

Programação Tradicional x Programação Calistênica

Mário Leite

Uma das premissas básicas para ser um bom programador (não somente codificador) é dominar as três estruturas básicas de programação: *Decisão*, *Seleção* e *Repetição*. Sem saber como usá-las corretamente o programa pode se tornar ineficiente, comprometendo seriamente a sua eficácia. E como programar é, em última análise, tomar decisões em função de dados de entrada ou obtidos através de leitura em arquivos ou mesmo fruto de processamento, as estruturas de decisão se tornam fundamentais em qualquer tipo de programa, envolvendo questões baseadas em situações condicionantes do tipo:

Se(condição) Então

Instruções se “condição” for verdadeira

Senão

Instruções se “condição” for falsa

FimSe (para linguagens que possuem algum terminador)

Qual é o problema que pode aparecer nesse tipo de estrutura?

O problema, embora não prejudique seriamente a velocidade de processamento, é quando esse tipo de estrutura contém muitos **Se’s** (If’s) e **Senão’s** (Else’s), aumentando muito a “profundidade” das instruções alternativas; quando aparecem muitas linhas de código aninhadas. Esse tipo de programação tradicional deve ser trocado (sempre que puder) por um tipo de programação chamada “**Calistênia**”, onde os “**Senão’s**” (else’s) são eliminados. E embora isto não torna, necessariamente, o código mais rápido, a legibilidade do programa fica bem melhor, mais plano e oferecendo uma melhor manutenibilidade. Os dois programas “**ProgComElse**” e “**ProgSemElse**” ambos codificados em Python, são exemplos que mostram a diferença entre as estruturas de decisão: com programação tradicional e com programação Calistênia.

```
'''
ProgComElse.py
-----
Programação tradicional, com "else's" aninhados, aprofundando a estrutura.
-----
'''
nota = 11
while((nota<0) or (nota>10)):
    nota = abs(float(input("Entre com a nota do aluno: ")))

print()
if(nota >= 9):
    print("Nível Excelente.")
else:
    if(nota >= 7):
        print("Nível Bom.")
    else:
        if(nota >= 5):
            print("Nível Regular.")
        else:
            print("Nível Insuficiente.")
#Fim do programa "ProgComElses" -----
```

```
'''
ProgSemElse.py
-----
Agora, usando a filosofia Calistênica de Programação" (sem else, plano, direto):
Cada condição é avaliada isoladamente. Se satisfaz, executa e sai da função com return.
Sem nenhum else. Código muito mais plano, limpo e fácil de manter.
-----
'''
nota = 11
#Avaliação sem 'else'
nota = 11
while((nota<0) or (nota>10)):
    nota = abs(float(input("Entre com a nota do aluno: ")))

print()

if(nota >= 9):
    print("Nível Excelente.")

if(nota >= 7 and nota < 9):
    print("Nível Bom.")

if(nota >= 5 and nota < 7):
    print("Nível Regular.")

if(nota < 5):
    print("Nível Insuficiente.")
#Fim do programa "ProgSemEles" -----
```

=====

Nota: A "Programação Calistênica" (inspirada no conceito de calistenia corporal) é uma filosofia que propõe regras para escrever código limpo, disciplinado e de alta qualidade - usando práticas simples, mas poderosas. Ela foi originalmente proposta por "Christian Grobmeier" em um artigo chamado "Object Calisthenics" (Calistenia de Objetos), com algumas regras para treinar o "código limpo"; mas a melhor aplicação é evitar o excesso de aprofundamento nas opções de estruturas de decisão, facilitando a legibilidade do código.

=====