

AULA 03

Estrutura Condicional

Existem várias situações em que temos que tomar decisões no andamento do algoritmo. Essas decisões interferem diretamente no andamento do programa.

Os comandos de decisão ou desvio fazem parte das técnicas de programação que conduzem a estruturas de programas que não são totalmente sequenciais. Com as instruções de SALTO ou DESVIO pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores.

A estrutura de decisão sempre vem acompanhada de uma ação. Por exemplo, você pode verificar se a nota de um determinado aluno é maior ou igual a sete para dizer se ele está ou não aprovado na disciplina.



1. Estrutura Condicional Simples

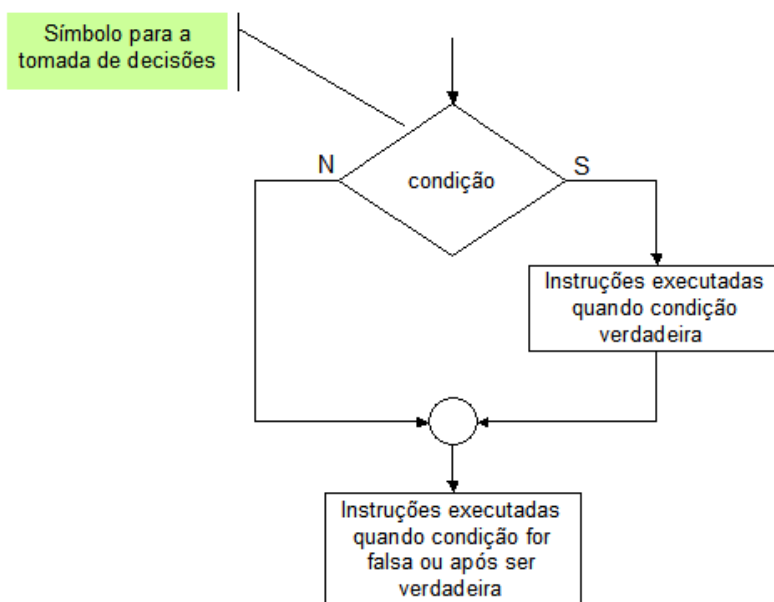
Na estrutura condicional simples, o **comando** entre as instruções “se...então” e “fim_se” só será executado se a condição for **verdadeira**. Se a condição for falsa, serão executados os comandos que estiverem após a instrução “fim_se”. Uma condição é uma comparação que possui dois valores possíveis, verdadeiro ou falso, como vimos anteriormente.

Se existir somente um comando a ser executado entre as instruções “se...então” e “fim_se”, a sintaxe do pseudo-código deve ser a seguinte:

```
se (<condição>) então  
    <comando 1>  
fim_se
```

Caso existam mais comandos a serem executados entre as instruções “se...então” e “fim_se”, a sintaxe do pseudo-código deverá ser da seguinte forma:

```
se (<condição>) então  
    início  
        <comando 1>  
        <comando 2>  
        <comando 3>  
    fim  
fim_se
```



Como um exemplo, vamos considerar o seguinte problema: “*Ler duas notas de um aluno, efetuar a média aritmética e, caso a média seja maior ou igual a 7, informar que o aluno foi aprovado.*”

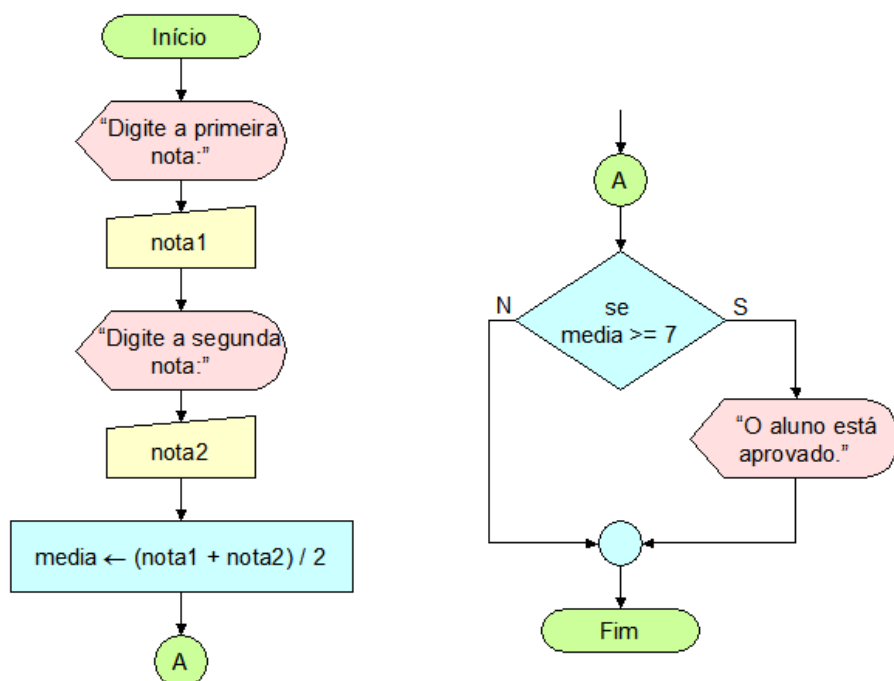
Algoritmo

1. Leia a nota 1 do aluno
2. Leia a nota 2 do aluno
3. Some a nota 1 com a nota 2 e calcule a média das notas
4. Verificar se a média é maior do que 7; se sim, exiba na tela “O aluno foi aprovado”.

Pseudo-código

```
programa media  
declare  
    nota1, nota2, media : real  
início  
    escreva ("Digite a primeira nota:")  
    leia (nota1)  
    escreva ("Digite a segunda nota:")  
    leia (nota2)  
    media ← (nota1 + nota2) / 2  
    se (media >= 7) então  
        escreva ("O aluno está aprovado.")  
    fim_se  
fim
```

Diagrama de bloco



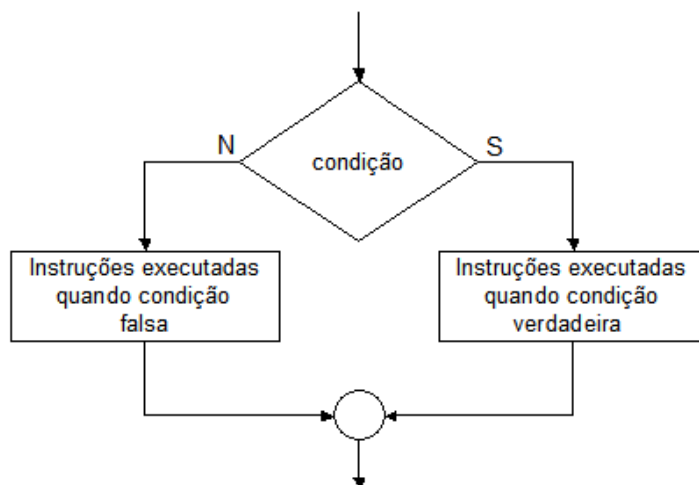
2. Estrutura Condicional Composta

A estrutura condicional composta nada mais é do que uma estrutura condicional simples com mais alguma ação caso a condição seja falsa. Portanto, se a condição for verdadeira, será executado o <comando 1>, senão, se a condição for falsa, será executado o <comando 2>.

```
se (<condição>) então
    <comando 1>
senão
    <comando 2>
fim_se
```

Como vimos anteriormente, se houver mais de um comando a ser executado quando a condição for falsa ou verdadeira, é necessário utilizar as palavras *início* e *fim* entre tais comandos, conforme a sintaxe abaixo:

```
se (<condição>) então
    início
        <comando 1>
        <comando 2>
    fim
senão
    início
        <comando 3>
        <comando 4>
    fim
fim_se
```



Para ilustrar a estrutura condicional composta, utilizaremos o problema anterior: *“Ler duas notas de um aluno, efetuar a média aritmética e, caso a média seja maior ou igual a 7, informar que o aluno foi aprovado, senão, informar que o aluno foi reprovado.”*

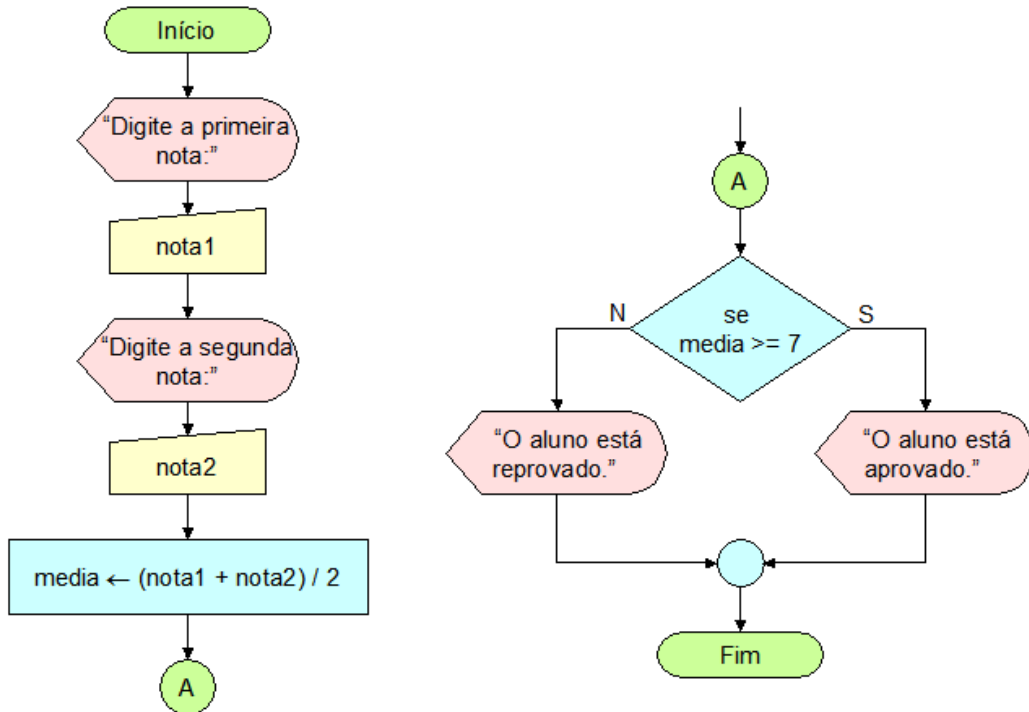
Algoritmo

1. Leia a nota 1 do aluno
2. Leia a nota 2 do aluno
3. Some a nota 1 com a nota 2 e calcule a média das notas
4. Verificar se a média é maior do que 7; se sim, exiba na tela “O aluno foi aprovado”, senão exiba na tela: “O aluno foi reprovado.”

Pseudo-código

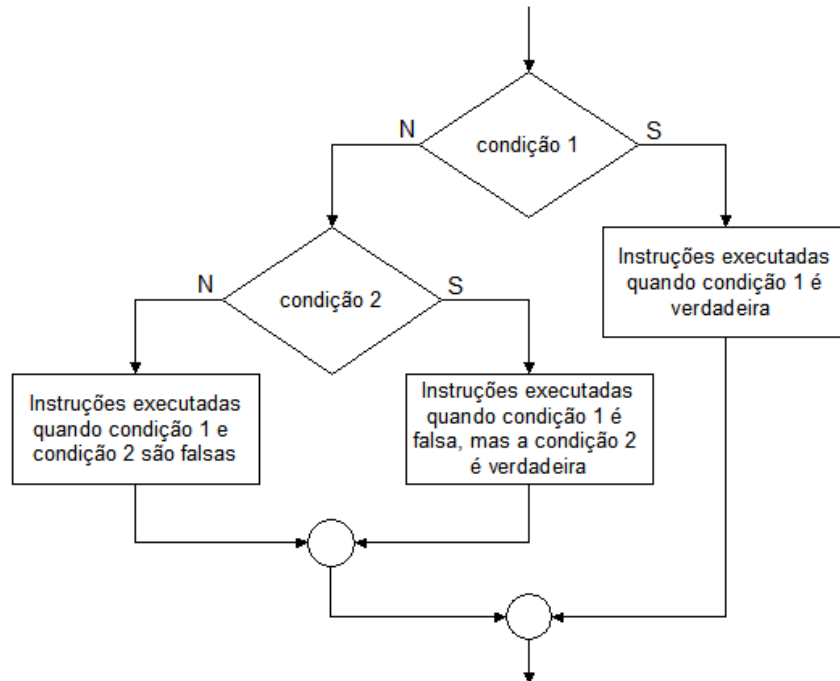
```
programa media
declare
    nota1, nota2, media : real
início
    escreva ("Digite a primeira nota:")
    leia (nota1)
    escreva ("Digite a segunda nota:")
    leia (nota2)
    media ← (nota1 + nota2) / 2
    se (media >= 7) então
        escreva ("O aluno está aprovado.")
    senão
        escreva ("O aluno está reprovado.")
    fim_se
fim
```

Diagrama de bloco



3. Estrutura Condicional Encadeada (ou Aninhada)

Existem casos em que é necessário se estabelecerem verificações de condições sucessivas, onde uma determinada ação poderá ser executada se um conjunto anterior de instruções ou condições for satisfeito. Sendo a ação executada, esta poderá ainda estabelecer novas condições. Isto significa utilizar uma condição dentro de outra condição. Este tipo de estrutura poderá possuir diversos níveis de condição, sendo chamadas de encadeamentos.



Neste exemplo, está sendo adotado o encadeamento para a <condição1> falsa, mas poderá, dependendo do problema a ser resolvido, ser colocado no outro lado, como poderá ocorrer de termos a necessidade de utilizar condição dos dois lados.

```
se (<condição1>) então  
    <instruções para condição1 verdadeira>  
senão  
    <instruções para condição1 falsa>  
    se (<condição2>) então  
        <instruções para condição2 verdadeira>  
    senão  
        <instruções para condição2 falsa>  
    fim_se  
fim_se
```

Para ilustrar a estrutura condicional encadeada, utilizaremos novamente o problema anterior, mas desta vez com algumas modificações: *“Ler duas notas de um aluno, efetuar a média aritmética e, se a média for maior ou igual a 7, informar que o aluno foi aprovado; se a média for maior ou igual a 5 mas menor do que 7, informar que o aluno está de exame; se a média for menor do que 5 informar que o aluno foi reprovado.”*

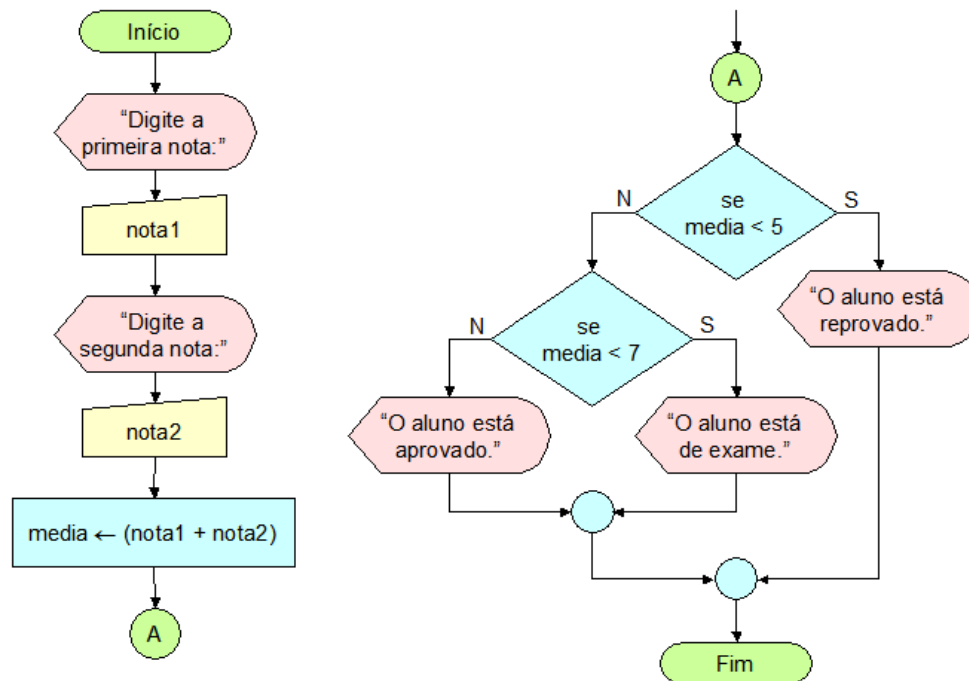
Algoritmo

1. Leia a nota 1 do aluno
2. Leia a nota 2 do aluno
3. Some a nota 1 com a nota 2 e calcule a média das notas
4. Verificar se a média é menor do que 5; se sim, exiba na tela “O aluno está reprovado”
5. Verificar se a média é menor do que 7; se sim, exiba na tela “O aluno está de exame”
6. Verificar se a média é maior ou igual a 7; se sim, exiba na tela “O aluno foi aprovado”.

Pseudo-código

```
programa media
declare
    nota1, nota2, media : real
início
    escreva ("Digite a primeira nota:")
    leia (nota1)
    escreva ("Digite a segunda nota:")
    leia (nota2)
    media ← (nota1 + nota2) / 2
    se (media < 5) então
        escreva ("O aluno está reprovado.")
    senão
        se (média < 7) então
            escreva ("O aluno está de exame.")
        senão
            escreva ("O aluno está aprovado.")
        fim_se
    fim_se
fim
```


Diagrama de bloco



Note que a referência feita no passo 6 do algoritmo não é escrita no pseudocódigo e no diagrama de bloco, uma vez que a mesma fica subentendida, ou seja, qualquer valor que não seja menor do que 5 ou que não seja menor do que 7, está conseqüentemente acima de 7.

4. Práticas

Escreva o algoritmo, pseudo-código e diagrama de bloco para os seguintes exercícios:

1. Leia dois números, faça a soma e apresente o número caso seja maior que 15.
2. Leia dois números, faça a multiplicação e apresente o número caso esteja entre 50 e 100.
3. Leia dois números, faça a subtração e caso seja menor que zero apresentar “número negativo”.
4. Escreva um algoritmo que leia dois números reais e um caractere que representa a operação aritmética a ser realizada (+, -, *, /). O algoritmo deve apresentar o resultado final da operação ou uma mensagem de erro para o caso da divisão de um número por zero ou uma mensagem de erro caso não seja escolhida uma das quatro operações aritméticas básicas.
5. Elaborar um programa que efetue o cálculo do reajuste de salário de um funcionário. Considere que o funcionário deverá receber um reajuste de 15% caso seu salário seja menor que 500. Se o salário for maior ou igual a 500 mas menor ou igual a 1000, seu reajuste será de 10%, e caso seja ainda maior que 1000, o reajuste deverá ser de 5%.
6. Em uma vídeo-locadora, o preço de locação do DVD por dia depende de uma cor de selo colada na embalagem do DVD. Sendo dados o número de dias locados e a cor do selo, calcular o preço total da locação, considerando-se a seguinte tabela:

Cor do selo	Preço por dia (R\$)
Verde	5.50
Azul	5.70
Branco	4.00
Rosa	3.50