

## Testando Valor de Entrada

Mário Leite

...

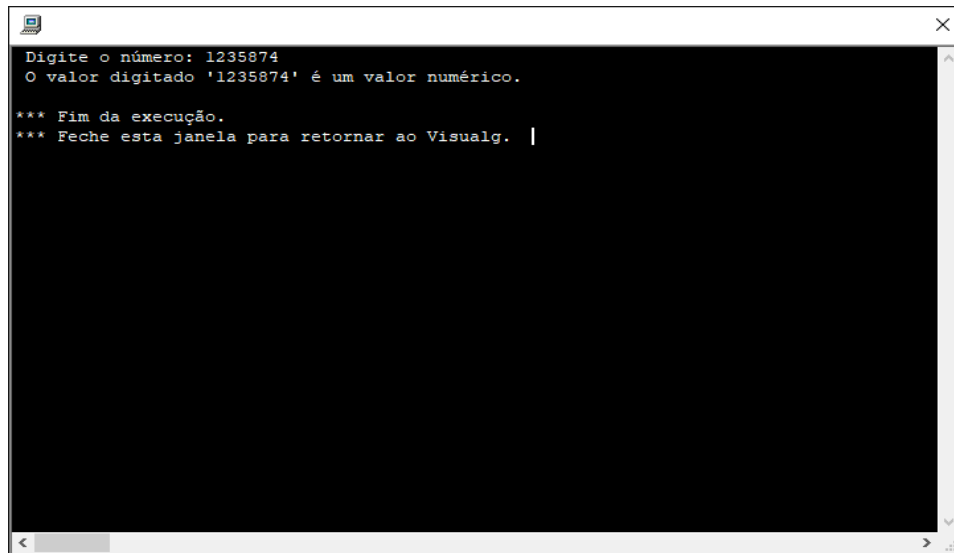
Qualquer sistema aberto possui três etapas: **Entrada - Processamento - Saída**. E normalmente a **Saída** representa a informação desejada pelo usuário, para que possa tomar alguma decisão a respeito de determinada questão. E embora a segunda etapa (**Processamento**) seja importante, o seu tipo é que vai determinar a eficiência e eficácia de todo o processo de obtenção da informação esperada. Nos eventos que exigem valores numéricos na **Entrada** a validação dos dados é muito importante, pois pode prejudicar seriamente todo o sistema, e o processamento pode até não ocorrer. Então, verificar o tipo de entrada de dado é fundamental.

A maioria das linguagens possui recursos próprios (*funções*) que podem ser usadas para verificar se um dado de entrada é numérico, ou não. A maioria dessas funções tem como identificador a palavra-chave **IsNumeric(x)**, sendo **x** um parâmetro de entrada: um valor que lhe é passado. Em algumas delas, como no caso do **Python** e **C#**, a verificação está definida como um método do objeto “valor” analisado e não como uma função da linguagem: **valor.Isnumeric(string)**; no caso do Python pode ser usado o método **isdigit()**. De qualquer forma, para o programador esta *função/método* é muito importante na implementação. Mas aqui pode ser feita uma pergunta: e se o programador quiser criar seu próprio recurso para verificar um valor numérico!? Como isto poderia ser feito!? Muito simples: em TODAS as linguagens cada caractere tem um valor inteiro associado, denominado “código ASCII”. Assim, por exemplo, as letras maiúsculas (A-Z) têm códigos que vai de 65 (“A”) até 90 (“Z”) e as minúsculas de 97 (“a”) até 122 (“z”. No caso dos números de 48 (0) até 57 (9).

Deste modo, é muito fácil verificar cada caractere do pretenso número digitado: se estiver na faixa **[48-57]** é número; caso contrário não é. O programa “**VerificaNumerico**”, implementado em **Visualg**, mostra um exemplo de como isto pode ser feito: o programa principal recebe o valor de entrada digitado pelo usuário que chama a função **EhNumero()** e passa-lhe este “valor”. A função analisa cada caractere do valor passado; se pelo menos um dos caracteres desse “valor” tiver um código ASCII fora da faixa o **[48-57]** não é um valor numérico, e retorna **Falso**; e se todos os caracteres estiverem nesta faixa retorna **Verdadeiro**. As **figuras 1, 2 e 3** mostram exemplos de saída deste programa.

Portanto, se sua linguagem preferida não possuir recurso próprio de fazer esta checagem, seja um PROGRAMADOR e crie sua própria função; este exemplo dado em **Visualg** pode ser facilmente convertido na sua linguagem preferida

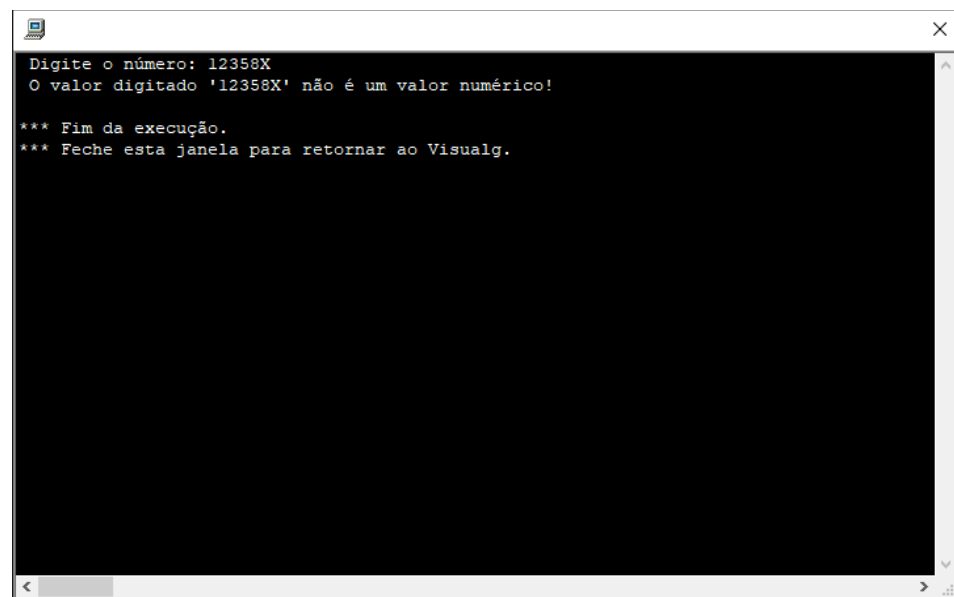
---



```
Digite o número: 1235874
O valor digitado '1235874' é um valor numérico.

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg. |
```

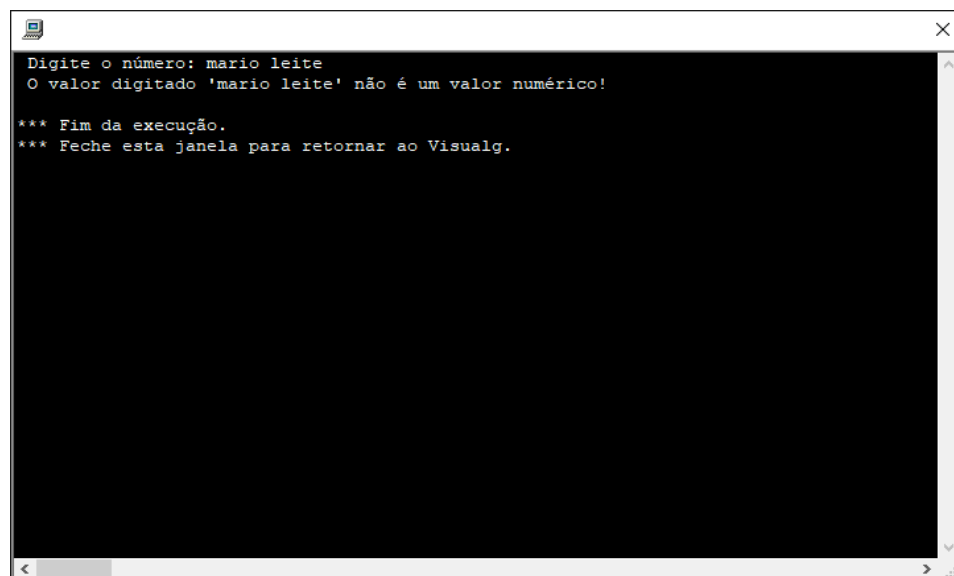
**Figura 1 - Valor numérico digitado**



```
Digite o número: 12358X
O valor digitado '12358X' não é um valor numérico!

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

**Figura 2 - Valor não numérico digitado**



```
Digite o número: mario leite
O valor digitado 'mario leite' não é um valor numérico!

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

**Figura 2 - Valor não numérico digitado**

Algoritmo **"VerificaNumerico"**

```
// Programa para testar função que verifica se um valor é numérico.  
// Em Visualg  
// Autor: Mário Leite  
// Data: 05/03/2023  
//-----
```

**Funcao** **EhNumero** (strNum: caractere): logico

**var** j, nCar, tamStr: inteiro  
        sCar: caractere  
        ret: logico

**Inicio**

```
tamStr <- Compr(strNum) //pega o comprimento da string  
ret <- Verdadeiro  
Para j De 1 Ate tamStr Faca  
    sCar <- Copia(strNum,j,1) //pega o código ASCII do caractere atual  
    nCar <- Asc(sCar)  
    Se ((nCar < 4) OU (nCar > 57)) Entao //verifica se está na faixa numerica  
        ret <- Falso //não é um número  
        Interrompa //sai do loop incondicionalmente  
    FimSe  
FimPara  
Retorne ret //retorno da função
```

**FimFuncao**

```
//=====
```

```
//Programa Principal: testa a função
```

**Var** num: caractere  
        EhNum: logico

**Inicio**

```
    Escreva (" Digite o número: ")  
    Leia (num)  
    EhNum <- EhNumero (num)  
    Se (EhNum) Entao  
        Escreval (" O valor digitado ", "'", num, "'", " é um valor numérico.")  
    Senao  
        Escreval (" O valor digitado ", "'", num, "'", " não é um valor numérico!")  
    FimSe
```

**FimAlgoritmo**