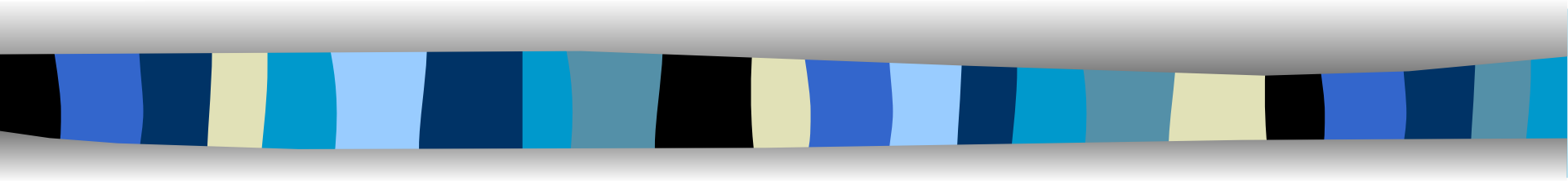
Estruturas de Controle



João Frederico Roldan Viana

jfredrv@gmail.com

(85)99231.2777

Agenda

- Estruturas de Controle
  - Estruturas de Decisão
    - if-else
    - switch
    - comando ternário

Estruturas de Decisão

- *if - else*
  - Utilizado quando for necessário escolher entre dois caminhos.
  - Sintaxe:

```
if (expressão) {
    bloco_de_comandos1;
} else {
    bloco_de_comandos2;
}
```

  - Comando *if*: expressão verdadeira ($\neq 0$).
  - Comando *else*: expressão falsa ($= 0$).

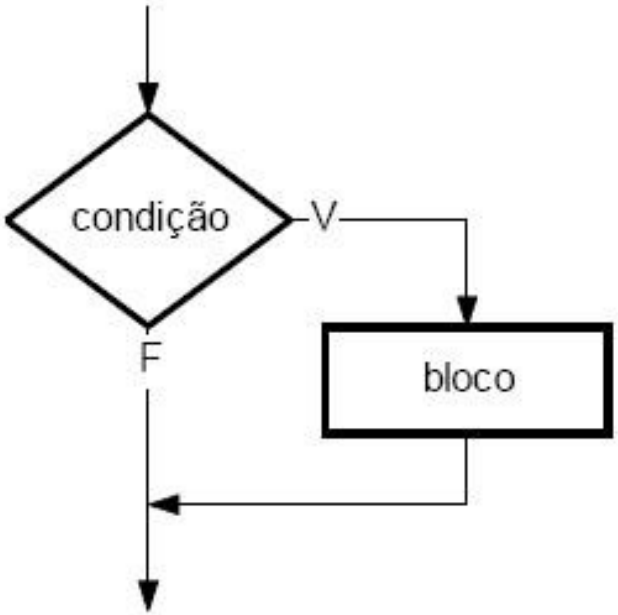
Estruturas de Decisão

- *if - else*
  - Decisão de um bloco (if . . .)
    - Sintaxe

```
if (expressão) {
    bloco_de_comandos1;
}
```

Estruturas de Decisão

- *if - else*
  - Decisão de um bloco (if . . .)
    - Fluxograma



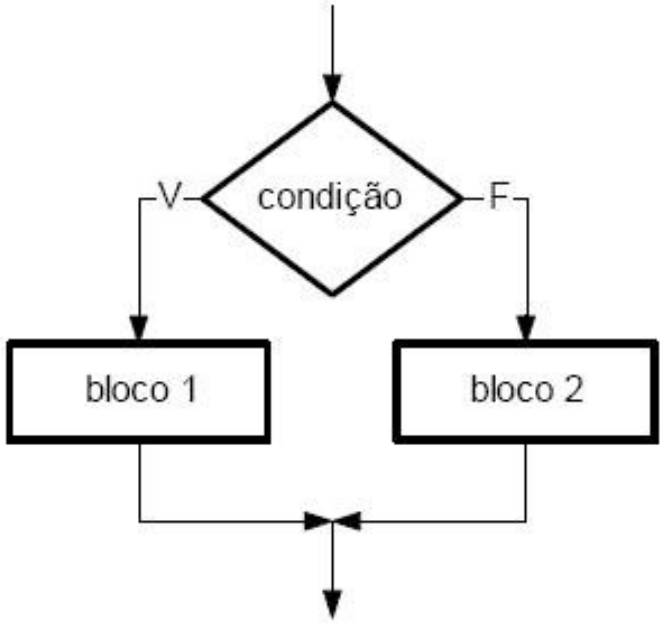
Estruturas de Decisão

- *if - else*
  - Decisão de dois blocos (if . . . else)
    - Sintaxe

```
if (expressão) {
    bloco_de_comandos1;
} else {
    bloco_de_comandos2;
}
```

Estruturas de Decisão

- *if - else*
  - Decisão de dois blocos (if . . . else)
    - Fluxograma



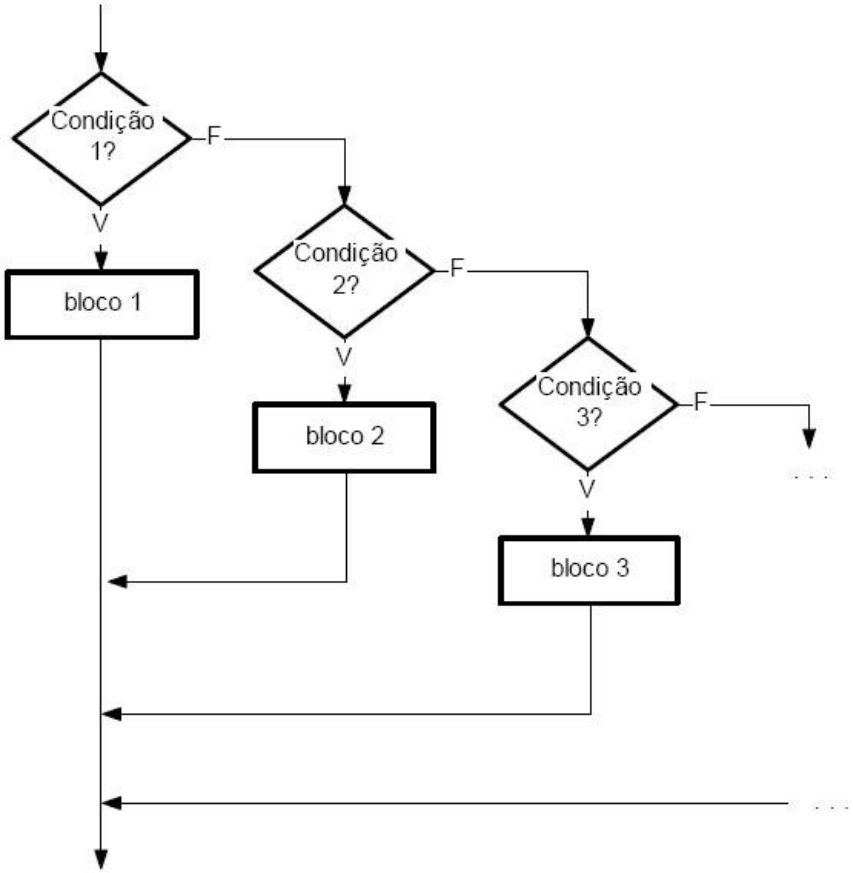
Estruturas de Decisão

- *if - else*
  - Decisão de múltiplos blocos (if . . . else if . . .)
    - Sintaxe

```
if (expressão) {
    bloco_de_comandos1;
. . .
} else {
    if (expressão N) {
        bloco_de_comandosN;
    } else {
        bloco_de_comandosP;
        }
}
```

Estruturas de Decisão

- *if - else*
  - Decisão de múltiplos blocos (if . . . else if . . .)
    - Fluxograma



Estruturas de Decisão

- *if - else*

Exemplo

```
if ((num1 >= num2) && (num1 >= num3)){
   printf("Maior numero: %d", num1);
}else {
     if (num2 >= num3){
        printf("Maior numero: %d", num2);
     }else {
          printf("Maior numero: %d", num3);
     }
}
```

Estruturas de Decisão

- *switch*
  - Utilizado para facilitar a escrita de trechos de programas em que a seleção deve ser feita entre várias alternativas.

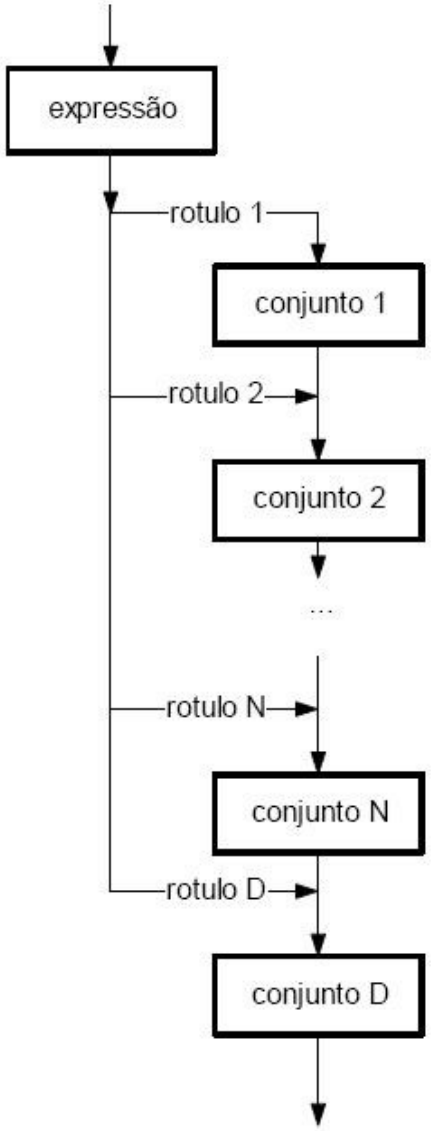
Estruturas de Decisão

- *switch*
  - Sintaxe:

```
switch (expressão) {
case const1:
bloco_de_comandos1;
break;
case const2:
bloco_de_comandos2;
break;
. . .
default:
 bloco_de_comandosP;
}
```

Estruturas de Decisão

- *switch*
  - Fluxograma:



Estruturas de Decisão

- *switch*
  - Passos da execução do comando *switch*:
    - 1. A expressão é avaliada;
    - 2. O resultado da expressão é comparado com os valores das constantes que aparecem nos case;
    - 3. Caso o resultado seja igual a uma das constantes, a execução se inicia a partir do comando associado a ela e continua até o fim do comando *switch*, ou até encontrar um *break*;
    - 4. Caso contrário, os comandos associados ao comando *default* são executados. Se o *default* não aparecer, nenhum comando será executado.

Estruturas de Decisão

- *switch*
  - Observações importantes:
    - O resultado da expressão deve ser um tipo enumerável: *int* ou *char*.
    - Caso não apareça um comando de desvio, todas as instruções seguintes ao teste *case* que teve sucesso serão executadas, mesmo as relacionadas com outros testes *case*.
    - O comando *switch* só pode testar igualdade.
    - Não podem aparecer duas constantes iguais em um *case*.

Estruturas de Decisão

- *switch*

Exemplo

```
switch (oper) {
   case '+':
      printf ("%f + %f = %f", num1, num2, num1+num2);
      break;
   case '-':
      printf ("%f - %f = %f", num1, num2, num1-num2);
      break;
   case '*':
      printf ("%f * %f = %f", num1, num2, num1*num2);
      break;
   default:
      printf ("Operacao invalida !");
}
```

Estruturas de Decisão

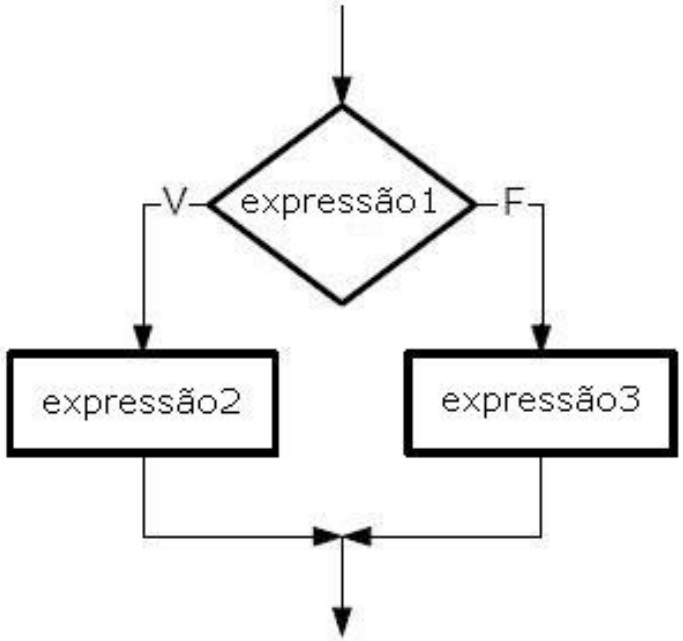
- *Comando ternário*
  - Necessita de três operadores para ser avaliado.
  - Sintaxe:

    expressão1 ? expressão2 : expressão3;

  - A execução desse comando ocorre da seguinte forma:
    - A expressão1 é avaliada.
    - Caso seja verdadeira ($\neq 0$), o resultado do comando será igual ao resultado da expressão2.
    - Caso contrário ($= 0$), o resultado do comando será igual ao resultado da expressão3.

Estruturas de Decisão

- *Comando ternário*
  - Fluxograma



Estruturas de Decisão

- *Comando ternário*

Exemplo

```
max = (num1 > num2) ? num1 : num2;
```