

# Apostila

## Algoritmos e Lógica de Programação Estruturas de Controle

Professor Eros Schettini Roman  
IFSP-Catanduva

**1 – Introdução**

Durante milhares de anos procuramos entender como pensamos. Como um mero punhado de matéria pode perceber, compreender, prever e manipular um mundo muito maior? (Russel e Norvig)

Como a mente humana trabalha por dentro? Por meio de introspecção ou experimentos psicológicos.

A partir dessas questões, começamos nosso estudo da Lógica, ciência que se originou a partir dos filósofos da Grécia antiga, por volta do século IV antes de Cristo. Seu início aconteceu do princípio filosófico de que as leis do pensamento deveriam governar as operações da mente.

Os filósofos tornaram a Lógica concebível, considerando as ideias de que a mente é, em alguns aspectos, semelhante a uma máquina, que opera sobre o conhecimento codificado em alguma linguagem interna e que o pensamento pode ser usado para escolher as ações que deverão ser executadas.

Como o cérebro habilita o pensamento ainda é um mistério para a ciência.

Observações indicavam que o cérebro estava envolvido no pensamento. Pancadas na cabeça levavam à incapacitação mental.

Aristóteles escreveu: "De todos os animais, o homem é o que tem o maior cérebro em proporção ao seu tamanho".

Aristóteles foi o primeiro filósofo grego a tentar codificar o pensamento por meio de processos de raciocínio irrefutáveis. Ele é o criador do conceito de Silogismo, que é uma argumentação onde existem premissas, que são afirmações verdadeiras, gerando conclusões corretas. Exemplo:

"Sócrates é um homem" e "Todo homem é mortal" são premissas (ou afirmações verdadeiras).

"Sócrates é mortal" é uma conclusão válida.

Essas leis propostas por Aristóteles deram origem a um campo de estudo chamado **Lógica**.

O desenvolvimento da lógica formal aconteceu entre os séculos 19 e 20 e ela fornece uma notação precisa para declarar sobre todos os tipos de coisas do mundo e as relações entre elas.

Essas ideias iniciais são importantes para a contextualização da abrangência do campo de estudo relativo à Lógica. Ela é uma ciência que envolve outras ciências, tais como: Filosofia, Matemática, Neurociência, Psicologia, etc..

## **2 – O Que é Lógica?**

***"É a disciplina que trata das formas de pensamento,  
da linguagem descritiva do pensamento,  
das leis de argumentação e do raciocínio correto,  
dos métodos e dos princípios que regem o pensamento humano.  
Portanto, não se trata somente de uma arte,  
mas também de uma ciência."  
Bastos e Keller***

A lógica envolve os processos de raciocínio utilizados na resolução dos problemas. O correto uso das leis que regem o nosso pensamento proporcionam soluções adequadas e eficientes para os mais diversos tipos de situações do mundo real. Coerência e racionalidade são quesitos fundamentais na solução de problemas estando profundamente relacionados com a lógica.

Veja algumas definições para lógica:

- ⤴ "Correção do pensamento/raciocínio";
- ⤴ "Arte de bem pensar";
- ⤴ "Ciência das formas do pensamento";
- ⤴ "Ordem da razão".

Usamos constantemente a lógica no nosso cotidiano, para decidirmos qual o melhor caminho até um dado destino, para trocar um pneu, etc.

### 3 – Lógica de Programação

Lógica de programação é a técnica de organizar pensamentos para atingir determinado objetivo.

O nosso objetivo em usar a lógica é a criação de programas de computadores que usem técnicas de desenvolvimento coerentes e válidas na solução de problemas



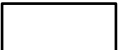
### 4 – Ferramentas para Desenvolvimento





Para o desenvolvimento de programas computacionais devemos usar alguma ferramenta teórica para a organização lógica do nosso raciocínio na resolução dos problemas.

A linguagem de representação é chamada de pseudo-código, que pode ser expressado de várias maneiras. As duas mais utilizadas são o Algoritmo e o Fluxograma. Podemos representar o pseudo-código por meio de uma descrição narrativa, que não é muito comum e prática no desenvolvimento de programas de computadores.

#### 4.1 - Fluxograma

Fluxogramas ou Diagramas de Blocos são representações gráficas que indicam o fluxo de execução de um programa. Existem algumas figuras que representam as ações que serão executadas:

	Representa o início ou fim de programa.
	Indica o sentido do fluxo de dados, conectando as figuras.
	Processamento, cálculo ou atribuição de valores.

	Indica leitura ou entrada de dados. Pode ser utilizado também como saída, substituindo a próxima figura
	Representa saída de dados.
	Indica a possibilidade de uma tomada de decisão, um desvio no programa.
	Conexão

### Exemplo

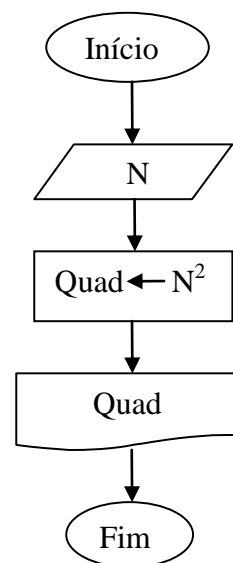
Início do fluxograma

Entrada do número

Cálculo do quadrado do número

Apresentação do resultado

Fim do fluxograma



## 4.2 - Algoritmo

Um algoritmo é formalmente uma sequência finita de passos que levam a execução de uma tarefa. Podemos pensar em algoritmo como uma receita, uma sequência de instruções que executam uma meta específica. Estas tarefas devem ser objetivas, com precisão e clareza.

Algoritmos são realizados cotidianamente em nossas ações: escovar os dentes, tomar um banho, trocar um pneu ou lâmpada, uma receita de bolo, etc. Todas essas atividades requerem uma organização de ações no sentido de atingir o objetivo estabelecido, seja ele qual for. Na lógica de programação, os algoritmos têm vital importância no desenvolvimento inicial dos estudos, pois é uma ferramenta teórica de fácil aprendizado e proporciona um desprendimento dos detalhes computacionais inerentes às linguagens de programação. Exemplos:

Algoritmo de Escovar os dentes:

- Pegar a escova

- Abrir a torneira
- Lavar a escova
- Fechar a torneira
- Pegar a pasta
- Abrir a pasta
- Passar a pasta na escova
- Fechar a pasta
- Escovar os dentes
- Abrir a torneira
- Enxaguar a boca e a escova
- Fechar a torneira
- Secar a boca e guardar a escova e a pasta

Calcular a área de um retângulo:

- Informar as medidas dos dois lados do retângulo (leia l1, l2)
- Calcular a área do retângulo (área  $\leftarrow l1 * l2$ )
- Mostrar o resultado (Escreva área)

Ambos os exemplos e conseqüentemente qualquer algoritmo podem ser resolvidos de maneiras diferentes, seguindo uma organização particular para uma determinada solução. O que importa é realizar passos de maneira coerente e atingir o objetivo de maneira eficiente.

### **Regras para Construção de um Algoritmo**

Para escrever um algoritmo precisamos descrever a sequência de instruções, de maneira simples e objetiva. Para isso utilizaremos algumas técnicas:

- Usar somente um verbo por frase e no infinitivo ou no imperativo
- Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática
- Usar frases curtas e simples
- Ser objetivo

Um algoritmo normalmente tem três atividades principais a serem desenvolvidas: **entrada, processamento e saída dos dados**.

### **Método para Construção de Algoritmos**

Os passos necessários para a construção de um algoritmo são:

- ler atentamente o enunciado do problema, compreendendo-o e destacando os pontos mais importantes;
- definir os dados de entrada, ou seja, quais dados serão fornecidos;
- definir os dados de saída, ou seja, quais dados serão gerados depois do processamento;
- definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela obtenção dos dados de saída com base nos dados de entrada;
- definir as variáveis necessárias para armazenar as entradas e efetuar o processamento;
- elaborar o algoritmo;
- testar o algoritmo realizando simulações.

### **Tipos de Algoritmos**

Sabemos que algoritmos são nada mais que regras formais que levam à obtenção de uma solução ou resultado de um problema, portanto, qualquer instrução representada de alguma forma e que conduza a algo, pode ser chamada de algoritmo.

Os tipos de algoritmos mais usados são:

- Descrição Narrativa
- Fluxograma
- Pseudocódigo

#### **Descrição Narrativa**

Consiste em analisar a descrição do problema e escrever em uma linguagem natural o passo a passo a ser seguido para a resolução do problema.

A vantagem da utilização desse tipo de algoritmo está na facilidade, pois usa uma língua natural, como o português.

A desvantagem é que, a língua natural é passível de interpretações diferentes, o que pode dificultar na transcrição do algoritmo para o programa.

#### **Fluxograma**

Um dos tipos de algoritmos mais usados que consiste em analisar a descrição do problema e escrever o passo a passo para sua solução através de símbolos gráficos pré-definidos.

A vantagem do uso desse tipo de algoritmo está no entendimento facilitado pela utilização de elementos gráficos.

#### **Pseudocódigo ou português**

Consiste em analisar a descrição do problema e escrever por meio de regras predefinidas o passo a passo a ser seguido para a resolução do problema.

Trata-se de um dos tipos de algoritmos que mais se aproxima da forma escrita das linguagens de programação, por isso, sua transcrição é bem mais rápida.

É nesse tipo de algoritmo que os nossos estudos irão se focar.

### **4.3 - Linguagem de Programação C**

Para exemplificar de maneira prática nossos exemplos utilizaremos a Linguagem de Programação C. Esta linguagem é utilizada há mais de 40 anos no ensino de programação, nos mais diversos níveis de estudo, oferecendo comandos para a programação estruturada. O intuito não é aprender todos os seus recursos, mas sim utilizá-la como apoio ao ensino das principais estruturas de programação.

## **5 - Conceitos Básicos**

### **Operadores Lógicos**

Atuam sobre expressões retornando sempre valores lógicos como Falso ou Verdadeiro.

<b>E</b>	RETORNA VERDADEIRO SE AMBAS AS PARTES FOREM VERDADEIRAS.
<b>OU</b>	BASTA QUE UMA PARTE SEJA VERDADEIRA PARA RETORNAR VERDADEIRO.
<b>NÃO</b>	INVERTE O ESTADO, DE VERDADEIRO PASSA PARA FALSO E VICE-VERSA.

### **Tabela Verdade**

A	B	A e B	A ou B	não (A)
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

### Variáveis

Variáveis são elementos que são utilizados para armazenar valores de maneira temporária. Tais valores podem ser mudados ao longo da execução do programa.

Ao definimos uma variável, a linguagem de programação juntamente com o sistema operacional da máquina reservam um espaço na memória do computador suficiente para armazenar valores correspondentes ao tamanho da variável, que é definido com base no tipo da variável.

### Tipo de uma Variável

O tipo de uma variável determina qual é o conjunto de valores que uma determinada variável pode receber, uma espécie de domínio da variável. Os tipos considerados primitivos são: inteiro, real, texto e lógico ou booleano.

**Inteiro:** pode armazenar valores negativos e positivos correspondentes aos números inteiros da matemática. Exemplos: no Pascal é *integer* e no C é *int*.

**Real:** armazena valores em ponto flutuante ou decimal, podendo armazenar valores negativos e positivos dentro dos números reais. Exemplos: no Pascal é *real* e no C *float* ou *double*.

**Texto:** pode conter valores chamados alfanuméricos, que são letras, caracteres especiais, símbolos e números, misturados ou não. Exemplos: no Pascal é *string* e no C é *char*.

**Lógico:** tipo utilizado para expressar valores lógicos ou booleanos, que possuem apenas dois valores: verdadeiro (*true*) ou falso (*false*). No Pascal é *boolean* e no C não existe.

### Identificadores

Os identificadores são os nomes que os programadores dão às variáveis, constantes, programas, subprogramas, unidades, etc. Sugere-se sempre que os identificadores tenham nomes adequados e apropriados para as situações de programa e que possam apontar de imediato o significado do objeto do



programa em questão. Temos algumas regras básicas de formação de identificadores que normalmente devemos seguir:

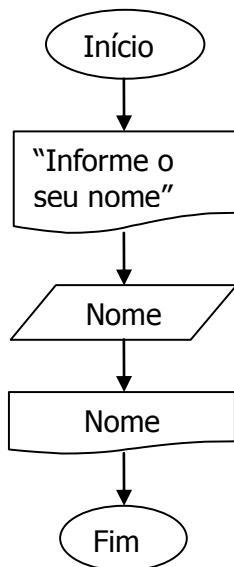
- Os caracteres utilizados são: números, letras maiúsculas e minúsculas e o caractere de sublinhado;
- O primeiro caractere deve ser sempre uma letra ou caractere sublinhado;
- Não são permitidos espaços em branco nem caracteres especiais (?, #, %, @, \*);
- As palavras reservadas de uma linguagem também não podem ser utilizadas para esse fim.

## 6 – Estrutura de Controle Sequencial

Sequencial ou sequenciamento é uma estrutura de controle de fluxo de execução de um algoritmo ou programa. Ela indica que um comando vem após outro, ordenados de maneira linear, de cima para baixo, no algoritmo. Nas principais linguagens de programação o que indica sequenciamento é o ponto-e-vírgula (;).

Exemplo 1: crie um fluxograma e um programa em C para receber o nome de uma pessoa e, em seguida, mostrá-lo.

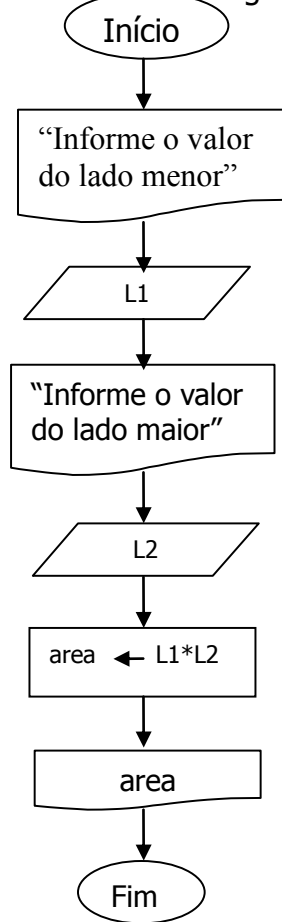
Fluxograma



Programa em C

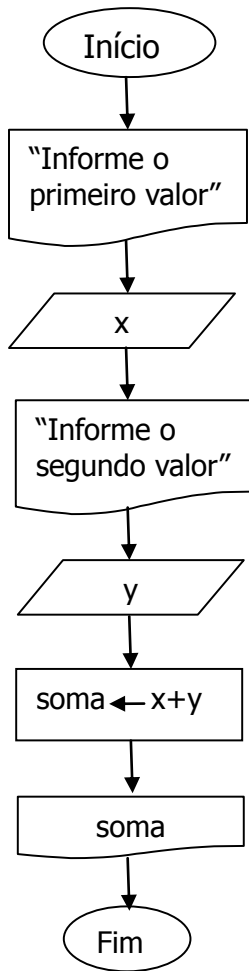
```
#include <stdio.h>
int main()
{
    char nome[15];
    printf("Informe o seu nome:");
    scanf("%s", &nome);
    printf("O seu nome eh: %s\n", nome);
    system("pause");
    return 0;
}
```

Exemplo 2: crie um fluxograma e um programa em C que recebe a medida dos lados de um retângulo e calcule e mostre a sua área.



```
#include <stdio.h>
int main()
{
    float L1, L2, area;
    printf("Informe o valor do lado menor: ");
    scanf("%f", &L1);
    printf("Informe o valor do lado maior: ");
    scanf("%f", &L2);
    area = L1 * L2;
    printf("A área do retângulo é: %6.2f\n", area);
    system("pause");
    return 0;
}
```

Exemplo 3: crie um fluxograma e um programa em C para receber dois valores do usuário e calcular a soma deles.



```
#include <stdio.h>
#include <locale.h>
int main()
{
    int x, y, soma;
    setlocale(LC_ALL, "Portuguese");
    printf("Informe o primeiro valor:");
    scanf("%d", &x);
    printf("Informe o segundo valor:");
    scanf("%d", &y);
    soma = x + y;
    printf("A soma de %d e %d é: %d\n", x, y, soma);
    system("pause");
    return 0;
}
```

### Exercícios

1 - Faça um fluxograma e um programa em C que leia dois valores e faça as quatro operações aritméticas básicas (+, -, x, /) com os dois valores, mostrando os resultados.

2 - Crie um fluxograma e um programa em C que receba três notas de um aluno, calcule e mostre a média aritmética entre elas.

3 - Crie um fluxograma e um programa em C que receba o ano de nascimento de uma pessoa e o ano atual, calcule e mostre:

- a idade dessa pessoa;
- quantos anos ela terá em 2050.

4 - Faça um fluxograma e um programa em C que receba o preço de um produto, calcule e mostre o preço reajustado, sabendo-se que este sofreu um aumento de 15%.

5 - Desenvolva um fluxograma e um programa em C que receba o preço de um produto e o percentual de desconto, calcule e mostre o valor do desconto e o novo preço do produto.

6 - Crie um fluxograma e um programa em C que receba o salário base de um funcionário, calcule e mostre o salário a receber, sabendo-se que o funcionário tem gratificação de 10% sobre o salário base e paga imposto de 8% sobre o salário base também.

7 - Faça um fluxograma e um programa em C que receba o salário base de um funcionário, calcule e mostre o seu salário a receber, sabendo-se que o funcionário tem gratificação de R\$ 50 e paga imposto de 7% sobre o salário base.

8 - Desenvolva um fluxograma e um programa em C que receba o valor de um depósito e o valor da taxa de juros, calcule e mostre o valor do rendimento e o valor total depois do rendimento.

9 - O custo ao consumidor de um carro novo é a soma do preço de fábrica com o percentual de lucro do distribuidor e dos impostos aplicados ao preço de fábrica. Crie um fluxograma e um programa em C que receba o preço de fábrica de um veículo, o percentual de lucro do distribuidor e o percentual de impostos, calcule e mostre:

- a) o valor correspondente ao lucro do distribuidor;
- b) o valor correspondente aos impostos;
- c) o preço final do veículo.

10 - Sabe-se que:

1 pé = 12 polegadas

1 jarda = 3 pés

1 milha = 1760 jardas

Faça um fluxograma e um programa em C que receba uma medida em pés, faça as conversões a seguir e mostre os resultados em:

- a) polegadas;
- b) jardas;
- c) milhas.

11 - Uma firma contrata um encanador a R\$ 100,00 por dia. Desenvolva um fluxograma e um programa em C que solicite o número de dias trabalhados pelo encanador e imprima a quantia líquida que deverá ser paga, sabendo-se que são descontados 8% para imposto de renda.

12 – Sabendo-se que o latão é obtido fundindo-se sete partes de cobre com três partes de zinco, crie um fluxograma e um programa em C que solicite quantos quilos de latão se quer produzir e imprima quantos quilos de cobre e de zinco são necessários.

13 – A importância de R\$ 1.000.000,00 será dividida entre os dois primeiros colocados de um concurso, em partes diretamente proporcionais aos pontos conseguidos por eles. Faça um fluxograma e um programa em C que solicite o número de pontos dos dois primeiros colocados e imprima a importância que caberá a cada um.

14 – Três amigos jogam toda semana na loteria. Escreva um fluxograma e um programa em C que solicite com quanto cada um entrou em dinheiro e o valor do prêmio que deve ser rateado em partes diretamente proporcionais às quantias com que cada um entrou. O programa deve imprimir quanto receberá cada um se eles ganharem.

15 - O coração humano bate em média uma vez por segundo. Desenvolva um algoritmo para calcular e escrever quantas vezes o coração de uma pessoa baterá se viver X anos. Dado de entrada: idade da pessoa (inteiro em anos). Considerações: 1 ano = 365,25 dias, 1 dia = 24 horas, 1 hora = 60 minutos e 1 minuto = 60 segundos.

16 - Um fabricante de latas deseja desenvolver um algoritmo para calcular o custo de uma lata cilíndrica de alumínio. O custo do alumínio é R\$ 100,00 por m<sup>2</sup>. Dados de Entrada: raio e altura da lata (em metros). Área da lata = área da base ( $\pi * \text{raio}^2 * 2$ ) + área do lado ( $2 * \pi * \text{raio} * \text{altura}$ ). Considere o preço do alumínio por m<sup>2</sup> como sendo uma constante.

17 - Fazer um algoritmo para executar as seguintes conversões:

- Ler uma temperatura dada em graus *Fahrenheit* e imprimir o equivalente em Celsius.

$$C = 5/9 * (F - 32).$$

- Ler uma quantidade de chuva dada em polegadas e imprimir o equivalente em milímetros (1 polegada = 25,4 mm).

18 - Fazer um algoritmo para ler os valores dos coeficientes A, B e C de uma equação quadrática. Calcular e imprimir o valor do discriminante (delta).

$$\text{Delta} = B^2 - 4 * A * C.$$

19 - Dada a base e a altura de uma pirâmide, fazer um algoritmo que calcule e escreva o seu volume.

$$\text{Volume} = 1/3 * \text{base} * \text{altura}.$$

20 - Fazer um algoritmo que obtenha o raio e a altura de um cilindro e que calcule e escreva sua área e seu volume.

Área =  $2 \cdot \pi \cdot \text{raio} \cdot (\text{altura} + \text{raio})$  e Volume =  $\pi \cdot \text{raio}^2 \cdot \text{altura}$ .

21 - Uma locadora de charretes cobra R\$ 8,50 de taxa para cada 3 horas de uso destas e R\$ 3,50 para cada hora abaixo destas 3 horas. Fazer um algoritmo que, dado a quantidade de horas que a charrete foi usada, calcule e escreva quanto o cliente tem de pagar.

22 - Escreva um algoritmo que leia três números inteiros positivos (A, B, C) e calcule a seguinte expressão:  $D = (R + S)/2$ , onde  $R = (A + B)^2$  e  $S = (B + C)^2$ .

23 - Faça um algoritmo que leia a idade de uma pessoa expressa em anos, meses e dias e mostre-a, expressa apenas em dias.

24 - Faça um algoritmo que leia a idade de uma pessoa expressa em dias e mostre-a, expressa em anos, meses e dias.

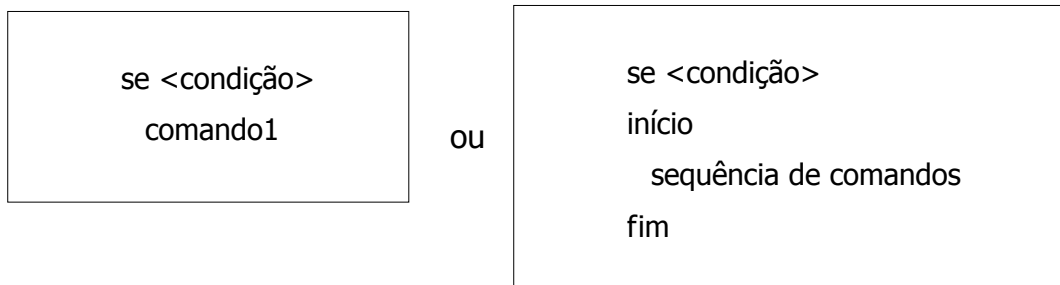
25 - Faça um algoritmo que leia o tempo de duração de um evento em uma fábrica expresso em segundos e mostre-o expresso em horas, minutos e segundos.

## 7 - Estruturas Seletivas

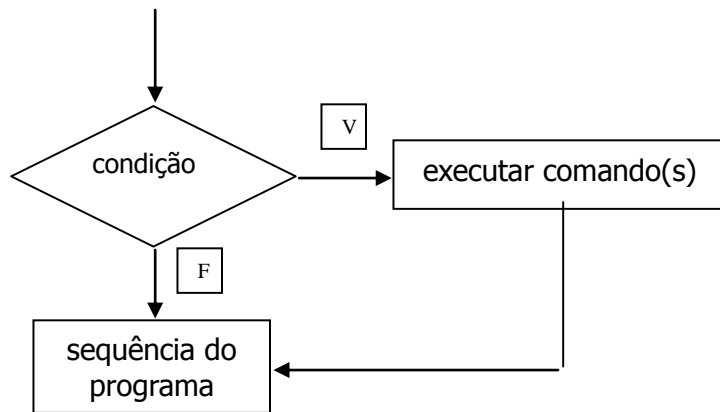
Estas estruturas também são chamadas de Estruturas Condicionais ou de Decisão. Elas são utilizadas para selecionar uma entre várias alternativas apresentadas e desviam o fluxo do programa.

### Estrutura Seletiva Simples

Sintaxe (Algoritmo):



Sintaxe (Fluxograma):



Sintaxe (Linguagem C):

```

if (condição)
    comando;
  
```

ou

```

if (condição)
{
    sequência de comandos;
}
  
```

### Exemplos

1 – Criar um algoritmo, um fluxograma e um programa na linguagem C para pedir a idade de uma pessoa e informar se ela está habilitada para dirigir um carro.

Algoritmo Habilitação1

Início

Declare idade inteiro

Escreva "Informe a sua idade"

Leia idade

Se idade  $\geq$  18

Escreva "Pode tirar habilitação"

Fim Algoritmo

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int idade;
```

```
    printf("Informe a sua idade");
```

```
    scanf("%d",&idade);
```

```
    if (idade  $\geq$  18)
```

```
        printf("Pode tirar habilitação");
```

```
    system ("pause");
```

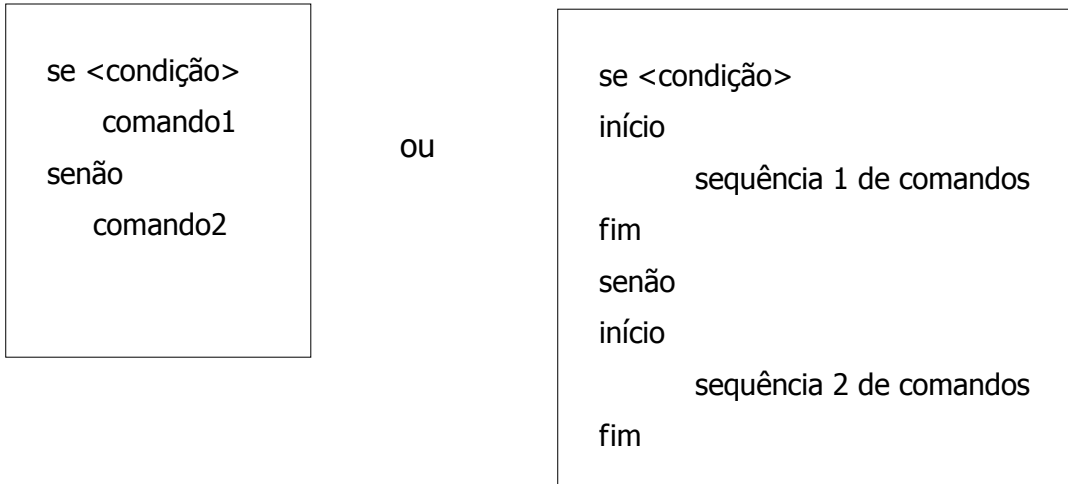
```

    return 0;
}

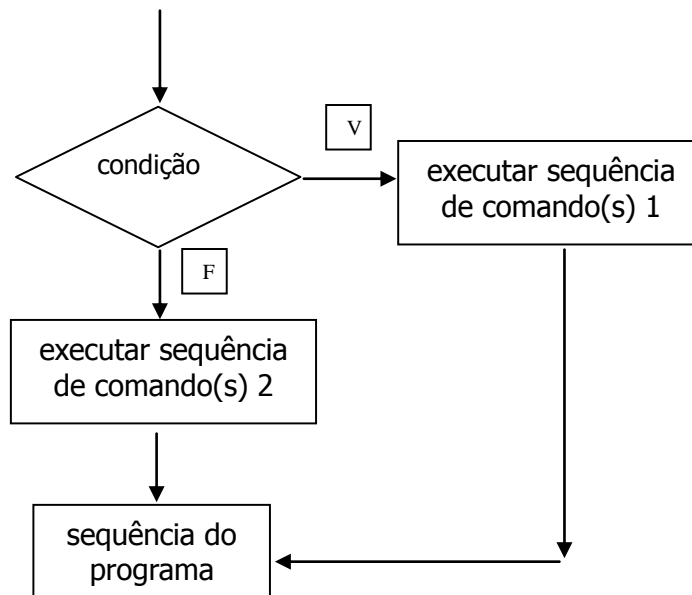
```

## Estrutura Seletiva Composta

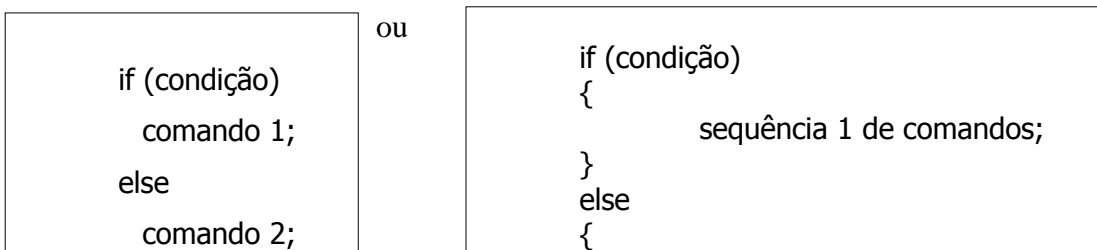
Sintaxe (Algoritmo):



Sintaxe (Fluxograma):



Sintaxe (Linguagem C):





## Exemplos

1. Criar um algoritmo e um programa na linguagem C para pedir a idade de uma pessoa e informar se ela está habilitada **ou não** para dirigir um carro.

Algoritmo Habilitação2

Declare idade inteiro

Escreva "Informe a sua idade"

Leia idade

Se idade  $\geq 18$

Escreva "Pode tirar habilitação"

senão

Escreva "Não pode tirar habilitação"

Fim Algoritmo

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int idade;
```

```
    printf("Informe a sua idade");
```

```
    scanf("%d",&idade);
```

```
    if (idade  $\geq 18$ )
```

```
        printf("Pode tirar habilitação");
```

```
    else
```

```
        printf("Não pode tirar habilitação");
```

```
    system("pause");
```

```
    return 0;
```

```
}
```

2. Criar um algoritmo e um programa na linguagem C que receba duas notas e calcule e mostre a média aritmética do aluno. Mostre também se o aluno está reprovado (média  $< 5$ ), se está de exame (média  $\geq 5$  e média  $< 7$ ) ou se está aprovado (média  $\geq 7$ ).

Algoritmo Calcula\_Media

Início

Declare Media, N1, N2 numéricos

Escreva "Informe a 1ª nota:"

Leia N1

Escreva "Informe a 2ª nota:"

Leia N2

Media  $\leftarrow (N1+N2)/2$

Escreva "Sua média foi de ", Media

```

Se Media < 5
Início
    Escreva "Reprovado"
Fim
Senão
Início
    Se Media >= 5 e Media < 7
    Início
        Escreva "Exame"
    Fim
    Senão
    Início
        Escreva "Aprovado"
    Fim
Fim
Fim Algoritmo

```

```

#include <stdio.h>
#include <stdlib.h>
int main()
{
    float Media, N1, N2;
    printf("Informe a 1ª nota:");
    scanf("%f",&N1);
    printf("Informe a 2ª nota:");
    scanf("%f",&N2);
    Media = (N1+N2)/2;
    printf("Sua média foi de %f", Media);
    if (Media < 5)
    {
        printf("Reprovado");
    }
    else
    {
        if (Media >= 5 && Media <7)
        {
            printf("Exame");
        }
        else
        {
            printf("Aprovado");
        }
    }
}

```

```

    system ("pause");
    return 0;
}

```

## Estrutura Seletiva de Múltipla Escolha

Esta estrutura é utilizada em situações onde a variável a ser testada pode ser comparada a uma lista de alvos enumeráveis com valores escalares e discretos. Certas aplicações envolvem um grande número de testes lógicos, e para tratá-los é preciso aninhamento de SE's. Existe uma estrutura que aplicada a certos casos de aninhamento produz o mesmo resultado e torna o algoritmo mais inteligível e com menos vulnerabilidade a ambigüidades. Esta estrutura é chamada de Estrutura Condicional de Múltipla Escolha.

Sintaxe:

Escolha x

Início

```

    Caso Condição 1
        Comando(s)_1
    Caso Condição 2
        Comando(s)_2
        .
        .
        .
    Caso Condição n
        Comando(s)_n
    [ Caso contrário / senão
        Comando(s)_s ]

```

Fim Escolha

## Exemplo:

Algoritmo Estado\_Civil

Declare EC inteiro

Escrever "Digite o estado civil (1 – solteiro, 2 – casado, 3 – outros)"

Ler EC

Escolha EC

Início

```

    Caso 1:
        Escrever "Solteiro"
    Caso 2:
        Escrever "Casado"
    Caso 3:

```

```

        Escrever "Outros"
    Caso Contrário:
        Escrever "Valor Inválido"
    Fim Escolha
Fim Algoritmo

#include <stdio.h>
#include <stdlib.h>
int main()
{
    int EC;
    printf("Digite o estado civil (1 – solteiro, 2 – casado, 3 até 5 –
    outros");
    scanf("%d",&EC);
    switch(EC)
    {
        case 1:    printf("Solteiro");
                  break;
        case 2:    printf("Casado");
                  break;
        case 3 ... 5: printf("Outros");
                  break;
        default:   printf("Valor Inválido");
    }
    system ("pause");
    return 0;
}

```

### **Exercícios de Estruturas de Seleção**

1. Faça um algoritmo e o programa em C que leia 3 números e escreva o maior.
2. Faça um algoritmo e o programa em C para verificar se um nº inteiro fornecido é par ou ímpar.
3. As maçãs custam R\$ 1,30 cada se forem compradas menos de uma dúzia, e R\$ 1,00 se forem compradas pelo menos 12. Escreva um algoritmo e o programa em C que leia o número de maçãs compradas, calcule e escreva o custo total da compra.
4. O custo para enviar um pacote por correio expresso é R\$ 10,00 para até um quilo e R\$ 3,75 para cada meio quilo. Se o pacote pesar mais que 35 quilos, uma taxa de peso adicional de R\$ 10,00 é adicionada ao custo. Nenhum pacote com mais de 50 quilos é aceito. Escreva um algoritmo e o programa em C que

aceite o peso de um pacote em quilos e calcule o custo de enviar o pacote ou não.

5. Um ano é bissexto se for divisível por 4 mas não por 100. Um ano é bissexto também se for divisível por 400. Faça um algoritmo e o programa em C que verifique se um ano fornecido pelo usuário é ou não bissexto.

6. A jornada de trabalho semanal de um funcionário é de 40 horas. O funcionário que trabalhar mais de 40 horas receberá hora extra, cujo cálculo é o valor da hora regular com um acréscimo de 50%. Escreva um algoritmo e o programa em C que leia o número de horas trabalhadas em um mês, o salário por hora e escreva o salário total do funcionário, que deverá ser acrescido das horas extras, caso tenham sido trabalhadas (considere que o mês possua 4 semanas exatas).

7. Elabore um algoritmo e o programa em C que calcule o que deve ser pago por um produto, considerando o preço normal de etiqueta e a escolha da condição de pagamento. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetuar o cálculo adequado.

<b>Código</b>	<b>Condição de Pagamento</b>
1	À vista em dinheiro ou cheque, recebe 10% de desconto
2	À vista no cartão de crédito, recebe 5% de desconto
3	Em duas vezes, preço normal de etiqueta sem juros
4	Em três vezes, preço normal de etiqueta mais juros de 10%

8. Faça um algoritmo que receba o salário de um funcionário, calcule e mostre o novo salário deste funcionário, acrescido da bonificação e de auxílio-escola.

<b>Salário</b>	<b>Bonificação</b>	<b>Salário</b>	<b>Auxílio-escola</b>
Até R\$ 500,00	5% do salár.	Até R\$ 600,00	R\$ 150,00
Entre R\$ 500,01 e R\$ 1200,00	12% do sal.	Mais que R\$ 600	R\$ 100,00
Acima de R\$ 1200,00	Sem bonific.		

9. Faça um programa em C que receba três números obrigatoriamente em ordem crescente e um quarto número que não siga esta regra. Mostre, em seguida, os quatro números em ordem decrescente.

10. Uma agência bancária possui dois tipos de investimentos, conforme o quadro a seguir. Faça um programa em C utilizando a estrutura *switch* que receba o tipo de investimento e o valor do investimento e calcule e mostre o valor corrigido de acordo com o tipo de investimento.

<b>TIPO</b>	<b>DESCRIÇÃO</b>	<b>RENDIMENTO MENSAL</b>
1	Poupança	3%

2	Fundos de Renda Fixa	4%
---	----------------------	----

11. Um supermercado deseja reajustar os preços de seus produtos usando o seguinte critério: o produto poderá ter seu preço aumentado ou diminuído. Para alterar o preço, o produto deve preencher um dos requisitos a seguir:

Requisitos		Reajustes	
Venda Média Mensal	Preço Atual	% de Aumento	% de Diminuição
< R\$500	< R\$ 30,00	10	---
>=R\$500 e < R\$1200	>= R\$30 e <R\$80	15	---
>=R\$1200	>= R\$80,00	---	20

Faça um programa em C que receba o preço atual e a venda mensal média do produto, calcule e mostre o novo preço.

12. Faça um programa em C que mostre o menu de opções a seguir, receba a opção do usuário e os dados necessários para executar cada operação.

Menu de Opções:

1 – Cálculo do resto do primeiro pelo segundo;

2 – Cálculo da raiz quadrada dos dois números.

Digite a opção desejada

## 8 - Estruturas de Repetição

Quando algum trecho de um programa precisa ser repetido para se alcançar um resultado temos que usar uma estrutura de repetição (laços). Podemos ter três tipos de laços:

### Estrutura de Repetição com Contador (For)

Tal estrutura tem como característica ser um laço contador, possuindo uma variável controladora que possui um valor inicial sendo incrementado ou decrementado até chegar em um valor final.

Sintaxe (algoritmo):

Para (var ← valor\_inicial; condição de parada; in(de)cremento)

Início

Sequência de comandos

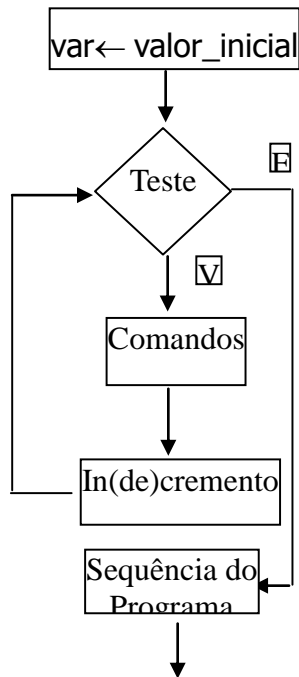
Fim

Seção 1 → Inicialização

Seção 2 → Teste Condicional

Seção 3 → In(de)cremento

Sintaxe (Fluxograma):



Sintaxe (Linguagem C):

```

for(var = valor_inicial; condição de parada; in[de]cremento)
{
    Sequência de comandos
}
  
```

Exemplo: Crie um algoritmo e um programa que faça a soma dos números entre 1 e 10.

Algoritmo Somatória1

Declare i, soma inteiros

soma ← 0

Para (i ← 1; i ≤ 10; i ← i + 1)

Início

soma ← soma + i

Fim

Escreva soma

Fim Algoritmo

```

#include <stdio.h>
#include <stdlib.h>
int main()
  
```

```

{
    int i, soma;
    soma = 0;
    for (i=1; i <= 10; i++)
    {
        soma = soma + i;
    }
    printf("A soma dos números entre 1 e 10 é %d", soma);
    system ("pause");
    return 0;
}

```

### Estrutura de Repetição com Teste no Início (While)

Esta estrutura tem a característica de possuir uma condição de teste ou parada logo no início da estrutura. Esta condição determina se os comandos que estão dentro da estrutura serão repetidos ou não.

Sintaxe (algoritmo):

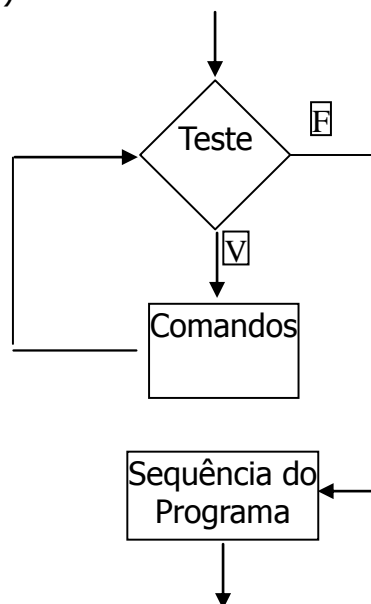
Enquanto (condição)

Início

Sequência de comandos

Fim

Sintaxe (Fluxograma):





Sintaxe (Linguagem C):

while (condição)

```
{  
    Sequência de comandos  
}
```

Exemplo: Crie um algoritmo e um programa que faça a soma dos números entre 1 e 10.

Algoritmo Somatória2

Declare i, soma inteiros

soma  $\leftarrow$  0

i  $\leftarrow$  1

Enquanto (i  $\leq$  10)

Início

soma  $\leftarrow$  soma + i

i  $\leftarrow$  i + 1

Fim

Escreva soma

Fim Algoritmo

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
int main()
```

```
{
```

```
    int i, soma;
```

```
    soma = 0;
```

```
    i=1;
```

```
    while (i  $\leq$  10)
```

```
    {
```

```
        soma = soma + i;
```

```
        i = i + 1;
```

```
    }
```

```
    printf("A soma dos números entre 1 e 10 é %d", soma);
```

```
    system ("pause");
```

```
    return 0;
```

```
}
```

### **Estrutura de Repetição com Teste no Fim (Do...While)**

A característica principal desta estrutura é possuir uma condição de teste ou parada no final da estrutura. Esta condição determina se os comandos que estão dentro da estrutura serão repetidos ou não. A diferença básica para a estrutura

anterior é que esta estrutura executa pelo menos uma vez as instruções dentro do laço.

Sintaxe (algoritmo):

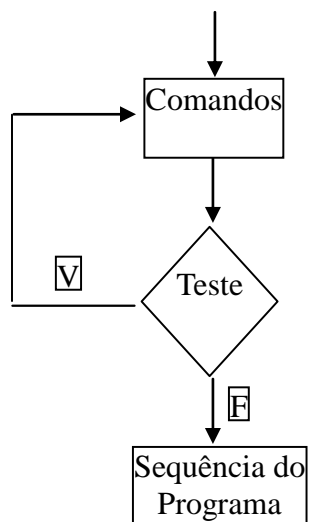
Faça  
Início

Sequência de comandos

Fim

Enquanto (condição)

Sintaxe (Fluxograma):



Sintaxe (Linguagem C):

do  
{

Sequência de comandos

} while (condição);

Exemplo: Crie um algoritmo e um programa que faça a soma dos números entre 1 e 10.

Algoritmo Somatória3

Declare i, soma inteiros

soma  $\leftarrow$  0

i  $\leftarrow$  1

Faça

Início

```
soma ← soma + i
i ← i + 1
Fim Enquanto(i <= 10)
Escreva soma
Fim Algoritmo
```

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int i, soma;
    soma = 0;
    i=1;
    do
    {
        soma = soma + i;
        i = i + 1;
    }while ( i <= 10);
    printf("A soma dos números entre 1 e 10 é %d", soma);
    system ("pause");
    return 0;
}
```

### **Exercícios de Estruturas de Repetição**

1. Faça um programa que converta uma temperatura em graus Celsius para graus Fahrenheit. A fórmula de conversão é:

$$F = C * (9.0/5.0) + 32.0$$

sendo F a temperatura em Fahrenheit. A temperatura Celsius varia de 10 a 20, de 1 em 1.

2. Preparar um Programa em C que imprima a soma dos quadrados dos 10 primeiros números inteiros.

3. Faça um programa onde o usuário digita um valor, e imprima na tela todos os valores entre 0 e o valor digitado.

4. Faça um programa onde o usuário digita dois valores, e imprima na tela todos os valores no intervalo entre os valores digitados.

5. Faça um programa que imprima na tela todos os números pares de um intervalo informado pelo usuário.

6. Um funcionário de uma empresa recebe aumento salarial anualmente. Sabe-se que:

- a) Esse funcionário foi contratado em 2015, com salário inicial de R\$ 1000,00;
- b) Em 2016 recebeu aumento de 1,0% sobre seu salário inicial;
- c) A partir de 2016, os aumentos corresponderam ao dobro do percentual anterior.

Faça um algoritmo e o programa em C que determine o salário atual desse funcionário.

7. Faça um algoritmo e o programa em C que receba o salário de um funcionário X. Sabe-se que o funcionário Y tem um salário equivalente a 1/3 do salário de X. X aplicará seu salário integralmente na caderneta de poupança, rendendo 2% ao mês e Y aplicará seu salário integralmente no fundo de renda fixa, que está rendendo 5% ao mês. Calcule e mostre a quantidade de meses necessários para que o valor pertencente a Y iguale ou ultrapasse o valor pertencente a X.

8. Supondo que a população de um país A seja da ordem de 90.000.000 de habitantes com uma taxa anual de crescimento de 3% e que a população de um país B, seja aproximadamente de 200.000.000 de habitantes com uma taxa anual de crescimento de 1,5%, fazer um algoritmo e o programa em C que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou se iguale a população do país B, mantidas estas taxas de crescimento.

9. Faça um algoritmo e o programa em C que leia um número indeterminado de linhas contendo cada um a idade de um indivíduo. O último valor, que não entrará nos cálculos, contém o valor da idade igual a zero. Calcule e mostre a idade média deste grupo de indivíduos.

10. Faça um algoritmo e o programa em C que receba duas notas de cinco alunos, calcule e mostre:

- a) a média aritmética das duas notas de cada aluno;
- b) a mensagem que está na tabela a seguir:

Média Aritmética	Mensagem
Até 3	Reprovado
Entre 3 e 7	Exame
Acima de 7	Aprovado

O total de alunos aprovados, de exame e reprovados; a média da classe.

11. Faça um algoritmo e o programa em C que leia uma quantidade indeterminada de números positivos e conte quantos deles estão nos seguintes intervalos: [0-25], [26-50], [51-75] e [76-100]. A entrada dos dados deverá terminar quando for lido um número negativo.

12. Sendo  $H = 1/1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ , prepare um algoritmo e o programa em C para gerar o número H. O número N é fornecido pelo usuário.

13. Faça um algoritmo e o programa em C que receba um número e que calcule e mostre a tabuada desse número.

14. Faça um programa que leia as notas de um aluno e calcule a média das notas. Não se sabe, de antemão, o número de notas a serem lidas, e a leitura deve encerrar quando uma nota negativa for informada.

15. Faça um programa que leia valores inteiros positivos e diga, no final, quantos números pares foram lidos, quantos números ímpares foram lidos, o maior número lido e o menor número lido. A entrada de dados para com a digitação do valor -1.