

Estruturas de Dados I

Apresentação do Plano de Curso

Revisão de estruturas básicas de programação:

- Sequenciais
- Condicionais
- Repetição

Ementa

- Tipos abstratos de dados.
- Estruturas de dados estáticas
 - vetores e matrizes.
- Estruturas de dados dinâmicas
 - listas lineares, pilhas, filas, listas circulares e duplamente ligadas, grafos e árvores.



Divisão do Conteúdo

27 de setembro

Primeira Avaliação

- Tipos abstratos de dados.
- Estruturas de dados estáticas:
 - vetores e matrizes.

01 de novembro

Segunda Avaliação

- Estruturas de dados dinâmicas
- uso de ponteiros e structs.
 - listas encadeadas, pilhas e filas.

06 de dezembro

Terceira Avaliação

- Estruturas de dados dinâmicas
- grafos
 - árvores



Divisão de Aulas e Nota Final



Aula teórica - sexta-feira - sala 204 - CAE (7h)

Aula prática - sexta-feira - Lab. 100 - CAE (9h) *

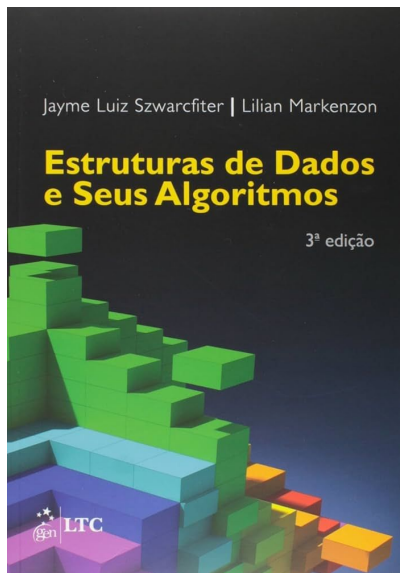
* com Aldo Díaz

$$\text{NF} = (0,7 * P) + (0,3 * AL)$$

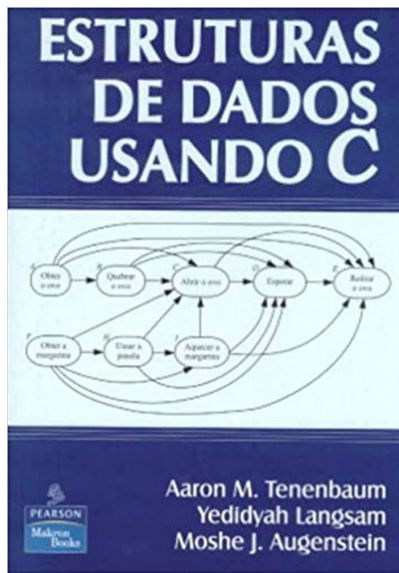
Reprovado < 6,0 > Aprovado

Onde: P = média das duas provas e AL = média das atividades laboratório.

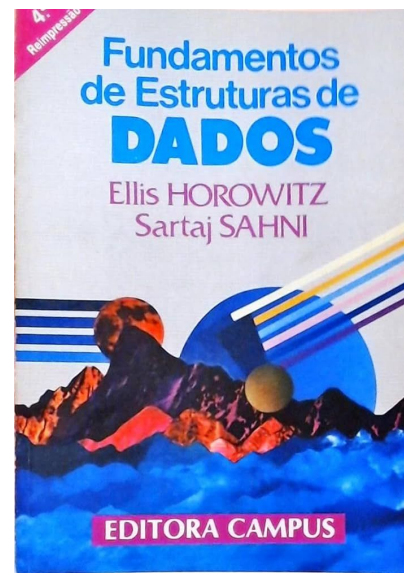
Bibliografia



Estruturas de Dados e Seus Algoritmos
Jayme Luiz Szwarcfiter e Lilian Markenzon



Estruturas de Dados Usando C
Aaron M. Tenenbaum



Fundamentos de Estruturas de Dados
Ellis Horowitz E Sartaj Sahni



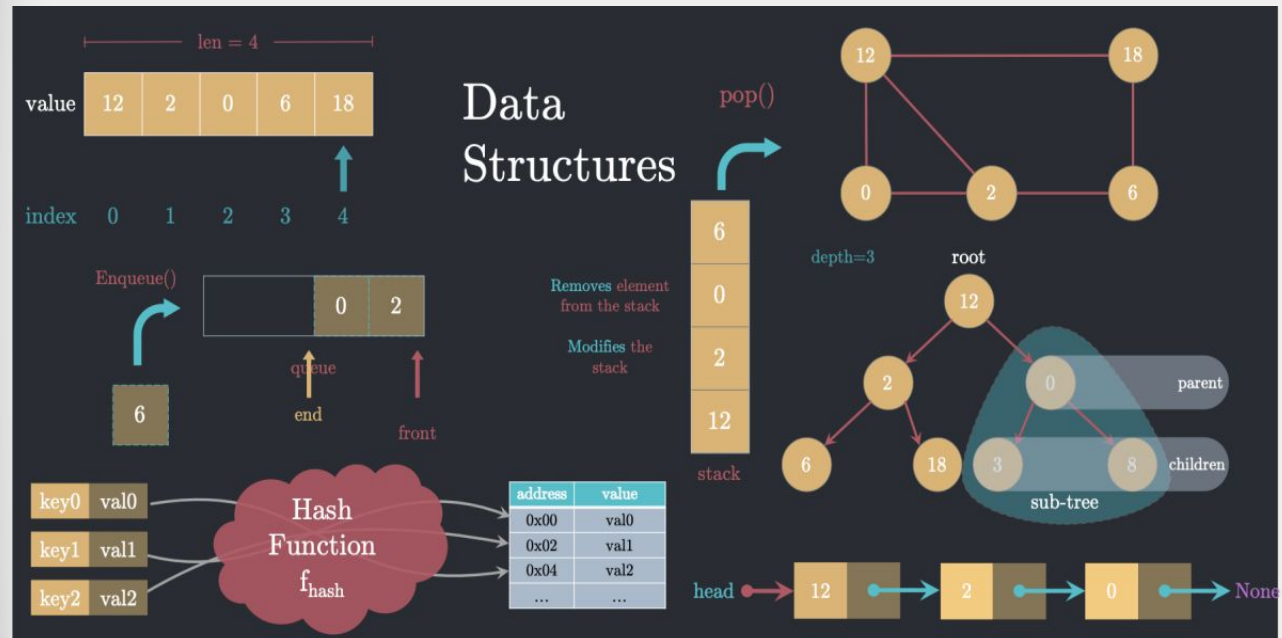


ESTRUTURA DE...

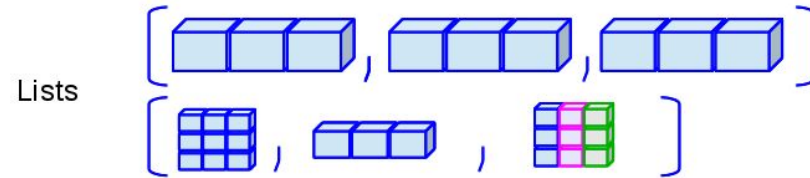
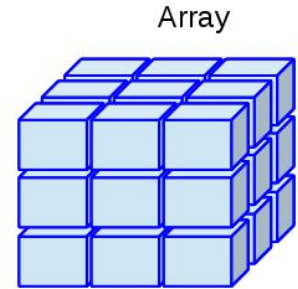
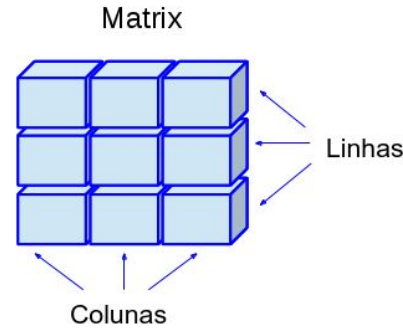
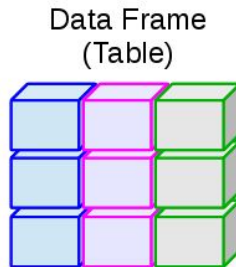
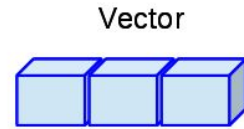
...DADOS



Estrutura de Dados
fornece as bases
para **organizar e
manipular dados
na memória** do
computador de
maneira eficiente.



As estruturas de dados, junto com o estudo de algoritmos, fazem parte dos fundamentos da programação.



Fundamentos da Programação

- Algoritmos
- Tipos de Dados
- Variáveis e Constantes
- Operadores Aritméticos,
Relacionais ou Lógicos.
- Estruturas de Controle
- Funções
- Estruturas de Dados
- Entrada e Saída (scanf , printf)
- Depuração e Teste
- Documentação



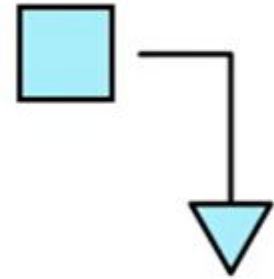
Estruturas de Controle

- Componentes fundamentais de qualquer linguagem de programação
- Permitem que os programas façam escolhas, repitam tarefas e controlem o fluxo de execução do código
 - Estruturas sequenciais
 - Estruturas condicionais
 - Estruturas de repetição



Estruturas Sequenciais

Representam uma sequência de ações desenvolvidas em uma ordem específica, realizando uma ação depois da outra até que todas tenham sejam concluídas.

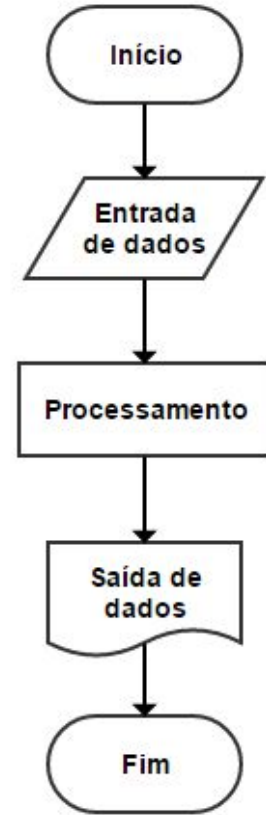


Sequenciais



Estruturas Sequenciais

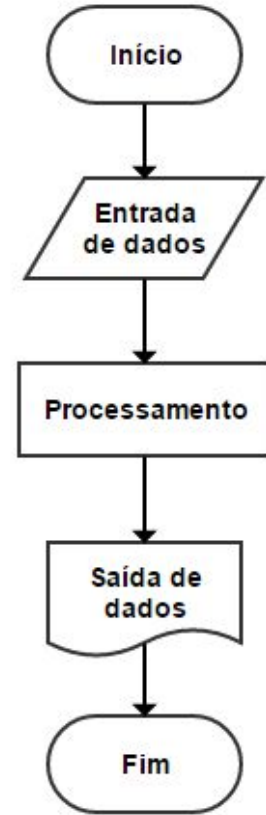
- **Início:** começo da execução do fluxo de operações.
- **Entrada de Dados:** coleta das informações necessárias para o processamento.
- **Processamento:** manipulação dos dados de entrada para produzir um resultado.
- **Saída de Dados:** apresentação final do que foi calculado ou processado.
- **Fim:** ponto final do algoritmo ou programa.



Estruturas Sequenciais

Aplicações

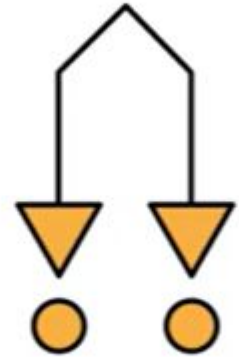
- Algoritmos de Leitura de Dados
- Algoritmos para execução de cálculos
 - Média de Notas
 - Conversão de Temperatura
 - Conversão de Moedas
 - Cálculo do IMC (Índice de Massa Corporal)



Estruturas Condicionais

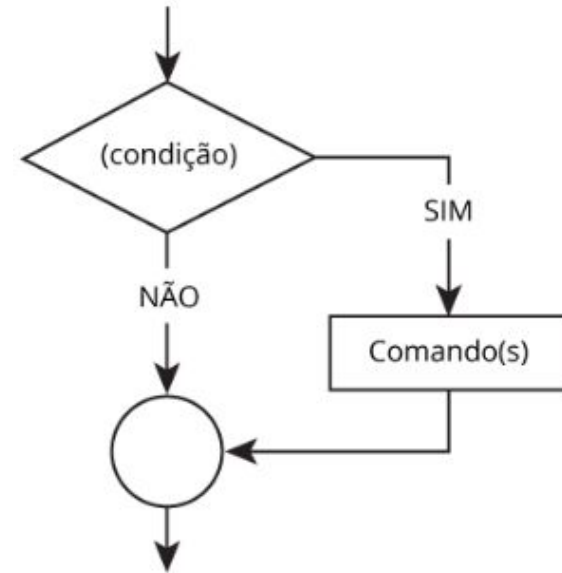
As estruturas condicionais permitem que o programa tome decisões baseadas em condições. O código só executa um determinado bloco de instruções se uma condição específica for verdadeira.

Condicionais



Estruturas Condicionais

Permitir a execução de diferentes blocos de código em resposta a diferentes condições. Isso permite ao programa reagir de maneira dinâmica a diferentes entradas ou estados.



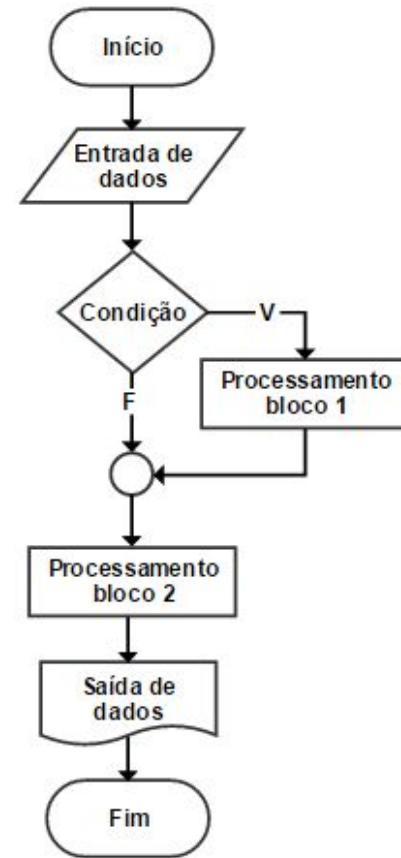
Referência: [Estruturas de decisão](#)



Estruturas Condicionais

Estrutura Condicional Simples

A estrutura condicional simples expõe que uma instrução ou bloco de instruções só sejam executados se determinada condição for verdadeira.



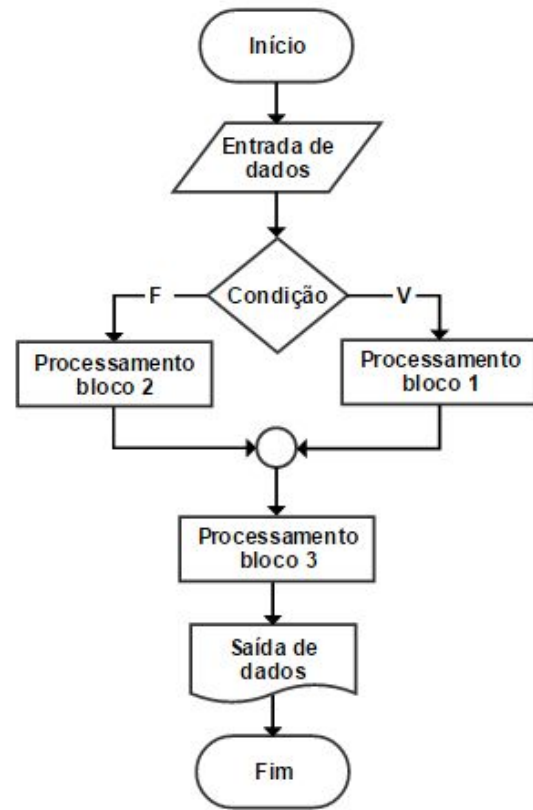
Referência: [Aula - Estrutura Condicional](#)



Estruturas Condicionais

Estrutura Condicional Composta

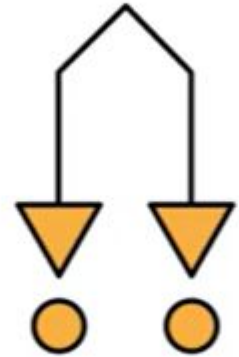
A estrutura condicional composta expõe que uma instrução ou bloco de instruções só sejam executados se determinada condição for satisfeita.



Estruturas Condicionais

As estruturas condicionais resolvem o problema de como lidar com diferentes situações, como validar entradas de usuários, responder a eventos em tempo real, ou aplicar diferentes regras de negócio em diferentes circunstâncias.

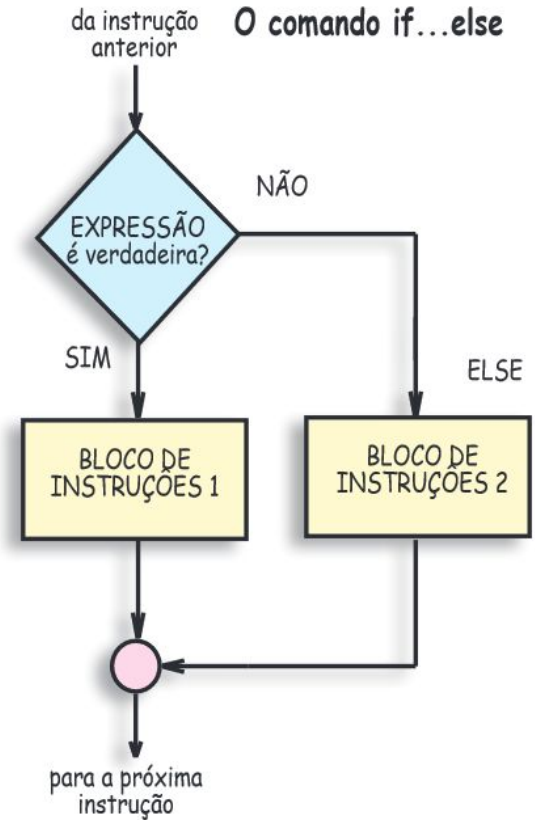
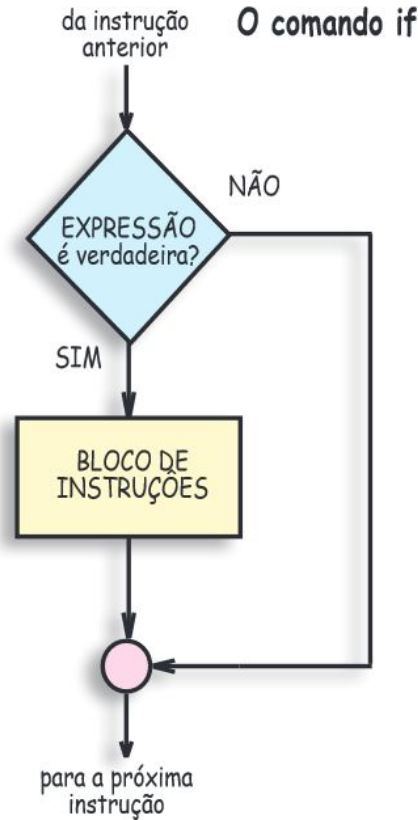
Condicionais



Estruturas Condicionais

Comandos de decisão:

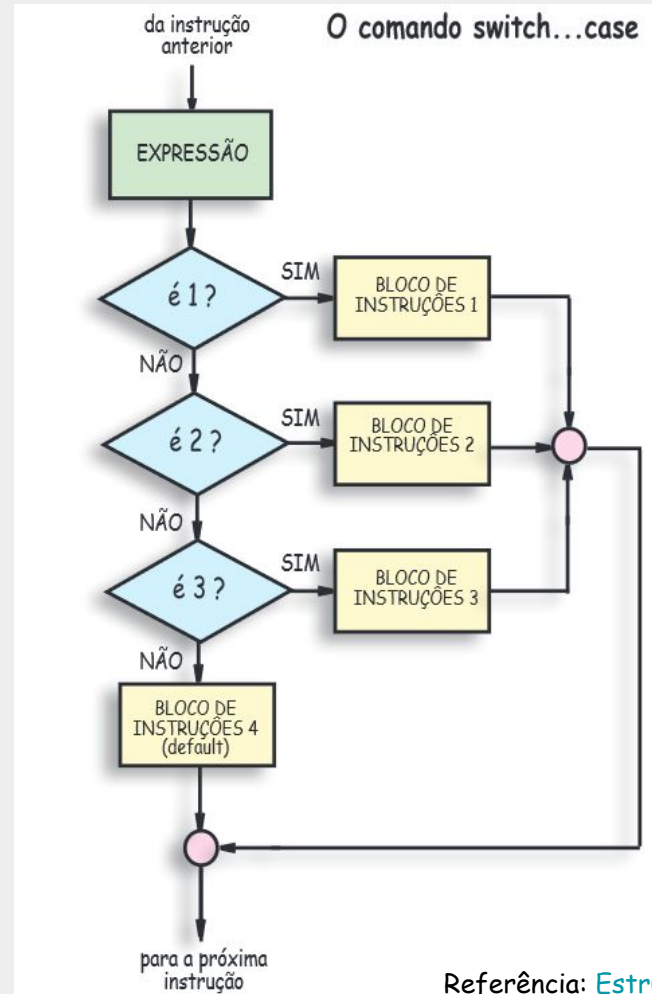
- `if()`
- `if() – else`



Estruturas Condicionais

Comandos de decisão:

- switch – case



Estruturas Condicionais (switch-case)



- O comando **switch-case** testa uma expressão que deve resultar em um número inteiro.
- Uma sequência de cláusulas case permite executar um conjunto de instruções conforme o valor da expressão.
 - É uma forma de reduzir a complexidade de vários if ... else encadeados.

Estruturas de Repetição

As estruturas de repetição permitem que um bloco de código seja executado várias vezes, repetindo a mesma operação até que uma condição específica seja atendida.



Estruturas de Repetição

As estruturas de repetição resolvem problemas como percorrer *listas* de dados, executar operações em massa, realizar cálculos iterativos, e outras tarefas que requerem múltiplas execuções de um mesmo bloco de código.





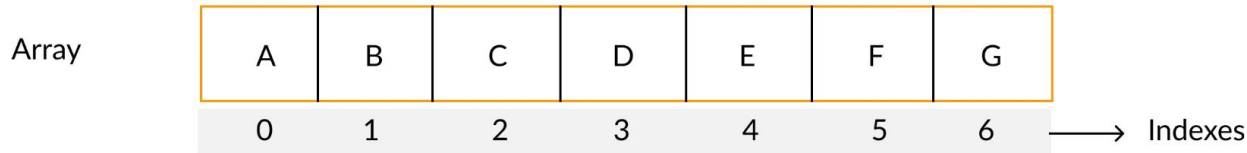
Estruturas de Repetição

- Estrutura de Repetição para número definido de repetições:
 - Comando **for()**
- Estrutura de Repetição para número indefinido de repetições:
 - Comando **while**
- Estrutura de Repetição para número indefinido de repetições e teste no início:
 - Comando **do-while**



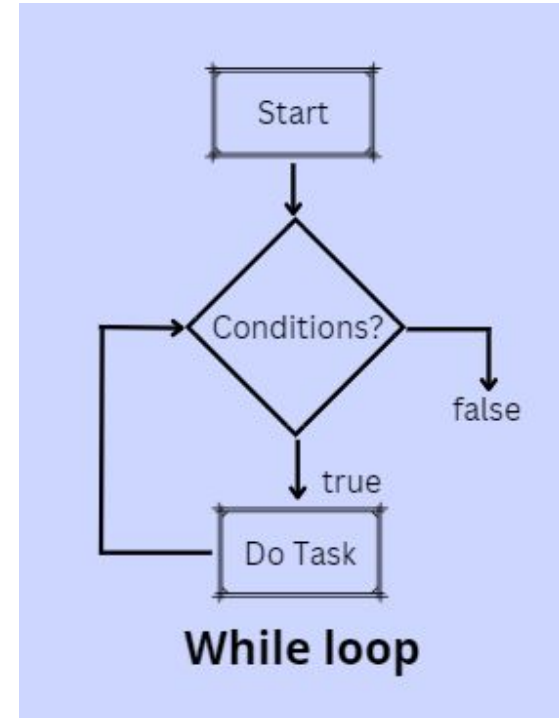
Estruturas de Repetição

A estrutura **for()** é usada quando o número de repetições é conhecido antes da execução do laço. É ideal para situações em que você sabe exatamente quantas vezes deseja que o bloco de código seja executado.



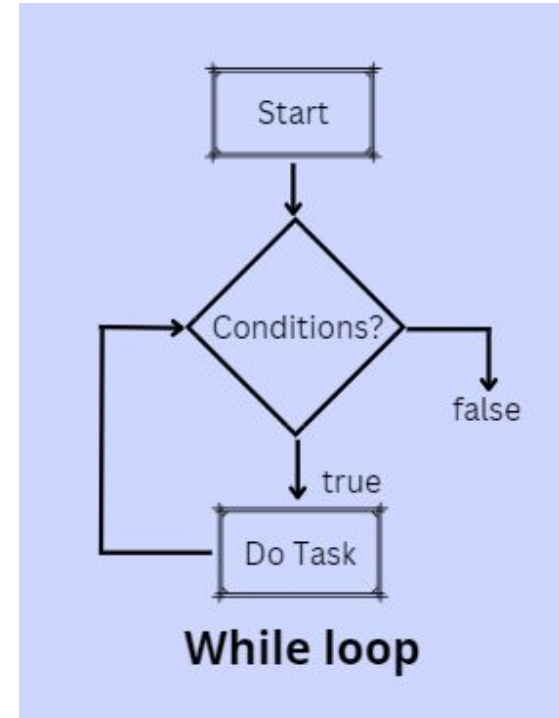
Estruturas de Repetição

A estrutura **while** é utilizada quando o número de repetições não é conhecido inicialmente. O laço continuará executando enquanto a condição especificada for verdadeira.



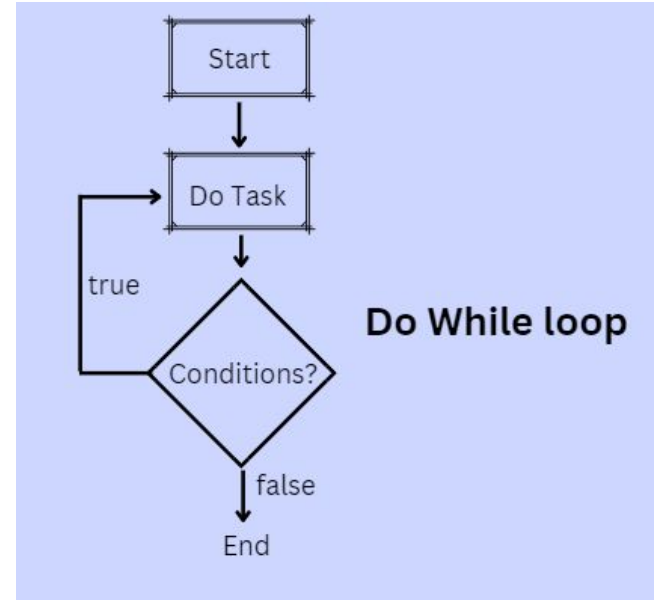
Estruturas de Repetição

Útil para situações onde o número de repetições depende de uma condição que pode mudar durante a execução. A **condição é verificada antes de cada iteração**, então o bloco pode nunca ser executado.



Estruturas de Repetição

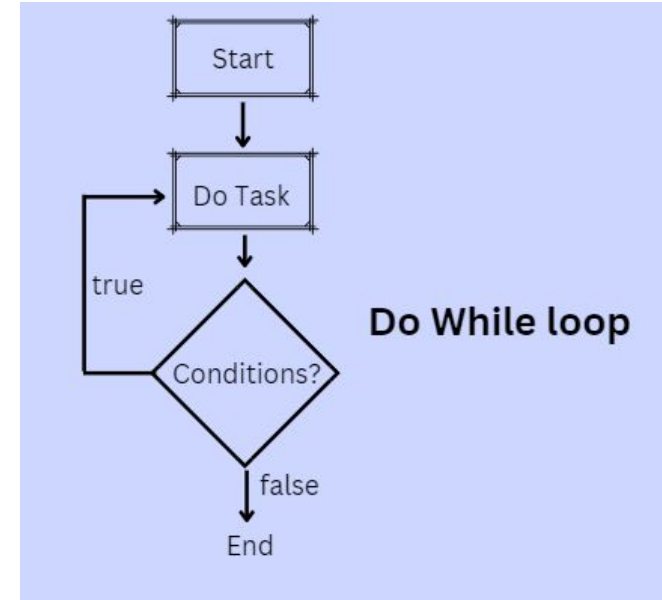
A estrutura **do-while** é semelhante ao *while*, mas garante que o bloco de código seja executado **pelo menos uma vez**, pois a condição é verificada após a execução do bloco.



Estruturas de Repetição

Aplicações da estrutura **do-while**

- Leitura de Input com Validação
- Menu de Opções
- Simulação de Jogos
- Repetir Pedidos de Confirmação





Estruturas de Repetição

As estruturas de controle foram criadas para tornar os programas mais flexíveis, eficientes e capazes de lidar com situações complexas.

- **Sequência:** Seguir uma ordem específica de operações.
- **Condicional:** Tomar decisões baseadas em condições variáveis.
- **Repetição:** Automatizar tarefas repetitivas sem duplicar código.

Dúvidas?



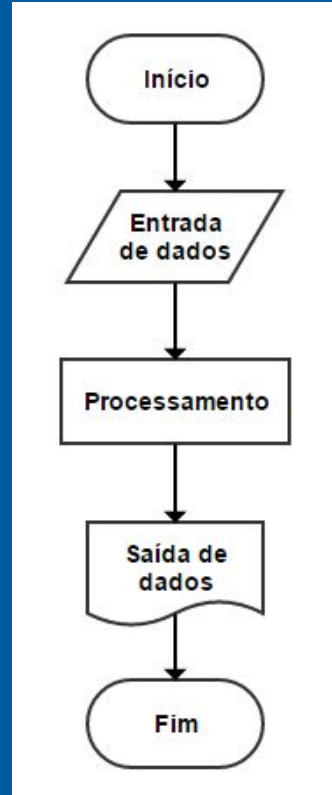
Bruna Michelly
brunamichelly@ufg.br



Prática














Estrutura Sequencial



E01. Hello world!

Hello, World! in 11 different languages

	Java	<code>System.out.println("Hello World!!");</code>
	Python	<code>print('Hello, world!')</code>
	C	<code>printf ("Hello, world!");</code>
	C++	<code>cout<<"Hello World";</code>
	C#	<code>System.Console.WriteLine("Hello World!");</code>
	JavaScript	<code>console.log('Hello World');</code>
	Php	<code>echo "Hello, World";</code>
	R	<code>cat('Hello World')</code>
	Swift	<code>println('Hello World');</code>
	Kotlin	<code>println("Hello World")</code>
	Go	<code>fmt.Println("Hello, World!")</code>



E02. Média

Faça um programa que calcule a média, padrão UFG, de 3 números inteiros, sendo que um desses números é 20 e os demais deverão ser digitados durante a execução.



E03. Área do círculo

Leia o valor do raio de um círculo e calcule e imprima a área dele.

A área do círculo é πr^2 , considere $\pi = 3.141592$



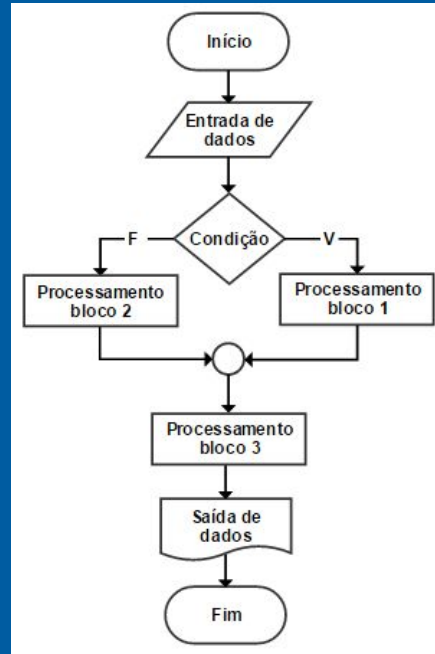
E04. Distância

Escreva um programa que dadas duas coordenadas $(x1, y1)$ e $(x2, y2)$ de pontos em \mathbb{R}^2 , calcule sua distância



Estruturas Condicionais

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



if(transform.rotation>180) 0 else 100;

E05. Idade

Fazer um programa que ao entrar com sua idade diga se você é maior de idade ou não



E06. Média

Fazer um programa que calcule a média de 3 notas. Caso a média seja igual ou superior a 6, o aluno estará aprovado, caso contrário, estará reprovado.

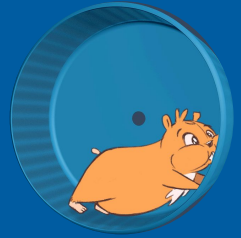


E07. ASCII

Fazer um programa que mostre o código ASCII correspondente aos caracteres
gamer WASD



Estruturas de Repetição



```
for(i = 1; i <= 10; i++)  
    comando;
```

```
x = 1;  
y = 5;  
while(x < y) {  
    x = x + 2;  
    y = y + 1;  
}
```

```
x = 1;  
y = 5;  
do{  
    x = x + 2;  
    y = y + 1;  
}while(x <= y);
```

E08. Dígitos

Faça um programa que escreva na tela de 0 até 9, utilizando as 3 estruturas aprendidas nesta aula



E09. Maior e menor

Elabore um programa que faça leitura de vários números inteiros, até que se digite um número negativo. O programa deve mostrar todos os números digitados (sem o negativo)



E10. Maior e menor

BONUS TRACK! Elabore um programa que faça leitura de vários números inteiros, até que se digite um número negativo. O programa tem que retornar o maior e o menor número lido



Referências

SILVA, G. B. **REA-AED**, 2024. Disponível em:

<https://gabrielbueno072.github.io/rea-aed/index.html>

FEOFILOFF, P. **Projeto de Algoritmos** (em C), 2018. Disponível em:

<https://www.ime.usp.br/~pf/algoritmos/>





Chega!

Aldo Díaz
aldo.diaz@ufg.br