



*Serviço Nacional de Aprendizagem Industrial*  
**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# **LÓGICA DE PROGRAMAÇÃO**

## **PSEUDOCÓDIGO**



## **CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI**

*Robson Braga de Andrade*  
Presidente

### **GABINETE DA PRESIDÊNCIA**

*Teodomiro Braga da Silva*  
Chefe do Gabinete - Diretor

### **DIRETORIA DE EDUCAÇÃO E TECNOLOGIA - DIRET**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor de Educação e Tecnologia

### **SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL - SENAI**

*Robson Braga de Andrade*  
Presidente do Conselho Nacional

### **SENAI – Departamento Nacional**

*Rafael Esmeraldo Lucchesi Ramacciotti*  
Diretor-Geral

*Julio Sergio de Maya Pedrosa Moreira*  
Diretor-Adjunto

*Gustavo Leal Sales Filho*  
Diretor de Operações



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

SÉRIE TI - SOFTWARE

# LÓGICA DE PROGRAMAÇÃO

PSEUDOCÓDIGO



© 2020. SENAI – Departamento Nacional

© 2020. SENAI – Departamento Regional de Santa Catarina

A reprodução total ou parcial desta publicação por quaisquer meios, seja eletrônico, mecânico, fotocópia, de gravação ou outros, somente será permitida com prévia autorização, por escrito, do SENAI.

Esta publicação foi elaborada pela equipe de Educação a Distância do SENAI de Santa Catarina, com a coordenação do SENAI Departamento Nacional, para ser utilizada por todos os Departamentos Regionais do SENAI nos cursos presenciais e a distância.

**SENAI Departamento Nacional**

Unidade de Educação Profissional e Tecnológica - UNIEP

**SENAI Departamento Regional de Santa Catarina**

Gerência de Educação

**SENAI**

Serviço Nacional de  
Aprendizagem Industrial  
Departamento Nacional

**Sede**

Setor Bancário Norte • Quadra 1 • Bloco C • Edifício Roberto  
Simonsen • 70040-903 • Brasília – DF • Tel.: (0xx61) 3317-  
9001 Fax: (0xx61) 3317-9190 • <http://www.senai.br>

# Lista de Ilustrações

---

Figura 1 - Tela do VisualG – Partes do Algoritmo em Pseudocódigo .....	10
Figura 2 - Tela do VisualG – Pseudocódigo para Somar dois valores inteiros quaisquer.....	10
Figura 3 - Fluxograma SE e SENAO.....	12
Figura 4 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura SE com SENAO.....	12
Figura 5 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura SE SENAO SE.....	13
Figura 6 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ESCOLHA .....	14
Figura 7 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura PARA.....	15
Figura 8 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura PARA (Resultado).....	16
Figura 9 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ENQUANTO .....	17
Figura 10 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ENQUANTO (Resultado).....	17
Figura 11 - Tela do VisualG – Pseudocódigo Contador de Tipo de Peças.....	18
Figura 12 - Tela do VisualG – Pseudocódigo Ficar no Parque ou ir embora .....	20

Tabela 1 - Interpretação das linhas de código em Portugol da Pseudocódigo Contador de Tipo de Peças ....	19
--	----



# Sumário

---

Pseudocódigo .....	9
Apresentação .....	9
Definição .....	9
Estrutura .....	10
Estrutura de Decisão .....	11
Estrutura de Escolha .....	13
Estrutura de Repetição Para .....	15
Estrutura de Repetição Enquanto .....	16
Aplicação na indústria .....	17
Exemplos .....	19
Referências.....	22







## APRESENTAÇÃO

Olá!

Preparado para explorar Pseudocódigo! Neste tópico, você irá acompanhar a construção de pseudocódigos completos para solucionar determinadas situações-problema, que vão desde o mapeamento das variáveis, suas declarações, definições de tipos, atribuição de valores, interação com o usuário para captar novas informações para as variáveis e, finalmente, a utilização das expressões lógicas e aritméticas para obter um comportamento adequado para essas aplicações.

Bons estudos!

## DEFINIÇÃO

Pseudocódigo é uma das possíveis formas de representar algoritmos computacionais, sendo que os algoritmos são resoluções propostas para determinadas situações-problema. Nesta unidade, será utilizado Portugol, que é um pseudocódigo baseado em regras e definições muito próximas ao universo das linguagens de programação tradicionais, mas com a vantagem de utilizar palavras em português. Este é um ponto de partida interessante para quem está iniciando no universo da computação.

Obviamente, seria muito improvável encontrar no mundo real aplicações escritas em Portugol, pois seu propósito é muito mais didático. É por isso que, inclusive, sempre se faz um paralelo entre o que é apresentado em Portugol com uma linguagem tradicional e profissional de programação, no caso o Java. Neste sentido, utiliza-se essa estratégia para que você possa ter sempre as duas experiências, a do nível formativo introdutório e também possa ver como um problema seria solucionado no mundo real da computação.

## ESTRUTURA

A estrutura de um pseudocódigo é muito simples. Acompanhe agora como um programa completo em Portugol é construído, suas partes e o propósito de cada uma.

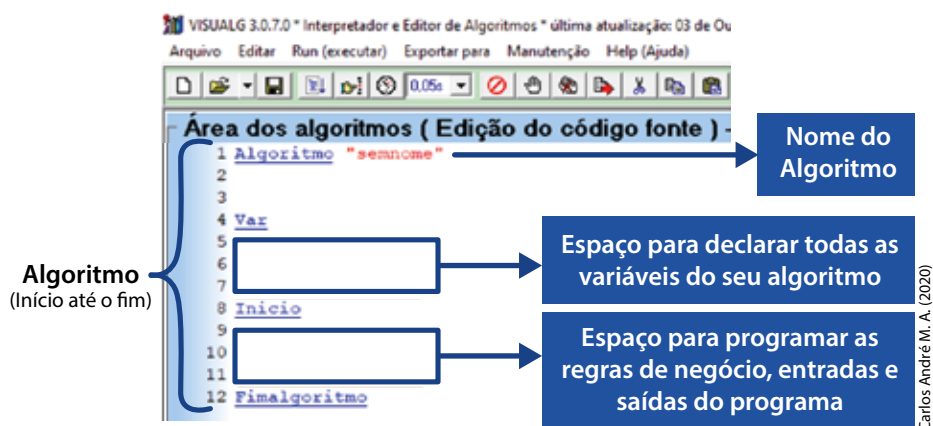


Figura 1 - Tela do VisualG – Partes do Algoritmo em Pseudocódigo

Fonte: do Autor (2020)

Então, imagine uma aplicação simples, para entender a elaboração de um pseudocódigo completo que possa ser capaz de solicitar dois números inteiros quaisquer ao usuário, somá-los e apresentar o resultado desta soma.

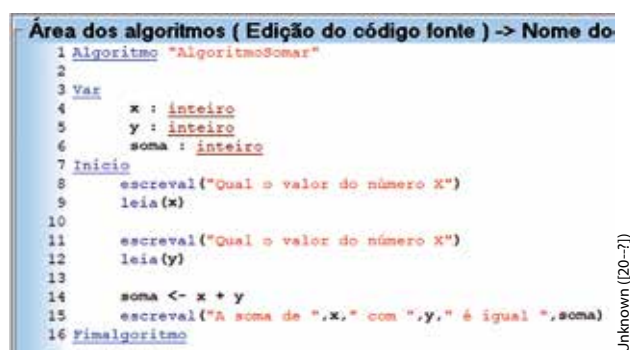


Figura 2 - Tela do VisualG – Pseudocódigo para Somar dois valores inteiros quaisquer

Fonte: do Autor (2020)

Veja que, conforme apresentado na Figura "Partes do Algoritmo em Pseudocódigo", e na imagem anterior, identifica-se que: na linha 1, ocorre a definição do nome do algoritmo; entre as linhas 4 e 6, a declaração das variáveis; e, por fim, entre as linhas 8 e 15, o desenvolvimento das regras do programa deste exemplo. Note que, na linha 8, é feita uma interação com o usuário, enviando uma mensagem para que ele informe o valor da variável x. O valor que for digitado será capturado pela instrução de leitura na linha 9 e armazenado na variável x.

Quando se fala em pseudocódigo, é importante conhecer outras estruturas utilizadas e que são extremamente importantes para que ele tenha as capacidades necessárias e desejadas. Por isso, acompanhe, a seguir, outras estruturas que definem comportamentos especiais no pseudocódigo.

## ESTRUTURA DE DECISÃO

A estrutura de decisão é muito importante para implementar comportamentos de fluxos em programas. A partir de um ou mais testes encadeados (por operadores lógicos E ou OU) na estruturas de decisão, é viável fornecer dois ou mais possíveis caminhos para os programas.

A estrutura-base para uma única decisão (com uma ou várias condições) é a seguinte:

```
se <expressão-lógica> entao
    <sequência-de-comandos>
fimse
```

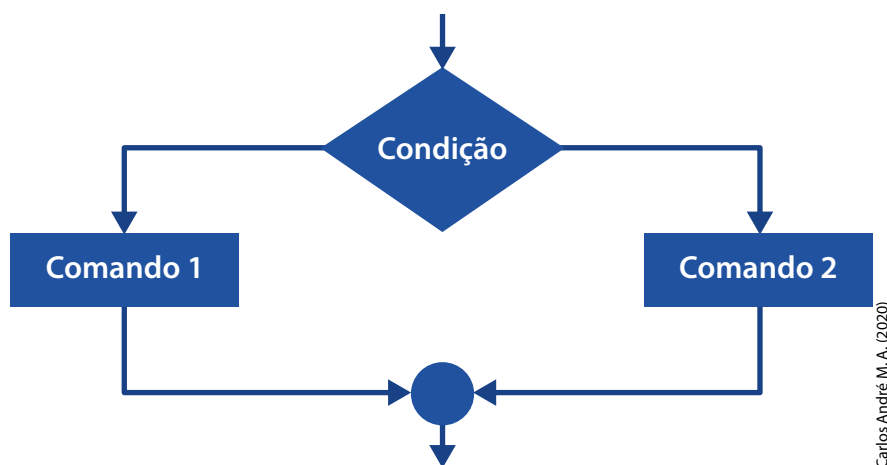
Carlos André M. A.  
(2020)

É possível também definir qual o fluxo, caso o teste não seja satisfatório, ou seja, caso a condição do SE resultar em um valor falso, fornecendo o caminho do SENAO:

```
se <expressão-lógica> entao
    <sequência-de-comandos-1>
senao
    <sequência-de-comandos-2>
fimse
```

Carlos André M. A. (2020)

Assim, é possível adicionar múltiplas estruturas de decisões alternativas em cada SENAO, colocando estruturas de decisões na estrutura de decisões, além de criar uma árvore de decisões hierárquicas. Ao reproduzir essa estrutura de decisão em uma forma gráfica de fluxogramas, tem-se:



Carlos André M. A. (2020)

Figura 3 - Fluxograma SE e SENAO  
Fonte: do Autor (2020)

Imagine, por exemplo, que se pretende criar um controlador de acesso por restrição de idade, ou seja, se as pessoas forem maiores de idade (18 anos ou mais), serão encaminhadas para a sala dos adultos; e caso elas sejam menores de idade (qualquer idade inferior a 18 anos), serão encaminhadas para a sala das crianças.

```

1 Algoritmo "EstruturaSE 01"
2
3 Var
4   idade : inteiro
5 Inicio
6   escreva("Digite sua idade")
7   leia(idade)
8
9   SE (idade >= 18) ENTÃO
10    escreva("Encaminhado para Sala dos Adultos")
11  SENÃO
12    escreva("Encaminhado para Sala das Crianças")
13  FIMSE
14
15 Fimalgoritmo
  
```

Unknown (20--?)

Figura 4 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura SE com SENAO  
Fonte: do Autor (2020)

Observe que a estrutura SE usou apenas um teste lógico (linha 9 do código) e depois tratou todos os outros possíveis resultados para idade numa condição única do SENAO (linha 11). Mas, é possível melhorar ainda mais esse pseudocódigo, adicionando condições para identificar adolescentes, crianças e também idosos (este é apenas um exemplo para ilustrar estruturas de decisão, pois, obviamente, todas as pessoas com 65 anos ou mais são também adultos).

```

1 Algoritmo "EstruturaSE 02"
2
3 Var
4   idade : inteiro
5 Inicio
6   escreva("Digite sua idade")
7   leia(idade)
8
9   SE (idade >= 18) E (idade < 65) ENTAO
10    escreva("Encaminhado para Sala dos Adultos")
11  SENAO
12    SE (idade < 18) E (idade >= 13) ENTAO
13     escreva("Encaminhado para Sala dos Adolescentes")
14   SENAO
15     SE (idade < 13) ENTAO
16      escreva("Encaminhado para Sala das Crianças")
17    SENAO
18      escreva("Encaminhado para Sala dos Idosos")
19    FIMSE
20  FIMSE
21
22 FIMSE
23
24 Fimalgoritmo

```

Figura 5 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura SE SENA SE  
Fonte: do Autor (2020)

## ESTRUTURA DE ESCOLHA

Existe outra estrutura bem interessante que pode determinar o caminho que o fluxo do programa pode adotar. É a estrutura de escolha a partir de determinados valores predefinidos pelo programador. Utiliza-se essa estrutura principalmente quando há um bom controle das possibilidades de escolha que o usuário (ou sistema) poderá fornecer.

A estrutura utilizada na escolha é basicamente um agrupamento de possíveis resultados (casos), dependendo do que for satisfatório em uma expressão lógica de seleção.

```

escolha <expressão-de-seleção>
caso <escolha1>
    <sequência-de-comandos-1>
caso <escolha2>
    <sequência-de-comandos-2>
...
outrocaso
    <sequência-de-comandos-extra>
fimescolha

```

Carlos André M. A. (2020)

Observe que é possível ter quantos casos forem necessários e que, por fim, é fornecido um caso genérico (outrocaso) para tratar a situação em que nenhuma das alternativas anteriores seja satisfatória.



Figura 6 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ESCOLHA  
Fonte: do Autor (2020)

No caso de se executar o programa da figura anterior, se a pessoa digitar o número 1, será executado o comando da linha 12. Caso digite o número 2, o resultado será o que está na linha 14 e, caso forneça o número 3 na digitação, o comando que será escolhido será o da linha 16. Por fim, caso qualquer outro número seja digitado (exceto 1, 2 e 3), será executado o comando da linha 18.

Note que, nessa estrutura, o programador necessita conhecer previamente os possíveis casos de resultado a partir da expressão fornecida. Então, é uma estrutura para ser utilizada quando a lógica de negócio desenvolvida está mais definida ao domínio dos resultados comuns. Alguns bons exemplos são:

- Dias da semana;
- Meses do ano;
- Horas do dia;
- Tipos de produto em uma lanchonete (bebida, lanche ou porção).

Em todos esses exemplos anteriores, sabe-se de forma muito mais controlada quais os possíveis resultados, pois é de conhecimento prévio que os dias da semana vão de domingo até sábado, os meses do ano vão de 1 até 12 e as horas do dia vão de 00 até 23. No caso do último exemplo, os tipos de produtos serão determinados pela análise de requisitos que o sistema precisa atender, o que será feito em um momento anterior ao da escrita do programa.

## ESTRUTURA DE REPETIÇÃO PARA

E, se for necessário que um programa execute um determinado comando ou um conjunto de comandos diversas vezes seguidas? Nesse caso, é pertinente utilizar estruturas de repetição, que possuem a característica de executar diversas vezes uma sequência de comandos a partir de uma estratégia (limite ou condição). As estruturas de repetição são também chamadas de laços, uma vez que permitem que um mesmo trecho de código seja executado em ciclos (e estes são também chamados de iterações).

No caso da estrutura de repetição PARA, repete-se um comando (ou conjunto de comandos) a partir de um valor de referência inicial até um valor de referência final, utilizando, para tanto, uma variável:

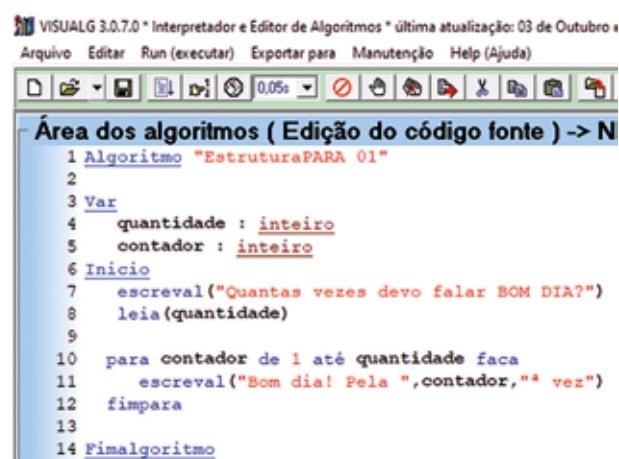
```
para <variável> de <valor-inicial> ate <valor-limite> faca
    <sequência-de-comandos>
fimpara
```

Carlos André M.  
A. (2020)

Para entender melhor, imagine um programa em que o usuário irá informar um número inteiro positivo qualquer X, e o sistema vai escrever X mensagens de “Bom dia!”, até chegar à mensagem “Bom dia! Pela Xª vez”. Ou seja, se o usuário informar o número 3, a saída esperada seria algo como:

- a) “Bom dia! Pela 1ª Vez”;
- b) “Bom dia! Pela 2ª Vez”;
- c) “Bom dia! Pela 3ª Vez”.

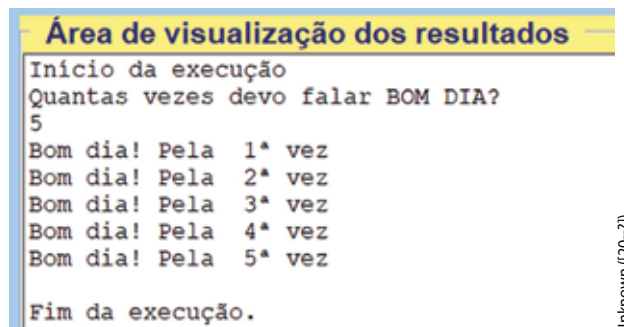
Observe esse algoritmo em Portugol.



Unknown (20--?)

Figura 7 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura PARA  
Fonte: do Autor (2020)

Quando executado o programa da imagem anterior, tem-se o seguinte resultado:



```

Área de visualização dos resultados
Início da execução
Quantas vezes devo falar BOM DIA?
5
Bom dia! Pela 1ª vez
Bom dia! Pela 2ª vez
Bom dia! Pela 3ª vez
Bom dia! Pela 4ª vez
Bom dia! Pela 5ª vez
Fim da execução.
  
```

Figura 8 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura PARA (Resultado)  
Fonte: do Autor (2020)

Essa estrutura de repetição é muito utilizada no desenvolvimento de sistemas, principalmente quando se tem uma determinada coleção de dados em uma lista, ou matriz, e se pretende efetuar um processamento para cada item existente nessa coleção. Você já imaginou quão trabalhoso seria para um programador ter que atualizar uma informação para cada usuário de um determinado sistema se esse tivesse, por exemplo, mais de mil usuários? Ficaria inviável ele ter que efetuar um comando para cada usuário existente. Porém, com o recurso de repetição, ele poderia fazer, em poucas linhas, uma estrutura PARA que iria de 1 até 1000 repetindo o código de atualização que ele quer.

## ESTRUTURA DE REPETIÇÃO ENQUANTO

Existem outras estruturas de repetição, como a estrutura ENQUANTO. Essa estrutura, ao contrário do PARA, não irá produzir uma repetição até um determinado limite de expressão, mas produzirá uma repetição enquanto uma determinada condição for verdadeira. Ou seja, enquanto aquela expressão lógica for satisfatória (resultar em um valor verdadeiro), o programa irá executar a sequência de comandos determinada, mas, quando a expressão não for mais satisfatória, o programa irá parar de executar essa repetição.

```

enquanto <expressão-lógica> faca
    <sequência-de-comandos>
fimenquanto
  
```

Carlos André M. A.  
(2020)

Como exemplo, acompanhe a seguinte situação: imagine que se tem um robô, que vai caminhar enquanto você digitar o número 1 e ele só vai parar de andar caso você informe qualquer número diferente do número 1. Nesse caso, não se sabe previamente quantos metros o robô irá andar, como seria no caso da estrutura de repetição PARA. Na estrutura ENQUANTO, somente se sabe que ele irá repetir alguma ação enquanto uma determinada situação for verdadeira, podendo, inclusive, andar infinitamente se, nesse caso, sempre estiver o número 1 acionado.



```

1 Algoritmo "EstruturaENQUANTO 01"
2
3 Var
4   metros : inteiro
5   numero : inteiro
6 Inicio
7   metros <- 0
8   numero <- 1
9   enquanto (numero = 1) faça
10    metros <- metros + 1
11    escreval("O Robô andou ",metros," metros.")
12    escreval("Digite 1 para ele continuar a andar:")
13    leia(numero)
14   fimenquanto
15
16 Fimalgoritmo
  
```

Figura 9 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ENQUANTO  
Fonte: do Autor (2020)

Agora veja como ficaria o resultado da execução do código anterior. Note que no exemplo da execução a seguir, por 3 vezes é digitado o número 1 e, na última vez, é inserido o número 0, dando fim à repetição.

```

Área de visualização dos resultados
Digite 1 para ele continuar a andar:
1
O Robô andou  2 metros.
Digite 1 para ele continuar a andar:
1
O Robô andou  3 metros.
Digite 1 para ele continuar a andar:
0
Fim da execução.
  
```

Figura 10 - Tela do VisualG – Pseudocódigo Exemplo de Estrutura ENQUANTO (Resultado)  
Fonte: do Autor (2020)

## APLICAÇÃO NA INDÚSTRIA

Com tantas estruturas apresentadas, fica muito fácil perceber a aplicação no universo do setor produtivo industrial. Imagine, por exemplo, que, em uma determinada indústria, há uma grande esteira que executa a verificação de 10 peças, classificando-as a partir de uma câmera que é capaz de filmar a peça e gerar um código de 1 a 3, dependendo do tipo da peça. No final, o objetivo é saber quantas peças de cada tipo passaram pela esteira.

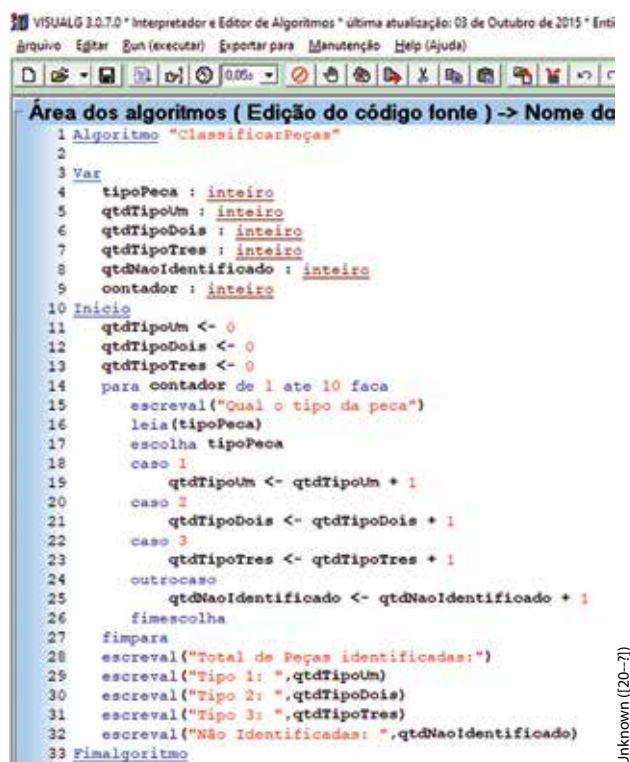
Perceba que, nesse exemplo, a indústria não sabe ao certo quantas peças de cada tipo irão atravessar a esteira, da mesma forma que não sabe a sequência. A única informação que ela possui é que serão 10 itens, e que as possibilidades serão entre 3 tipos de peças. Parece complexo? Mas, isso pode ser resolvido

usando apenas duas das estruturas já apresentadas até agora: a estrutura de repetição PARA e a estrutura ESCOLHA.

Obviamente, para este exemplo, foi diminuída bastante a escala de quantidade de peças, que poderia ser um número muito maior (10 mil, talvez) e a forma de identificação do tipo da peça, que, ao invés de usar uma câmera com inteligência artificial para reconhecimento de padrões de imagem, é implementada por meio da digitação do usuário. Porém, imagine que, no lugar de um usuário que digitará um valor entre 1 e 3, um outro sistema estará fazendo esse mesmo papel de identificar o tipo da peça e gerar o input com valores entre 1 e 3 para o sistema.

Da mesma forma, pode-se imaginar que, no início da execução, será informada a quantidade total de peças colocada na esteira para identificação, em vez de definir apenas 10 peças. Salvo essas proporções de escala e de integração com outros sistemas, fica fácil vislumbrar a aplicação dessas estruturas em um universo produtivo real.

Observe agora a implementação em Portugol.



```

1 Algoritmo "ClassificarPeças"
2
3 Var
4   tipoPeca : inteiro
5   qtdTipoUm : inteiro
6   qtdTipoDois : inteiro
7   qtdTipoTres : inteiro
8   qtdNaoIdentificado : inteiro
9   contador : inteiro
10 Inicio
11   qtdTipoUm <- 0
12   qtdTipoDois <- 0
13   qtdTipoTres <- 0
14   para contador de 1 ate 10 faca
15     escreva("Qual o tipo da peça")
16     leia(tipoPeca)
17     escolha tipoPeca
18     caso 1
19       qtdTipoUm <- qtdTipoUm + 1
20     caso 2
21       qtdTipoDois <- qtdTipoDois + 1
22     caso 3
23       qtdTipoTres <- qtdTipoTres + 1
24     outrocaso
25       qtdNaoIdentificado <- qtdNaoIdentificado + 1
26     fimsecolha
27   fimpara
28   escreva("Total de Peças identificadas:")
29   escreva("Tipo 1: ", qtdTipoUm)
30   escreva("Tipo 2: ", qtdTipoDois)
31   escreva("Tipo 3: ", qtdTipoTres)
32   escreva("Não Identificadas: ", qtdNaoIdentificado)
33 Fimalgoritmo
  
```

Figura 11 - Tela do VisualG – Pseudocódigo Contador de Tipo de Peças  
Fonte: do Autor (2020)

Acompanhe agora a análise de algumas linhas do código anterior, proposto na figura anterior.

LINHA	RESULTADO NO JAVA - IDE ECLIPSE
4	Declaração da variável <b>tipoPeca</b> utilizada para receber o <i>input</i> da leitura da câmera, classificando as peças de 1 a 3, ou qualquer outro número, no caso de inválido.
5	Declaração da variável <b>qtdTipoUm</b> para a contagem de peças do tipo 1 encontradas na esteira.
6	Declaração da variável <b>qtdTipoDois</b> para a contagem de peças do tipo 2 encontradas na esteira.
7	Declaração da variável <b>qtdTipoTres</b> para a contagem de peças do tipo 3 encontradas na esteira.
8	Declaração da variável <b>qtdNaoidentificado</b> para a contagem de peças não identificadas na esteira.
9	Declaração da variável <b>contador</b> utilizada na estrutura do PARA, iniciando em 1 até um limite de 10.
11 até 13	Iniciamos as variáveis <b>qtdTipoUm</b> , <b>qtdTipoDois</b> e <b>qtdTipoTres</b> com o valor 0.
14 até 27	Bloco da estrutura <b>PARA</b> , que vai repetir por 10 vezes a execução do código que estiver entre a linha 15 até a linha 26.
15 e 16	Simulação de <i>input</i> da câmera, informando o número de classificação do <b>tipoPeca</b> .
17 até 26	Bloco da estrutura <b>ESCOLHA</b> , que, dependendo do número informado na variável <b>tipoPeca</b> , irá incrementar uma das 4 alternativas de variáveis de controle de quantidades: <b>qtdTipoUm</b> , <b>qtdTipoDois</b> , <b>qtdTipoTres</b> ou <b>qtdNaoidentificado</b> .
28 até 32	Saídas finais do sistema, informando o valor total de contagem de peças identificadas (nos respectivos tipos) e quantidade de peças não identificadas.

Tabela 1 - Interpretação das linhas de código em Portugol da Pseudocódigo Contador de Tipo de Peças  
Fonte: do Autor (2020)

## EXEMPLOS

Neste tópico, que apresentou questões interessantes e pertinentes ao universo da programação, serão apresentados mais alguns exemplos que fortalecerão o aprendizado relacionado ao conhecimento da elaboração de pseudocódigos.

Imagine um sistema que irá verificar determinadas condições climáticas e financeiras de uma pessoa. Dependendo das respostas fornecidas, significará que ela continuará determinada em se divertir em um parque de diversões e que só irá embora se chover ou se acabar o dinheiro.

```

1 Algoritmo "FicarNoParque"
2
3 Var
4   chovendo : logico
5   dinheiro : logico
6   modificou : caractere
7 Inicio
8   chovendo <- falso
9   dinheiro <- verdadeiro
10  enquanto (chovendo = falso) E (dinheiro = verdadeiro) faca
11      escreval("A pessoa está se divertindo no Parque")
12      escreval("Agora está chovendo? S ou N")
13      leia(modificou)
14      SE (modificou = "S") ENTÃO
15          chovendo <- verdadeiro
16          escreval("Começou a Chover! A Pessoa vai embora")
17      FIMSE
18      escreval("Acabou o Dinheiro? S ou N")
19      leia(modificou)
20      SE (modificou = "S") ENTÃO
21          dinheiro <- falso
22          escreval("Acabou o Dinheiro! A Pessoa vai embora")
23      FIMSE
24  fimenquanto
25 Fimalgoritmo
  
```

Figura 12 - Tela do VisualG – Pseudocódigo Ficar no Parque ou ir embora  
Fonte: do Autor (2020)

Neste caso, utilizou-se as estruturas ENQUANTO e SE, que irão ser responsáveis por basicamente dois comportamentos nesse código:

- Repetir infinitamente o código enquanto não estiver chovendo ou enquanto a pessoa continuar tendo dinheiro, sendo que no momento que começar a chover ou acabar dinheiro a pessoa irá embora.
- Cuidar das condições que irá testar e modificar as variáveis que controlam a estrutura de repetição, testando a condição **se chove** ou **se acaba o dinheiro** em duas estruturas.

Observe no programa que, caso se informe que começou a chover, será apresentada a mensagem “Começou a chover! A Pessoa vai embora”. Mesmo assim, será ainda solicitada resposta para a questão “Acabou o Dinheiro?”. Perceba que não há necessidade desta questão ser realizada, pois já está decidido que a pessoa vai embora por causa da chuva. Pense no que precisa ser alterado no algoritmo para evitar que esta pergunta seja feita nesse caso.

**SAIBA  
MAIS**

A importância do pseudocódigo é estruturante, pois ele fundamenta o pensamento lógico e fortalece os conhecimentos nas estruturas básicas existentes em praticamente todas as linguagens de programação. Aliás, esse tem sido um dos grandes diferenciais para programadores que conseguem efetivamente evoluir e migrar para inúmeras linguagens de programação e, conseqüentemente, crescer em suas carreiras, em relação àqueles que meramente se aventuram focados em uma linguagem de programação que esteja na moda.

Quando o profissional possui bom embasamento em lógica de programação, e o pseudocódigo pode ajudar nisso, ele possui maior facilidade para ler e compreender as estruturas utilizadas nos blocos de código, funções, procedimentos, classes, objetos etc. É como comparar se você tentar aprender a falar uma nova língua estrangeira, como o inglês, por exemplo. Uma coisa é você decorar algumas frases e palavras em inglês, outra completamente diferente é você compreender profundamente as estruturas da sintaxe da língua inglesa, suas regras e aplicações.

Aprofunde seus conhecimentos em pseudocódigos e terá benefícios nessa fase inicial no universo da programação. Pesquise em sites e fóruns um pouco mais sobre Portugal. Exercite os exemplos apresentados nessas outras fontes de informação. Além disso, pratique e invente novos desafios. Isso pode ser determinante na construção do seu perfil profissional no futuro.



## REFERÊNCIAS

---

SOUZA, Marco Antonio de; GOMES, Marcelo Marques; SOARES, Márcio José; CONCILIO, Ricardo. **Algoritmos e lógica de programação**. São Paulo (SP): Thomson Pioneira, 2005.

ARAÚJO, Everton Coimbra de. **Algoritmos**: fundamentos e pratica. Florianópolis: Visual Books, 2005.

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo de. **Algoritmos**: lógica para desenvolvimento de programação de computadores. 24. ed. São Paulo (SP): Érica, 2011.





*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**