

## Fibonacci.mql

```
1. #property link      "cleitoncsg@gmail.com"
2. #include "suporteResistencia.mq4"
3.
4. #define TAKE_PROFIT 500
5. #define STOP_LOSS 500
6. #define ALAVANCAGEM 0.25
7. #define FATOR_RETRACAO 0.38
8. #define MAX_CANDLES 34
9. #define SEXTA_FEIRA
10. #define ESTADO_VALIDO 0.01
11. double retracao_fibo;
12.
13. int ticket=0;
14. string nome = "CSG";
15.
16. int start(){
17.     bool venda;
18.
19.     if(NormalizeDouble(retracao_fibonacci()+ suporte(),4) == Bid
        && estadoMercado(MAX_CANDLES) > ESTADO_VALIDO){
20.         venda = true;
21.     }
22.
23.     if( ((venda == true && OrdersTotal() == 0)) ){
```

```

24.         RefreshRates();
25.         while (IsTradeContextBusy()) Sleep(5);
26.
27.         ticket= OrderSend(Symbol(),OP_SELL,ALAVANCAGEM,Bid,0,Bid
+ STOP_LOSS*Point,
28.         Bid - TAKE_PROFIT*Point,nome,AccountNumber(),0,Green);
29.     }
30.
31.         double ponto_positivo, ponto_negativo;
32.
33.         for(int j=0; j < OrdersHistoryTotal();j++){
34.             OrderSelect(j,SELECT_BY_POS,MODE_HISTORY);
35.
36.             if(OrderSymbol()!=Symbol()) continue;
37.             if(OrderMagicNumber() != AccountNumber())
continue;
38.             if(OrderProfit() > 0){
39.                 ponto_positivo++;
40.             }
41.             else{
42.                 ponto_negativo++;
43.             }
44.         }
45.
46.     Comment(

```

```

47.         "Margem da Conta = ", AccountMargin() , "\n",
48.         "Ordens em lucro = ", ponto_positivo , "\n",
49.         "Ordens em prejuizo = ", ponto_negativo , "\n",
50.         "Suporte = ", suporte() , "\n",
51.         "Resistencia = ", resistencia() , "\n",
52.         "Retracao Fibo = ", retracao_fibonacci() , "\n",
53.         ""
54.     );
55.
56.     return(0);
57. }
58.
59. double estadoMercado(int tempoCorrelacao){
60.     double soma_ordenadas = 0, soma_abscissas = 0;
61.     double numero_abscissa, numero_ordenada;
62.
63.     for(int c=0; c<tempoCorrelacao; c++){
64.         numero_abscissa = NormalizeDouble(Open[c],5);
65.         numero_ordenada =NormalizeDouble(Close[c],5);
66.         soma_abscissas = soma_abscissas +
            numero_abscissa;
67.         soma_ordenadas = soma_ordenadas +
            numero_ordenada;
68.     }
69.

```

```
70.         return ( soma_abcissas - soma_ordenadas);
71.     }
72.
73.     double retracao_fibonacci(){
74.         double retracao = ( resistencia() -
            suporte())*FATOR_RETRACAO;
75.
76.         return (retracao);
77.     }
```