



Estrutura de decisão simples e Composta



professor (a)

Hilson Silva

Curso:

Engenharia de Software

Disciplina:

Algoritmo e Programação



Agenda

- Objetivos da aula
- Conceitos de Desvio condicional
- Estrutura Simples
- Estrutura composta
- Aninhada
- Encadeada



Objetivo da aula

- Compreender as estruturas Condicionais
- Reforçar a compreensão da Lógica de Programação
- Destacar onde pode ser aplicada essas estruturas



Desvio condicional (Simples e Composta)



Conceito

Desvio condicional ou estrutura de decisão uma estrutura de desvio do fluxo de controle presente nas linguagens de programação que realizam diferentes ações, tendo como base um valor booleano ou lógico (verdadeira ou falsa).



Estrutura de Decisão

A estrutura condicional permite a escolha de um grupo de ações a ser executado quando determinada condição, representada por uma expressão lógica é ou não satisfeita.

- Esta estrutura pode se apresentar de duas formas:

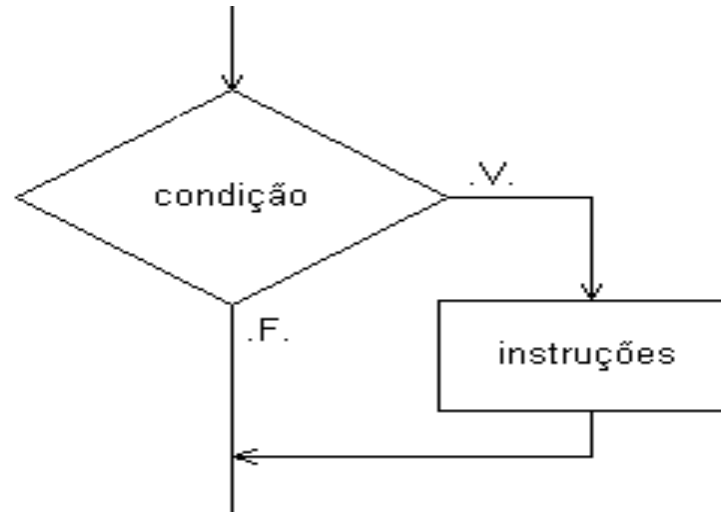
1.Estrutura Condicional Simples

2.Estrutura Condicional Composta



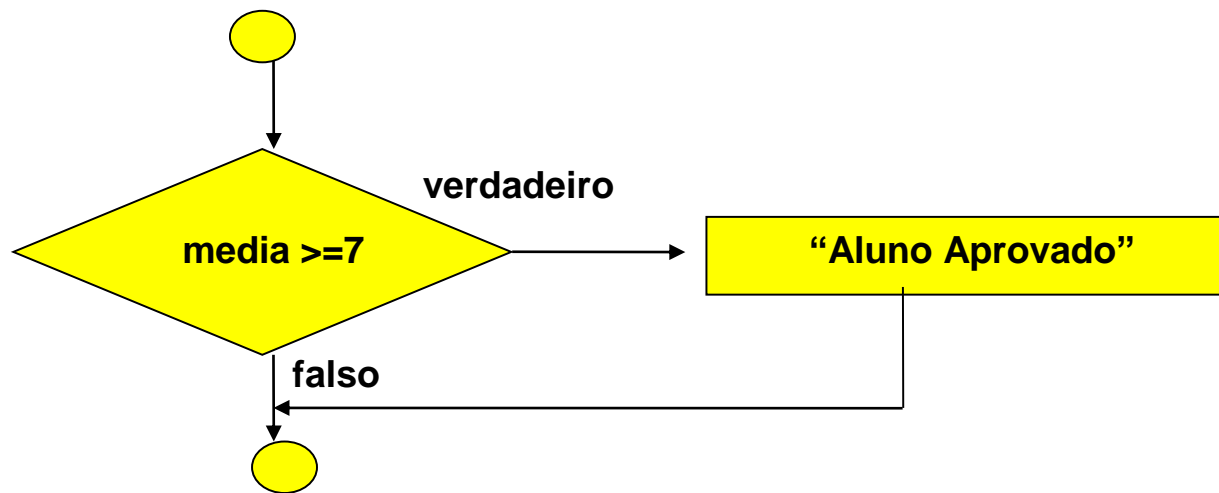
Desvio Condicional-Simples

No desvio condicional simples uma condição é avaliada e, se o resultado for verdadeiro, um conjunto de instruções é executado.





Conceito

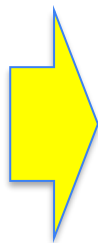




Desvio Condicional-Simples

Exemplos: Portugol e C

```
se (condição for Verdade) então  
    //comando1;  
    /*ou bloco de comandos;*/  
fim_se;
```

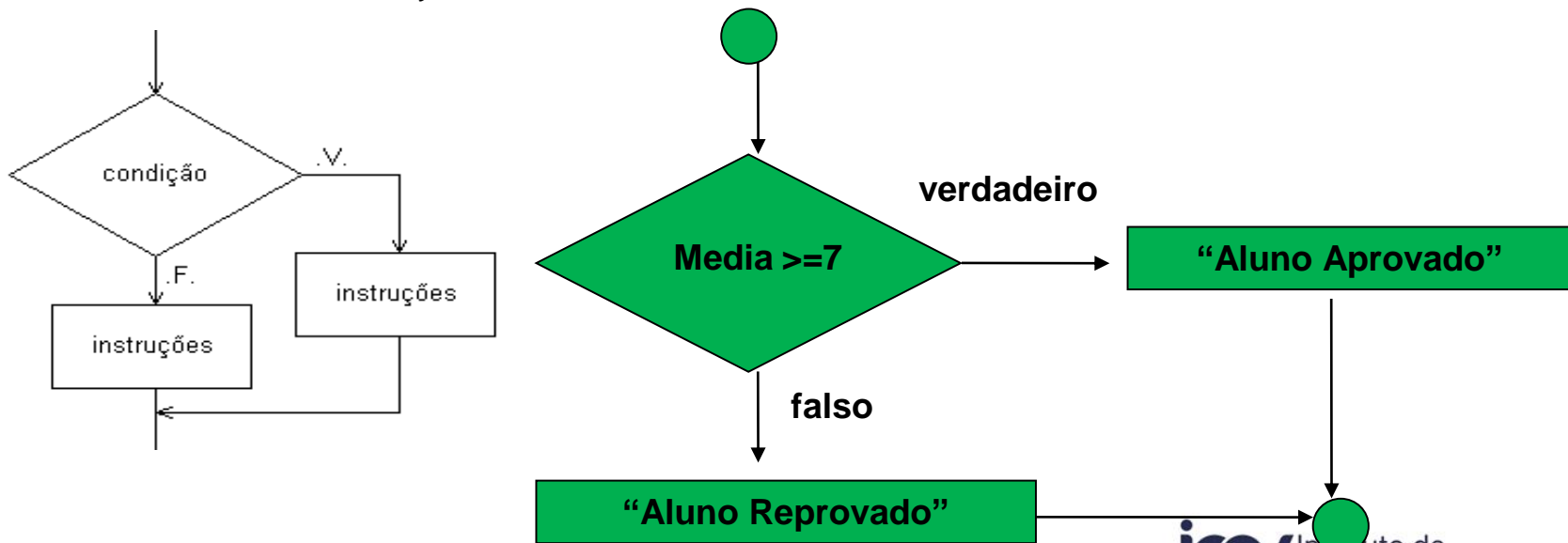


```
if (condição != 0) {  
    //comando1;  
    /*ou bloco de comandos;*/  
}
```



Desvio Condicional-Composto

No desvio condicional composto, uma condição é avaliada e, se o resultado for verdadeiro, um conjunto de instruções é executado. Caso contrário, outro conjunto de instruções serão executados.





Desvio Condicional-Composto

- Exemplos: Portugal e C

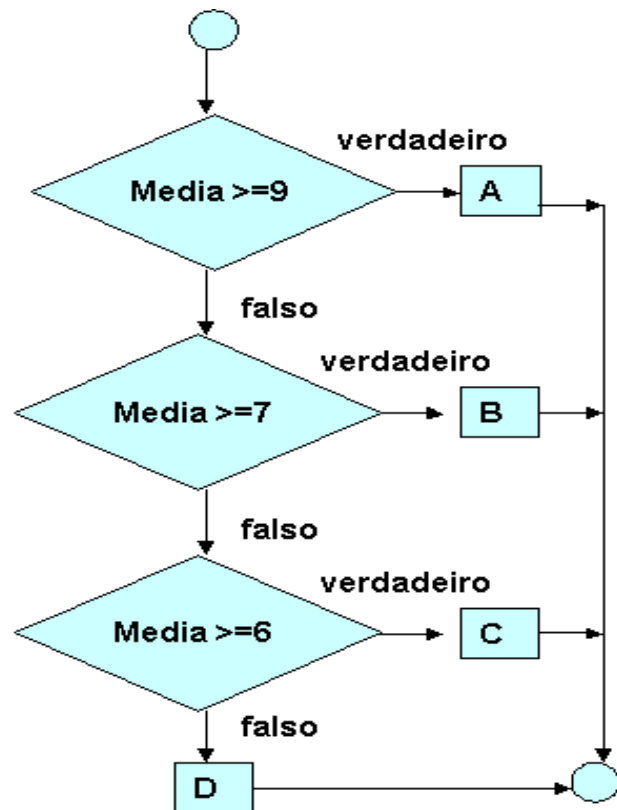
```
se (condição for Verdade) então  
    comando1;  
    {ou bloco de comandos};  
senão  
    comando2;  
    {ou bloco de comandos};  
fim_se;
```

```
if (condição != 0)  
{  
    comando1;  
    {ou bloco de comandos};  
}  
else  
{  
    comando2;  
    {ou bloco de comandos};  
}
```



Estrutura Condicional-Encadeada

Sequências de estruturas em que o **If-Else** são usados para um teste com múltiplos casos.





Estrutura Condicional-Encadeada

- Exemplos: Portugol e C

```
se (condição1 for Verdade) então  
  se (condição2 for Verdade) então  
    se (condição3 for Verdade) então  
      //comando1;  
      /*ou bloco de comandos*/  
    fim_se;  
  fim_se;  
fim_se;
```

```
if (condição1 != 0)  
{  
  if (condição2 != 0)  
  {  
    if (condição3 != 0)  
    {  
      //comando1  
      /*ou bloco de comandos*/  
    }  
  }  
}
```



Estrutura Condicional-Encadeada

```
function register()
{
    if (!empty($_POST)) {
        $msg = '';
        if ($_POST['user_name']) {
            if ($_POST['user_password_new']) {
                if ($_POST['user_password_new'] === $_POST['user_password_repeat']) {
                    if (strlen($_POST['user_password_new']) > 5) {
                        if (strlen($_POST['user_name']) < 65 && strlen($_POST['user_name']) > 1) {
                            if (preg_match('/^[a-z\d]{2,64}$/i', $_POST['user_name'])) {
                                $user = read_user($_POST['user_name']);
                                if (!isset($user['user_name'])) {
                                    if ($_POST['user_email']) {
                                        if (strlen($_POST['user_email']) < 65) {
                                            if (filter_var($_POST['user_email'], FILTER_VALIDATE_EMAIL)) {
                                                create_user();
                                                $_SESSION['msg'] = 'You are now registered so please login';
                                                header('Location: ' . $_SERVER['PHP_SELF']);
                                                exit();
                                            } else $msg = 'You must provide a valid email address';
                                        } else $msg = 'Email must be less than 64 characters';
                                    } else $msg = 'Email cannot be empty';
                                } else $msg = 'Username already exists';
                            } else $msg = 'Username must be only a-z, A-Z, 0-9';
                        } else $msg = 'Username must be between 2 and 64 characters';
                    } else $msg = 'Password must be at least 6 characters';
                } else $msg = 'Passwords do not match';
            } else $msg = 'Empty Password';
        } else $msg = 'Empty Username';
        $_SESSION['msg'] = $msg;
    }
    return register_form();
}
```





Estrutura Condicional-Encadeada

• Outro exemplo de IF/ELSE encadeado:

```
if (numero%2 == 0) //se for verdadeiro imprime O numero eh PAR
{
    printf("O numero eh multiplo de 2 \n");
}
else if(numero%3 == 0)
{
    printf("O numero eh multiplo de 3 \n");
}
else if(numero%5 == 0)
{
    printf("O numero eh multiplo de 5 \n");
}
else
{
    printf("O numero nao eh multiplo de 2,3 ou 5 \n");
}
```




Estrutura Condicional-Aninhada

- Exemplos: Portugal e C

```
se (condição1 for Verdade) então  
  se (condição2 for Verdade) então  
    se (condição3 for Verdade) então  
      /*bloco de comandos*/  
    senão  
      /*bloco de comandos*/  
  fim_se; //condição 3  
fim_se; //condição 2  
senão  
  /*bloco de comandos*/  
fim_se; //condição 1
```



```
if (condição1 != 0) {  
  if (condição2 != 0) {  
    if (condição3 != 0) {  
      /*bloco de comandos*/  
    }  
    else  
    {  
      /*bloco de comandos*/  
    }  
  }  
}  
else  
{  
  /*bloco de comandos*/  
}
```



Estrutura Múltipla Escolha

- Estrutura de múltipla escolha

escolha (X)

caso 1:

/*bloco de comandos*/

caso 2:

/*bloco de comandos*/

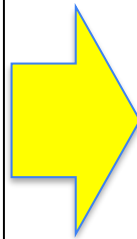
caso 3:

/*bloco de comandos*/

caso Contrário: //pode ser omitido

/*bloco de comandos*/

fim_escolha;



switch (X)

{

case 1:

/*bloco de comandos*/

break;

case 2:

/*bloco de comandos*/

break;

case 3:

/*bloco de comandos*/

break;

default: //pode ser omitido

/*bloco de comandos*/

break;

}



Estrutura Multipla Escolha

- Exemplo:

```
23 Inicio
24 // Seção de Comandos, procedimento, funções, operadores, etc...
25 repita
26     escreval("")
27     escreval("| 1-Cadastrar")
28     escreval("| 2-Alterar ")
29     escreval("| 3-Consultar")
30     escreval("| 4-Remover ")
31     escreval("| 5-Sair ")
32     escreval("informe a operação desejada")
33     leia(operacao)
34     escolha (operacao)
35     caso 1
36         cadastrar()
37     caso 2
38         alterar()
39     caso 3
40         Consultar()
41     caso 4
42         Remover()
43     fimescolha
44     ate operacao=5
```



```
114 int main(){
115     do{
116
117         TelaPrincipal();
118
119         switch(op){
120
121             case 1:{
122                 system("cls");
123                 TelaCadastros();
124                 break;
125             }
126             case 2:{
127                 system("cls");
128                 TelaImprimir();
129                 break;
130             }
131         }
132     }while(op!=4);
133 }
```



Atividade

1. Implemente um algoritmo que represente a situação a seguir: Se salário maior que 1800 reais e menor que 8000 mil reais, deverá ser apresentada a mensagem de saída 'Salário válido para financiamento'. Caso contrário deverá ser apresentada a mensagem de saída 'Salário fora da faixa permitida para financiamento'.
2. Implemente um algoritmo que represente a situação a seguir: Se a idade for menor que 18 anos ou idade for maior que 95 anos, apresentar a mensagem de saída 'Você não pode tirar carteira de motorista'. Caso contrário apresentar a mensagem de saída 'Você pode possuir carteira de motorista'.



Atividade

3. Implemente um algoritmo que represente a situação a seguir: Se a idade for maior ou igual 18 anos e idade menor ou igual 95 anos e o resultado do exame de legislação for APROVADO, então apresentar a mensagem de saída 'Sua carteira de motorista estará pronta em uma semana'. Caso contrário apresentar a mensagem de saída 'Você não possui idade permitida ou não passou no teste de legislação'.
4. Implemente um algoritmo que represente a situação a seguir: Lê três números inteiros positivos apresentando como saída o maior valor digitado entre eles. O algoritmo deverá prever os valores iguais, caso isso aconteça apresentar a mensagem: "Todos os valores digitados são iguais".



Atividade

5. Implemente uma algoritmo que capaz de imprimir o mês do ano, sendo que quando o usuário digitar entre o intervalo de 1 a 12 será impresso o mês correspondente ao número (Ex. Se o usuário digitar o número 3 deverá ser impresso o mês de MARÇO). O algoritmo deverá prever os valores digitados fora desse intervalo estabelecido.

6. Implemente uma algoritmo que calcule a raiz quadrada de um numero descartando a possibilidade do número ser igual a zero. Caso contrário isso aconteça apresentar a mensagem de saída 'Informe um valor maior que zero'.



Atividade

7. Escreva um algoritmo que determine o grau de obesidade de uma pessoa, sendo fornecido o peso e a altura da pessoa. O grau de obesidade é determinado pelo índice de massa corpórea(IMC) ($\text{massa} = \text{peso} / \text{altura}^2$) através da tabela abaixo:

Índice Massa corpórea	Grau de obesidade
<26	Normal
≥ 26 e <30	Obeso
≥ 30	Obeso mórbido



Exemplo de Portugol

```
1 Algoritmo "Soma"
2
3 Var
4 // Seção de Declarações das variáveis
5 valproA : real
6 valproB : real
7 valproC : real
8
9 Inicio
10 // Seção de Comandos, procedimento, funções, operadores, etc...
11     escreval("Informe o valor do produto A")
12     leia(valproA)
13     escreval("Informe o valor do produto B")
14     leia(valproB)
15     escreval("Informe o valor do produto C")
16     leia(valproC)
17
18     se (valproA+valproB)>valproC entao
19         escreval("Soma produto A+B",valproA+valproB)
20     senao
21         escreval("Valor do produto C",valproC)
22     fimse
23     se (valproB+valproC)>valproA entao
24         escreval("Soma produto B+C",valproB+valproC)
25     senao
26         escreval("Valor do produto A",valproA)
27     fimse
28
29 Fimalgoritmo
```




Bibliografia Básica:

1. ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos de Programação de Computadores - Algoritmos, Pascal, C/C++ e Java**. 3. ed. São Paulo: Pearson Education do Brasil, 2012.
2. SOUZA, M. GOMES, M. SOARES, M. CONCÍLIO, R. **Algoritmos e Lógica de Programação: Um texto introdutório para a engenharia**. 3. Ed. São Paulo: Cengage, 2020.
3. PIVA JR., D.; ENGERBRECHT, A. M.; NAKAMITI, G. S.; BIANCHI, F. **Algoritmos e Programação de Computadores**. 2. ed. Rio de Janeiro: Elsevier, 2019.



Bibliografia complementar

1. LOPES, A.; GARCIA, G. **Introdução à Programação: 500 algoritmos resolvidos**. São Paulo: Campus, 2002
2. EDELWEISS, Nina, LIVI, Castro, M. A. **Algoritmos e Programação com Exemplos em Pascal e C - Série Livros Didáticos UFRGS - Volume 23**. Porto Alegre: Bookman, 2014.
3. MANZANO, G., J.A. N., OLIVEIRA, de, J. F. **Algoritmos - Lógica para Desenvolvimento de Programação de Computadores**. 29. ed. São Paulo: Érica, 2019.
4. BACKES, André. **Linguagem C - Completa e Descomplicada**. 2. ed. Rio de Janeiro: LTC, 2020.
5. MENEZES, Nilo Ney Coutinho. **Introdução à Programação com Python**. 3. ed. São Paulo: Novatec, 2019.



icev Instituto de
Ensino Superior