



INTRODUÇÃO À COMPUTAÇÃO

***UD II - ALGORITMOS
CONDICIONAIS***

UD II - INTRODUÇÃO À ALGORITMOS

Condicionais



ELEMENTOS DE COMPETÊNCIA

- Empregar recursos para operar em ambientes humanizados, integrando as dimensões física, humana e informacional deste ambiente operacional.
- Tomar decisões e conduzir ações, em situações de crise.

Condicionais



OBJETIVOS

1. Identificar os tipos de condicionais. (FACTUAL)
2. Compreender a aplicação de condicionais na construção de algoritmos. (CONCEITUAL)
3. Aplicar as condicionais na elaboração de algoritmos. (PROCEDIMENTAL)

UD II - INTRODUÇÃO À ALGORITMOS

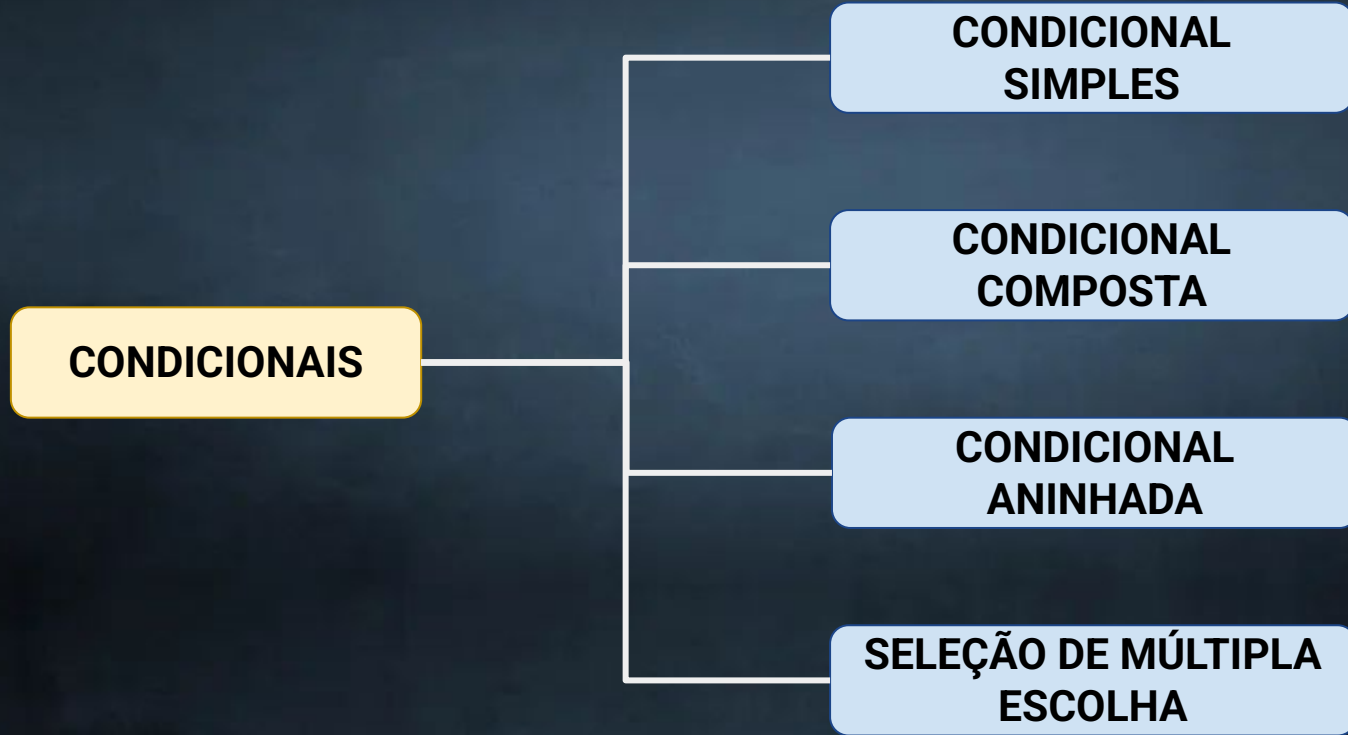
Entrada e Saída



ATITUDES

1. Organização: capacidade de desenvolver atividades de forma sistemática e eficiente.
2. Dedicação: agir, realizando espontaneamente, com empenho e entusiasmo, as atividades necessárias ao cumprimento da missão.
3. Responsabilidade: capacidade de cumprir suas atribuições assumindo e enfrentando as consequências de suas atitudes e decisões.

- Também chamadas de estruturas de seleção ou estruturas de decisão.
- Uma das estruturas de controle de fluxo de instruções.
- Permitem o desvio do fluxo de execução do algoritmo, fazendo com que o caminho a ser seguido durante a execução do programa seja determinado por meio de testes realizados. Esses testes são constituídos por expressões relacionais e/ou lógicas, que retornarão como resultado um valor lógico (verdadeiro ou falso).



Condicional simples

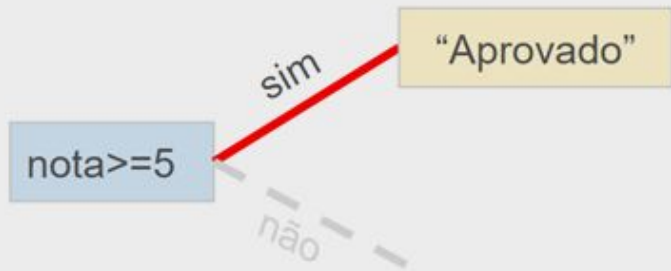
Uma instrução ou um bloco de instruções será executado apenas se uma determinada condição for verdadeira.

Para construir estruturas condicionais simples, faremos uso de algumas palavras reservadas (“se”, “então” e “fimse”)

A sintaxe da condicional simples em pseudocódigo é:

```
Se <condição> então
    <instrução/bloco de instruções>
Fimse
```

Condicional simples



```
Se nota >= 5 então  
|   Escreva("Aprovado")  
FimSe
```

Condicional simples

Algoritmo "06_01 media_aluno2"

*\\ Algoritmo que calcula a média do aluno a partir das notas dos dois semestres e
\\ apresenta a mensagem de aprovação, caso sua nota seja igual ou superior a 5.*

Var

nota1, nota2, media: real

Início

Escreva ("Digite a nota do primeiro semestre: ")

Leia (nota1)

Escreva ("Digite a nota do segundo semestre: ")

Leia (nota2)

$media \leftarrow (nota1 + nota2)/2$

Escreval ("A média é: ",media)

Se media >= 5 **então**

 Escreva ("Parabéns!!!! Você foi aprovado.")

Fimse

FimAlgoritmo

Condicional composta

É destinada ao caso em que se necessita que sejam executadas algumas instruções quando o resultado lógico da expressão for verdadeiro e outras quando o resultado for falso. Palavras reservadas utilizadas neste tipo de condicional no pseudocódigo: “se”, “então”, “senão” e “fimse”.

Sintaxe:

Se <condição> **então**

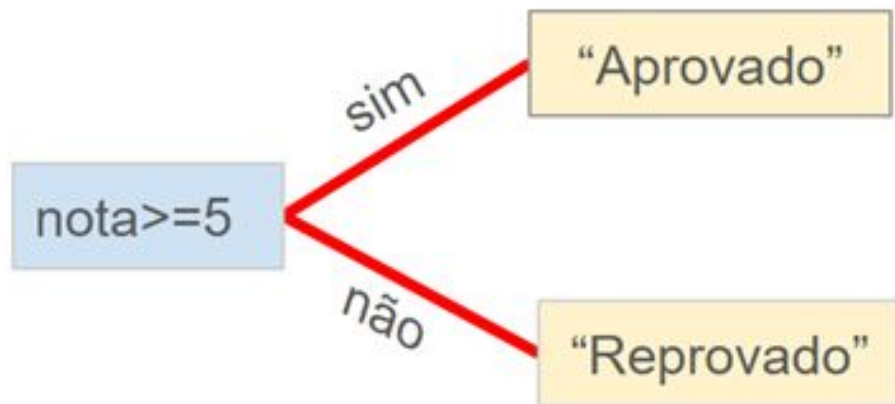
<instrução/bloco de instruções (verdade)>

Senão

<instrução/bloco de instruções (falso)>

FimSe

Condicional composta



Se `nota >= 5` então

| Escreva("Aprovado")

Senão

| Escreva("Reprovado")

FimSe

Algoritmo "06_02 media_aluno3"

// Algoritmo que calcula a média do aluno a partir das notas dos dois semestres e apresenta
// a mensagem de aprovação ou reprovação, caso sua nota seja igual ou superior a 5 ou
// inferior a 5, respectivamente.

Var

nota1, nota2, media: real

Início

Escreva ("Digite a nota do primeiro semestre: ")

Leia (nota1)

Escreva ("Digite a nota do segundo semestre: ")

Leia (nota2)

$media \leftarrow (nota1 + nota2)/2$

Escreval ("A média é: ",media)

Se media \geq 5 **então**

 Escreva ("Parabéns!!!! Você foi aprovado.")

Senão

 Escreva ("Sinto muito, mas você foi reprovado.")

FimSe

FimAlgoritmo

Condicional aninhada

São caracterizadas pela existência de condicionais dentro de outras condicionais, havendo, portanto, mais de uma condição a ser analisada.

Sintaxe:

```
Se <condição> então
|   <instrução/bloco de instruções (verdade)>
Senão
|   Se <condição> então
|   |   <instrução/bloco de instruções (verdade)>
|   Senão
|   |   <instrução/bloco de instruções (falso)>
|   FimSe
FimSe
```

Condicionnal aninhada



```
Se media=10 então
|   Escreva("Aprovado com louvor")
Senão
|   Se media>=5 então
|   |   Escreva("Aprovado")
|   Senão
|   |   Escreva("Reprovado")
|   FimSe
FimSe
```

Início

// Entrada de dados

Escreva ("Digite a nota do primeiro semestre: ")

Leia (nota1)

Escreva ("Digite a nota do segundo semestre: ")

Leia (nota2)

// Processamento

$media \leftarrow (nota1 + nota2)/2$

Escreva ("A média é: ",media)

// Saída de dados

Se media = 10 **então**

Escreva ("Sensacional!!!! Você foi aprovado com louvor.")

// Saída de dados

Senão

Se media \geq 5 **E** media $<$ 10 **então**

Escreva ("Parabéns!!!! Você foi aprovado.")

// Saída de dados

Senão

Escreva ("Sinto muito, mas você foi reprovado.")

// Saída de dados

Fimse

Fimse

FimAlgoritmo

Algoritmo "06_03 media_aluno4"

// Algoritmo que calcula a média do aluno a partir das notas dos dois semestres

// e apresenta a mensagem de aprovação, reprovação ou aprovação com louvor, caso sua

// nota seja igual ou superior a 5, inferior a 5 ou igual a 10, respectivamente.

Var

nota1, nota2, media: real

Início

// Entrada de dados

Escreva ("Digite a nota do primeiro semestre: ")

Leia (nota1)

Escreva ("Digite a nota do segundo semestre: ")

Leia (nota2)

// Processamento

media \leftarrow (nota1 + nota2)/2

Escreva ("A média é: ",media)

// Saída de dados

Se media = 10 **então**

Escreva ("Sensacional!!!! Você foi aprovado com louvor.")

// Saída de dados

Senão

Se media \geq 5 **E** media $<$ 10 **então**

Escreva ("Parabéns!!!! Você foi aprovado.")

// Saída de dados

Senão

Escreva ("Sinto muito, mas você foi reprovado.")

// Saída de dados

Fimse

Fimse

FimAlgoritmo

Algoritmo "06_03
media_aluno4"

Algoritmo no
Visualg

Seleção de múltipla escolha

Condicional utilizada em muitas linguagens de programação, quando existe um conjunto de condições a serem testadas, sendo que para cada condição escolhida são executadas instruções diferentes. Similar a um “Menu”, em que, dependendo da opção escolhida, teríamos os resultados associados àquela escolha.

Seleção de múltipla escolha

Sintaxe:

Escolha <Expressão/variável de escolha>

caso <valor 1>

| <instruções referentes ao caso valor 1>

caso <valor 2>

| <instruções referentes ao caso valor 2>

caso <valor 3>

| <instruções referentes ao caso valor 3>

outrocaso

| <instruções referentes ao outrocaso>

Fimescolha

Condicional aninhada

Algoritmo "06_04 Calculadora Básica"

*// Algoritmo que executa uma das quatro operações básicas, dependendo da
// escolha feita pelo usuário, conforme menu de opções apresentado.*

Var

numero1, numero2, resultado: real

operacao: inteiro

Início

Escreva ("Digite o primeiro número: ")

Leia (numero1)

Escreva ("Digite o segundo número: ")

Leia (numero2)

Escreva ("Menu: 1 – soma; 2 – subtração; 3 – multiplicação; 4 – divisão")

Escreva ("Escolha a operação, conforme Menu acima: ")

```
1 Algoritmo "06_04 Calculadora Básica"
2 // Algoritmo que executa uma das quatro operações básicas, dependendo da
3 // escolha feita pelo usuário, conforme menu de opções apresentado.
4
5 Var
6 numero1, numero2, resultado: real
7 operacao: inteiro
8
9 Início
10 Escreva ("Digite o primeiro número: ")
11 Leia (numero1)
12 Escreva ("Digite o segundo número: ")
13 Leia (numero2)
14 Escreva ("Menu: 1 – soma; 2 – subtração; 3 – multiplicação; 4 – divisão")
15 Escreva ("Escolha a operação, conforme Menu acima: ")
16 Leia (operacao)
17 Escolha operacao
18   caso 1
19     resultado ← (numero1+numero2)
20     Escreva ("O resultado da soma é: ",resultado)
21   caso 2
22     resultado ← (numero1-numero2)
23     Escreva ("O resultado da subtração é: ",resultado)
24   caso 3
25     resultado ← (numero1*numero2)
26     Escreva ("O resultado da multiplicação é: ",resultado)
27   caso 4
28     resultado ← (numero1/numero2)
29     Escreva ("O resultado da divisão é: ",resultado)
30   outro caso
31     Escreva ("Opção inválida")
32 FimEscolha
33 FimAlgoritmo
```

Condicional aninhada

```
15 Escolha operacao
16     caso 1
17         resultado ← (numero1+numero2)
18         Escreva ("O resultado da soma é: ",resultado)
19     caso 2
20         resultado ← (numero1-numero2)
21         Escreva ("O resultado da subtração é: ",resultado)
22     caso 3
23         resultado ← (numero1*numero2)
24         Escreva ("O resultado da multiplicação é: ",resultado)
25     caso 4
26         resultado ← (numero1/numero2)
27         Escreva ("O resultado da divisão é: ",resultado)
28     outro caso
29         Escreva ("Opção inválida")
30 FimEscolha
31 FimAlgoritmo
```

```
1 Algoritmo "06_04 Calculadora_Básica"
2 // Algoritmo que executa uma das quatro operações básicas, dependendo da
3 // escolha feita pelo usuário, conforme menu de opções apresentado.
4
5 Var
6     numero1, numero2, resultado: real
7     opcao: inteiro
8
9 Inicio
10 Escreva ("Digite o primeiro número: ")
11 Leia (numero1)
12 Escreva ("Digite o segundo número: ")
13 Leia (numero2)
14 Escreva ("Menu: 1 – soma; 2 – subtração; 3 – multiplicação; 4 – divisão")
15 Escreva ("Escolha a operação, conforme Menu acima: ")
16 Leia (opcao)
17 Escolha opcao
18     caso 1
19         resultado ← (numero1+numero2)
20         Escreva ("O resultado da soma é: ",resultado)
21     caso 2
22         resultado ← (numero1-numero2)
23         Escreva ("O resultado da subtração é: ",resultado)
24     caso 3
25         resultado ← (numero1*numero2)
26         Escreva ("O resultado da multiplicação é: ",resultado)
27     caso 4
28         resultado ← (numero1/numero2)
29         Escreva ("O resultado da divisão é: ",resultado)
30     outro caso
31         Escreva ("Opção inválida")
32 FimEscolha
33 FimAlgoritmo
```

Exercícios em sala

Lista 6.1

Desenvolva um algoritmo, nomeado `Estrutura_Decisao_Simples`, que solicita ao usuário que digite um número no teclado e, caso o número digitado seja par, mostre na tela o seguinte texto: “O número digitado é par”.

Lista 6.2

Modifique o algoritmo do exercício anterior, alterando seu nome para `Estrutura_Decisao_Composta` e acrescente a possibilidade de mostrar o texto “O número digitado é ímpar”, quando for essa a situação.

Exercícios em sala

Algoritmo "Lista 06_01 Estrutura_Decisao_Simples"

// Exercício 2 Cap 6, em sala, disciplina de IC

Var

numero: inteiro

Inicio

escreva("Digite um número inteiro: ")

leia(numero)

se (numero%2=0) então

 escreva(numero, " é número par")

fimse

Fimalgoritmo

Exercícios em sala

VISUALG 3.0.7.0 * Interpretador e Editor de Algoritmos * última atualização: 03 de Outubro de 2010

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)

0,05s

Área dos algoritmos (Edição do código fonte)

```
1 Algoritmo "Estrutura_Decisao_Simples"  
2 // Exercício 2 Cap 6, em sala, disciplina de IC  
3 Var  
4   numero: inteiro  
5 Inicio  
6   escreva("Digite um número inteiro: ")  
7   leia(numero)  
8   se (numero%2=0) então  
9     escreva(numero, " é número par")  
10  fimse  
11 Fimalgoritmo  
12
```

C:\> Console simulando o modo texto do MS-DOS

```
Digite um número inteiro: 4  
4 é número par  
>>> Fim da execução do programa !
```

C:\> Console simulando o modo texto do MS-DOS

```
Digite um número inteiro: 3  
>>> Fim da execução do programa !
```

Exercícios em sala

Lista 6.1

Desenvolva um algoritmo, nomeado `Estrutura_Decisao_Simples`, que solicita ao usuário que digite um número no teclado e, caso o número digitado seja par, mostre na tela o seguinte texto: “O número digitado é par”.

Lista 6.2

Modifique o algoritmo do exercício anterior, alterando seu nome para `Estrutura_Decisao_Composta` e acrescente a possibilidade de mostrar o texto “O número digitado é ímpar”, quando for essa a situação.

Algoritmo "Lista 06_02 Estrutura_Decisao_Composta"

// Exercício 2 Cap 6, em sala, disciplina de IC

Var

numero: inteiro

Inicio

escreva("Digite um número inteiro: ")

leia(numero)

se (numero%2=0) então

 escreva(numero, " é número par")

senão

 escreva(numero, " é número impar")

fimse

Fimalgoritmo

Exercícios em sala

VISUALG 3.0.7.0 * Interpretador e Editor de Algoritmos * última atualização: 03 de Outubro de 2010

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)

0,05s

Área dos algoritmos (Edição do código fonte)

```
1 Algoritmo "Estrutura_Decisao_Simples"
2 // Exercício 2 Cap 6, em sala, disciplina de IC
3 Var
4   numero: inteiro
5 Inicio
6   escreva("Digite um número inteiro: ")
7   leia(numero)
8   se (numero%2=0) então
9     escreva(numero, " é número par")
10  senão
11    escreva(numero, " é número impar")
12  fimse
13 Fimalgoritmo
14
```

C:\> Console simulando o modo texto do MS-DOS

```
Digite um número inteiro: 8
8 é número par
>>> Fim da execução do programa !
```

C:\> Console simulando o modo texto do MS-DOS

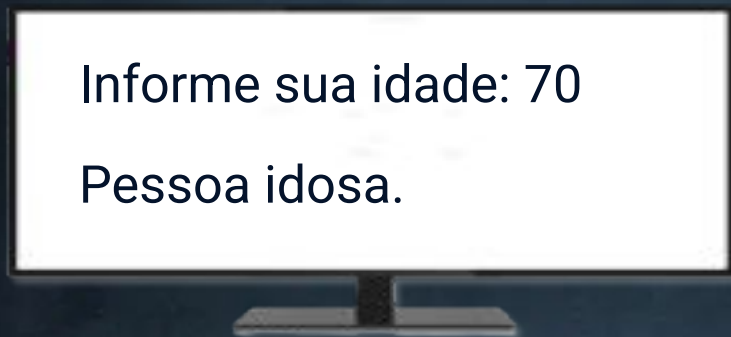
```
Digite um número inteiro: 3
3 é número impar
>>> Fim da execução do programa !
```

Exercícios em sala

Lista 6.4

Escreva e implemente um algoritmo que receba como entrada, via teclado, a idade de uma pessoa e exiba a informação se ela é menor de idade, maior de idade ou idoso, conforme ela possua menos que 18 anos, entre 18 e 65 anos ou acima de 65 anos, respectivamente.

Exemplo:



Algoritmo "Lista 06_04 Ver Faixa Etária"

// Exercício avulso do Cap 6, em sala, disciplina de IC

Var

idade: inteiro

Início

escreva("Digite a idade: ")

leia(idade)

se (idade<=18) então

| escreva(idade, ": jovem!")

senão

| se (idade>18) e (idade<=65) então

| | escreva(idade, ": adulto!")

| senão

| | escreva(idade, ": idoso!")

| fimse

fimse

Fimalgoritmo

Exercícios em sala

VISUALG 3.0.7.0 * Interpretador e Editor de Algoritmos * última atualização: 03 de Outubro de 2015 * Es

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)



Área dos algoritmos (Edição do código fonte) -> Nome do

```
1 Algoritmo "Lista 06_04 Ver Faixa Etária"  
2 // Exercício avulso do Cap 6, em sala, disciplina de IC  
3 Var  
4   idade: inteiro  
5 Inicio  
6   escreva("Digite a idade: ")  
7   leia(idade)  
8   se (idade<=18) então  
9     escreva(idade, ": jovem!")  
10  senão  
11    se (idade>18) e (idade<=65) então  
12      escreva(idade, ": adulto!")  
13    senão  
14      escreva(idade, ": idoso!")  
15    fimse  
16  fimse  
17 Fimalgoritmo  
18
```

Console simulando o modo texto do MS-DOS

```
Digite a idade: 12  
12: jovem!  
>>> Fim da execução do programa !
```

Console simulando o modo texto do MS-DOS

```
Digite a idade: 22  
22: adulto!  
>>> Fim da execução do programa !
```

Console simulando o modo texto do MS-DOS

```
Digite a idade: 70  
70: idoso!  
>>> Fim da execução do programa !
```

Exercícios em sala

Exercício extra-lista

Escreva um algoritmo no qual se informe um número (código) e o programa devolverá a Arma correspondente, conforme a tabela:

1	Infantaria
2	Cavalaria
3	Artilharia
4	Engenharia
5	Comunicações

Exemplo:

Informe o código: 2

Arma: Cavalaria

Exercícios em sala

Algoritmo "Escolha de Arma"

// Exercício avulso do Cap 6, IC

Var

codigo: inteiro

Inicio

```
escreva("Digite o código: ")
leia(codigo)
escolha codigo
    caso 1
        escreva("Infantaria")
    caso 2
        escreva("Cavalaria")
    caso 3
        escreva("Artilharia")
    caso 4
        escreva("Engenharia")
    caso 5
        escreva("Comunicações")
    outro caso
        escreva("Código incorreto")
fimEscolha
Fimalgoritmo
```

Exercícios

C:\ Console simulando o modo texto do MS-DOS

```
Digite o código: 1
Infantaria
>>> Fim da execução do programa !
```

C:\ Console simulando o modo texto do MS-DOS

```
Digite o código: 2
Cavalaria
>>> Fim da execução do programa !
```

C:\ Console simulando o modo texto do MS-DOS

```
Digite o código: 5
Comunicações
>>> Fim da execução do programa !
```

VISUALG 3.0.7.0 * Interpretador e Editor de Algoritmos * última atualização: 03 de Outubro de 2015 *

Arquivo Editar Run (executar) Exportar para Manutenção Help (Ajuda)

0,05s

Área dos algoritmos (Edição do código fonte) -> Nome d

```
1 Algoritmo "Escolha de Arma"
2 // Exercício avulso do Cap 6, em sala, disciplina de IC
3 Var
4     codigo: inteiro
5 Início
6     escreva("Digite o código: ")
7     leia(codigo)
8     escolha codigo
9         caso 1
10            escreva("Infantaria")
11        caso 2
12            escreva("Cavalaria")
13        caso 3
14            escreva("Artilharia")
15        caso 4
16            escreva("Engenharia")
17        caso 5
18            escreva("Comunicações")
19        outro caso
20            escreva("Código incorreto")
21    fimEscolha
22 Fimalgoritmo
```



Bug: "outro caso" inoperante