

Jhonatan Willian dos Santos Silva 240025870

1. adicionando dois numeros

```
    INP      ;recebe o primeiro numero do usuario e reserva no acumulador
    STA num1 ;guarda no espaco de memoria num1
    INP      ;recebe o segundo numero que o usuario digitou e reserva no acumulador
    ADD num1 ;adiciona o que esta no acumulador com o que esta reservado no espaco de memoria num1 e reserva no acumulador
    OUT      ;printa o que esta no acumulador
    HLT      ;encerra o programa
```

```
num1 DAT      ;reserva/declara um espaco na memoria num1
```

2. maior entre dois numeros

```
    INP      ;recebe o primeiro numero do usuario e reserva no acumulador
    STA num1 ;guarda no espaco de memoria num1
    INP      ;pede o segundo numero ao usuario
    STA num2 ;coloca esse segundo numero que o usuario digitou na pos de memoria num2 e no acumulador
    SUB num1 ;subtrai do acumulador "num2" o primeiro numero que o usuario digitou
    BRP pos  ;se o resultado de num2 - num1 for positivo significa que o num2 é maior que num1...
              portando devemos carregar o num2 no acumulador e exibir na tela
    LDA num1 ;caso a subtracao de num2 - num1 nao tenha dado positivo significa que o num1...
              é maior ou os numeros sao iguais entao devemos exibi-lo, assim resgatamos na memoria o num1
    OUT      ;e printamos na tela o que foi resgatado que tambem esta no acumulador
    BRA exit ;apos printarmos que o num1 é o maior numero precisamos encerrar o programa por isso pulamos para o "exit"
pos  LDA num2 ;se o num2 - num1 for positivo vem parar aqui e é resgatado na memoria o num2 e armazenado no acumulador
    OUT      ;exibimos na tela o valor que esta no acumulador
exit HLT      ;encerra o programa
```

```
num1 DAT      ;reserva um espaco na memoria para o num1
```

```
num2 DAT      ;reserva um espaco na memoria para o num2
```

### 3. countdown

```
    INP      ;recebe um numero que o usuario digitar e guarda no acumulador
loop  OUT      ;printa na tela o que esta no acumulador
    STA count ;guarda o que esta no acumulador no espaco de memoria count para reservar o a contagem atual
    SUB one   ;subtrai 1 do acumulador pois estamos fazendo um contador ate o 0
    STA count ;agora o numero que sera armazenado no count sera menor que o de antes em 1
    BRP loop  ;se o acumulador estiver positivo significa que a contagem ainda nao terminou ...
            ;e precisaremos voltar para o loop ate que o numero seja diferente de 0, ...
            ;quando for nao sera preciso exibir mais nada e o o programa se encerrará
    HLT      ;encerra o programa

one   DAT 1    ;reserva um espaco na memoria com o valor 1
count DAT      ;reserva um espaco na memoria
```

### 4. multiplicar dois numeros

```
    INP      ;recebe o primeiro numero que o usuario digitou
    STA NUM1  ;guarda esse numero na posicao de memoria num1
    INP      ;recebe o segundo numero que o usuario digitou para ser multiplicado pelo primeiro
    STA NUM2  ;guarda esse segundo numero na posicao de memoria num2
LOOP  LDA TOTAL ;como nao conseguimos simplesmente multiplicar esses 2 numeros teremos que fazer uma soma sucessiva,...
            ;por isso declaramos um loop, e carregamos no acumulador o total dessa soma que inicialmente é 0
    ADD NUM1  ;adicionamos nesse total o primeiro numero que o usuario digitou
    STA TOTAL ;guardamos na posicao de memoria total o num1 que no caso é a primeira vez que sera somad
    LDA NUM2  ;agora carregamos no acumulador o segundo numero que o usuario digitou que é quantas vezes a soma ...
            ;sucessiva irá ocorrer
    SUB ONE   ;subtraímos 1 do numero2 pois ja fizemos a soma 1 vez
    STA NUM2  ;atualizamos agora o contador de somas no num2
    BRP LOOP  ;se o num2 for positivo significa que nao terminamos a soma sucessiva e teremos que repetir o processo
    LDA TOTAL ;se chegar aqui significa que o num2 é negativo ou seja acabamos de somar o mesmo ...
            ;numero varias vezes, porem ele somou uma vez a mais pois ele foi para o loop novamente...
            ;quando o num2 era 0, pois 0 é positivo, portanto devemos anular uma soma
    SUB NUM1  ;subtraímos do total o num1, ou seja tiramos a ultima adicao indesejada
    STA TOTAL ;atualizamos o total
    OUT      ;exibimos o total na tela
    HLT      ;encerramos o programa

NUM1  DAT      ;reserva um espaco na memoria para o num1 "numero esse que sera somado varias vezes"
NUM2  DAT      ;reserva um espaco na memoria para o num2 "numero esse que sera um contador"
ONE   DAT 1    ;reserva um espaco na memoria com o valor 1
TOTAL DAT 0    ;reserva um espaco na memoria para o total da multiplicacao
```

##### 5.numeros triangulares

```
loop    LDA number    ;carrega no acumulador o valor armazenado em number
        ADD counter   ;adicona no numero de bolinhas o valor do contador que inicialmente é 1 "o primeiro triangulo
        OUT           ;printa na tela o numero de bolinha do triangulo em questao, sera sempre valor inicial + (n + 1) ex: 1, 3 ,6, 10, 15
        STA number    ;armazena no espaco de memoria o numero de bolinhas do primeiro triangulo
        LDA counter   ;carrega no acumulador o contador auxiliar
        ADD one       ;adiciona 1 nele
        STA counter   ;armazena o valor atualizado do contador
        LDA ten       ;carrega o valor predefinido 10 pois iremos mostrar apenas a quantidade de bolinhas ate os 10 primeiros triangulos
        SUB counter   ;subtrai o valor do contador dele e verifica se é um numero positivo,caso for ainda traingulos a serem determinados
        BRP loop      ;pula para o loop novamente para determinar no numero de bolinhas do proximo triangulo
        HLT           ;caso a subtracao de cima de um numero negativo encerra o programa

counter DAT 1         ;reserva um espaco na memoria para o contador
number  DAT 0         ;reserva um espaco na memoria para o numero de bolinha do triangulo
one     DAT 1         ;reserva um espaco na moria com o valor 1
ten     DAT 10        ;reserva um espaco na memoria com o valor 10
```

# 6. fatorial de um numero

```

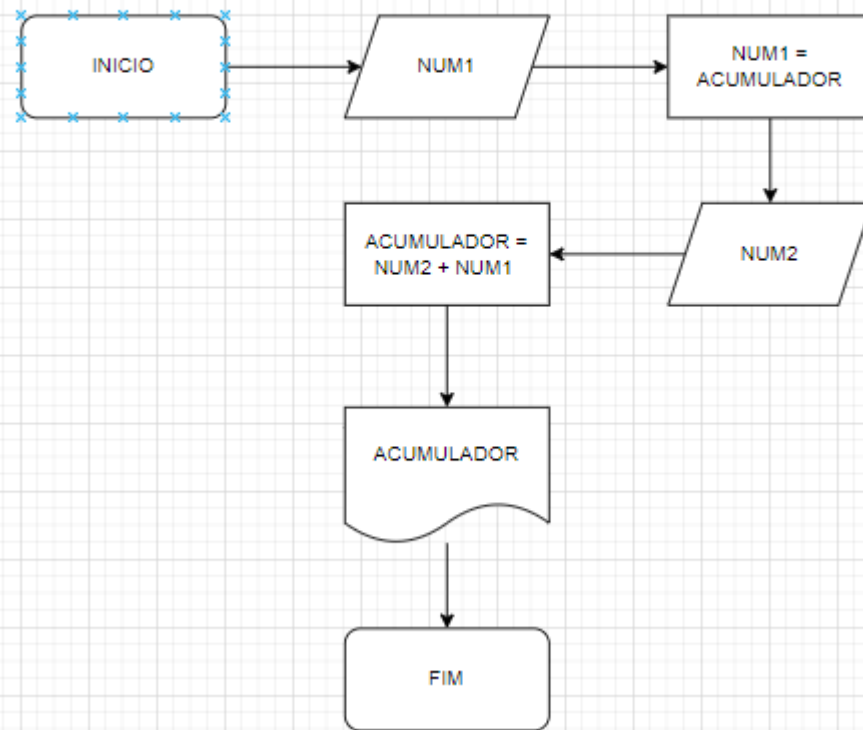
    INP                ;recebe um numero que o usuario digitou
    STA final          ;armazena esse numero na variavel "final" e carrega ele no acumulador, sera o valor total do nosso fatorial
    BRZ oneval         ;verifica se o numero é 0 e se for significa que o usuario quer o fatorial de 0 que no caso é 1,...
                        ;entao pulamos direto para exibir o resultado sem fazer nenhuma conta
    SUB one            ;se nao for 0 a gente subtrai um do acumulador e do da variavel "final"
    STA iteration      ;colocamos esse valor na variavel "iteration" que indica quantas iteracao ainda serao feitas
    STA counter        ;coloca no counter o mesmo valor
    LDA final          ;carrega o final
    STA num            ;armazena o valor do acumulador em num
mult  LDA iteration    ;carrega o valor atual da variavel iteration
    BRZ end            ;se iteracao esta com um significa que vai ser uma multiplicacao do resultado por 1
    SUB one            ;entao subtraimos 1 da itecao pois é irrelevante
    BRZ end            ;se a iteracao chegou em chego signifca que ja nao temos mais que multiplicar , entao exibimos o resultado final, caso nao
    LDA final          ;carregamos o final
    ADD num            ;e adicionamos num nele que é o valor inicial que digitamos
    STA final          ;atualizamos o valor final
    LDA counter        ;carregamos no acumulador a variavel counter
    SUB one            ;subtraimos um dele
    STA counter        ;atualizamos o seu valor
    SUB one            ;subtraimos um dele novamente
    BRZ next           ;se o acumulador for 0 pulamos para nextm, para mulktiplcar o valor final pelo proximo numero, se nao
    BRA mult           ;faca mais uma multiplicacao com o valor que ja estava antes

next  LDA final        ;carrega o valor final
    STA num            ;coloca na variavel num
    LDA iteration      ;carrega a iteracao
    SUB one            ;subtrai 1
    STA iteration      ;coloca o valor na iteracao
    STA counter        ;coloca o valor no counter
    SUB one            ;subtrai 1
    BRZ end            ;se o valor do acumulador for 0 acabar o contador estamos no ultimo caso
    BRA mult           ;se nao faca mais multiplicacao
end   LDA final        ;carrega o valor final
    OUT               ;printa na tela
    HLT               ;encerra o programa
oneval LDA one         ;carrega no acumulador o valor 1
    OUT               ;printa na tela esse valor
    HLT               ;encerra o programa

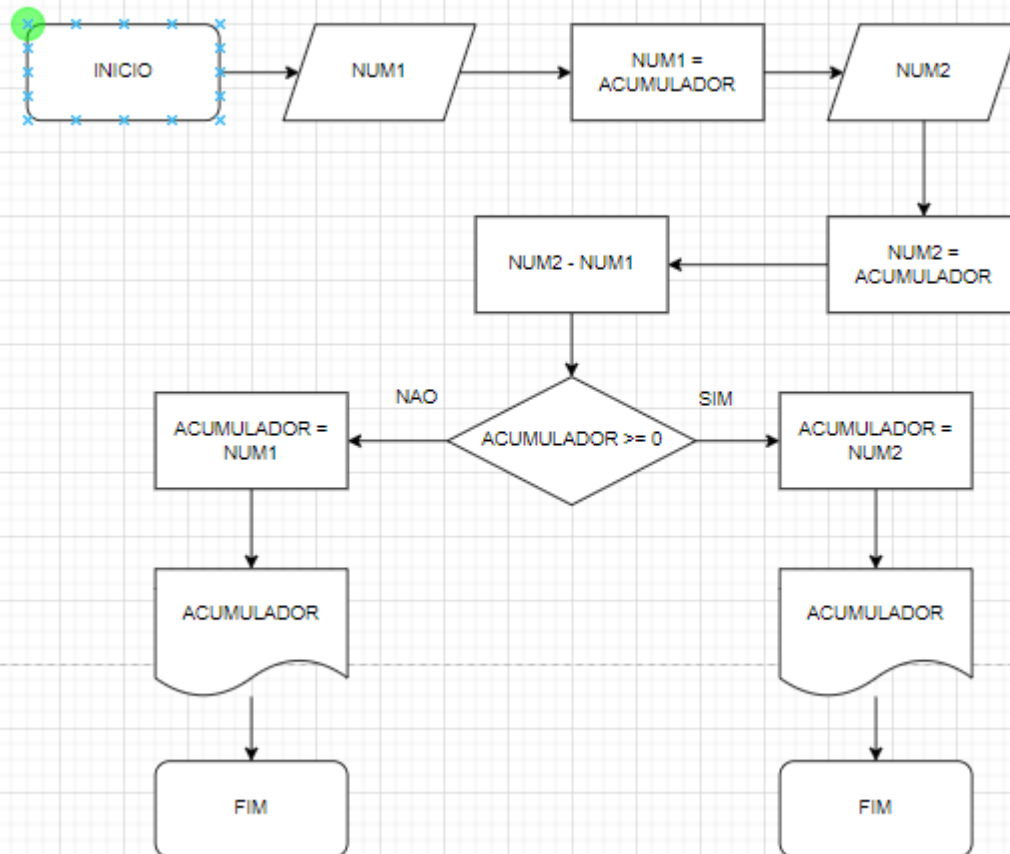
final  DAT 0           ;declara a variavel que vai ser o resultado
counter DAT 0          ;declara a variavel counter (para multiplicar pelo numero antecessor ao atual)
one    DAT 1           ;declara uma contante com o valor 1
iteration DAT 0         ;declara a variavel iteration quantas vezes vai fazer com o mesmo numero
num    DAT 0           ;decalra variavel auxiliar para somar com a final

```

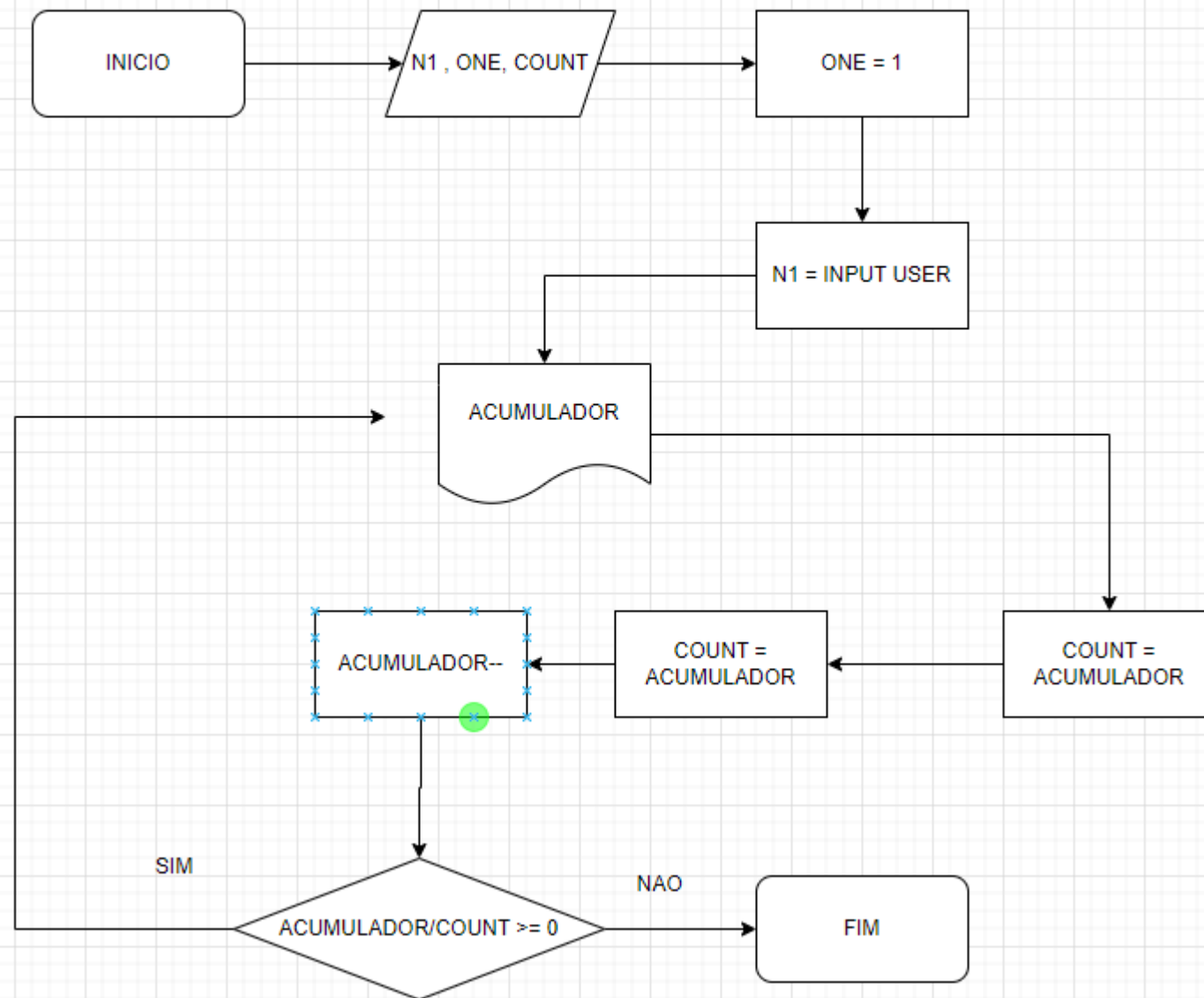
# ADICIONANDO DOIS NUMEROS



MAIOR ENTRE DOIS NUMEROS

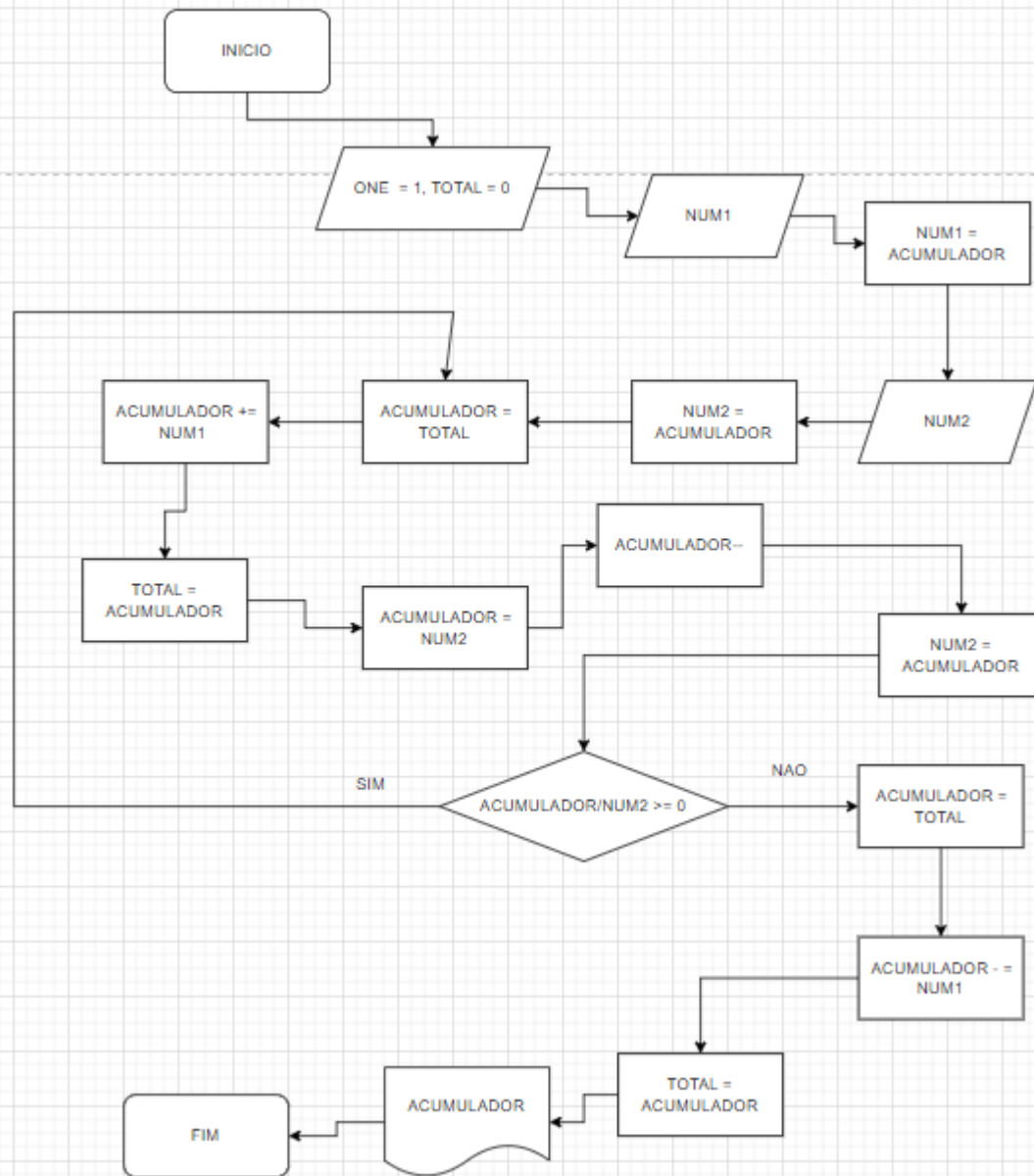


# COULDOWN



MULTIPLICANDO DOIS NUMEROS

```
graph TD
    INICIO([INICIO]) --> Init[/ONE = 1, TOTAL = 0/]
    Init --> Read1[/NUM1/]
    Read1 --> Assign1[ACUMULADOR = NUM1]
    Assign1 --> Read2[/NUM2/]
    Read2 --> Assign2[ACUMULADOR = NUM2]
    Assign2 --> Decision{ACUMULADOR/NUM2 >= 0}
    Decision -- NAO --> Assign3[ACUMULADOR = TOTAL]
    Assign3 --> Assign4[ACUMULADOR -= NUM1]
    Assign4 --> Assign5[TOTAL = ACUMULADOR]
    Assign5 --> Decision
    Decision -- SIM --> Assign6[ACUMULADOR = TOTAL]
    Assign6 --> Assign7[ACUMULADOR += NUM1]
    Assign7 --> Assign8[TOTAL = ACUMULADOR]
    Assign8 --> Decision
    Decision -- SIM --> Read3[/ACUMULADOR/]
    Read3 --> FIM([FIM])
```





# NUMEROS TRIANGULARES

