

```

//+-----+
//|                                     experimeto.mq5 |
//|                                     Copyright 2023, MetaQuotes Ltd. |
//|                                     https://www.mql5.com |
//+-----+
#property copyright "Copyright 2023, MetaQuotes Ltd."
#property link      "https://www.mql5.com"
#property version   "1.00"
//+-----+
//| Expert initialization function |
//+-----+
#include <trade\trade.mqh>
CTrade trade;
ulong BilheteDeVenda = 0; // Inicializado com 0 para indicar que não há ordem de venda aberta
ulong BilheteDeCompra = 0; // Inicializado com 0 para indicar que não há ordem de compra aberta
ulong ticketvenda = 0; // Inicializado com 0 para indicar que não há ordem de venda aberta
ulong ticketcompra = 0; // Inicializado com 0 para indicar que não há ordem de compra aberta
input bool surfada = false;
input datetime target_time = D'2023.12.05 20:10:00';
input int MagicNumber = 12345; // Unique identifier for the EA
input string password = "jossias"; // password
input ENUM_TIMEFRAMES PeríodoOperacional = PERIOD_M15; // Período Operacional
input bool ativar_ou_desativar_venda = true; // ativar_ou_desativar_venda
input bool ativar_ou_desativar_compra = true; // ativar_ou_desativar_compra
input double compra; // preço de compra
input double venda; // preço de venda
input double lot = 0.01; // lot
input double santo; // pontos para fora da caixa
input bool Modificar_Sl_Para_Ox0 = true; // colocar a ordem no 0
input double dedo = 0; // pontos para rompimento da vela
input bool condicao_De_rompimento_v = true; // abrir a condição para rompimento da vela na venda
input bool condicao_De_rompimento_c = true; // abrir a condição para rompimento da vela na compra
input bool Costura = false;
input int Nives = 1;
input string _____novas_funcoes_____ ;
input bool NOVAS_CONDICOES = false; // NOVAS_CONDICOES
input double preco_para_compra = 0.0; // preco_para_compra em NOVAS_CONDICOES
input double preco_para_venda = 0.0; // preco_para_venda em NOVAS_CONDICOES
double fora; // variavel para santo
double pc; // preço de compra pra linha
double pv; // preço de venda pra linha
double PrecoDeCompra; // preço de compra pra ordem
double PrecoDeVenda; // preço de venda pra ordem
double var1; // possível preço de compra
double var2; // possível preço de venda
double Pontos; // pontos para linhas
double divisao; // divisão
double Buytake; // take de compra
double Buystop; // estop de compra
double Selltake; // teke de venda
double Sellestop; //estop de venda
double SellModif; // modificar estop sell
double Buymodif; // modificar estop buy
double Buysubsicul; // subsicol de compra
double Sellsuscul; // subsicol de venda
int tpAtingido = 0; // contegem de take profit
int slAtingido = 0; // contegem de stop loss
int slzero = 0; //contegem de zero zero
double preco_de_abertura_de_venda; // preço de abertura de veda
double preco_de_abertura_de_compra; // preço de abertura de compra
double nivelzerobuy; //preco sem perda para compra
double nivelzerosell; // preço sem perda para venda
double nivelstoplos_sell; // preço de stop loss de venda
double nivelstoplos_buy; // preço de stop loss de compra
double linhasell; // criar linha de venda para linha
double linhabuy; // criar linha de compra para linha
double linhasubsiculv; // criar linha de subsicol de venda
double linhasubsiculc; //criar linha de subsicol de compra
double pontosf; // pontos para ordem
double takbuy; // nivel de take buy
double taksell; // nivel de take sell
double tamanho_DA_vela_para_compra = 0.0;
double tamanho_DA_vela_para_venda = 0.0;
//int canais = 0;
int p;
//+-----+
//| |
//+-----+
double precosArray [];
//+-----+
//| |
//+-----+
//+-----+
//| controles |

```

```

//+-----+
bool controlbuy = false;          // control de ordem de compra
bool controlsell = false;         // control de ordem de venda
bool controlbuymdf = false;       // control de modificação de ordem de compra
bool controlsellmdf = false;      // control de modificação de ordem de venda
bool linhasc = false; // control de linha de compra
bool linhasv = false; // control de linha de venda
bool segundo_control_de_taksell = false;
bool segundo_control_de_takbuy = false ;
bool cont = false ;
bool cont1 = false ;
bool cont2 = false ;
bool controlsurfada1 = false;
bool controlsurfada2 = false;
bool control_de_venda = false; // segundo control para venda
bool control_de_compra = false; // segundo control para compra
bool control_de_venda2 = false;
bool control_de_compra2 = false;

//+-----+
//|
//+-----+
void criarlinhascompra(string precocompra, double valor)
{
    ObjectCreate(0, precocompra, OBJ_HLINE, 0, 0, valor);
    ObjectSetInteger(0, precocompra, OBJPROP_STYLE, STYLE_SOLID);
    ObjectSetInteger(0, precocompra, OBJPROP_COLOR, clrRed);
}
//+-----+
void criarlinhavenda(string precovanda, double valor)
{
    ObjectCreate(0, precovanda, OBJ_HLINE, 0, 0, valor);
    ObjectSetInteger(0, precovanda, OBJPROP_STYLE, STYLE_SOLID);
    ObjectSetInteger(0, precovanda, OBJPROP_COLOR, clrRed);
}
//
//+-----+
//|
//+-----+
int OnInit()
{
    if(password == "jossias")
    {
    }
    else
    {
        ExpertRemove();
    }
    var1 = compra;
    var2 = venda;
    PrecoDeCompra = var1;
    PrecoDeVenda = var2;
    pontosf = var1 - var2;
    divisao = pontosf / 2;
    Buytake = PrecoDeCompra + (pontosf * Nives);
    Buystop = PrecoDeCompra - pontosf;
    Buysubsicul = PrecoDeCompra + divisao;
    Buymodif = PrecoDeCompra;
    Selltake = PrecoDeVenda - (pontosf * Nives);
    Sellestop = PrecoDeVenda + pontosf;
    Sellsubsicul = PrecoDeVenda - divisao;
    SellModif = PrecoDeVenda;

    double santinho = SymbolInfoDouble(_Symbol, SYMBOL_POINT);
    fora = santinho * santo ;

    // Definir o símbolo e o período
    string symbol = Symbol();

    // Obter a hora da vela atual
    datetime currentTime = iTime(symbol, PeríodoOperacional, 0);

//+-----+
//| Expert deinitialization function
//+-----+
    var1 = compra;
    var2 = venda;
    pc = var1 ;
    pv = var2 ;
    Pontos = pc - pv;
    divisao = Pontos / 2;
    linhabuy = pc + Pontos;

```

```

        linhasubsiculc = pc + divisao;
//+-----+
//| Expert deinitialization function |
//+-----+
        pc = var1 ;
        pv = var2 ;
        Pontos = pc - pv;
        divisao = Pontos / 2;
        linhasell = pv - Pontos;
        linhasubsiculv = pv - divisao;

//+-----+
//|
//+-----+
        criarlinhavenda("prevodevenda", pv);
        criarlinhascompra("precode compra ", pc);
        return(INIT_SUCCEEDED);
    }
//+-----+
//| Expert deinitialization function |
//+-----+
void OnDeinit(const int reason)
{
    ObjectsDeleteAll(0, "precocompra", 0, OBJ_HLINE);
    ObjectsDeleteAll(0, "precovanda", 0, OBJ_HLINE);
    ObjectsDeleteAll(0, "linha3_", 0, OBJ_HLINE);
    ObjectsDeleteAll(0, "linha2_", 0, OBJ_HLINE);
    ObjectsDeleteAll(0, "linha1_", 0, OBJ_HLINE);
    ObjectsDeleteAll(0, "linha_", 0, OBJ_HLINE);
//---
}
//+-----+
//| Expert tick function |
//+-----+
void OnTick()
{
    double nivel = MathAbs(PrecioDeCompra - PrecioDeVenda) / SymbolInfoDouble(_Symbol, SYMBOL_POINT);
    double precocurrentecB = SymbolInfoDouble(_Symbol, SYMBOL_BID);
    double precocurrentevA = SymbolInfoDouble(_Symbol, SYMBOL_ASK);
    double closeprice = iClose(Symbol(), PeriodoOperacional, 1);
    double precodiferentev = MathAbs(closeprice - PrecioDeVenda) / Point();
    double precodiferentec = MathAbs(PrecioDeCompra - closeprice) / Point();
    datetime current_time = TimeLocal();
    string symbol = Symbol();
    datetime currentTime = iTime(symbol, PeriodoOperacional, 0);
    int totalPositions = PositionsTotal() ;

    int bingala_de_alta = int(tamanho_DA_vela_para_compra);
    int espassao = int (pontosf / Point());
    int canais = (bingala_de_alta / espassao);

    int bingala_de_baixa = int (tamanho_DA_vela_para_venda);
    int canaisv = (bingala_de_baixa / espassao);

    if(current_time >= target_time)
    {
    }

//+-----+
//| CRIAÇÃO DE LINHAS |
//+-----+
    if(closeprice > pc)
    {
        criarlinhaBuy1(linhasubsiculc);
        criarlinhaBuy2(linhabuy) ;
    }
//+-----+
//| CRIAÇÃO DE LINHAS |
//+-----+

    if(closeprice > linhabuy)
    {
        pc = pc + Pontos ;
        pv = pv + Pontos ;
        Pontos = pc - pv;
        divisao = Pontos / 2;
        linhabuy = pc + Pontos;
        linhasubsiculc = pc + divisao;
        linhasell = pv - Pontos;
        linhasubsiculv = pv - divisao;
    }

```

```

//+-----+
//|                CRIAÇÃO DE LINHAS                |
//+-----+

    if(closeprice < pv)
    {
        criarlinhassell1(linhasubsiculv);
        criarlinhassell2(linhasell);
    }
//+-----+
//|                CRIAÇÃO DE LINHAS                |
//+-----+

    if(closeprice < linhasubsiculv)
    {
        pc = pc - Pontos ;
        pv = pv - Pontos ;
        Pontos = pc - pv;
        divisao = Pontos / 2;
        linhasell = pv - Pontos;
        linhasubsiculv = pv - divisao;
        linhabuy = pc + Pontos;
        linhasubsiculc = pc + divisao;
    }
//+-----+
//|                |
//+-----+
    if(precocurrentecB >= takbuy)
    {
        if(!segundo_control_de_takbuy)
        {
            PrecoDeVenda = PrecoDeCompra - pontosf;
            Buytake = PrecoDeCompra + (pontosf * Nives);
            Buystop = PrecoDeCompra - pontosf;
            Buysubsicul = PrecoDeCompra + divisao;
            Buymodif = PrecoDeCompra;
            Selltake = PrecoDeVenda - (pontosf * Nives);
            Sellestop = PrecoDeVenda + pontosf;
            Sellsubsicul = PrecoDeVenda - divisao;
            SellModif = PrecoDeVenda;

            /* Print("novo preço de compra. :", PrecoDeCompra);
            Print(" novo preço de venda. :", PrecoDeVenda);
            Print(" novo Buysubsicul. :", Buysubsicul);
            Print("novo Buytake. :", Buytake);
            Print("novo Buystop. :", Buystop);
            Print(" novo Sellsubsicul. :", Sellsubsicul);
            Print(" novo selltake : ", Selltake);
            Print("novo Sellestop. :", Sellestop);
            */segundo_control_de_takbuy = true;
        }
    }
//+-----+
//|                |
//+-----+
    if(precocurrentevA <= taksell)
    {
        if(!segundo_control_de_taksell)
        {
            PrecoDeCompra = PrecoDeVenda + pontosf;
            Buytake = PrecoDeCompra + (pontosf * Nives);
            Buystop = PrecoDeCompra - pontosf;
            Buysubsicul = PrecoDeCompra + divisao;
            Buymodif = PrecoDeCompra;
            Selltake = PrecoDeVenda - (pontosf * Nives);
            Sellestop = PrecoDeVenda + pontosf;
            Sellsubsicul = PrecoDeVenda - divisao;
            SellModif = PrecoDeVenda;

            /* Print("novo preço de compra. :", PrecoDeCompra);
            Print(" novo preço de venda. :", PrecoDeVenda);
            Print(" novo Buysubsicul. :", Buysubsicul);
            Print("novo Buytake. :", Buytake);
            Print("novo Buystop. :", Buystop);
            Print(" novo Sellsubsicul. :", Sellsubsicul);
            Print(" novo selltake : ", Selltake);
            Print("novo Sellestop. :", Sellestop);
            */segundo_control_de_taksell = true;
        }
    }
    if(closeprice < PrecoDeVenda)
    {
    }
}

```

```

        if(pecocurrentecB || pecocurrentevA <= nivelstoplos_buy)
        {
            if(totalPositions == 0)
            {
                controlsell = false;
            }
        }
    }
}

if(closeprice > PrecoDeCompra)
{
    if(pecocurrentecB || pecocurrentevA >= nivelstoplos_sell)
    {
        if(totalPositions == 0)
        {
            controlbuy = false;
        }
    }
}

//+-----+
//|
//+-----+
// Obter a última vela
MqlRates rates[1];

// Copiar os dados da última vela para a matriz 'rates'
int copied = CopyRates(symbol, PeríodoOperacional, 0, 1, rates);

if(copied > 0)
{
    // Última vela
    double lastOpen = rates[0].open;
    double lastClose = rates[0].close;

    // Calcular o tamanho da vela
    if(lastOpen > lastClose)
    {
        tamanho_DA_vela_para_compra = MathAbs(lastClose - PrecoDeCompra) / Point();
    }
    if(lastOpen < lastClose)
    {
        tamanho_DA_vela_para_venda = MathAbs(lastClose - PrecoDeVenda) / Point();
    }

    // Imprimir o tamanho da vela
    // Print("Tamanho da última vela: ", tamanho_DA_vela_para_compra);
}
else
{
    Print("Erro ao copiar os dados da vela: ", GetLastError());
}

}

bool condicao2 = (condicao_De_rompimento_c) ? // condição para rompimento da vela de compra
                (precodiferentec < dedo) : true ; //
if(surfada)
{
    if(closeprice > PrecoDeCompra)
    {
        if(! controlbuy)
        {
            if(totalPositions == 1)
            {
                if(bingala_de_alta > espassao)
                {
                    for(int i = 1 ; i <= canais ; i++)
                    {
                        PrecoDeVenda = PrecoDeCompra;
                        PrecoDeCompra = Buytake;
                    }
                }
            }
        }
    }
}

```

```

        Buytake = PrecoDeCompra + (pontofsf * Nives);
        Buystop = PrecoDeCompra - pontofsf;
        Buysubsicul = PrecoDeCompra + divisao;
        Buymodif = PrecoDeVenda;
        Selltake = PrecoDeVenda - (pontofsf * Nives);
        Sellestop = PrecoDeVenda + pontofsf;
        Sellsubsicul = PrecoDeVenda - divisao;
        SellModif = PrecoDeVenda;

        Print("novo preço de compra. :", PrecoDeCompra);
        Print(" novo preço de venda. :", PrecoDeVenda);
        Print(" novo Buysubsicul. :", Buysubsicul);
        Print(" precos multiplicados de compra aaaaaaaaaaaaaaaaaa");
        Print("novo Buytake. :", Buytake);
        Print("novo Buystop. :", Buystop);
        Print(" novo Sellsubsicul. :", Sellsubsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);

    }

}

PrecoDeCompra = PrecoDeCompra ;
PrecoDeVenda = PrecoDeCompra - pontofsf ;
Buytake = PrecoDeCompra + (pontofsf * Nives);
Buystop = PrecoDeCompra - pontofsf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeVenda;
Selltake = PrecoDeVenda - (pontofsf * Nives);
Sellestop = PrecoDeVenda + pontofsf;
Sellsubsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeVenda;

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print(" so maior que compra");
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{
    control_de_venda = false;
    control_de_compra = false;
}
controlsell = false ;
controlbuymdf = false;
segundo_control_de_takbuy = false;
segundo_control_de_taksell = false;
cont = false;

controlbuy = true;
}
else
{

    if(! control_de_compra)
    {
        if(bingala_de_alta > espasao)
        {
            for(int i = 1 ; i <= canais ; i++)
            {
                PrecoDeVenda = PrecoDeCompra;
                PrecoDeCompra = Buytake;
                Buytake = PrecoDeCompra + (pontofsf * Nives);
                Buystop = PrecoDeCompra - pontofsf;
                Buysubsicul = PrecoDeCompra + divisao;
                Buymodif = PrecoDeVenda;
                Selltake = PrecoDeVenda - (pontofsf * Nives);
                Sellestop = PrecoDeVenda + pontofsf;
                Sellsubsicul = PrecoDeVenda - divisao;
                SellModif = PrecoDeVenda;

                Print("novo preço de compra. :", PrecoDeCompra);
                Print(" novo preço de venda. :", PrecoDeVenda);
                Print(" novo Buysubsicul. :", Buysubsicul);
                Print("novo Buytake. :", Buytake);
                Print("novo Buystop. :", Buystop);
                Print(" novo Sellsubsicul. :", Sellsubsicul);
                Print(" novo selltake : ", Selltake);
            }
        }
    }
}

```

```

        Print("novo Sellestop. :", Sellestop);
    }

}

if(ativar_ou_desativar_compra && condicao2)
{
    BuyMarkt();
}

PrecoDeCompra = PrecoDeCompra ;
PrecoDeVenda = PrecoDeCompra - pontosf ;
Buytake = PrecoDeCompra + (pontosf * Nives);
Buystop = PrecoDeCompra - pontosf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeVenda;
Selltake = PrecoDeVenda - (pontosf * Nives);
Sellestop = PrecoDeVenda + pontosf;
Sellsubsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeVenda;

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{

}

controlsell = false ;
controlbuymdf = false;
segundo_control_de_takbuy = false;
segundo_control_de_taksell = false;
cont = false;
controlbuy = false;
control_de_compra = true;
}

}

}

}

else
{
    if(!controlbuy)
    {
        if(closeprice > PrecoDeCompra)
        {

            if(bingala_de_alta > espasao)
            {
                for(int i = 1 ; i <= canais ; i++)
                {
                    PrecoDeVenda = PrecoDeCompra;
                    PrecoDeCompra = Buytake;
                    Buytake = PrecoDeCompra + (pontosf * Nives);
                    Buystop = PrecoDeCompra - pontosf;
                    Buysubsicul = PrecoDeCompra + divisao;
                    Buymodif = PrecoDeVenda;
                    Selltake = PrecoDeVenda - (pontosf * Nives);
                    Sellestop = PrecoDeVenda + pontosf;
                    Sellsubsicul = PrecoDeVenda - divisao;
                    SellModif = PrecoDeVenda;

                    Print("novo preço de compra. :", PrecoDeCompra);
                    Print(" novo preço de venda. :", PrecoDeVenda);
                    Print(" novo Buysubsicul. :", Buysubsicul);
                    Print("novo Buytake. :", Buytake);
                    Print("novo Buystop. :", Buystop);
                    Print(" novo Sellsubsicul. :", Sellsubsicul);
                    Print(" novo selltake : ", Selltake);
                    Print("novo Sellestop. :", Sellestop);
                }
            }
        }
    }

    if(!control_de_compra2)
    {
        if(totalPositions == 0)
        {

```

```

        if(ativar_ou_desativar_compra && condicao2)
        {
            BuyMarkt();
            control_de_compra2 = true;
        }
    }
}

```

```

PrecoDeVenda = PrecoDeCompra - pontosf ;
Buytake = PrecoDeCompra + (pontosf * Nives);
Buystop = PrecoDeCompra - pontosf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeCompra;
Selltake = PrecoDeVenda - (pontosf * Nives);
Sellestop = PrecoDeVenda + pontosf;
Sellsubsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeVenda;

```

```

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);
Print("nível de stop loss buy :", nivelstoplos_buy);

```

```

if(Costura)
{
    controlsell = false ;
}
controlbuymdf = false;
segundo_control_de_takbuy = false;
segundo_control_de_taksell = false;
cont = false;
controlbuy = true;
}

```

```

}

```

```

}

```

```

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

```

```

if(surfada)

```

```

{
    if(!controlbuymdf)
    {
        if(closeprice > Buysubsicul)
        {
            if(totalPositions == 1)
            {

```

```

                if(bingala_de_alta > espasao)
                {
                    for(int i = 1 ; i <= canais ; i++)
                    {
                        PrecoDeVenda = PrecoDeCompra;
                        PrecoDeCompra = Buytake;
                        Buytake = PrecoDeCompra + (pontosf * Nives);
                        Buystop = PrecoDeCompra - pontosf;
                        Buysubsicul = PrecoDeCompra + divisao;
                        Buymodif = PrecoDeVenda;
                        Selltake = PrecoDeVenda - (pontosf * Nives);
                        Sellestop = PrecoDeVenda + pontosf;
                        Sellsubsicul = PrecoDeVenda - divisao;
                        SellModif = PrecoDeVenda;

```

```

                        Print("novo preço de compra. :", PrecoDeCompra);
                        Print(" novo preço de venda. :", PrecoDeVenda);
                        Print(" novo Buysubsicul. :", Buysubsicul);
                        Print(" multiplicado ", i);
                        Print("novo Buytake. :", Buytake);
                        Print("novo Buystop. :", Buystop);

```



```

        Print(" novo Sellsubsicul. :", Sellsubsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);
    }

}

PrecoDeVenda = PrecoDeCompra;
PrecoDeCompra = Buytake;
Buytake = PrecoDeCompra + (pontofsf * Nives);
Buystop = PrecoDeCompra - pontofsf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeVenda;
Selltake = PrecoDeVenda - (pontofsf * Nives);
Sellestop = PrecoDeVenda + pontofsf;
Sellsubsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeVenda;

ModifyBuyOrder();

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print(" modificado de compra ");
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{
    control_de_venda = false;
    control_de_compra = false;
}
controlbuy = false;
controlsell = false;

controlbuymdf = true;
}
else
{

    if(bingala_de_alta > espasao)
    {
        for(int i = 1 ; i <= canais ; i++)
        {
            PrecoDeVenda = PrecoDeCompra;
            PrecoDeCompra = Buytake;
            Buytake = PrecoDeCompra + (pontofsf * Nives);
            Buystop = PrecoDeCompra - pontofsf;
            Buysubsicul = PrecoDeCompra + divisao;
            Buymodif = PrecoDeVenda;
            Selltake = PrecoDeVenda - (pontofsf * Nives);
            Sellestop = PrecoDeVenda + pontofsf;
            Sellsubsicul = PrecoDeVenda - divisao;
            SellModif = PrecoDeVenda;

            Print("novo preço de compra. :", PrecoDeCompra);
            Print(" novo preço de venda. :", PrecoDeVenda);
            Print(" novo Buysubsicul. :", Buysubsicul);
            Print("novo Buytake. :", Buytake);
            Print("novo Buystop. :", Buystop);
            Print(" novo Sellsubsicul. :", Sellsubsicul);
            Print(" novo selltake : ", Selltake);
            Print("novo Sellestop. :", Sellestop);
        }

    }

    // ModifyBuyOrder(); // isso nao faz sentido
    PrecoDeVenda = PrecoDeCompra;
    PrecoDeCompra = Buytake;
    Buytake = PrecoDeCompra + (pontofsf * Nives);
    Buystop = PrecoDeCompra - pontofsf;
    Buysubsicul = PrecoDeCompra + divisao;
    Buymodif = PrecoDeVenda;
    Selltake = PrecoDeVenda - (pontofsf * Nives);
    Sellestop = PrecoDeVenda + pontofsf;

```

```

        Sellsubsicul = PrecoDeVenda - divisao;
        SellModif = PrecoDeVenda;

        Print("novo preço de compra. :", PrecoDeCompra);
        Print(" novo preço de venda. :", PrecoDeVenda);
        Print(" novo Buysubsicul. :", Buysubsicul);
        Print("novo Buytake. :", Buytake);
        Print("novo Buystop. :", Buystop);
        Print(" novo Sellsubsicul. :", Sellsubsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);
        if(Costura)
        {
            control_de_venda = false;
            control_de_compra = false;
        }
        controlbuy = false;
        controlsell = false;
        controlbuymdf = true;
    }
}
else
{
    if(!controlbuymdf)
    {
        if(closeprice > Buysubsicul)
        {

            if(bingala_de_alta > espassao)
            {
                for(int i = 1 ; i <= canais ; i++)
                {
                    PrecoDeVenda = PrecoDeCompra;
                    PrecoDeCompra = Buytake;
                    Buytake = PrecoDeCompra + (pontof * Nives);
                    Buystop = PrecoDeCompra - pontof;
                    Buysubsicul = PrecoDeCompra + divisao;
                    Buymodif = PrecoDeVenda;
                    Selltake = PrecoDeVenda - (pontof * Nives);
                    Sellestop = PrecoDeVenda + pontof;
                    Sellsubsicul = PrecoDeVenda - divisao;
                    SellModif = PrecoDeVenda;

                    Print("novo preço de compra. :", PrecoDeCompra);
                    Print(" novo preço de venda. :", PrecoDeVenda);
                    Print(" novo Buysubsicul. :", Buysubsicul);
                    Print("novo Buytake. :", Buytake);
                    Print("novo Buystop. :", Buystop);
                    Print(" novo Sellsubsicul. :", Sellsubsicul);
                    Print(" novo selltake : ", Selltake);
                    Print("novo Sellestop. :", Sellestop);
                }

            }
            ModifyBuyOrder(); // isso nao faz sentido
            PrecoDeVenda = PrecoDeCompra;
            PrecoDeCompra = Buytake;
            Buytake = PrecoDeCompra + (pontof * Nives);
            Buystop = PrecoDeCompra - pontof;
            Buysubsicul = PrecoDeCompra + divisao;
            Buymodif = PrecoDeCompra;
            Selltake = PrecoDeVenda - (pontof * Nives);
            Sellestop = PrecoDeVenda + pontof;
            Sellsubsicul = PrecoDeVenda - divisao;
            SellModif = PrecoDeVenda;

            Print("novo preço de compra. :", PrecoDeCompra);
            Print(" novo preço de venda. :", PrecoDeVenda);
            Print(" novo Buysubsicul. :", Buysubsicul);
            Print("novo Buytake. :", Buytake);
            Print("novo Buystop. :", Buystop);
            Print(" novo Sellsubsicul. :", Sellsubsicul);
            Print(" novo selltake : ", Selltake);
            Print("novo Sellestop. :", Sellestop);

            if(Costura)
            {

```

```

        control_de_compra2 = false;
        control_de_venda2 = false;
    }
    controlsell = false;
    controlbuy = false;
    controlbuymdf = true;
}
}
}

//+-----+
//|
//+-----+
if(surfada)
{
    if(! controlsellmdf)
    {
        if(closeprice < Sellsubsicul)
        {
            if(totalPositions == 1)
            {

                if(bingala_de_baixa > espasao)
                {
                    for(int i = 1 ; i <= canaisv ; i++)
                    {
                        PrecoDeCompra = PrecoDeVenda;
                        PrecoDeVenda = Selltake;
                        Buytake = PrecoDeCompra + (pontof * Nives);
                        Buystop = PrecoDeCompra - pontof;
                        Buysubsicul = PrecoDeCompra + divisao;
                        Buymodif = PrecoDeVenda;
                        Selltake = PrecoDeVenda - (pontof * Nives);
                        Sellestop = PrecoDeVenda + pontof;
                        Sellsubsicul = PrecoDeVenda - divisao;
                        SellModif = PrecoDeCompra;

                        Print("novo preço de compra. :", PrecoDeCompra);
                        Print(" novo preço de venda. :", PrecoDeVenda);
                        Print(" novo Buysubsicul. :", Buysubsicul);
                        Print("preços multiplicados do subsicol");
                        Print("novo Buytake. :", Buytake);
                        Print("novo Buystop. :", Buystop);
                        Print(" novo Sellsubsicul. :", Sellsubsicul);
                        Print(" novo selltake : ", Selltake);
                        Print("novo Sellestop. :", Sellestop);

                    }

                }

                PrecoDeCompra = PrecoDeVenda;
                PrecoDeVenda = Selltake;
                Buytake = PrecoDeCompra + (pontof * Nives);
                Buystop = PrecoDeCompra - pontof;
                Buysubsicul = PrecoDeCompra + divisao;
                Buymodif = PrecoDeVenda;
                Selltake = PrecoDeVenda - (pontof * Nives);
                Sellestop = PrecoDeVenda + pontof;
                Sellsubsicul = PrecoDeVenda - divisao;
                SellModif = PrecoDeCompra;

                ModifySellOrder();

                Print("novo preço de compra. :", PrecoDeCompra);
                Print(" novo preço de venda. :", PrecoDeVenda);
                Print(" novo Buysubsicul. :", Buysubsicul);
                Print("preços multiplicados do subsicol2222");
                Print("novo Buytake. :", Buytake);
                Print("novo Buystop. :", Buystop);
                Print(" novo Sellsubsicul. :", Sellsubsicul);
                Print(" novo selltake : ", Selltake);
                Print("novo Sellestop. :", Sellestop);

            if(Costura)
            {
                control_de_compra = false;
                control_de_venda = false;
            }
            controlbuy = false;
            controlsell = false;

            controlsellmdf = true;

```

```

    }
else
{

    if(bingala_de_baixa > espasao)
    {
        for(int i = 1 ; i <= canaisv ; i++)
        {
            PrecoDeCompra = PrecoDeVenda;
            PrecoDeVenda = Selltake;
            Buytake = PrecoDeCompra + (pontof * Nives);
            Buystop = PrecoDeCompra - pontof;
            Buysubsicul = PrecoDeCompra + divisao;
            Buymodif = PrecoDeVenda;
            Selltake = PrecoDeVenda - (pontof * Nives);
            Sellestop = PrecoDeVenda + pontof;
            Sellsubsicul = PrecoDeVenda - divisao;
            SellModif = PrecoDeCompra;

            Print("novo preço de compra. :", PrecoDeCompra);
            Print(" novo preço de venda. :", PrecoDeVenda);
            Print(" novo Buysubsicul. :", Buysubsicul);
            Print("novo Buytake. :", Buytake);
            Print("novo Buystop. :", Buystop);
            Print(" novo Sellsubsicul. :", Sellsubsicul);
            Print(" novo selltake : ", Selltake);
            Print("novo Sellestop. :", Sellestop);

        }

    }

    // ModifySellOrder(); // isso nao faz sentido
    PrecoDeCompra = PrecoDeVenda;
    PrecoDeVenda = Selltake;
    Buytake = PrecoDeCompra + (pontof * Nives);
    Buystop = PrecoDeCompra - pontof;
    Buysubsicul = PrecoDeCompra + divisao;
    Buymodif = PrecoDeVenda;
    Selltake = PrecoDeVenda - (pontof * Nives);
    Sellestop = PrecoDeVenda + pontof;
    Sellsubsicul = PrecoDeVenda - divisao;
    SellModif = PrecoDeCompra;

    Print("novo preço de compra. :", PrecoDeCompra);
    Print(" novo preço de venda. :", PrecoDeVenda);
    Print(" novo Buysubsicul. :", Buysubsicul);
    Print("novo Buytake. :", Buytake);
    Print("novo Buystop. :", Buystop);
    Print(" novo Sellsubsicul. :", Sellsubsicul);
    Print(" novo selltake : ", Selltake);
    Print("novo Sellestop. :", Sellestop);
    if(Costura)
    {
        control_de_compra = false;
        control_de_venda = false;

    }
    controlbuy = false;
    controlsell = false;
    controlsellmdf = true;
}

}

}
else
{
    if(!controlsellmdf)
    {
        if(closeprice < Sellsubsicul)
        {

            if(bingala_de_baixa > espasao)
            {
                for(int i = 1 ; i <= canaisv ; i++)
                {
                    PrecoDeCompra = PrecoDeVenda;
                    PrecoDeVenda = Selltake;
                    Buytake = PrecoDeCompra + (pontof * Nives);

```

```

        Buystop = PrecoDeCompra - pontosf;
        Buysubsicul = PrecoDeCompra + divisao;
        Buymodif = PrecoDeVenda;
        Selltake = PrecoDeVenda - (pontosf * Nives);
        Sellestop = PrecoDeVenda + pontosf;
        Sellsubsicul = PrecoDeVenda - divisao;
        SellModif = PrecoDeCompra;

        Print("novo preço de compra. :", PrecoDeCompra);
        Print(" novo preço de venda. :", PrecoDeVenda);
        Print(" novo Buysubsicul. :", Buysubsicul);
        Print("novo Buytake. :", Buytake);
        Print("novo Buystop. :", Buystop);
        Print(" novo Sellsubsicul. :", Sellsubsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);

    }

}

ModifySellOrder();
PrecoDeCompra = PrecoDeVenda;
PrecoDeVenda = Selltake;
Buytake = PrecoDeCompra + (pontosf * Nives);
Buystop = PrecoDeCompra - pontosf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeCompra;
Selltake = PrecoDeVenda - (pontosf * Nives);
Sellestop = PrecoDeVenda + pontosf;
Sellsubsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeCompra;

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{
    control_de_venda2 = false;
    control_de_compra2 = false;
}
controlbuy = false;
controlsell = false;
controlsellmdf = true;
}
}
}

```

```

//+-----+
//|
//+-----+
bool condicao = (condicao_De_rompimento_v) ?
                (precodiferentev < dedo) : true ;

//+-----+
//|
//+-----+
if(surfada)
{
    if(closeprice < PrecoDeVenda)
    {
        if(!controlsell)
        {
            if(totalPositions == 1)
            {

                if(bingala_de_baixa > espasao)
                {
                    for(int i = 1 ; i <= canaisv ; i++)
                    {
                        PrecoDeCompra = PrecoDeVenda;
                        PrecoDeVenda = Selltake;
                        Buytake = PrecoDeCompra + (pontosf * Nives);
                        Buystop = PrecoDeCompra - pontosf;
                        Buysubsicul = PrecoDeCompra + divisao;
                        Buymodif = PrecoDeVenda;
                        Selltake = PrecoDeVenda - (pontosf * Nives);
                    }
                }
            }
        }
    }
}

```

```

        Sellestop = PrecoDeVenda + pontosf;
        Sellsbsicul = PrecoDeVenda - divisao;
        SellModif = PrecoDeCompra;

        Print("novo preço de compra. :", PrecoDeCompra);
        Print(" novo preço de venda. :", PrecoDeVenda);
        Print(" novo Buysbsicul. :", Buysbsicul);
        Print(" preços multiplicados feixamento de venda");
        Print("novo Buytake. :", Buytake);
        Print("novo Buystop. :", Buystop);
        Print(" novo Sellsbsicul. :", Sellsbsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);
    }
}

PrecoDeVenda = PrecoDeVenda ;
PrecoDeCompra = PrecoDeVenda + pontosf;
Buytake = PrecoDeCompra + (pontosf * Nives);
Buystop = PrecoDeCompra - pontosf;
Buysbsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeVenda;
Selltake = PrecoDeVenda - (pontosf * Nives);
Sellestop = PrecoDeVenda + pontosf;
Sellsbsicul = PrecoDeVenda - divisao;
SellModif = PrecoDeCompra;

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysbsicul. :", Buysbsicul);
Print("preços maior que feixamento de venda");
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsbsicul. :", Sellsbsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{
    control_de_compra = false;
    control_de_venda = false;

}
controlbuy = false;
controlsellmdf = false;
segundo_control_de_takbuy = false;
segundo_control_de_taksell = false;
cont1 = false ;

controlsell = true;
}
else
{

    if(!control_de_compra)
    {

        if(bingala_de_baixa > espasao)
        {
            for(int i = 1 ; i <= canaisv ; i++)
            {
                PrecoDeCompra = PrecoDeVenda;
                PrecoDeVenda = Selltake;
                Buytake = PrecoDeCompra + (pontosf * Nives);
                Buystop = PrecoDeCompra - pontosf;
                Buysbsicul = PrecoDeCompra + divisao;
                Buymodif = PrecoDeVenda;
                Selltake = PrecoDeVenda - (pontosf * Nives);
                Sellestop = PrecoDeVenda + pontosf;
                Sellsbsicul = PrecoDeVenda - divisao;
                SellModif = PrecoDeCompra;

                Print("novo preço de compra. :", PrecoDeCompra);
                Print(" novo preço de venda. :", PrecoDeVenda);
                Print(" novo Buysbsicul. :", Buysbsicul);
                Print("novo Buytake. :", Buytake);
                Print("novo Buystop. :", Buystop);
                Print(" novo Sellsbsicul. :", Sellsbsicul);
                Print(" novo selltake : ", Selltake);
                Print("novo Sellestop. :", Sellestop);

            }
        }
    }
    if(ativar_ou_desativar_venda && condicao)

```

```

        {
            SellMarkt();
        }
        PrecoDeVenda = PrecoDeVenda ;
        PrecoDeCompra = PrecoDeVenda + pontosf;
        Buytake = PrecoDeCompra + (pontosf * Nives);
        Buystop = PrecoDeCompra - pontosf;
        Buysubsicul = PrecoDeCompra + divisao;
        Buymodif = PrecoDeVenda;
        Selltake = PrecoDeVenda - (pontosf * Nives);
        Sellestop = PrecoDeVenda + pontosf;
        Sellsubsicul = PrecoDeVenda - divisao;
        SellModif = PrecoDeCompra;

        Print("novo preço de compra. :", PrecoDeCompra);
        Print(" novo preço de venda. :", PrecoDeVenda);
        Print(" novo Buysubsicul. :", Buysubsicul);
        Print("novo Buytake. :", Buytake);
        Print("novo Buystop. :", Buystop);
        Print(" novo Sellsubsicul. :", Sellsubsicul);
        Print(" novo selltake : ", Selltake);
        Print("novo Sellestop. :", Sellestop);

        if(Costura)
        {

        }
        controlbuy = false;
        controlsellmdf = false;
        segundo_control_de_takbuy = false;
        segundo_control_de_taksell = false;
        cont1 = false ;
        controlsell = false;
        control_de_venda = true;

    }
}
}
}
else
{
    if(!controlsell)
    {
        if(closeprice < PrecoDeVenda)
        {
            if(bingala_de_baixa > espasao)
            {
                for(int i = 1 ; i <= canaisv ; i++)
                {
                    PrecoDeCompra = PrecoDeVenda;
                    PrecoDeVenda = Selltake;
                    Buytake = PrecoDeCompra + (pontosf * Nives);
                    Buystop = PrecoDeCompra - pontosf;
                    Buysubsicul = PrecoDeCompra + divisao;
                    Buymodif = PrecoDeVenda;
                    Selltake = PrecoDeVenda - (pontosf * Nives);
                    Sellestop = PrecoDeVenda + pontosf;
                    Sellsubsicul = PrecoDeVenda - divisao;
                    SellModif = PrecoDeCompra;

                    Print("novo preço de compra. :", PrecoDeCompra);
                    Print(" novo preço de venda. :", PrecoDeVenda);
                    Print(" novo Buysubsicul. :", Buysubsicul);
                    Print("novo Buytake. :", Buytake);
                    Print("novo Buystop. :", Buystop);
                    Print(" novo Sellsubsicul. :", Sellsubsicul);
                    Print(" novo selltake : ", Selltake);
                    Print("novo Sellestop. :", Sellestop);

                }
            }
        }
        if(! control_de_venda2)
        {
            if(totalPositions == 0)
            {
                if(ativar_ou_desativar_venda && condicao)
                {
                    SellMarkt();
                }
                control_de_venda2 = true;
            }
        }
    }
}

```

```

    }
}

PrecoDeCompra = PrecoDeVenda + pontosf;
Buytake = PrecoDeCompra + (pontosf * Nives);
Buystop = PrecoDeCompra - pontosf;
Buysubsicul = PrecoDeCompra + divisao;
Buymodif = PrecoDeCompra;
Selltake = PrecoDeVenda - (pontosf * Nives);
Sellestop = PrecoDeVenda + pontosf;
Sellsubsicul = PrecoDeVenda - divisao;
Sellmodif = PrecoDeVenda;

Print("novo preço de compra. :", PrecoDeCompra);
Print(" novo preço de venda. :", PrecoDeVenda);
Print(" novo Buysubsicul. :", Buysubsicul);
Print("novo Buytake. :", Buytake);
Print("novo Buystop. :", Buystop);
Print(" novo Sellsubsicul. :", Sellsubsicul);
Print(" novo selltake : ", Selltake);
Print("novo Sellestop. :", Sellestop);

if(Costura)
{
    controlbuy = false;
}

controlsellmdf = false;
segundo_control_de_takbuy = false;
segundo_control_de_taksell = false;
cont1 = false ;
controlsell = true;
}
}
}

```

```

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

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

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

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

bool BuyMarkt()
{
    MqlTradeRequest request = { };
    MqlTradeResult result = { };
    request.action = TRADE_ACTION_DEAL;
    request.volume = lot;
    request.symbol = Symbol();
    request.type = ORDER_TYPE_BUY;

    if(surfada)
    {
        request.sl = Buystop - fora ;
    }
    else
    {
        request.sl = Buystop - fora ;
        request.tp = Buytake - fora ;
    }

    takbuy = request.tp;
    nivelstoplos_buy = request.sl;
    if(OrderSend(request, result))
    {

```



```

        BilheteDeCompra = result.order;
        preco_de_abertura_de_compra = result.price;
        Print("ordem de compra enviada -Bilhete : ", BilheteDeCompra);
        return true;
    }
else
{
    Print("Erro ao enviar a ordem de compra -erro :", GetLastError());
    return false;
}
}
//+-----+
bool SellMarkt()
{
    MqlTradeRequest request = { };
    MqlTradeResult result = { };
    request.action = TRADE_ACTION_DEAL;
    request.volume = lot;
    request.symbol = Symbol();
    request.type = ORDER_TYPE_SELL;

    if(surfada)
    {
        request.sl = Sellestop + fora ;
    }
else
{
    request.sl = Sellestop + fora ;
    request.tp = Selltake + fora ;
}

    taksell = request.tp;
    nivelstoplos_sell = request.sl;
    if(OrderSend(request, result))
    {
        BilheteDeVenda = result.order;
        preco_de_abertura_de_venda = result.price;
        Print("ordem de venda enviada -Bilhete : ", BilheteDeVenda);
        return true;
    }
else
{
    Print("Erro ao enviar a ordem de venda -erro :", GetLastError());
    return false;
}
    return true;
}
//+-----+
bool ModifySellOrder()
{
    if(PositionSelectByTicket(BilheteDeVenda))
    {
        {
            MqlTradeRequest request = { };
            MqlTradeResult result = { };
            request.action = TRADE_ACTION_SLTP;
            request.symbol = Symbol();
            if(surfada)
            {
                request.sl = SellModif;
            }
else
{
            request.tp = Selltake + fora;
            if(Modificar_SL_Para_0x0)
            {
                request.sl = preco_de_abertura_de_venda ;
            }
else
            {
                request.sl = SellModif ;
            }
        }
        request.position = BilheteDeVenda;
        nivelzerosell = request.sl;
        taksell = request.tp;
        if(OrderSend(request, result))
        {
            Print("Ordem de venda modificada - Bilhete: ", BilheteDeVenda, " Novo SL: ", SellModif);
            return true;
        }
else
{

```

```

        Print("Erro ao modificar a ordem de venda - Bilhete: ", BilheteDeVenda);
        return false;
    }
}
}
return true;
}
//+-----+
//|
//+-----+
bool ModifyBuyOrder()
{
    if(PositionSelectByTicket(BilheteDeCompra))
    {
        {
            MqlTradeRequest request = { };
            MqlTradeResult result = { };
            request.action = TRADE_ACTION_SLTP;
            request.symbol = Symbol();
            if(surfada)
            {
                request.sl = Buymodif;
            }
            else
            {
                request.tp = Buytake - fora;
                if(Modificar_SL_Para_0x0)
                {
                    request.sl = preco_de_abertura_de_compra;
                }
                else
                {
                    request.sl = Buymodif;
                }
            }

            request.position = BilheteDeCompra;
            nivelzerobuy = request.sl;
            takbuy = request.tp;
            if(OrderSend(request, result))
            {
                Print("Ordem de compra modificada - Bilhete: ", BilheteDeCompra, " Novo SL: ", Buymodif);
                return true;
            }
            else
            {
                Print("Erro ao modificar a ordem de compra - Bilhete: ", BilheteDeCompra);
            }
            return false;
        }
    }
    return true;
}
//+-----+
void criarlinhassell1(double valor)
{
    string subsicul = "linha3_" + DoubleToString(valor, 4);
    bool unica3 = true;
    for(int i = 0; i < ArraySize(precosArray); i++)
    {
        if(precosArray[i] == valor)
        {
            unica3 = false ;
            break;
        }
    }
    if(unica3)
    {
        int currentSize = ArraySize(precosArray);
        ArrayResize(precosArray, currentSize + 1);
        precosArray[currentSize] = valor;
        string subsicul = "linha3_" + DoubleToString(valor, 4);
        ObjectCreate(0, subsicul, OBJ_HLINE, 0, 0, valor);
        ObjectSetInteger(0, subsicul, OBJPROP_STYLE, STYLE_SOLID);
        ObjectSetInteger(0, subsicul, OBJPROP_COLOR, clrBlue);
    }
}
//+-----+
void criarlinhassell2(double valor)
{
    string precosell = "linha2_" + DoubleToString(valor, 4);
    bool unica2 = true;
    for(int i = 0; i < ArraySize(precosArray); i++)
    {

```

```

        if(precosArray[i] == valor)
        {
            unica2 = false ;
            break;
        }
    }
    if(unica2)
    {
        int currentSize = ArraySize(precosArray);
        ArrayResize(precosArray, currentSize + 1);
        precosArray[currentSize] = valor;
        string precosell = "linha2_" + DoubleToString(valor, 4);
        ObjectCreate(0, precosell, OBJ_HLINE, 0, 0, valor);
        ObjectSetInteger(0, precosell, OBJPROP_STYLE, STYLE_SOLID);
        ObjectSetInteger(0, precosell, OBJPROP_COLOR, clrRed);
    }
}
//+-----+
//|                // criar linha de compra  vermelho
//+-----+
void criarlinhaBuy1(double valor)
{
    string subsiculbuy = "linha1_" + DoubleToString(valor, 4);
    bool unica1 = true;
    for(int i = 0; i < ArraySize(precosArray); i++)
    {
        if(precosArray[i] == valor)
        {
            unica1 = false ;
            break;
        }
    }
    if(unica1)
    {
        int currentSize = ArraySize(precosArray);
        ArrayResize(precosArray, currentSize + 1);
        precosArray[currentSize] = valor;
        string subsiculbuy = "linha1_" + DoubleToString(valor, 4);
        ObjectCreate(0, subsiculbuy, OBJ_HLINE, 0, 0, valor);
        ObjectSetInteger(0, subsiculbuy, OBJPROP_STYLE, STYLE_SOLID);
        ObjectSetInteger(0, subsiculbuy, OBJPROP_COLOR, clrBlue);
    }
}
//+-----+
void criarlinhaBuy2(double valor)
{
    string precobuy = "linha_" + DoubleToString(valor, 4);
    bool unica = true;
    for(int i = 0; i < ArraySize(precosArray); i++)
    {
        if(precosArray[i] == valor)
        {
            unica = false ;
            break;
        }
    }
    if(unica)
    {
        int currentSize = ArraySize(precosArray);
        ArrayResize(precosArray, currentSize + 1);
        precosArray[currentSize] = valor;
        string precobuy = "linha_" + DoubleToString(valor, 4);
        ObjectCreate(0, precobuy, OBJ_HLINE, 0, 0, valor);
        ObjectSetInteger(0, precobuy, OBJPROP_STYLE, STYLE_SOLID);
        ObjectSetInteger(0, precobuy, OBJPROP_COLOR, clrRed);
    }
}

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

// +-----+
//+-----+
//+-----+
//+-----+
//+-----+

```