```mq4
//+------------------------------------------------------------------+
//| Trecho para codigo_Martingale_ordem_compra_venda.mq4 |
//| Copyright 2021, Marcelo Marques |
//| https:// |
//+------------------------------------------------------------------+
#property copyright "Copyright 2021, Marcelo Marques"
#property link "https://"
#property version "1.00"
#property strict
#define EXPERT_MAGIC 123456
enum Modos_xyz
{
xyz = MODE_ALTO
xyz = MODE_MEDIO
xyz = MODE_BAIXO
};
input Modos_XYZ linha1 = acima, //Linha para Compra
linha2 = abaixo; //Linha para Venda


input int Slippage = 0,
StopLoss = 100,
TakeProfit = 100;
input double lote = 0.01;
//+------------------------------------------------------------------+
//| Expert initialization function |
//+------------------------------------------------------------------+
int OnInit()
{
//---
//---
return(INIT_SUCCEEDED);
```

```
}
//+------------------------------------------------------------+
//| Expert deinitialization function |
//+------------------------------------------------------------+
void OnDeinit(const int reason)
{
//---
}
//+------------------------------------------------------------+
//| Expert tick function |
//+------------------------------------------------------------+
bool abrir_uma_vez = true;
int BUY_SELL;
double lote_modificado;


void OnTick()
{
//Sempre que estiver dúvida no indicador, clique nele e aperte F1 e pegue o exemplo//
```

```
//AQUI INDICADOR A SER USADO.
double Xyz_cima = iXyz(_Symbol,0,9,26,52,linha1,0),
Xyz_baixo = iXyz(_Symbol,0,9,26,52,linha2,0);
if(Bid >= INDICADOR A SER USADO(_ACIMA) && OrdersChart(-1) == 0 &&


abrir_uma_vez == true)


{
lote_modificado = lote*2;
Open_Order(_Symbol,OP_SELL,lote,Slippage,StopLoss,TakeProfit,NULL,
```

```
EXPERT_MAGIC);

Print("A Ordem foi aberta");

abrir_uma_vez = false;

BUY_SELL = OP_SELL;

}

else

if(Bid <= INDICADOR A SER USADO(_BAIXO) && OrdersChart(-1) == 0 &&


abrir_uma_vez == true)


{

lote_modificado = lote*2;

Open_Order(_Symbol,OP_BUY,lote,Slippage,StopLoss,TakeProfit,


NULL,EXPERT_MAGIC);


printf("A ordem foi aberta");

abrir_uma_vez = false;

BUY_SELL = OP_BUY;

}

}


//+----------------------------------------------------------------------+
//|------------------------- Martingale SELL -----------------------------|
//+----------------------------------------------------------------------+
if(OrdersChart(-1) == 0 && BUY_SELL == OP_SELL && Verificar_Fechamento(

StopLoss_s) == true )


{
```

```
Open_Order(_Symbol,OP_BUY,lote_modificado,Slippage,StopLoss,

TakeProfit,NULL,EXPERT_MAGIC);

lote_modificado = lote_modificado*2;
BUY_SELL = OP_BUY;
}
else
if(OrdersChart(-1) == 0 && BUY_SELL == OP_SELL &&

Verificar_Fechamento(TakeProfit_s) == true)

{
Open_Order(_Symbol,OP_SELL,lote,Slippage,StopLoss,TakeProfit,

NULL,EXPERT_MAGIC);

lote_modificado = lote*2;
}

//+------------------------------------------------------------------+
//|---------------------------- Martingale BUY ----------------------|
//+------------------------------------------------------------------+
if(OrdersChart(-1) == 0 && BUY_SELL == OP_BUY && Verificar_Fechamento(

StopLoss_s) == true )

{
```

```
Open_Order(_Symbol,OP_SELL,lote_modificado,Slippage,StopLoss,
```

```
                TakeProfit,NULL,EXPERT_MAGIC);


                lote_modificado = lote_modificado*2;

                BUY_SELL = OP_SELL;

                }

                else

                if(OrdersChart(-1) == 0 && BUY_SELL == OP_BUY && Verificar_Fechamento

                (TakeProfit_s) == true)


                {

                Open_Order(_Symbol,OP_BUY,lote,Slippage,StopLoss,TakeProfit,


                NULL,EXPERT_MAGIC);


                lote_modificado = lote*2;

                }

                }


//+--------------------------------------------------------------------+
//|------------------------------ ABRIR ORDENS -----------------------|
//+--------------------------------------------------------------------+
//|----------------------//Só vai usar copiar elas para usar//--------------|
//+--------------------------------------------------------------------+
void Open_Order(string simbolo,


ENUM_ORDER_TYPE type_order,

double lot,

int slippage,

int sl,

int tk,

string comment,
```

```
        int magic_n
        )

        {
        int c,v;
        switch(type_order)
        {
        case OP_SELL:
        {
        if(tk==0 && sl==0)
        {
        v = OrderSend(simbolo,type_order,lot,Bid,slippage,0,0,

        comment,magic_n,0,0);
        }
        if(tk!=0 && sl==0)
        {
        v = OrderSend(simbolo,type_order,lot,Bid,slippage,0,Bid-tk*

        Point(),comment,magic_n,0,0);

        }
        if(tk==0 && sl!=0)
        {
        v = OrderSend(simbolo,type_order,lot,Bid,slippage,Bid+sl*

        Point(),0,comment,magic_n,0,0);

        }
        if(tk!=0 && sl!=0)
        {
```

```
v = OrderSend(simbolo,type_order,lot,Bid,slippage,Bid+sl*

Point(),Bid-tk*Point(),comment,magic_n,0,0);

}
```

```
}
break;
case OP_BUY:
{
if(tk==0 && sl==0)
{
v = OrderSend(simbolo,type_order,lot,Ask,slippage,0,0,

comment,magic_n,0,0);
}
if(tk!=0 && sl==0)
{
v = OrderSend(simbolo,type_order,lot,Ask,slippage,0,Bid+tk*

Point(),comment,magic_n,0,0);

}
if(tk==0 && sl!=0)
{
v = OrderSend(simbolo,type_order,lot,Ask,slippage,Bid-sl*

Point(),0,comment,magic_n,0,0);
```

```
        }
        if(tk!=0 && sl!=0)
        {
        v = OrderSend(simbolo,type_order,lot,Ask,slippage,Bid-sl*

        Point(),Bid+tk*Point(),comment,magic_n,0,0);

        }
        }
        break;
        }
        Sleep(500);
        if(!OrderSelect(OrdersChart(-1)-1,SELECT_BY_POS,MODE_TRADES))
        {
        PrintFormat("Não pôde ser aberta a Ordem: "+GetLastError());
        }
        else
        {
        int ticket_Order = OrderTicket();
        ArrayResize(tickets,ArraySize(tickets)+1);
        ArrayFill(tickets,ArraySize(tickets)-1,1,ticket_Order);
        PrintFormat("Qtd Array: "+ArraySize(tickets)+"New Ticket: "+tickets

        [ArraySize(tickets)-1]);
        }
        }
///+------------------------------------------------------------------+
//|-------------------------------- FECHAR ORDENS -----------------------|
//+------------------------------------------------------------------+
void Close_Orders(ENUM_ORDER_TYPE type)
{
```

```
for(int i = OrdersTotal()-1; i >=0; i--)

{

OrderSelect(i,SELECT_BY_POS,MODE_TRADES);

switch(type)

{

case OP_SELL:

{
```

```
if(OrderSymbol() == Symbol() && OrderType() == OP_SELL)

OrderClose(OrderTicket(),OrderLots(),Ask,Slippage,0);

}

break;

case OP_BUY:

{

if(OrderSymbol() == Symbol() && OrderType() == OP_BUY)

OrderClose(OrderTicket(),OrderLots(),Bid,Slippage,0);

}

break;

case -1:

{

if(OrderSymbol() == Symbol())

OrderClose(OrderTicket(),OrderLots(),Bid,Slippage,0);

}

break;

}

}

}

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

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

```
//+-----------------------------------------------------------------+
int tickets[];
enum Type_Closed
{
TakeProfit_s,
StopLoss_s
};


//+-----------------------------------------------------------------+
//|------------------ VERIFICA SE ELA FECHOU NO STOP OU NO TAKE --------------|
//+-----------------------------------------------------------------+
bool Verificar_Fechamento(Type_Closed tipo)
{
bool ocorreu=false;
for(int i = ArraySize(tickets)-1; i >= 0; i--)
{
int j = OrdersHistoryTotal()-1;
while(j >= 0)
{
OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
double fechamento = OrderClosePrice();
double profit = 0;
profit+=OrderProfit();
profit+=OrderSwap();
profit+=OrderCommission();
if(ArraySize(tickets) != 0)
{
```

```
if(tipo == TakeProfit_s && OrderSymbol() == Symbol() &&
```

```
tickets[i] == OrderTicket())

{
if(fechamento != 0 && profit >=0)
{
ocorreu = true;
Comment("Deu take\n",ArraySize(tickets));
ArrayFill(tickets,i,1,-1);
break;
}
else
ocorreu = false;
}
else
if(tipo == StopLoss_s && OrderSymbol() == Symbol() &&

tickets[i] == OrderTicket())
{
if(fechamento != 0 && profit <=0)
{
ocorreu = true;
Comment("Deu Loss\n",ArraySize(tickets));
ArrayFill(tickets,i,1,-1);
break;
}
else
ocorreu = false;
}

}
```

```
        j--;

        }

    }

    return(ocorreu);

}



//+------------------------------------------------------------------+
//+------------------------------------------------------------------|
//+------------------------------------------------------------------+
int OrdersChart(int type)

{

int qtd=0;

switch(type)

{

case -1:

{

for(int i= OrdersTotal()-1; i>=0; i--)

{

OrderSelect(i,SELECT_BY_POS,MODE_TRADES);

{

if(OrderSymbol()==Symbol())

qtd++;

}

}

}

break;

case OP_SELL:

{

for(int i= OrdersTotal()-1; i>=0; i--)

{
```

```
OrderSelect(i,SELECT_BY_POS,MODE_TRADES);
{
if(OrderSymbol()==Symbol() && OrderType() == OP_SELL)
qtd++;
}
}
}
break;
case OP_BUY:
{
for(int i= OrdersTotal()-1; i>=0; i--)
{
OrderSelect(i,SELECT_BY_POS,MODE_TRADES);
{
if(OrderSymbol()==Symbol() && OrderType() == OP_BUY)
qtd++;
}
}
}
break;
}
return(qtd);
}
//+--------------------------------------------------------------------+
```