PUC Minas

Estruturas Condicionais

João Pedro Oliveira Batisteli

Fevereiro, 2025

Estrutura Sequencial

Forma Geral:

```
<Comando 1>;
<Comando 2>;
<Comando 3>;
     ...
<Comando n>;
```

Estrutura Sequencial

Forma Geral:

```
<Comando 1>;
<Comando 2>;
<Comando 3>;
     ...
<Comando n>;
```

Os comandos são separados por ponto e vírgula e executados de forma sequencial, ou seja, na ordem em que eles aparecem.

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...
- Se idade maior que 18, então, ...

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...
- Se idade maior que 18, então, ...
- Se eu ganhar na mega sena, então, ...

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...
- Se idade maior que 18, então, ...
- Se eu ganhar na mega sena, então, ...
- Se o meu time ganhar, então, ...

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...
- Se idade maior que 18, então, ...
- Se eu ganhar na mega sena, então, ...
- Se o meu time ganhar, então, ...
- Se eu passar em cálculo, então, ...

Estruturas Condicionais

Em nosso dia a dia, quase sempre, temos que tomar decisões:

- Se fizer sol, então, ...
- Se idade maior que 18, então, ...
- Se eu ganhar na mega sena, então, ...
- Se o meu time ganhar, então, ...
- Se eu passar em cálculo, então, ...

A programação é totalmente relacionada à tomada de decisões.

Estruturas Condicionais

- Em um programa, normalmente, os comandos são executados um após do outro, na sequência em que estiverem escritos.

Estruturas Condicionais

- Em um programa, normalmente, os comandos são executados um após do outro, na sequência em que estiverem escritos.
- Isso é chamado de execução sequencial.

Estruturas Condicionais

- Em um programa, normalmente, os comandos são executados um após do outro, na sequência em que estiverem escritos.
- Isso é chamado de execução sequencial.
- Existem comandos que permitem ao programador especificar o próximo comando a ser executado, que pode ser outro que não o próximo na sequência.

Estrutura Sequencial

- Um bloco de comandos é delimitado por ‘{‘ (indica início do bloco) e por ‘}’ (indica final do bloco).

Estrutura Sequencial

- Um bloco de comandos é delimitado por '{' (indica início do bloco) e por '}' (indica final do bloco).
- Qual componente dos códigos que vimos e escrevemos até agora tem essa característica?

Estrutura Sequencial

- Um bloco de comandos é delimitado por '{' (indica início do bloco) e por '}' (indica final do bloco).
- Qual componente dos códigos que vimos e escrevemos até agora tem essa característica?
- Os comandos pertencentes a esse bloco estarão dentro dessa delimitação.

Estrutura condicionais

- A estrutura de decisão "*if*" vem acompanhada de uma expressão, ou seja, se determinada condição for satisfeita pelo comando *if* então um determinado comando é executado.

Estrutura condicionais

- A estrutura de decisão "*if*" vem acompanhada de uma expressão, ou seja, se determinada condição for satisfeita pelo comando *if* então um determinado comando é executado.
- O comando a ser executado está dentro do bloco (delimitado por { ... }) do *if*.

Estrutura condicionais

- A estrutura de decisão "*if*" vem acompanhada de uma expressão, ou seja, se determinada condição for satisfeita pelo comando *if* então um determinado comando é executado.
- O comando a ser executado está dentro do bloco (delimitado por { ... }) do *if*.
- Dentro de cada bloco podem existir quantos comandos forem necessários.

Estrutura condicionais

- A estrutura de decisão "*if*" vem acompanhada de uma expressão, ou seja, se determinada condição for satisfeita pelo comando *if* então um determinado comando é executado.
- O comando a ser executado está dentro do bloco (delimitado por { ... }) do *if*.
- Dentro de cada bloco podem existir quantos comandos forem necessários.
- A estrutura condicional em algoritmos pode ser **simples** ou **composta**.

Estrutura condicionais

- A estrutura de decisão "*if*" vem acompanhada de uma expressão, ou seja, se determinada condição for satisfeita pelo comando *if* então um determinado comando é executado.
- O comando a ser executado está dentro do bloco (delimitado por { ... }) do *if*.
- Dentro de cada bloco podem existir quantos comandos forem necessários.
- A estrutura condicional em algoritmos pode ser **simples** ou **composta**.
- **Tipos:** if, if-else, if-else if

Estrutura condicional simples

Estrutura simples

```
SE condição
  ENTÃO:
  comando(s)
```

Exemplo em C

```
if(x>2){
  printf("X maior que 2");
}
```

Estrutura condicional simples

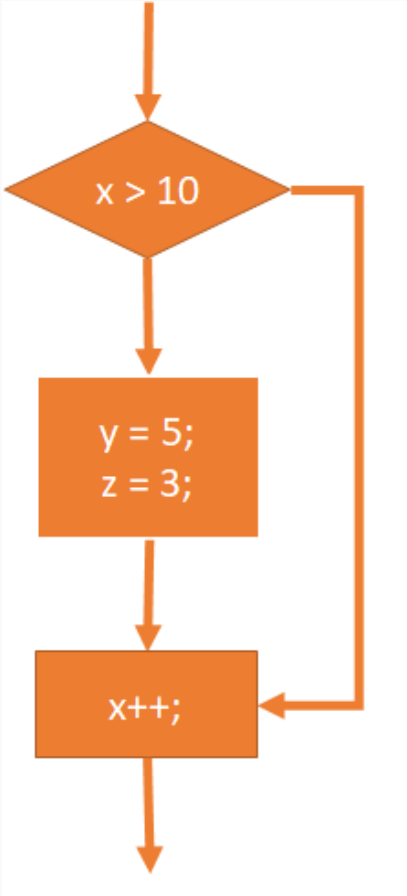
- O bloco de comandos 1 somente será executado se a condição for verdadeira, caso contrário o bloco inteiro será ignorado e o próximo comando depois da estrutura é avaliado.

```
if (condição A){
  comandos 1
}
.
.
.
```

Exemplo

Código em C

```
if(x>10){
  y=5;
  z=3;
}
x++;
if(x%2 == 0){
  y=z+1;
}
```



Exercício

1)Crie um programa que solicite ao usuário a digitação de um número inteiro. O programa deve verificar se o número digitado é menor que 20. Caso seja, o programa deve exibir a mensagem *"O número é menor que 20"* na tela.

Exercício

1)Crie um programa que solicite ao usuário a digitação de um número inteiro. O programa deve verificar se o número digitado é menor que 20. Caso seja, o programa deve exibir a mensagem *"O número é menor que 20"* na tela.

```
#include <stdio.h>
int main(){
  int x;
  printf("Digite um número inteiro");
  scanf("%d",&x);
  if(x<20){
    printf("O número recebido é menor que 20");
  }
  return 0;
}
```

Estrutura condicional composta

Estrutura composta

```
SE condição
  ENTÃO: comando(s)
SENÃO: comando(s)
```

Exemplo em C

```
if(x>2){
  printf("X maior que 2");
}
else{
  printf("X menor que 2");
}
```

Estrutura condicional composta

- O bloco de comandos 1 somente será executado se a condição for verdadeira, caso contrário (condição falsa) o bloco de comandos 1 será ignorado e o bloco de comandos 2 será executado.

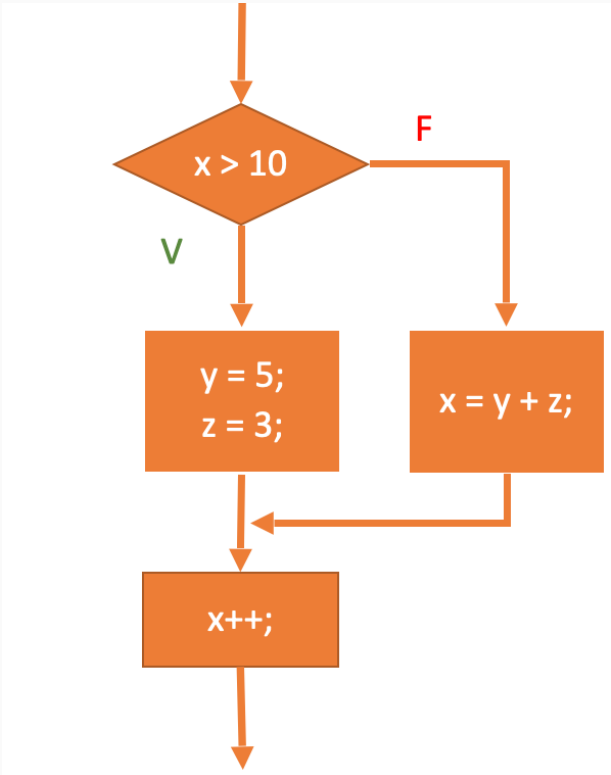
- **Somente um dos blocos é executado!**

```
if (condição A){
  comandos 1
}
else{
  comandos 2
}
```

Estrutura condicional composta

Código em C

```
if(x>10){
  y=5;
  z=3;
}
else{
  x=y+z;
}
x++;
```



Exercício

1) Crie um programa que solicite ao usuário a digitação de um número inteiro. O programa deve verificar se o número digitado é menor que 20. Caso seja, o programa deve exibir a mensagem *"O número é menor que 20"* na tela. Caso contrário, o programa não deve exibir a mensagem *"O número é maior ou igual a 20"*.

Exercício

1)Crie um programa que solicite ao usuário a digitação de um número inteiro. O programa deve verificar se o número digitado é menor que 20. Caso seja, o programa deve exibir a mensagem *"O número é menor que 20"* na tela. Caso contrário, o programa não deve exibir a mensagem *"O número é maior ou igual a 20"*.

```
#include <stdio.h>
int main(){
  int x;
  scanf("%d",&x);
  if(x<20){
    printf("O número é menor que 20");
  }
  else{
    printf("O número é maior ou igual a 20");
  }
}
```

Estrutura condicional composta

Estrutura composta

```
SE condição 1
  ENTÃO: comando(s)
SENÃO SE: condição 2
  ENTÃO: comando(s)
SENÃO SE: condição 3
  ENTÃO: comando(s)
...
SENÃO SE: condição N
  ENTÃO: comando(s)
SENÃO: comando(s)
```

Exemplo em C

```
if(x>2){
  //lista de comandos
}
else if(x<2){
  //lista de comandos
}
else if((x+5)==10){
  //lista de comandos
}
else{
  //lista de comandos
}
```

Estrutura if-else if

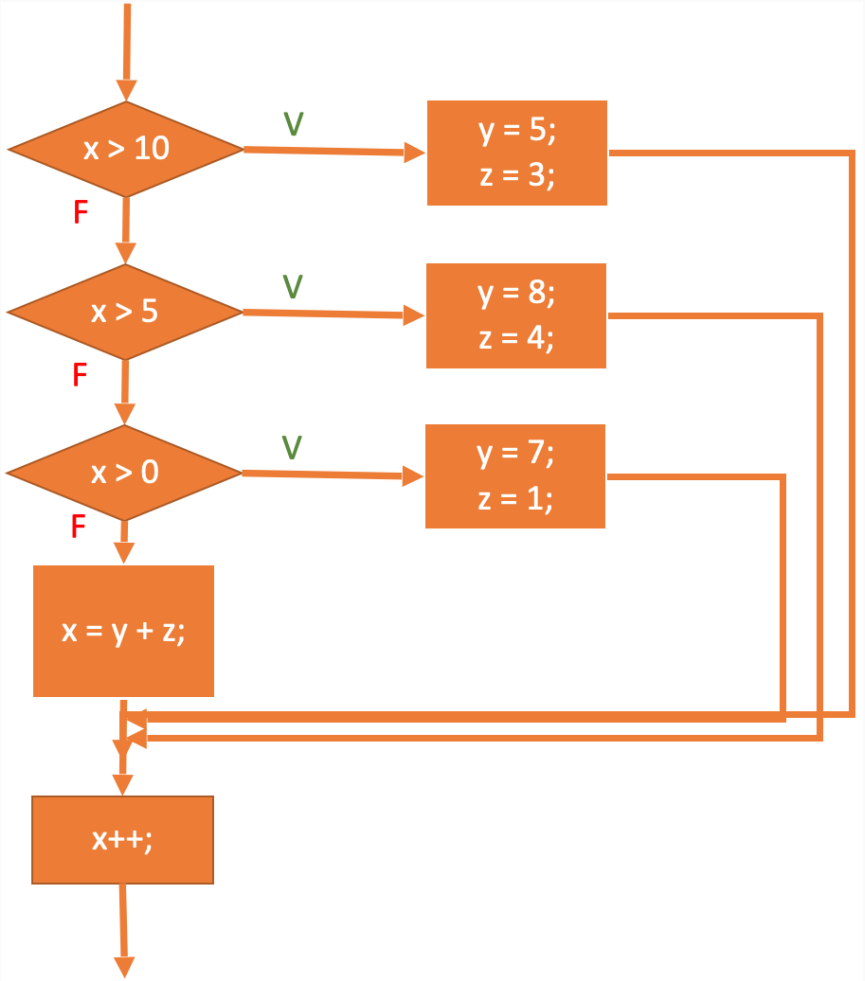
- Os blocos são avaliados na ordem de cima para baixo, desde que nenhuma condição anterior tenha sido verdadeira. Nesse caso, o bloco da condição verdadeira é executado e os demais são ignorados.
- O bloco do else é executado automaticamente caso todas as condições tenham sido avaliadas como falsas.
- O número de blocos *else-if* ou mesmo o bloco *else* são opcionais.

```
if (condição A){
  comandos 1
}
else if (condição B){
  comandos 2
}
...
else{
  comandos N
}
```

Estrutura if-else if

Código em C

```
if(x>10){
   y=5;
   z=3;
}
else if(x>5){
   y=8;
   z=4;
}
else if(x>0){
   y=7;
   z=1;
}
else{
   x=y+z;
}
x++;
```



Exercício

1) Crie um programa que solicite ao usuário a digitação de um número inteiro. O programa deve realizar as seguintes verificações e exibir a mensagem correspondente na tela:

1. Se o número for maior que 20, exibir a mensagem "O número é maior que 20".
2. Se o número for maior que 15, exibir a mensagem "O número é maior que 15".
3. Se o número for maior que 10, exibir a mensagem "O número é maior que 10".
4. Se o número for maior que 5, exibir a mensagem "O número é maior que 5".
5. Caso não atenda nenhuma das condições, exibir a mensagem "O número não atendeu a nenhuma verificação".

Exercício

```
int main(){
   int x;
   scanf("%d",&x);
   if(x>20){
      printf("O número é maior que 20");
   }
   else if(x>15){
      printf("O número é maior que 15");
   }
   else if(x>10){
      printf("O número é maior que 10");
   }
   else if(x>5){
      printf("O número é maior que 5");
   }
   else{
      printf("O número não atendeu a nenhuma verificação");
   }
}
```

Aninhamento de condicionais

- Como dentro de qualquer bloco podem ser inseridos comandos, a estrutura condicional também é um comando.

Aninhamento de condicionais

- Como dentro de qualquer bloco podem ser inseridos comandos, a estrutura condicional também é um comando.
- Dessa forma podem existir estruturas condicionais dentro de outras do mesmo tipo, quantos níveis forem necessários.

Aninhamento de condicionais

```
se ( expressão ) então
    se ( expressão ) então
      ...
    senão
      ...
    fim se
senão
    se ( expressão ) então
      ...
    senão
        se ( expressão ) então
          ...
    fim se
```

Aninhamento de condicionais

```
if(expressão1){
  comando1;
  if(expressão2){
    Comando2;
  }
  else{
    Comando3;
  }
}
```

Cuidados com estruturas condicionais

- O { e o } são obrigatórios quando o *if* ou o *else* tiverem mais de um comando.

Cuidados com estruturas condicionais

- O { e o } são obrigatórios quando o *if* ou o *else* tiverem mais de um comando.
- Quando eles tiverem exatamente um comando, o { e o } é facultativo.

Cuidados com estruturas condicionais

- O { e o } são obrigatórios quando o *if* ou o *else* tiverem mais de um comando.
- Quando eles tiverem exatamente um comando, o { e o } é facultativo.
- Uma ótima prática de programação é sempre utilizá-los

Cuidados com estruturas condicionais

- O { e o } são obrigatórios quando o *if* ou o *else* tiverem mais de um comando.
- Quando eles tiverem exatamente um comando, o { e o } é facultativo.
- Uma ótima prática de programação é sempre utilizá-los
- CUIDADO com ifs aninhados.

Cuidados com estruturas condicionais

```
int main(){
  int a;
  int b;
  printf("Digite um número inteiro \n")
  scanf("%d", a);
  printf("Digite outro número inteiro \n")
  scanf("%d", b);
  if(a>b)
    printf("O primeiro valor digitado é maior");
  else
    printf("O segundo valor digitado é maior");
  return 0;
}
```

Cuidados com estruturas condicionais

O **else** abaixo pertence a qual **if**?

```
if (n > 0)
  if(a > b)
     z = a;
else
  z = b;
```

Cuidados com estruturas condicionais

O **else** abaixo pertence a qual **if**?

```
if (n > 0)
  if(a > b)
     z = a;
else
  z = b;
```

Sempre associamos o else ao if mais interno (o mais próximo)

Desafio

Desenvolva um programa que leia três números inteiros do teclado e os armazene nas variáveis `a`, `b` e `c`. Em seguida, o programa deve reorganizar esses valores em ordem decrescente, de modo que `a` contenha o maior número, `b` o segundo maior e `c` o menor.

DICA: crie variáveis auxiliares para movimentar os valores.

Estrutura CASE - switch case

- Em alguns programas, existem condições mutuamente exclusivas, isto é, se uma situação for executada, as demais não serão.
- Condições mutuamente exclusivas: aquelas que não podem ser verdadeiras ao mesmo tempo.
- Quando este for o caso, um comando **seletivo (switch case)** é o mais indicado.
- Esse é um tipo de estrutura de seleção, na qual ocorre a avaliação da correspondência de uma expressão com as opções disponíveis.

```
switch (variável){
case valor1:
  lista de comandos;
  break;
case valor2:
  lista de comandos;
  break;
...
default:
  lista de comandos;
}
```

Estrutura CASE - switch case

- É opcional indicar a situação default, executada no caso de não ocorrer correspondência com as demais opções.

- O uso de **break** no final de cada caso é opcional, inclusive para default. Mas ele é necessário para se ter situações mutuamente exclusivas.

- Em programação, o comando **break** é uma instrução de controle de fluxo que permite interromper a execução de uma estrutura de seleção switch antes que sua condição normal de término seja atingida.

```
switch (variável){
case valor1:
  lista de comandos;
  break;
case valor2:
  lista de comandos;
  break;
...
default:
  lista de comandos;
}
```

Switch Case - Exemplo

```
int i;
scanf("%d", i);
switch(i){
  case 2:
    printf("Número 2");
    break;
  case 5:
    printf("Número 5");
    break;
  default:
    printf("Número diferente de 2 e de 5");
}
...
```

Observação

Tipos de valores permitidos para avaliar usando o comando *switch case*:

- int
- unsigned int
- short int
- long
- unsigned long
- char
- unsigned char

Resumindo, int ou char.

Exercício

Faça um programa que leia um caractere, identifique-o e escreva na tela se ele é um ponto, uma vírgula, um ponto e vírgula ou outro sinal.
Utilize obrigatoriamente o comando switch-case.

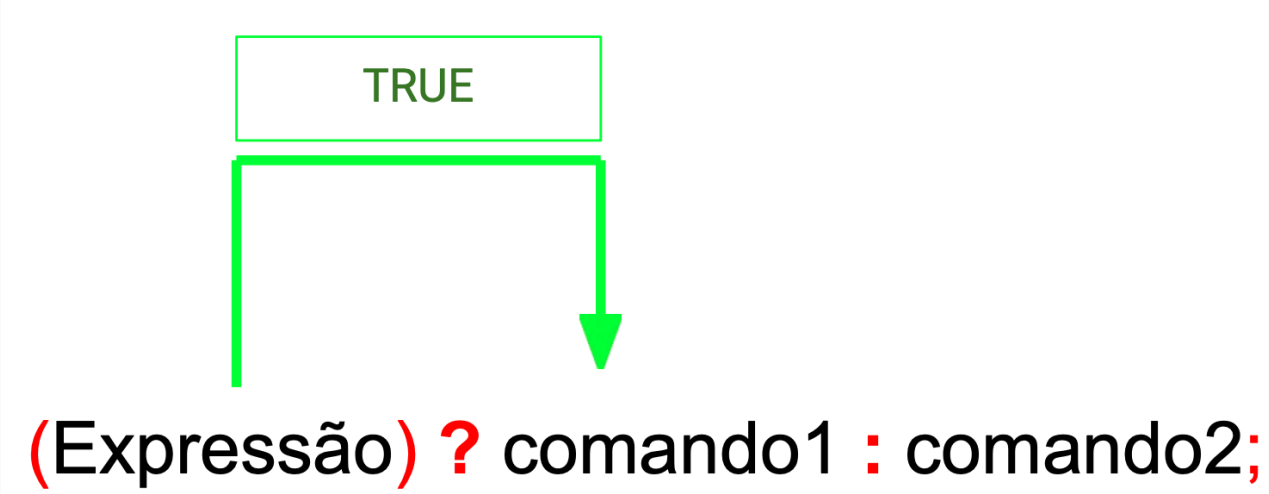
Exercício

```
char ch;
scanf(" %c",&ch);
switch( ch ) {
  case '.':
    printf("Ponto");
    break;
  case ',':
    printf("Vírgula");
    break;
  case ';':
    printf("Ponto e vírgula");
    break;
  default:
    printf("Não é pontuação");
}
```

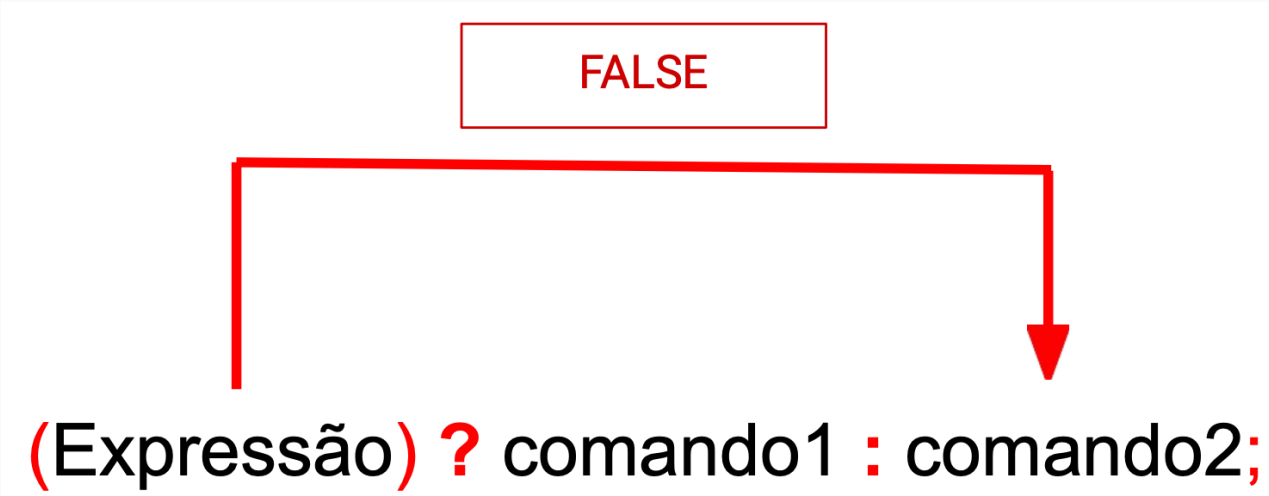
Operador ternário - Condicional enxuta

```
(Expressão) ? comando1 : comando2;
```

Operador ternário - Condicional enxuta



Operador ternário - Condicional enxuta



Operador Ternário - Exemplo

```
if(a>b) {
    c=a*a;
} else {
    c=b;
}
```

```
c = (a>b)?  a*a :  b
```

Dúvidas?