipoDeVela[i])=="SPIN") return(false);
        if (TipoDeVela[i])=="DOJI") return(false);
        if (TipoDeVela[i])=="HAMMER") return(false);
        if (TipoDeVela[i])=="HAMMER INVERTIDO") return(false);
    }

    if (Apertura[3]>Apertura[2]) return(false);
    if (Apertura[2]>Apertura[1]) return(false);

    if (Cierre[3]<Apertura[2]) return(false);
    if (Cierre[2]<Apertura[1]) return(false);

    if (Cuerpo[2] < Cuerpo[1]*.7) return(false);
    if (Cuerpo[3] < Cuerpo[2]*.7) return(false);

    if (MechaSuperior[3]>Cuerpo[3]*.2) return(false);
```

```
if (MechaSuperior[2]>Cuerpo[2]*.2) return (false);  
if (MechaSuperior[1]>Cuerpo[1]*.2) return (false);  
  
if (Apertura[0]<Cierre[1]) return(false);  
if (Cierre[0]>Cierre[3]) return(false);  
  
return (true);  
}
```