

## Sobre repetições em Lista: Visualg x Python

Mário Leite

...

Seguindo rigorosamente o que o Gustavo Guanabara sugeriu no seu curso de Python no You Tube (não pulando nenhuma aula, fazendo TODOS os exercícios dos desafios e não olhando previamente) consegui aprender muito bem a linguagem. Realmente é uma linguagem muito poderosa e com muitos recursos extras que outras não possuem. Um desses recursos que muito me agradou foi **Listas**, abordado na **aula 17**. E um dos desafios propostos ao final desta aula foi o de número **79** cujo enunciado é o seguinte: *“Criar um programa onde o usuário possa digitar vários valores numéricos e cadastre-os numa lista. Caso o número já exista lá dentro, ele não será adicionado. No final serão exibidos todos os valores únicos digitados em ordem crescente”*.

Confesso que a maior dos exercícios dos desafios que ele propôs eu já tinha feito em outras linguagens, e isto me facilitou muito na solução codificada em Python. Mas, esse de número **79** me chamou muito a atenção por ter sido um dos que mais me deram trabalho para criar o algoritmo, uma vez que é para criar uma lista de números inteiros dinamicamente e eliminando os elementos repetidos no ato da entrada dos números. Muito interessante; um ótimo desafio para os iniciantes. Então, consultando meus “papiros” encontrei um código em Visualg que havia feito para um dos meus livros. Assim, estou apresentando a vocês as duas versões: em **Visualg** e em **Python**; apenas fiz alguns ajustes na entrada para generalizar o tamanho da lista, e retirei a parte em que ele pede para apresenta a lista ordenada, o que é muito fácil. Entretanto, o que eu gostaria de passar para vocês é o que SEMPRE estou comentando aqui: **Programar** é uma coisa, **Codificar** é outra!

Observem que o código em Python ficou bem mais enxuto e menos complexo do que o código em Visualg; isto pode que, TEORICAMENTE, o código em Python é mais eficiente que o escrito em Visualg. Em princípio, eu concordo; uma vez que em Visualg existem muitas estruturas aninhadas e profundas; o que pode, TEORICAMENTE, forçar mais o processador. ENTRETANTO, como já afirmei aqui, o que interessa ao USUÁRIO (o verdadeiro dono do programa) é a eficácia do programa e não, necessariamente, a sua eficiência. É claro que o ideal é o programa ser eficiente e eficaz, simultaneamente. Mas, por outro lado, o que deve ser SEMPRE considerado pelo programador é que a solução do problema NÃO é dada pela linguagem, e sim pelo algoritmo que implementa a sua solução. Quando eu criei este programa segui um algoritmo lógico: “A CADA ELEMENTO LIDO DEVE-SE VERIFICAR SE ELE JÁ ESTÁ NA LISTA, ANTES DE FAZER A INSERÇÃO”; este é o ponto crucial, e que deve ser observado; DEPOIS, verifique se a linguagem que você conhece tem recursos para fazer isto mais eficientemente, que é o caso de Python. Mas isto NÃO É O CASO DA MAIORIA DAS LINGUAGENS! Então, como eu já sabia o algoritmo, foi muito fácil implementar em Python, que possui esse maravilhoso recurso, MAS... e se você não souber codificar em Python, o que você faria? Procuraria a solução pronta na Internet!!?? Desistiria de ser programador, ou pensaria numa solução lógica para o problema!?

---

***“Que a Lógica esteja com Vocês...!”***

**Algoritmo "ListaSemRep"**

```
//Cria dinamicamente uma lista de números inteiros sem elementos repetidos.
//Em Visualg
//Autor: Mário Leite
//-----
Const MAXTAM=100 //define um tamanho máximo para a lista
Var ListNum: vetor[1..MAXTAM] de inteiro
    i, j, k, N, m: inteiro
Inicio
  Repita
    Escreva("Quantidade de elementos do vetor [min 2", "-max", MAXTAM, "]:")
    Leia(n)
    n <- Int(Abs(n)) //garante N inteiro e positivo
  Ate((n>=2) e (n<=MAXTAM))
  j <- 1
  Enquanto (j<=n) Faca //loop para criar a lista
    Escreva("Digite um número: ")
    Leia(ListNum [j])
    Se(j>1) Entao //começa a partir do segundo elemento
      Para i De 1 Ate (j-1) Faca //loop para verificar repetições
        k <- i + 1
        Para m De k Ate j Faca //verifica repetição
          Enquanto (ListNum [i]= ListNum [j]) Faca //repetiu!!
            {Tenta um novo número para este elemento}
            Escreva("Repetido! Tente outro número: ")
            Leia(ListNum [j]) //lê o novo elemento digitado
          FimEnquanto
        FimPara
      FimPara //fim do loop de verificação de repetições
    FimSe
    j <- j + 1 //próximo elemento da lista
  FimEnquanto //fim do loop de criação da lista
  LimpaTela
  {Exibe a lista com seus elementos}
  Para i De 1 Ate N Faca
    Escreva(ListNum [i], " ")
  FimPara
FimAlgoritmo
```

Figura 1 - Código em Visualg 3.0.7

```

'''
ListaSemRep.py
Cria dinamicamente uma lista de números inteiros sem elementos repetidos.
Em Python
Autor: Mário Leite
'''

#Inicio do Programa-----
endif = "endif"
endwhile = "endwhile"
MAXTAM = 100 #define um tamanho máximo para a lista
print(f"Digite o tamanho do vetor [2 - {MAXTAM}]: ", end="")
n = int(input())
while((n < 2) or (n>MAXTAM)):
    print(f"Digite o tamanho do vetor [2 - {MAXTAM}]: ", end="")
    n = int(input())
endwhile
j = 1 #contador de elementos da lista
ListNum = []
while(j <= n): #loop para criar a lista
    Num = int(input("Digite um número: "))
    ListNum.append(Num)
    #Verifica repetições
    Vezes = ListNum.count(Num)
    if(Vezes > 1):
        print(f"Repetido!. Tente outro número!")
        ListNum.remove(Num) #remove o item repetido
        j = j - 1 #decrementa o contador de elementos
    endif
    j = j + 1
endwhile #fim do loop de criação da lista
print("\n")
#Exibe a lista com seus elementos
print(ListNum)
#FimPrograma-----

```