



**ANHANGUERA – CAMPO GRANDE/MS**

EZEQUIEL CUENGA – RA: 3500116601

**PORTFÓLIO – ALGORITMOS E PROGRAMAÇÃO ESTRUTURADA**

Mato Grosso do Sul

2023

## **PORTFÓLIO – ALGORITMOS E PROGRAMAÇÃO ESTRUTURADA**

Trabalho de portfólio apresentado como requisito parcial para a obtenção de pontos para a média 2º semestre.

Professor(a): Wesley Viana Pereira

Tutor à distância: Cristiano Giroldo

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>4</b>
<b>2 DESENVOLVIMENTO.....</b>	<b>4</b>
2.1 Primeiro Algoritmo – Maior/Menor Idade.....	4
2.1.1 Figura Visualg - Maior/Menor Idade.....	6
2.1.2 Figura Visualg- Resultado Maior.....	6
2.1.3 Figura Visualg- Resultado Menor.....	7
2.2 Segundo Algoritmo – Calcular Potenciação.....	8
2.2.1 Figura Visualg- Calcular Potenciação.....	8
2.2.2 Figura Visualg- Calculo 01.....	9
2.2.3 Figura Visualg- Calculo 02.....	10
2.3 Visualg- Link Código.....	10
<b>5 CONCLUSÃO.....</b>	<b>11</b>
<b>6 REFERÊNCIAS.....</b>	<b>12</b>

## 1. INTRODUÇÃO

Nesta breve apresentação, vamos introduzir dois algoritmos desenvolvidos utilizando a ferramenta VisualG.

O primeiro algoritmo trata da "Verificação de Idade", no qual o usuário fornece sua idade e recebe uma mensagem indicando se for menor ou maior de idade. Essa aplicação é valiosa para determinar critérios de acesso a certos conteúdo ou serviços.

Por outro lado, o segundo algoritmo aborda o "Cálculo de Potenciação". Nele, o usuário informa dois números inteiros e positivos, 'x' e 'y'. O algoritmo realiza a operação de potenciação, obtendo o resultado de 'x' elevado a 'y'. A estrutura de repetição com teste no início possibilita a realização eficiente de cálculos exponenciais.

Ambos os algoritmos são essenciais para compreender conceitos fundamentais de programação, como condicionais, repetições e operações matemáticas, enriquecendo a prática e o conhecimento em lógica de programação.

## 2. DESENVOLVIMENTO

O propósito deste algoritmo é possibilitar ao usuário inserir sua idade e, com base nessa entrada, receber uma mensagem que o categorize como menor ou maior de idade. Para sua elaboração, foi empregada a ferramenta VisualG, que oferece uma interface amigável e eficaz para a criação de algoritmos em linguagem portugol.

### 2.1 Primeiro Algoritmo – Maior/Menor Idade

O processo de desenvolvimento envolve a validação de uma variável (na linha 6 do exemplo abaixo), com o nome e o tipo de argumento/valor coletado (na linha 8 do exemplo abaixo). Posteriormente, solicita-se ao usuário que insira sua idade utilizando a fórmula (na linha 15 do exemplo abaixo):

```
6 Var
7 // Seção de Declarações das variaveis
8 idadeValida: inteiro
9 //-----

14 //Usuario deve informar idade atual
15 escreval("Digite sua Idade: ")
16 leia(idadeValida)
17 escreval("_____")
18 escreval("")
```

A partir deste ponto, a informação inserida na tela é armazenada na variável, como evidenciado na fórmula "leia(IdadeValida)" (na linha 16 do exemplo acima). Com esse valor inserido, podemos estabelecer uma regra para comparar a idade deste usuário.

```
20 //Compara a idade do Usuario sendo maior de idade
21 se (idadeValida >= 18) entao
22 escreval("Sua idade atual é ",idadeValida," Anos.")
23 escreva("Você já é MAIOR de idade.")
24 escreval("")
25
26 //Compara a idade do Usuario sendo menor de idade
27 senão
28 escreval("Sua idade atual é ",idadeValida," Anos.")
29 escreva("Você é MENOR de idade.")
30 escreval("")
31 FIMSE
32
33 Fimalgoritmo
```

Após a entrada do valor, estabelecemos a seguinte condição: se a idade for maior ou igual a 18 anos (entre as linhas 20 até a linha 23 do exemplo acima), exibimos o valor armazenado na variável acompanhado do texto:

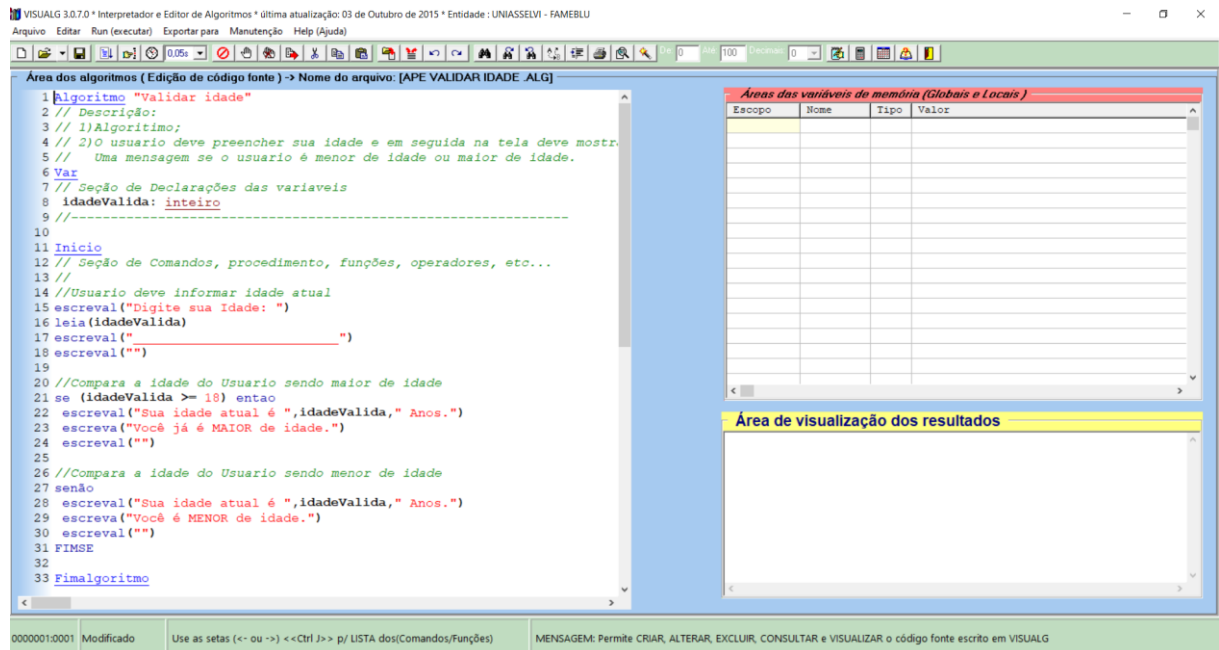
escreval("Sua idade atual é ",idadeValida," Anos.")(Linha 22);

Além de outra mensagem confirmando que é **maior de idade**(Linha 23).

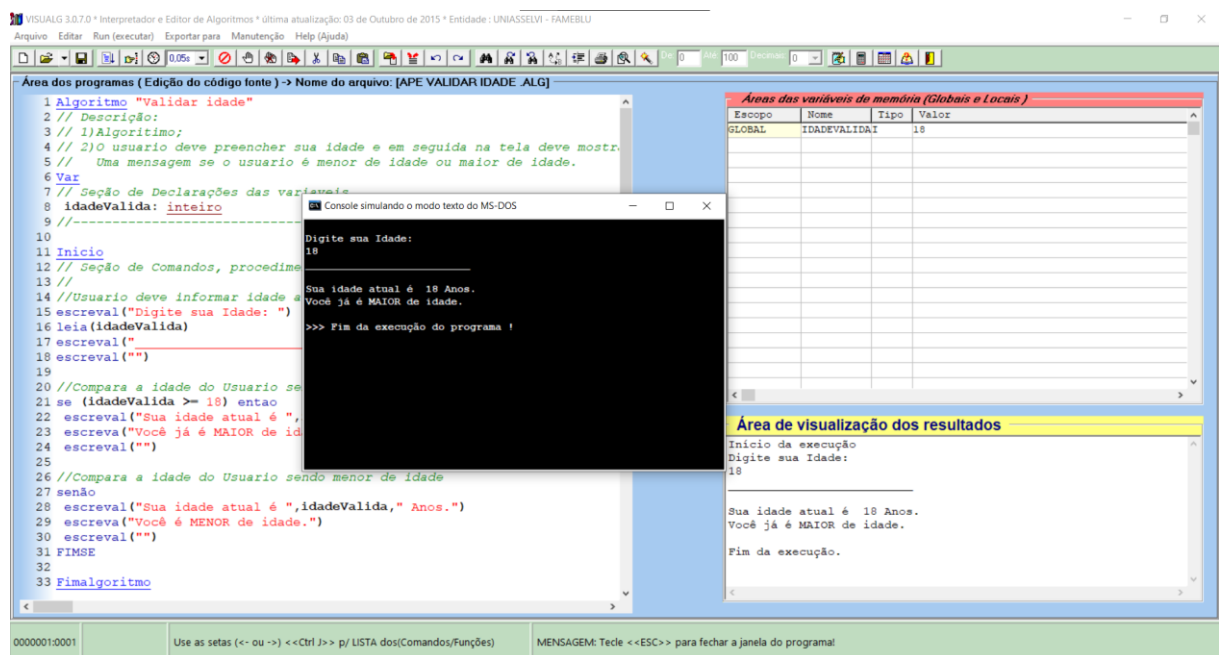
E por fim caso o usuário ao digitar sua idade for menor ou igual a 17(dezessete) anos(entre as linhas 26 até a linha 29, do exemplo acima), mostramos o valor imputado salvo na variável junto ao texto:

escreval("Sua idade atual é ",idadeValida," Anos.")(Linha 28); Além de outra mensagem confirmando que é **menor de idade**(Linha 29).

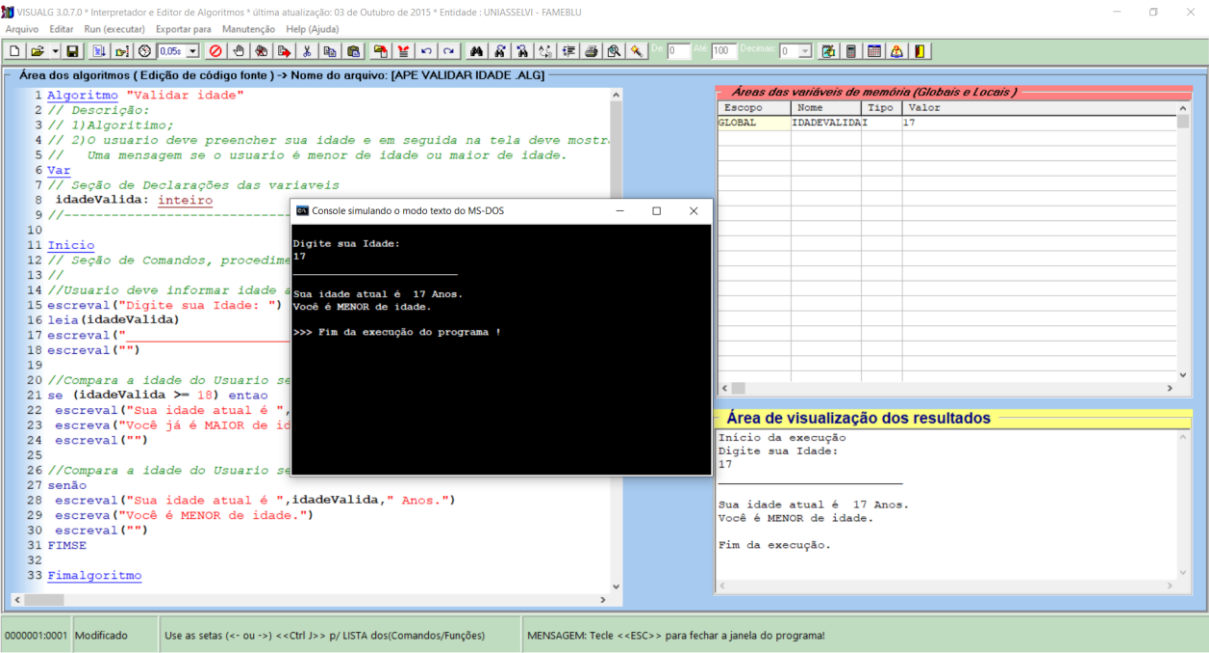
## 2.1.1 Figura Visualg - Maior/Menor Idade



## 2.1.2 Figura Visualg – Resultado Maior



### 2.1.3 Figura Visualg – Resultado Menor



## 2.2 Segundo Algoritmo – Calcular Potenciação

O segundo algoritmo tem como finalidade efetuar o cálculo da potência de um número 'x' elevado a um expoente 'y'. Ambos 'x' e 'y' são números inteiros e positivos (linha 16 do exemplo abaixo, juntamente com as variáveis "potencia" e "z"). Para realizar essa operação matemática, empregamos a ferramenta VisualG, que nos oferece a capacidade de criar algoritmos de forma eficaz e interativa.

### 2.2.1 Figura Visualg- Calcular Potenciação

```
6 Var
7 // Seção de Declarações das variaveis
8 y, potencia, x, z:inteiro
9 //-----
10 Inicio
11 //-----
12 // Primeiro escreve:
13 // base => "x" e expoente => "y"
14 escreva(" Digite y(Elevada Potencia): ")
15 leia(y)
16 escreval("")
17 escreva(" Digite x (Valor Elevado): ")
18 leia(x)
19 escreval("")
20 //-----
```

O processo de desenvolvimento envolve a solicitação ao usuário para que insira os valores de 'y' e 'x' por meio da função "leia".

Com esses valores em mãos, o algoritmo inicia uma variável chamada "z" com o valor 1 (linha 30, do exemplo abaixo, "z:=1").

```
30 z:= 1
31 para potencia de 1 ate x faca
32 //
33 //=====
34 //EXEMPLO 1:
35 //-----
36 //para potencia de 1 ate 3 faz:
37 // "z" := "1"
38 // "z" = "1*2"
39 // "z" : = "2"
40 //=====

62 z:= z*y
63 fimpara
64 escreva("Resultado P = ", z, ".")
65 fimalgoritmo
```



Em seguida, utilizamos uma estrutura de repetição "para...faca" para realizar o cálculo exponencial de forma iterativa (linha 31, do exemplo acima).

A repetição ocorre 'y' vezes, e em cada iteração, multiplicamos o valor atual de "resultado" pelo valor de 'x'. Dessa forma, ao final do laço, a variável "resultado" conterá o resultado de 'x' elevado a 'y'.

### 2.2.2 Figura Visualg- Calculo 01

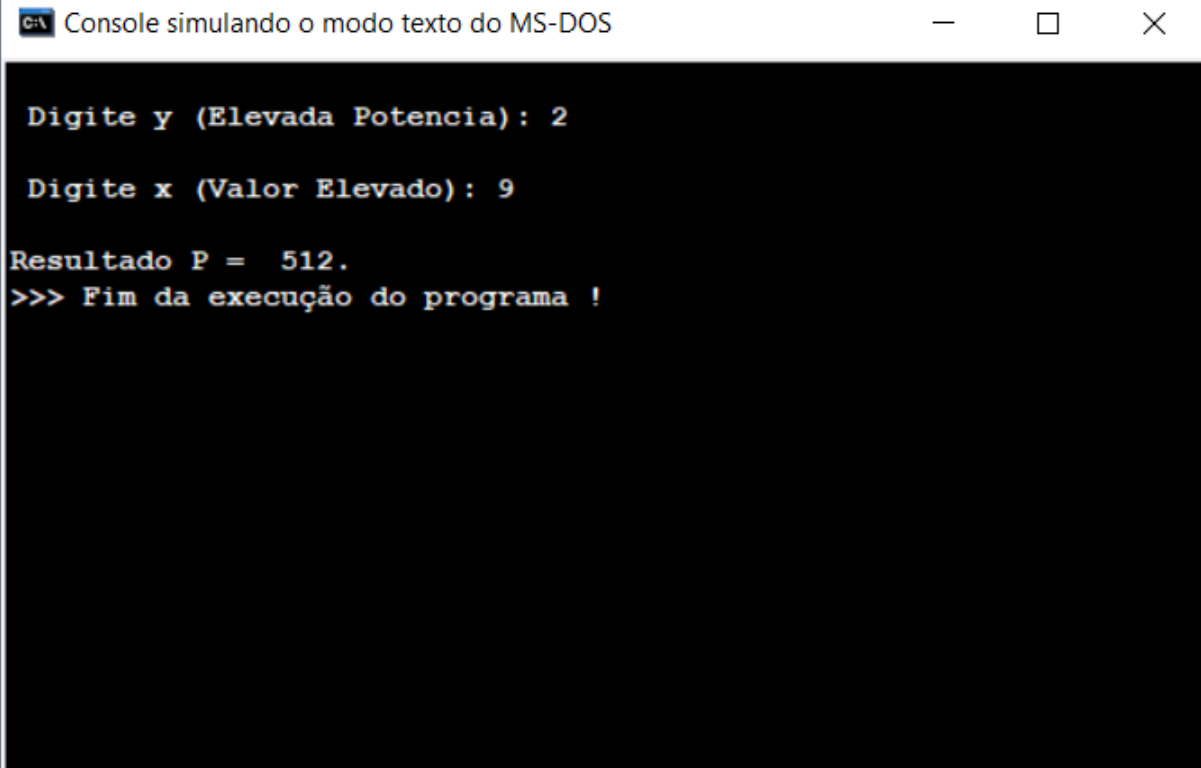
Áreas das variáveis de memória (Globais e Locais)			
Escopo	Nome	Tipo	Valor
GLOBAL	Y	I	9
GLOBAL	POTENCIA	I	2
GLOBAL	X	I	2
GLOBAL	Z	I	81

Console simulando o modo texto do MS-DOS

```
Digite y (Elevada Potencia): 9
Digite x (Valor Elevado): 2
Resultado P = 81.
>>> Fim da execução do programa !
```

### 2.2.3 Figura Visualg- Calculo 02

Áreas das variáveis de memória (Globais e Locais)			
Escopo	Nome	Tipo	Valor
GLOBAL	Y	I	2
GLOBAL	POTENCIA	I	9
GLOBAL	X	I	9
GLOBAL	Z	I	512

```
Digite y (Elevada Potencia): 2

Digite x (Valor Elevado): 9

Resultado P = 512.
>>> Fim da execução do programa !
```

A estrutura de repetição com teste no início é uma abordagem eficiente para calcular potências, evitando operações exponenciais complexas. Através dessa repetição, conseguimos diminuir o número de multiplicações necessárias para obter o resultado, tornando o algoritmo mais rápido e otimizado.

### 2.3 Visualg – Link Código

<https://github.com/ezequielkruz/FaculdadeADS/blob/main/visualg%20potencia>

<https://github.com/ezequielkruz/FaculdadeADS/blob/main/visualg%20validar%20idade>

## 4 CONCLUSÃO

Com a conclusão do desenvolvimento do segundo algoritmo para o cálculo de potenciação, ampliamos nossa compreensão em programação, tornando-nos aptos a abordar problemas matemáticos complexos e aplicar conceitos fundamentais em várias áreas da computação.

Neste relatório, examinamos dois algoritmos distintos, cada um com sua própria finalidade e lógica de programação. O primeiro algoritmo, focado na "Verificação de Idade", possibilitou a classificação dos usuários como menores ou maiores de idade com base na entrada fornecida. Já o segundo algoritmo tratou do "Cálculo de Potenciação", demonstrando de forma eficiente como elevar um número 'x' a um expoente 'y'.

Uma das principais vantagens de desenvolver esses algoritmos com o auxílio da ferramenta VisualG foi a facilidade de interação e visualização do processo. O VisualG oferece uma interface amigável que permite aos programadores criar algoritmos de maneira intuitiva, proporcionando uma visualização clara das etapas de execução. Esse aspecto é especialmente valioso para iniciantes, auxiliando na compreensão da lógica por trás do código de forma mais tangível.

Além disso, o aprendizado obtido ao desenvolver esses algoritmos é inestimável. A criação do primeiro algoritmo, que envolveu estruturas condicionais, ensinou a tomar decisões em um programa com base em condições específicas. O segundo algoritmo, por sua vez, introduziu o conceito de repetição controlada por um teste, mostrando como otimizar cálculos complexos de forma iterativa.

O uso do VisualG permitiu que nos aprofundássemos gradualmente no mundo da programação de forma compreensível. A familiaridade com as estruturas de controle, operações matemáticas e entrada/saída de dados enriqueceu nossa compreensão em lógica de programação. Essas habilidades são transferíveis para outras linguagens de programação, possibilitando a expansão do nosso repertório à medida que avançamos no campo da computação.

Em conclusão, explorar algoritmos por meio do VisualG não apenas nos proporcionou um meio eficaz de criar soluções práticas, mas também nos equipou com habilidades valiosas no universo da programação. O aprendizado adquirido durante essa jornada nos preparou para enfrentar desafios mais complexos, incentivando um crescimento contínuo e a construção de bases sólidas para futuras explorações no mundo da tecnologia."

## **5. REFERÊNCIAS**

Tuto Studio - <https://www.youtube.com/watch?v=XY3jyX6lZ6I>

Professor(a): Wesley Viana Pereira, Algoritmos e Programação Estruturada, 2023