



UNIVERSIDADE ANHANGUERA

TABOÃO DA SERRA

**PORTIFÓLIO – RELATÓRIO DE AULA PRÁTICA
ALGORITMO E PROGRAMAÇÃO ESTRUTURADA**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

Helen Barros Lopes / RA: 3474905901

Tutor Presencial: Welton

Tutor a Distância: Joice Siqueira Lima Romanini

SUMÁRIO

PRIMEIRO ALGORITIMO

1	INTRODUÇÃO.....	3
2	MÉTODO E DESENVOLVIMENTO	3
3	ETAPAS.....	4
4	RESULTADO.....	5
5	CONCLUSÃO.....	6

SEGUNDO ALGORITIMO

1	MÉTODO E DESENVOLVIMENTO.....	7
2	ETAPAS.....	7
3	RESULTADO.....	7
4	CONCLUSÃO.....	8

REFERÊNCIAS.....	9
------------------	---

PRIMEIRO ALGORITMO

1 INTRODUÇÃO

O desenvolvimento de algoritmos é uma das bases fundamentais da programação. Nesta atividade, será criado um algoritmo para calcular a diferença de idade para a identificar se a pessoa pode ou não tirar habilitação e apresentar uma mensagem na tela informando como foram essas diferenças. Serão utilizadas estruturas de condição e o comando ESCREVAL para tratar possíveis falhas e definir se a pessoa está com a idade certa para tirar ou não.

2 MÉTODO E DESENVOLVIMENTO

Para criar o algoritmo, serão necessários os dados da idade da pessoa. A partir desses dados, será calculada a diferença de idade e verificado se houve a possibilidade, de tirar a habilitação. Para isso, serão utilizadas estruturas de condição para tratar possíveis falhas, como a entrada de dados inválidos. Inicialmente, será solicitado a idade da pessoa. Em seguida, será feito o cálculo da diferença de idade utilizando a fórmula "escreval("Informe tua idade")". Após o cálculo, será utilizado o comando ESCREVAL para verificar o valor da diferença de idade e definir a mensagem que será exibida na tela. Caso a diferença de idade seja menor que 18, será apresentada a mensagem "Menor de idade - Faltam 1 para poder tirar habilitação". Se a diferença de idade for maior que 18, será apresentada a mensagem "Maior de idade - Pode tirar a habilitação". Este algoritmo recebe como entrada a idade, calcula a diferença delas, de acordo com a diferença, exibe uma mensagem indicando se a pessoa pode ou não tirar a habilitação.

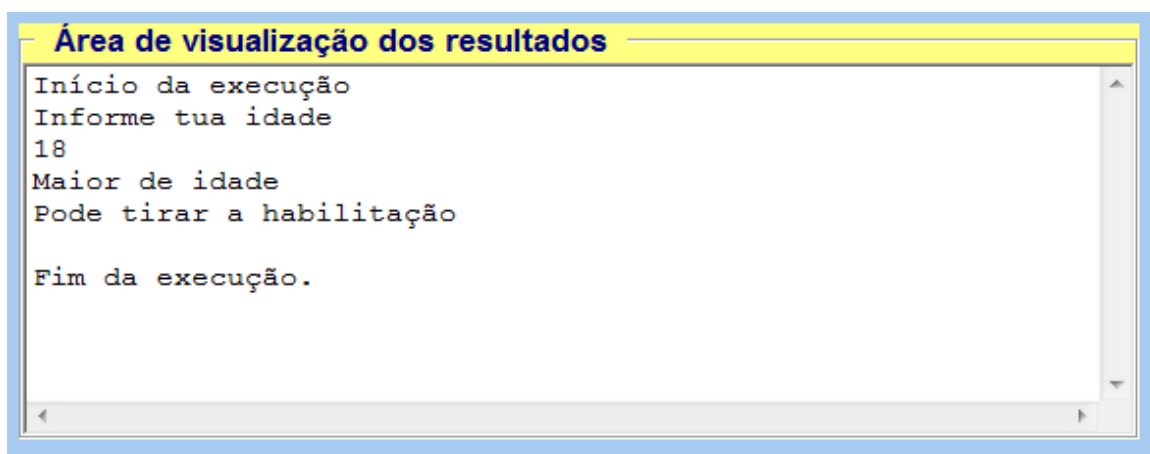
3 ETAPAS

```
Área dos algoritmos ( Edição do código fonte ) -> Nome do arquivo: [semnome]

1 Algoritmo "semnome"
2 //Receba a idade do usuário e verifique se ele tem 18 anos ou mais. Se a
3 //resposta for positiva escrever "maior de idade", senão "menor de idade".
4 //Se o usuário tiver idade maior igual a 18, escreva que pode tirar a
5 //habilitação, senão, deve indicar quantos anos que faltam para ele tirar a
6 //habilitação.
7
8 var
9     idade : inteiro
10
11 inicio
12     escreval("Informe tua idade")
13     leia(idade)
14     se (idade >= 18) entao
15         escreval("Maior de idade")
16         escreval("Pode tirar a habilitação")
17     senao
18         escreval("Menor de idade")
19         escreval("Faltam", (18 - idade), " para poder tirar habilitação")
20     fimSe
21
22 fimAlgoritmo
```

4 RESULTADOS:

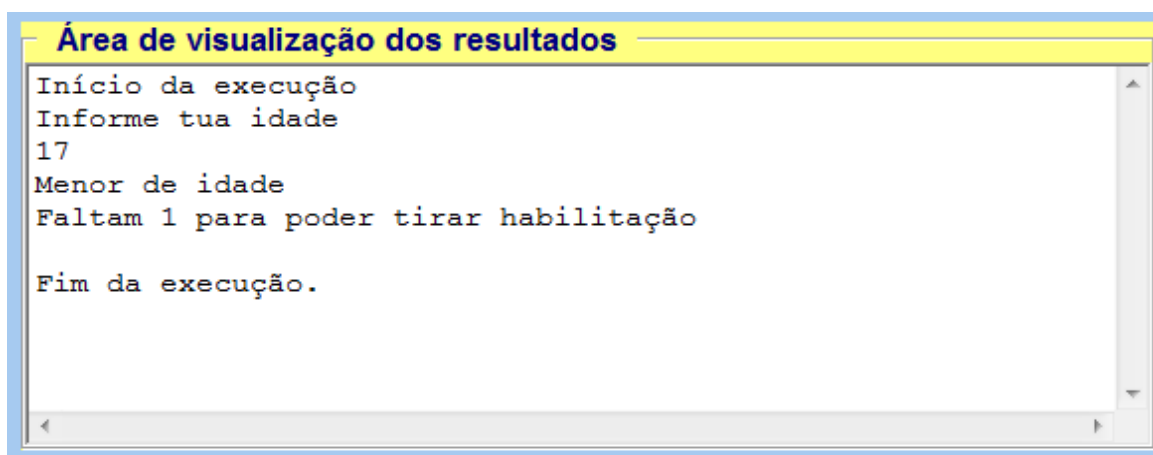
Maior de Idade:



```
Área de visualização dos resultados
Início da execução
Informe tua idade
18
Maior de idade
Pode tirar a habilitação

Fim da execução.
```

Menor de Idade:



```
Área de visualização dos resultados
Início da execução
Informe tua idade
17
Menor de idade
Faltam 1 para poder tirar habilitação

Fim da execução.
```

5 CONCLUSÃO

O desenvolvimento de algoritmos é uma atividade essencial para a programação e permite a criação de diversas soluções para problemas do dia a dia. Nesta atividade, foi criado um algoritmo simples para calcular a diferença de idade e apresentar uma mensagem na tela informando como foram essas diferenças. Utilizando estruturas de condição e o comando ESCREVAL, foi possível concluir e definir a mensagem correta a ser apresentada. É importante ressaltar que o desenvolvimento de algoritmos é uma habilidade que deve ser constantemente praticada e aperfeiçoada para a criação de soluções cada vez mais eficientes.

SEGUNDO ALGORITMO

1 INTRODUÇÃO

Algoritmo para operação de potenciação:

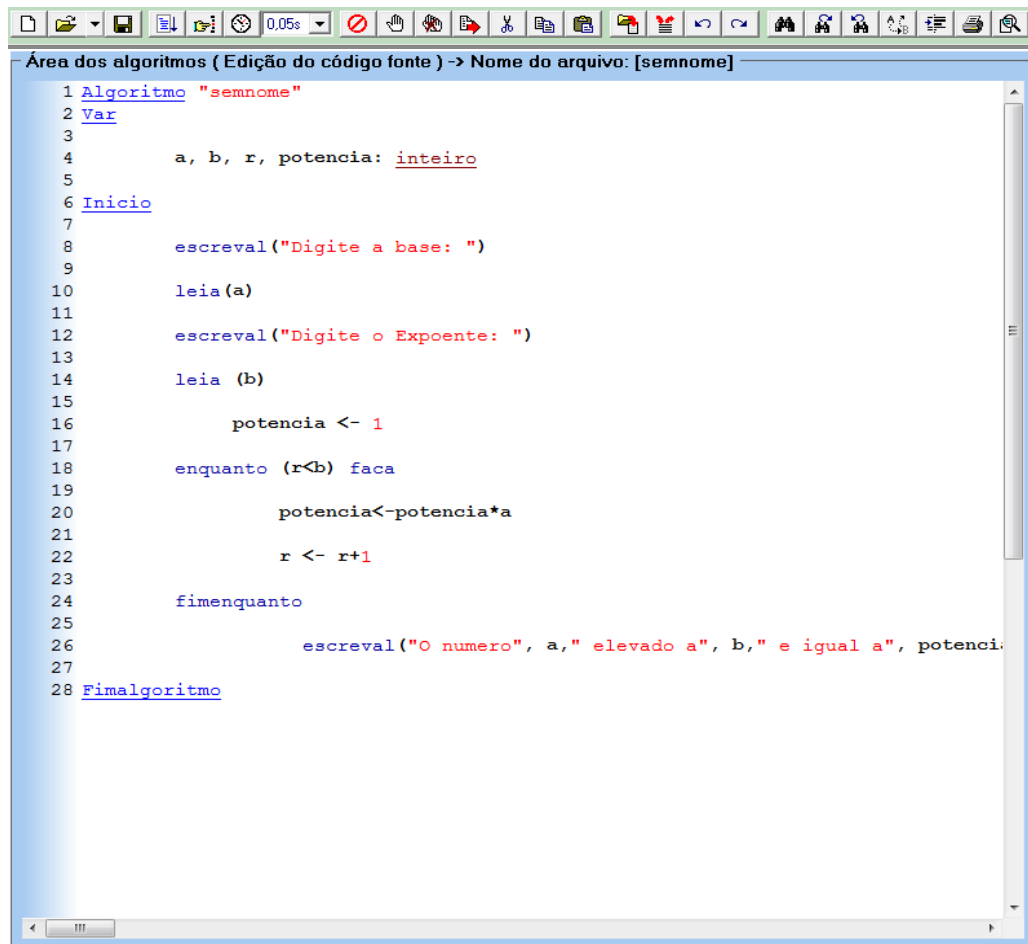
O algoritmo abaixo, em português, realiza a operação de potenciação entre os valores atribuídos a "a" e a "b" um número r de vezes. Para tanto, ele solicita que o usuário insira os números de servirão de base ("a") e de potência ("b"). A variável r recebe o número de vezes que o usuário deseja realizar a potenciação.

2 MÉTODO E DESENVOLVIMENTO

A estrutura de repetição tem como teste no início o valor de "r" que deve ser maior do que 0. Ele atinge esse valor ao ser decrementado em 1 dentro da estrutura de repetição a cada vez que as ações de inserção e cálculo da potência são efetivados.

3 ETAPAS E RESULTADO

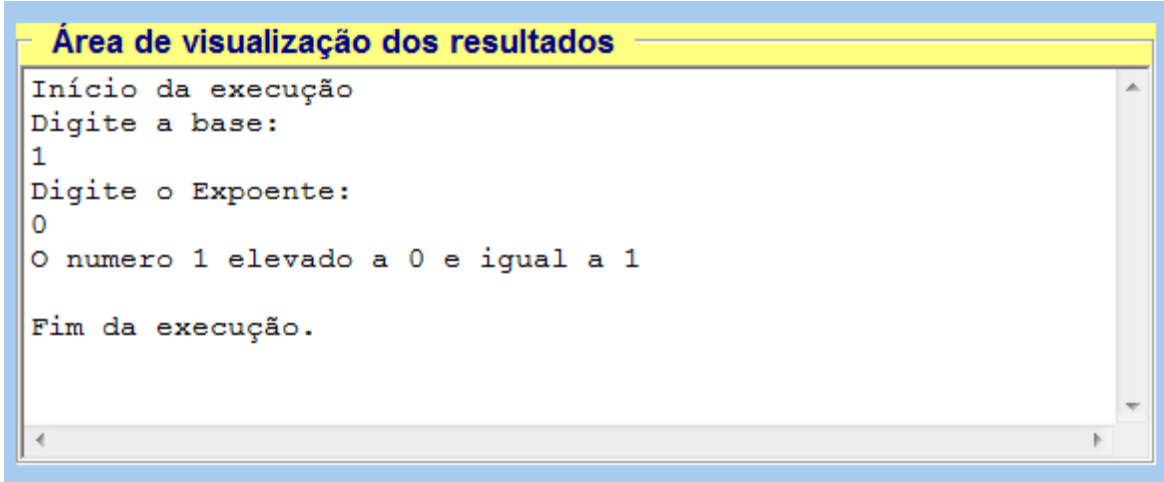
Etapas:



```

1 Algoritmo "semnome"
2 Var
3
4     a, b, r, potencia: inteiro
5
6 Inicio
7
8     escreval("Digite a base: ")
9
10    leia(a)
11
12    escreval("Digite o Expoente: ")
13
14    leia(b)
15
16    potencia <- 1
17
18    enquanto (r<b) faca
19
20        potencia<-potencia*a
21
22        r <- r+1
23
24    fimenquanto
25
26    escreval("O numero", a," elevado a", b," e igual a", potenci
27
28 Fimalgoritmo
  
```

Resultado:



```
Área de visualização dos resultados
Início da execução
Digite a base:
1
Digite o Expoente:
0
O numero 1 elevado a 0 e igual a 1
Fim da execução.
```

3 CONCLUSÃO

Este algoritmo permite que você calcule a potência de um número de forma eficaz, repetindo a multiplicação da base pela quantidade de vezes especificada pela potência.

Neste caso algo será repetidamente executado enquanto uma condição verdadeira for verificada, somente após a sua negativa essa condição será interrompida.

Na realização dessa condição, o comando iterativo "while", que significa "enquanto" em português realiza o teste no início, antes de executar as ações programadas.

REFERÊNCIAS

VISUALG

<https://visualg3.com.br/>