



Universidade Positivo

Algoritmos de Programação

Aula 07

Prof.^a Mariane Cassenote

mariane.cassenote@up.edu.br

Conteúdo



- Retomada dos principais pontos da aula anterior
- Operadores lógicos e relacionais
- Estruturas condicionais simples, compostas e aninhadas (SE)
- Continuidade de introdução à programação em linguagem C

Ao final do componente curricular será possível utilizar estruturas condicionais simples e compostas para resolver problemas algorítmicos.



Na aula anterior...

Comandos de entrada e saída

// Portugol WebStudio

```
programa {  
    funcao inicio() {  
  
        real altura, peso, imc  
  
        escreva("Digite o peso em quilos: ")  
        leia(peso)  
        escreva("Digite a altura em metros: ")  
        leia(altura)  
  
        imc = peso / (altura * altura)  
  
        escreva("O IMC da pessoa eh: ", imc)  
  
    }  
}
```

// Linguagem C

```
#include <stdio.h>  
  
int main() {  
    float altura, peso, imc;  
  
    printf("Digite o peso em quilos: ");  
    scanf("%f", &peso);  
    printf("Digite a altura em metros: ");  
    scanf("%f", &altura);  
  
    imc = peso / (altura * altura);  
  
    printf("O IMC da pessoa eh: %f", imc);  
  
    return 0;  
}
```

Expressões lógicas



- São expressões cujos resultados só podem ser **verdadeiro** ou **falso**
- São compostas por:
 - Operadores relacionais
 - Operadores lógicos

Operadores relacionais



- São aqueles que comparam dois valores do mesmo tipo
- O retorno da expressão indica se o resultado da comparação foi **VERDADEIRO** ou **FALSO**
- Exemplos:
 - $2 < 3$ é uma expressão válida que retorna VERDADEIRO
 - $2 = 8$ é uma expressão válida que retorna FALSO

Operadores relacionais



Operador	Símbolo
Igual a	==
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=
Diferente de	!=

Expressões aritméticas e relacionais



Exemplos:

- `a >= b`
- `(3 + 7) < (2 * 4)`
- `4 >= ((3 - 1) * 2)`
- `(consumo_medio * qtde_pessoas) <= qtde_disponivel`

Percebam que a maioria dos exemplos utiliza operadores relacionais em conjunto com operadores aritméticos

Exemplo do pedido

```
programa {  
    funcao inicio() {  
        inteiro pedido_minimo, qtde_pedido  
        inteiro qtde_estoque  
        logico respeita_minimo, respeita_estoque  
  
        pedido_minimo = 10  
        qtde_pedido = 20  
        qtde_estoque = 18
```

→ *(segue no quadro ao lado)*

```
        respeita_minimo = (qtde_pedido >=  
        pedido_minimo)  
  
        escreva("Respeita minimo: " + respeita_minimo  
        + "\n")  
  
        respeita_estoque = (qtde_pedido <=  
        qtde_estoque)  
  
        escreva("Respeita estoque: " +  
        respeita_estoque + "\n")  
    }  
}
```

```
Respeita minimo: verdadeiro  
Respeita estoque: falso
```

```
Programa finalizado. Tempo de execução: 29 ms
```

Expressões relacionais



Com o comando **escreva** é possível escrever valores de qualquer tipo existente, como valores reais, valores lógicos, valores inteiros, do tipo sequência de caracteres, resultados de expressões aritméticas, resultados de expressões lógicas, resultados de expressões relacionais.

```
escreva(8>9)
```

```
falso  
Programa finalizado. Tempo de execução: 22 ms
```

```
escreva("8>9")
```

```
8>9  
Programa finalizado. Tempo de execução: 20 ms
```

Operadores Lógicos



- São utilizados para representar situações lógicas que não podem ser representadas por operadores aritméticos
- Também são chamados de **conectivos lógicos** por unirem duas expressões simples em uma composta
- Podem ser operadores **binários** (que operam duas sentenças ou expressões) ou **unário** (que opera em uma só sentença)

Operadores Lógicos - NAO



- Operador unário de **negação** ou **nao** lógico, representado por **!** ou **NOT** ou **~**
- Gera como saída um valor inverso ao valor lógico de entrada
- Sempre deve aparecer à esquerda da variável ou expressão cujo valor deve modificar
- Exemplos:
 - **nao**(varA)
 - **nao**(5 > 3)

Operadores Lógicos - NAO

```
programa {  
    funcao inicio() {  
        inteiro x, y  
        logico l, m  
  
        x = 1  
        y = 2  
        l = (x > y)  // FALSO  
        m = nao(x > y)  // VERDADEIRO  
    }  
}
```

Operadores Lógicos - NAO

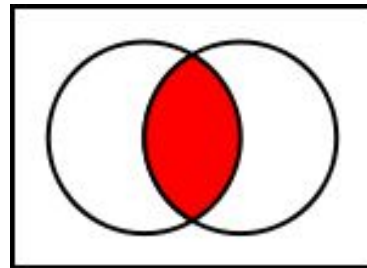


- Tabela verdade do operador **NAO**:

A	NAO (A)
V	F
F	V

Operadores Lógicos - E

- Operador binário de **conjunção** ou **E** lógico, representado por \wedge ou **AND** ou **E** ou **&&**
- Quando duas expressões são unidas por esse operador, a expressão resultante só é verdadeira se **ambas as expressões** que a constituem também são **verdadeiras**
- Se uma das expressões não ocorrer, a expressão como um todo é falsa
- Exemplos:
 - `(varA) e (varB)`
 - `(altura <= altura_maxima) e (largura <= largura_maxima)`



Operadores Lógicos - E

```
programa {  
    funcao inicio() {  
  
        inteiro x, y, z  
        logico l, m, n, o, p  
  
        x = 1  
        y = 2  
        z = 3
```

→ *(segue no quadro ao lado)*

```
        l = (x > y)  
        m = (z > y)  
        n = verdadeiro  
  
        o = (l) e (m)  
        p = (m) e (n)  
  
    }  
}
```


Operadores Lógicos - E

```
programa {  
    funcao inicio() {  
  
        inteiro x, y, z  
        logico l, m, n, o, p  
  
        x = 1  
        y = 2  
        z = 3
```

→ *(segue no quadro ao lado)*

```
l = (x > y)      // FALSO  
m = (z > y)      // VERDADEIRO  
n = verdadeiro  // VERDADEIRO  
  
o = (l) e (m)    // FALSO  
p = (m) e (n)    // VERDADEIRO  
  
    }  
}
```

Operadores Lógicos - E



- Tabela verdade do operador E:

A	B	(A) e (B)
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos - E

- Exemplo:

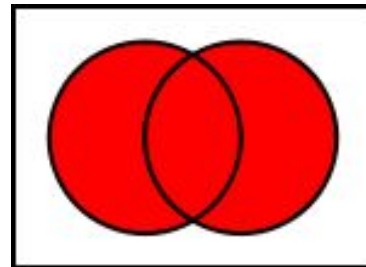
- $x = 10$
- $y = 20$

$x == y$	$x <= y$	$(x == y) \text{ e } (x <= y)$
F	V	F

$x != y$	$x <= y$	$(x != y) \text{ e } (x <= y)$
V	V	V

Operadores Lógicos - OU

- Operador binário de **disjunção** ou **OU** lógico, representado por V ou **OR** ou **OU** ou **||**
- Quando duas expressões são unidas por esse operador, a expressão resultante é verdadeira se **pelo menos uma** expressão que a constitui também é **verdadeira**
- Se nenhuma das expressões for verdadeira, a expressão como um todo é falsa
- Exemplos:
 - `(varA) ou (varB)`
 - `(altura > altura_maxima) ou (largura > largura_maxima)`



Operadores Lógicos - OU

```
programa {  
    funcao inicio() {  
  
        inteiro x, y, z  
        logico l, m, n, o, p  
  
        x = 1  
        y = 2  
        z = 3
```

→ *(segue no quadro ao lado)*

```
l = (x > y)
```

```
m = (z > y)
```

```
n = falso
```

```
o = (l) ou (m)
```

```
p = (l) ou (n)
```

```
}
```

```
}
```

Operadores Lógicos - OU

```
programa {  
    funcao inicio() {  
  
        inteiro x, y, z  
        logico l, m, n, o, p  
  
        x = 1  
        y = 2  
        z = 3
```

→ (segue no quadro ao lado)

```
        l = (x > y)    // FALSO  
        m = (z > y)    // VERDADEIRO  
        n = falso      // FALSO  
  
        o = (l) ou (m) // VERDADEIRO  
        p = (l) ou (n) // FALSO  
  
    }  
}
```

Operadores Lógicos - OU



- Tabela verdade do operador **ou**:

A	B	(A) ou (B)
V	V	V
V	F	V
F	V	V
F	F	F

Operadores Lógicos - OU

- Exemplo:

- $x = 10$
- $y = 20$

$x == y$	$x <= y$	$(x == y) \text{ ou } (x <= y)$
F	V	V

$x == y$	$x >= y$	$(x == y) \text{ ou } (x <= y)$
F	F	F

Operadores Lógicos



Exemplos:

$(2 < 3) \text{ E } (5 > 1)$

V E V

V

$(5 < 4) \text{ OU } (12 > 2)$

F OU V

V

Operadores Lógicos - Síntese



Operador lógico	Tipo	Resultado
NAO	unário	Verdadeiro se o operando for falso, falso se o operando for verdadeiro
E	binário	Verdadeiro somente se ambos os operandos são verdadeiros
OU	binário	Verdadeiro se pelo menos um dos operandos for verdadeiro

Operadores Lógicos - Síntese



A	B	NAO (A)	(A) E (B)	(A) OU (B)
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

Precedência geral de operadores

Ordem	Operador	Símbolo
1º	Parênteses	()
2º	Potenciação	^ ou **
3º	Multiplicação, divisão, resto e divisão inteira	*, /, % ou mod, \ ou div
4º	Adição, subtração	+ , -
5º	Relacional	==, <, >, <=, >=, !=
6º	Negação	nao, ~, !
7º	Conjunção	e, ^, &&
8º	Disjunção	ou, v,



Estruturas condicionais

simples, compostas e aninhadas

Estruturas condicionais



- **Estruturas condicionais** ou **estruturas de decisão** permitem a escolha de um conjunto de instruções a serem executadas quando determinada condição aconteça
- Qualquer condição / expressão lógica pode ser utilizada, desde que o resultado seja verdadeiro ou falso
 - Utilizar expressões com operadores relacionais, lógicos ou variáveis do tipo lógico

Estrutura condicional simples



- Estrutura condicional IF

```
if (<condição>) {  
  
    <bloco de comandos>  
  
}
```

ATENÇÃO
para o uso de {}

Caso o resultado da condição for **VERDADEIRO**, será executado o bloco de comandos que está dentro da estrutura

Caso o resultado seja **FALSO**, a execução do programa ignora o bloco de comandos

Estrutura condicional simples

// Portugol WebStudio

```
programa {
    funcao inicio() {

        inteiro idade

        escreva("Digite a idade da pessoa: ")
        leia(idade)

        se (idade < 18) {
            escreva("A pessoa eh menor de idade \n")
        }

        escreva("A idade digitada foi:", idade)

    }
}
```

// Linguagem C

```
#include <stdio.h>

int main() {

    int idade;

    printf("Digite a idade da pessoa: ");
    scanf("%d", &idade);

    if (idade < 18) {
        printf("A pessoa eh menor de idade \n");
    }

    printf("A idade digitada foi: %d", idade);

    return 0;

}
```


Estrutura condicional simples

// Portugol WebStudio

```
programa {
    funcao inicio() {

        inteiro idade

        escreva("Digite a idade da pessoa")
        leia(idade)

        se (idade < 18) {
            escreva("A pessoa eh menor de idade \n")
        }

        escreva("A idade digitada foi: ")

    }
}
```

```
Digite a idade da pessoa: 15
A pessoa eh menor de idade
A idade digitada foi: 15
```

```
Digite a idade da pessoa: 21
A idade digitada foi: 21
```

// Linguagem C

```
#include <stdio.h>

int main() {

    printf("Digite a idade da pessoa: ");
    scanf("%d", &idade);

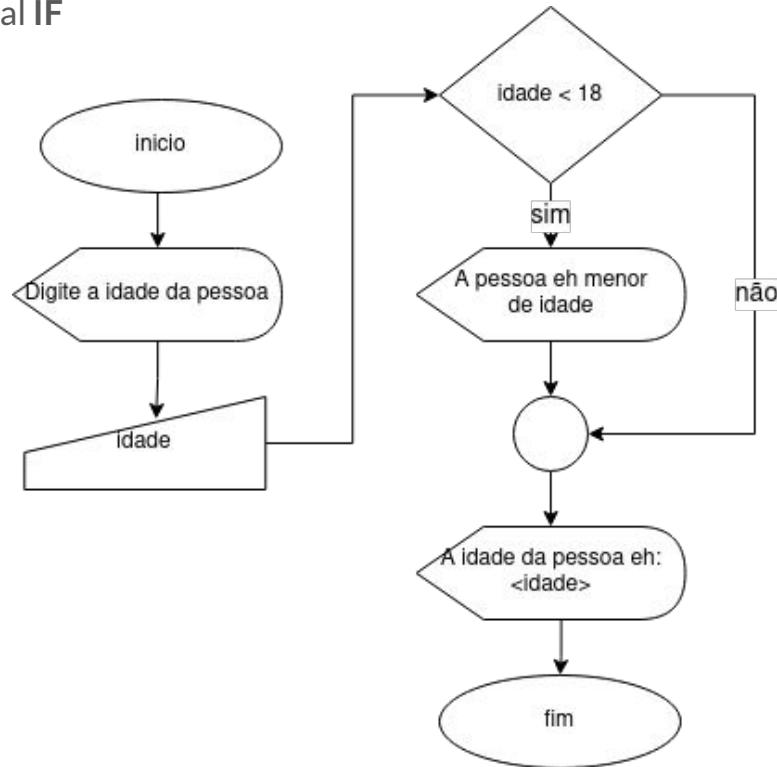
    if (idade < 18) {
        printf("A pessoa eh menor de idade \n");
    }

    printf("A idade digitada foi: %d", idade);

    return 0;
}
```

Estrutura condicional simples

- Estrutura condicional IF



Estrutura condicional composta

- Estrutura condicional IF-ELSE

```
if (<condição>) {  
    <bloco de comandos VERDADEIRO>  
} else {  
    <bloco de comandos FALSO>  
}
```

Caso o resultado da condição for **VERDADEIRO**, será executado o bloco de comandos VERDADEIRO

Caso o resultado seja **FALSO**, será executado o bloco de comandos FALSO

Estrutura condicional composta

// Portugol WebStudio

```
programa {
    funcao inicio() {
        real notaA, notaB, media

        escreva("Digite a notaA: ")
        leia(notaA)

        escreva("Digite a notaB: ")
        leia(notaB)

        media = (notaA + notaB) / 2

        se (media >= 6.0) {
            escreva("Aprovado")
        }
        senao {
            escreva("Reprovado")
        }
    }
}
```

// Linguagem C

```
#include <stdio.h>

int main() {
    float notaA, notaB, media;

    printf("Digite a notaA: ");
    scanf("%f", &notaA);

    printf("Digite a notaB: ");
    scanf("%f", &notaB);

    media = (notaA + notaB) / 2;

    if (media >= 6.0) {
        printf("Aprovado");
    }
    else {
        printf("Reprovado");
    }

    return 0;
}
```

Estrutura condicional composta

// Portugol WebStudio

```
programa {  
    funcao inicio() {  
        real notaA, notaB, media  
        escreva("Digite a notaA: ")  
        leia(notaA)  
  
        escreva("Digite a notaB: ")  
        leia(notaB)  
  
        media = (notaA + notaB) / 2  
  
        se (media >= 6.0) {  
            escreva("Aprovado")  
        }  
        senao {  
            escreva("Reprovado")  
        }  
    }  
}
```

```
Digite a notaA: 5.9  
Digite a notaB: 8.1  
Aprovado
```

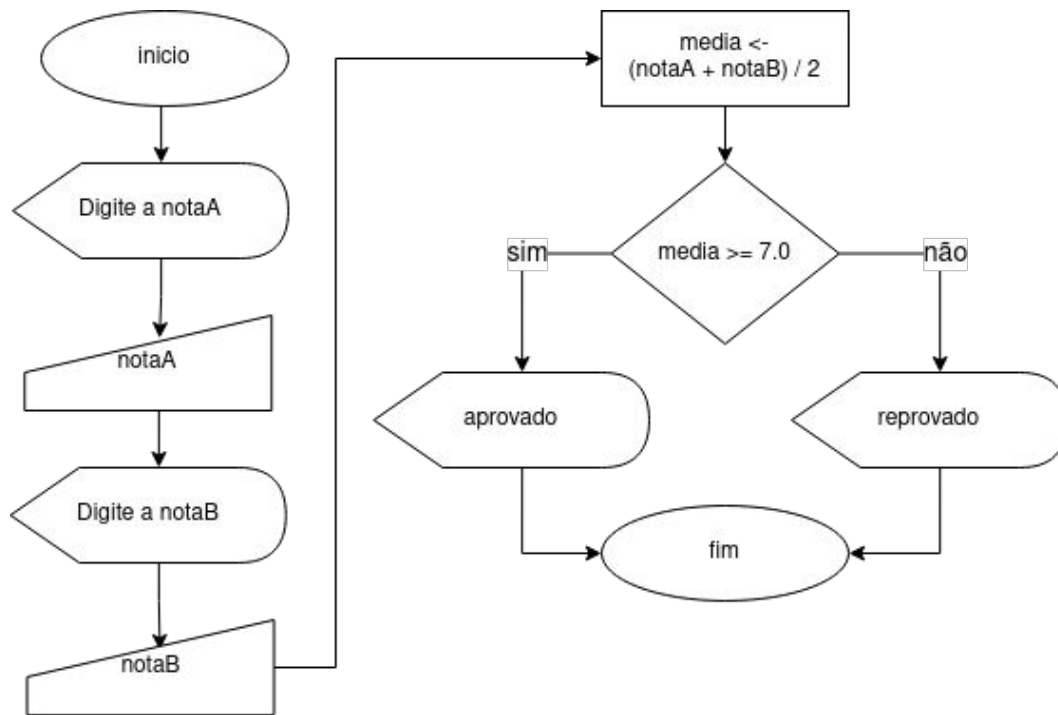
```
Digite a notaA: 5.6  
Digite a notaB: 3.5  
Reprovado
```

// Linguagem C

```
#include <stdio.h>  
  
int main() {  
    float notaA, notaB, media;  
    printf("Digite a notaA: ");  
    scanf("%f", &notaA);  
    printf("Digite a notaB: ");  
    scanf("%f", &notaB);  
  
    media = (notaA + notaB) / 2;  
  
    if (media >= 6.0) {  
        printf("Aprovado");  
    }  
    else {  
        printf("Reprovado");  
    }  
  
    return 0;  
}
```

Estrutura condicional composta

- Estrutura condicional IF-ELSE



Estruturas condicionais aninhadas



- Muitas vezes será necessário tomar uma nova decisão dentro de um fluxo condicional, ou
- Pode ser que tenhamos mais de duas opções de fluxo de execução
- Em ambos os casos, podemos utilizar estruturas condicionais aninhadas, que representam uma estrutura condicional inserida dentro de outra

Estruturas condicionais aninhadas

```
programa {  
    funcao inicio() {  
  
        real notaA, notaB, media  
  
        escreva("Digite a notaA: ")  
        leia(notaA)  
  
        escreva("Digite a notaB: ")  
        leia(notaB)  
  
        media = (notaA + notaB) / 2  
  
        se (media >= 6.0) {  
            escreva("Aprovado")  
        }  
        senao se (media >= 4.0) {  
            escreva("Em recuperacao")  
        }  
        senao {  
            escreva("Reprovado")  
        }  
    }  
}
```

```
#include <stdio.h>  
  
int main() {  
    float notaA, notaB, media;  
  
    printf("Digite a notaA: ");  
    scanf("%f", &notaA);  
  
    printf("Digite a notaB: ");  
    scanf("%f", &notaB);  
  
    media = (notaA + notaB) / 2;  
  
    if (media >= 6.0) {  
        printf("Aprovado");  
    }  
    else if (media >= 4.0) {  
        printf("Em recuperacao");  
    }  
    else {  
        printf("Reprovado");  
    }  
  
    return 0;  
}
```


Estruturas condicionais aninhadas

```
Digite a notaA: 3.7
Digite a notaB: 4.1
Reprovado
```

```
escreva("Digite a notaA: ")
leia(notaA)
```

```
escreva("Digite a notaB: ")
leia(notaB)
```

```
media = (notaA + notaB) / 2
```

```
se (media >= 6.0) {
    escreva("Aprovado")
}
senao se (media >= 4.0) {
    escreva("Em recuperacao")
}
senao {
    escreva("Reprovado")
}
}
```

```
Digite a notaA: 4.7
Digite a notaB: 5.8
Em recuperacao
```

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

```
int main() {
    float notaA, notaB, media;

    printf("Digite a notaA: ");
    scanf("%f", &notaA);

    printf("Digite a notaB: ");
    scanf("%f", &notaB);
```

```
    media = (notaA + notaB) / 2;

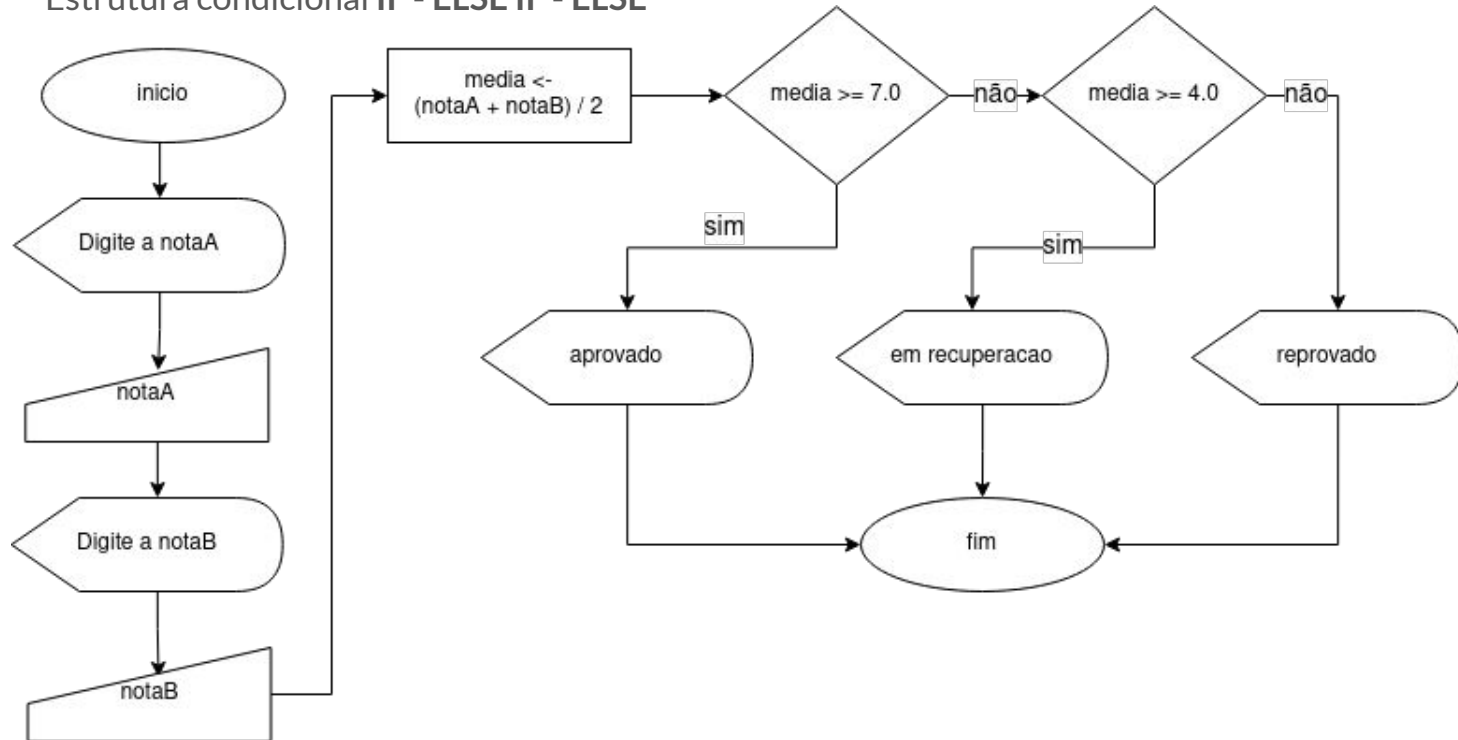
    if (media >= 6.0) {
        printf("Aprovado");
    }
    else if (media >= 4.0) {
        printf("Em recuperacao");
    }
    else {
        printf("Reprovado");
    }

    return 0;
}
```

```
Digite a notaA: 8.7
Digite a notaB: 6.8
Aprovado
```

Estruturas condicionais aninhadas

- Estrutura condicional IF - ELSE IF - ELSE



Condições compostas em estruturas condicionais



- É possível compor expressões lógicas (condições) utilizando operadores relacionais
 - `==`, `!=`, `<`, `>`, `<=`, `>=`
- Também é possível compor expressões lógicas utilizando operadores lógicos
 - `nao(!)`, `e(&&)` e `ou(||)`
- Então a expressão lógica que será avaliada também pode ser uma expressão lógica composta

Condições compostas em estruturas condicionais

```
programa {
    funcao inicio() {

        real freq, media

        escreva("Digite a media: ")
        leia(media)
        escreva("Digite a frequencia: ")
        leia(freq)

        se ((media >= 7.0) e (freq >= 75.0)) {
            escreva("Aprovado")
        }
        senao se ((media >= 4.0) e (freq >= 75.0)) {
            escreva("Em recuperacao")
        }
        senao {
            escreva("Reprovado")
        }
    }
}
```

```
#include <stdio.h>

int main() {

    float freq, media;

    printf("Digite a media: ");
    scanf("%f", &media);

    printf("Digite a frequencia: ");
    scanf("%f", &freq);

    if ((media >= 7.0) && (freq >= 75.0)) {
        printf("Aprovado");
    }
    else if ((media >= 4.0) && (freq >= 75.0)) {
        printf("Em recuperacao");
    }
    else {
        printf("Reprovado");
    }

    return 0;
}
```

Condições compostas em estruturas condicionais

```
programa {  
    funcao inicio() {  
  
        real freq, media  
        Digite a media: 8.7  
        escreva(media)  
        leia(freq)  
        Digite a frequencia: 94.0  
        escreva(freq)  
        leia(freq)  
  
        se ((media >= 7.0) e (freq >= 75.0)) {  
            escreva("Aprovado")  
        }  
        senao {  
            Digite a media: 9.25  
            Digite a frequencia: 87.5  
            escreva("Reprovado")  
        }  
    }  
}
```

```
#include <stdio.h>  
  
int main() {  
  
    float media, freq  
    Digite a media: 4.8  
    printf("Digite a frequencia: ");  
    scanf("%f", &freq)  
    Digite a frequencia: 95.0  
    Em recuperacao:   
  
    printf("Digite a frequencia: ");  
    scanf("%f", &freq);  
  
    if ((media >= 7.0) && (freq >= 75.0)) {  
        printf("Aprovado");  
    }  
    else if ((media < 7.0) && (freq < 75.0)) {  
        Digite a media: 3.8  
        Digite a frequencia: 55.0  
        Reprovado:   
    }  
    else {  
        Digite a media: 3.8  
        Digite a frequencia: 55.0  
        Reprovado:   
    }  
  
    return 0;  
}
```

Outro jeito de aninhar condições

```
#include <stdio.h>

int main() {

    float freq, media;

    printf("Digite a media: ");
    scanf("%f", &media);

    printf("Digite a frequencia: ");
    scanf("%f", &freq);
```

→ *(segue no quadro ao lado)*

```
    if (freq >= 75.0) {

        if (media >= 6.0) {
            printf("Aprovado");
        } else if (media >= 4.0) {
            printf("Em recuperacao");
        } else {
            printf("Reprovado por media");
        }

    } else {
        printf("Reprovado por faltas");
    }

    return 0;
}
```

Outro jeito de aninhar condições

```
programa {  
  funcao inicio() {  
  
    real freq, media  
  
    escreva("Digite a media: ")  
    leia(media)  
  
    escreva("Digite a frequencia: ")  
    leia(freq)
```

→ *(segue no quadro ao lado)*

```
    se (freq >= 75.0) {  
  
        se (media >= 6.0) {  
            escreva("Aprovado")  
        } senao se (media >= 4.0) {  
            escreva("Em recuperacao")  
        } senao {  
            escreva("Reprovado por media")  
        }  
  
    } senao {  
        escreva("Reprovado por faltas")  
    }  
  
}
```

Exemplo da média e comparação

```
#include <stdio.h>

int main() {

    float num1, num2, media;

    printf("Digite o numero1: ");
    scanf("%f", &num1);

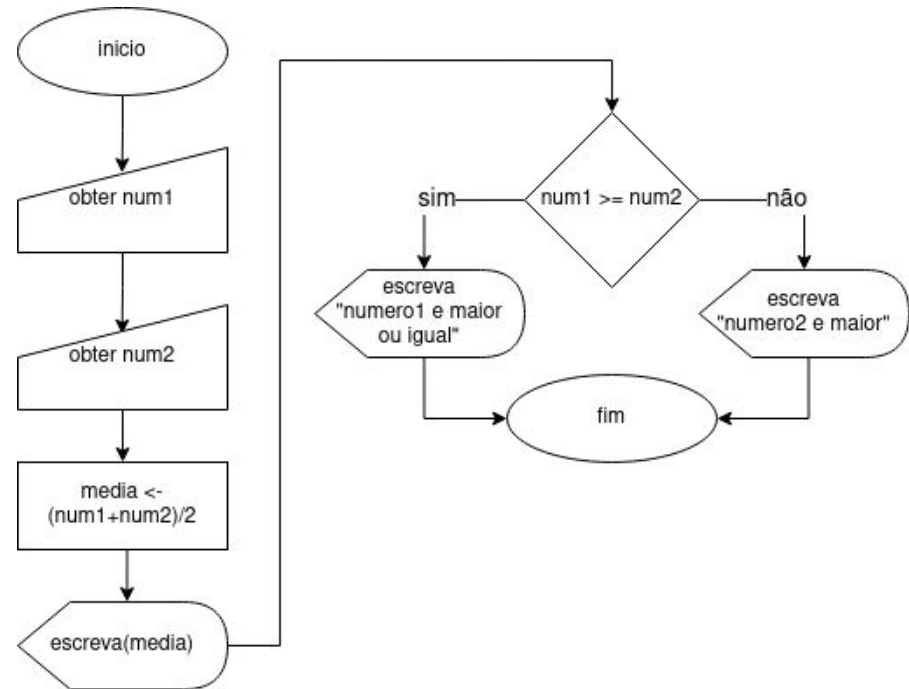
    printf("Digite o numero2: ");
    scanf("%f", &num2);

    media = (num1 + num2) / 2;

    printf("Media: %f \n", media);

    if (num1 >= num2) {
        printf("numero1 e maior ou igual");
    }
    else {
        printf("numero2 e maior");
    }

    return 0;
}
```



Estruturas condicionais – Síntese

ATENÇÃO para o uso de **{}**. O **else** é opcional e não tem condição.

```
if (<condição>) {  
    <bloco de comandos>  
}
```

```
if (<condição>) {  
    <bloco de comandos>  
} else {  
    <bloco de comandos>  
}
```

```
if (<condição1>) {  
    <bloco de comandos>  
} else if (<condição2>) {  
    <bloco de comandos>  
}
```

```
if (<condição1>) {  
    <bloco de comandos>  
} else if (<condição2>) {  
    <bloco de comandos>  
} else if (<condição3>) {  
    <bloco de comandos>  
} else {  
    <bloco de comandos>  
}
```

Estruturas condicionais – Síntese

ATENÇÃO para o uso de **{}**. O **senao** é opcional e não tem condição.

```
se (<condição>) {  
    <bloco de comandos>  
}
```

```
se (<condição>) {  
    <bloco de comandos>  
} senao {  
    <bloco de comandos>  
}
```

```
se (<condição1>) {  
    <bloco de comandos>  
} senao se (<condição2>) {  
    <bloco de comandos>  
}
```

```
se (<condição1>) {  
    <bloco de comandos>  
} senao se (<condição2>) {  
    <bloco de comandos>  
} senao se (<condição3>) {  
    <bloco de comandos>  
} senao {  
    <bloco de comandos>  
}
```

Próxima aula



- Retomada dos principais pontos desta aula
- Estruturas condicionais simples, compostas e aninhadas (SE, ESCOLHA)
- Continuidade da introdução à Linguagem C



Universidade Positivo

Algoritmos de Programação

Aula 07

Prof.^a Mariane Cassenote

mariane.cassenote@up.edu.br