

# Estruturas Condicionais

Eldrey Galindo



# Lembre-me:

- ✓ **Chamada** está disponível no Classroom
- ✓ Aula está sendo gravada e estará disponível para os alunos que solicitarem na secretaria.
- ✓ Abra a câmera



# Introdução

Até agora, todos os algoritmos vistos eram **sequenciais**

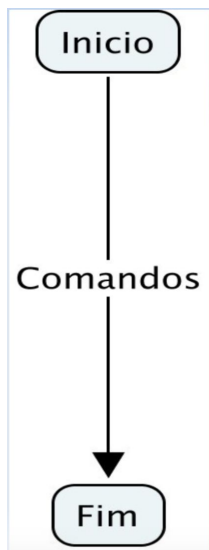
- Instruções são executadas uma após a outra
- De cima para baixo

Porém, problemas reais, em sua maioria, exigem uma **tomada de decisão** no algoritmo, onde há comandos que desviam o fluxo de execução

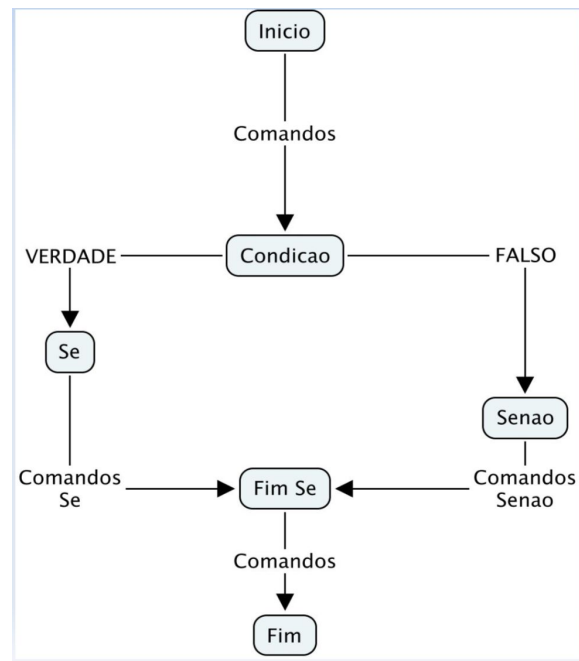
- Algumas instruções podem ser ignoradas
- Depende da condição dada

# Introdução

## Execução Normal:



## Instrução de Decisão / Desvio Condicional:



# Quando usar as instruções de decisão?

- Quando queremos que uma condição seja analisada;
- Dependem de uma **condição**
  - Resultado da condição deve retornar **VERDADEIRO** ou **FALSO**
- Caso esta condição seja verdadeira, um comando será executado;
- Caso esta condição seja falsa, outro comando será executado.

# Tipos de instruções de decisão

- Instruções SIMPLES;  
se
- Instruções COMPOSTAS;  
se - senão
- Instruções ALINHADA;  
se - senão se
- Instruções de MÚLTIPLA ESCOLHA;  
escolha - caso1, caso2, ..., casoN

# Instrução de Decisão Simples

- Utiliza a seguinte sintaxe:

```
se (CONDICAO) {
```

```
    comando a ser executado
```

```
}
```

- A expressão da **CONDICAO** é avaliada
- Se o resultado da avaliação é **VERDADEIRO**
  - Os comandos dentro do bloco indentado são executados
- Se o resultado da avaliação é **FALSO**
  - Tudo dentro do bloco indentado é ignorado seguindo para a linha após a decisão

```

1 programa
2 {
3
4     funcao inicio()
5     {
6         inteiro valor1, valor2
7         caracter operacao
8
9         escreva("entre com o primeiro valor \n")
10        leia(valor1)
11
12        escreva("entre com a operação s->soma, m->multiplicação, d->divisão \n")
13        leia(operacao)
14
15        escreva("entre com o segundo valor \n")
16        leia(valor2)
17
18        se (operacao == 's' e operacao != 'm' e operacao != 'd'){
19            escreva("Soma é ", (valor1 + valor2))
20        }
21
22        se (operacao != 's' e operacao == 'm' e operacao != 'd'){
23            escreva("Multiplicação é ", (valor1 * valor2))
24        }
25
26        se (operacao != 's' e operacao != 'm' e operacao == 'd'){
27            escreva("Divisão é ", (valor1 / valor2))
28        }
29    }
30 }

```



# Instrução de Decisão Composta

- Utiliza a seguinte sintaxe:

```
se (CONDICAO) {
```

```
    comando a ser executado
```

```
}
```

```
senao {
```

```
    comando a ser executado
```

```
}
```

- A expressão da **CONDICAO** é avaliada
- Se o resultado da avaliação é **VERDADEIRO**
  - Os comandos dentro do bloco **SE** são executados
- Se o resultado da avaliação é **FALSO**
  - Os comandos dentro do bloco **SENAO** são executados

# Instrução de Decisão Alinhada

- Utiliza a seguinte sintaxe:

```
se (CONDICAO1) {
```

```
    comando a ser executado
```

```
}
```

```
senao se (CONDICAO2){
```

```
    comando a ser executado
```

```
}
```

# Instrução de Decisão Alinhadas

- A expressão da **CONDICAO1** é avaliada
- Se o resultado da avaliação é **VERDADEIRO**
  - Os comandos dentro do bloco **SE** são executados
- Se o resultado da avaliação é **FALSO** é verificado a expressão da **CONDICAO2**
  - Se o resultado da avaliação é **VERDADEIRO**
    - Os comandos dentro do bloco **SENAO SE** são executados
  - Se o resultado da avaliação é **FALSO**
    - Tudo dentro do bloco indentado é ignorado segindo para a linha após a decisão

```

1 programa
2 {
3
4     funcao inicio()
5     {
6         inteiro valor1, valor2
7         caracter operacao
8
9         escreva("entre com o primeiro valor \n")
10        leia(valor1)
11
12        escreva("entre com a operação s->soma, m->multiplicação, d->divisão \n")
13        leia(operacao)
14
15        escreva("entre com o segundo valor \n")
16        leia(valor2)
17
18        se (operacao == 's'){
19            escreva("Soma é ", (valor1 + valor2))
20        }
21        senao se (operacao == 'm'){
22            escreva("Multiplicação é ", (valor1 * valor2))
23        }
24
25        senao {
26            escreva("Divisão é ", (valor1 / valor2))
27        }
28    }
29 }
30

```

# O PROBLEMA DE SER PROGRAMADOR



**MINHA MULHER DISSE:**

**– AMOR, VÁ ATÉ O MERCADO E COMPRE 1 GARRAFA DE LEITE.  
SE ELES TIVEREM OVOS, TRAGA 6**

**~ EU VOLTEI PARA CASA COM 6 GARRAFAS DE LEITE ~**

**ELA DISSE:**

**– PORQUE DIABOS VOCÊ COMPROU 6 GARRAFAS DE LEITE?**

**EU RESPONDI:**

**– PORQUE ELES TINHAM OVOS.**

# Instrução de Decisão de Múltipla Escolha

- Uma decisão de múltipla escolha pode ser construída com os comandos `se...senao` aninhados, porém, torna-se de **difícil leitura**;
- Há um comando, `escolha...caso`, que oferece uma melhoria na leitura do algoritmo;
- A proposta do `escolha...caso` é ser uma solução mais **elegante**. Levando o fluxo do programa direto ao bloco de código correto, dependendo do valor de uma variável de verificação.

# Instrução de Decisão de Múltipla Escolha

- Utiliza a seguinte sintaxe:

```
escolha (EXPRESSAO_DE_SELECAO) {
```

```
  caso (OPCAO1):
```

```
    comando a ser executado
```

```
  pare
```

```
  caso (OPCAO2):
```

```
    comando a ser executado
```

```
  pare
```

```
  caso contrario:
```

```
    comando a ser executado
```

```
}
```

# Instrução de Decisão de Múltipla Escolha

- A EXPRESSÃO DE SELEÇÃO é comparada com os casos
- Se o resultado da comparação for VERDADEIRO
  - Os comandos são executados até o computador encontrar o comando PARE
- Se o resultado da comparação é FALSO para todos os casos
  - Se o comando caso contrario existir
    - Executa os comandos internos e termina o escolha
  - Se não, termina o escolha.



```

1 programa
2 {
3     funcao inicio()
4     {
5         inteiro valor1, valor2
6         caracter opcao
7
8         escreva("entre com o primeiro valor \n")
9         leia(valor1)
10
11        escreva("entre com a operação s->soma, m->multiplicação, d->divisão \n")
12        leia(opcao)
13
14        escreva("entre com o segundo valor \n")
15        leia(valor2)
16
17        limpa()
18
19        escolha (opcao)
20        {
21            caso 's': // Se opcao == 's'
22                escreva("Soma é ", (valor1 + valor2))
23                pare // -> Impede que caso a opcao == s o código execute o caso seguinte
24            caso 'm': // Se opcao == 'm'
25                escreva("Multiplicação é ", (valor1 * valor2))
26                pare
27            caso 'd': // Se opcao == 'd'
28                escreva("Divisão é ", (valor1 / valor2))
29                pare
30            caso contrario: // Se opcao for diferente de todos os casos
31                escreva ("Opção Inválida !")
32        }
33    }
34 }

```

# Em grupos...

1. Leia 4 números inteiros e um carácter, mostre a soma dos números ímpares se o character for “i” ou dos pares se o character for “p”
2. Leia dois valores
  - Se a soma dos valores for menor ou igual a 10, deverá ser apresentado ao usuário o resultado da adição mais 5;
  - Se o valor somado não for menor ou igual a 10, deverá ser apresentado ao usuário o resultado da adição menos 7.
3. Escreva um algoritmo para ler um valor inteiro de 1 a 7 e escrever a descrição do dia da semana correspondente. Se for 1, imprimir DOMINGO, e assim sucessivamente; Caso o valor digitado esteja fora desse intervalo o usuário deverá ser alertado.

# Instrução de Decisão - Parte 01

- *Lista de Exercício 03 disponível!*