



Fundamentos de Programação e Dados

1 – Introdução

Dev & Data

Prof. Guilherme Assis

1.1. Introdução

Prof. Guilherme Assis

- Atuo com Desenvolvimento de Software desde 2009.
- Professor desde 2018.
- <https://ghdeassis.dev>

O que veremos?

- Introdução a computadores.
- Lógica de programação.
- Lógica de programação na prática.

Tópicos

- Componentes de um computador.
- CPU e Memória RAM.
- Introdução a programação.
- Introdução ao JavaScript.
- Google Chrome.
- Visual Studio Code.
- Variáveis, tipos de variáveis, operadores, funções, estruturas de decisão, lógica booleana, listas e estruturas de repetição.

Dicas

- Consuma todo o conteúdo.
- Codifique junto com o professor nas aulas gravadas.
- Pratique muito.

Conclusão

- Introdução.



Fundamentos de Programação e Dados

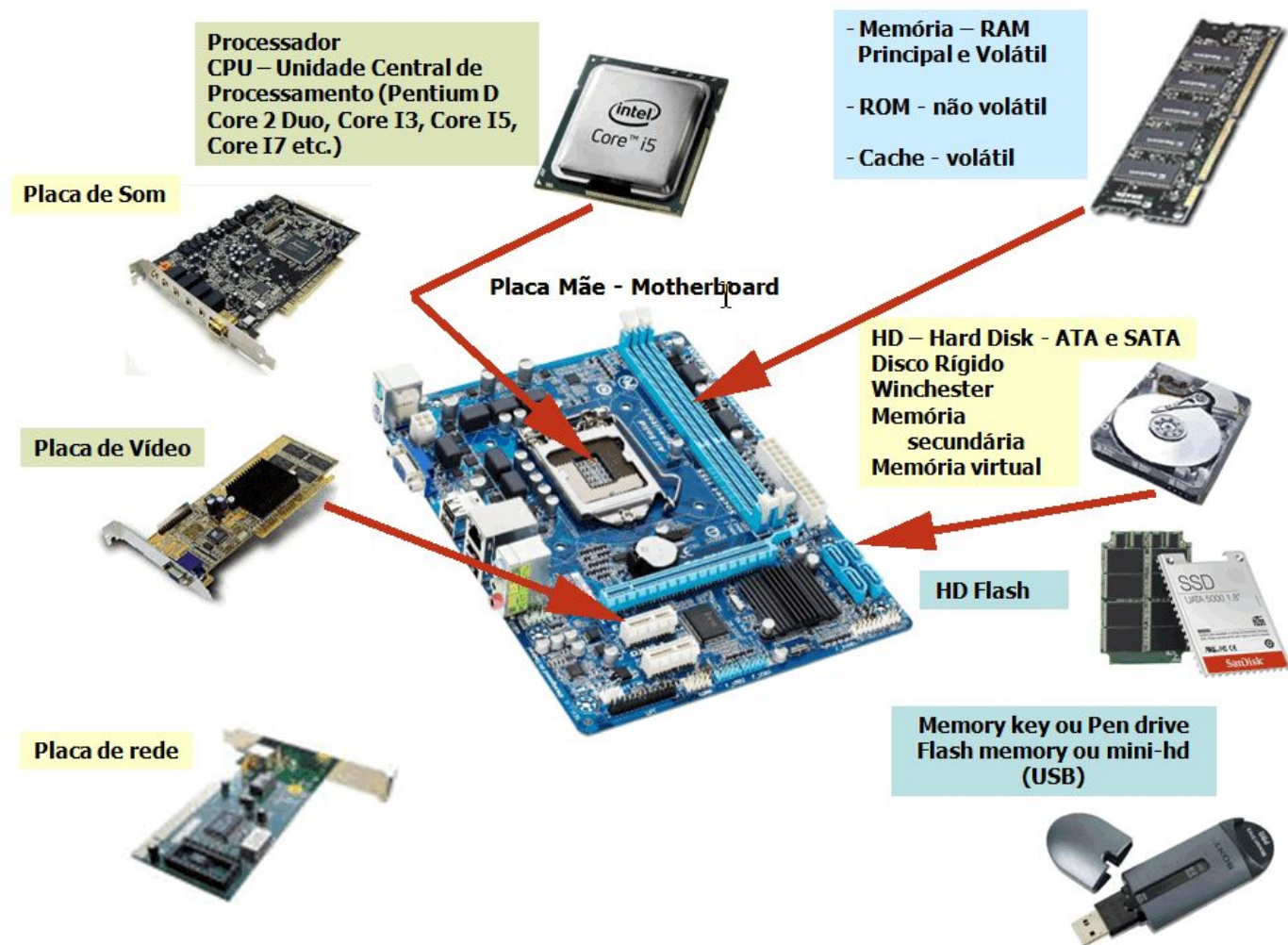
1 – Introdução

Dev & Data

Prof. Guilherme Assis

1.2. Componentes de um computador

Componentes de um computador



Conclusão

- Componentes de um computador.



Fundamentos de Programação e Dados

1 – Introdução
Dev & Data
Prof. Guilherme Assis

1.3. CPU e Memória RAM

CPU

- Central Processing Unit.
- Trabalha em conjunto com a memória RAM.
- Executa instruções que estiverem na memória RAM.
- Controla os demais componentes do computador.
- Exemplo: tecla digitada no teclado é enviada para a CPU (input), que por sua vez informa a interface de vídeo que deve exibir aquela caractere (output).
- Entrada -> Processamento -> Saída.

Bits e Bytes

- Sistema decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Sistema binário: 0, 1:
 - Bit: 0 ou 1.
 - Byte: grupo de 8 bits.
 - KB (kilobyte): 1.024 bytes.
 - MB (megabyte): 1.024 kb.
 - GB (gigabyte): 1.024 mb.
- Bytes são usados para representar caracteres:
 - Letra A: 01000001.
 - Letra B: 01000010.

Memória Principal (RAM)

- Acessada diretamente pelo processador.
- CPU utiliza a RAM para armazenar e executar programas.
- É uma memória volátil:
 - Quando o computador é desligado, todos os dados são perdidos.



Memória Secundária

- Memória de massa (permanente).
- Alta capacidade de armazenamento, mais barata que a primária.
- São muito mais lentos que a memória principal.
- Processador não trabalha com eles diretamente.
- Inviável de ser utilizado em conjunto com o processador.



Conclusão

- CPU.
- Bits e Bytes.
- Memória Principal.
- Memória Secundária.



Fundamentos de Programação e Dados

1 – Introdução

Dev & Data

Prof. Guilherme Assis

1.4. Introdução a Programação

Nesta aula

- O que é um programa de computador?
- O que é uma linguagem de programação?
- O que são algoritmos?

O que é um programa de computador?

- Um programa é uma sequência de passos definidos por um programador para alcançar um objetivo.
- Cada passo pode ser chamado de uma instrução.
- Exemplo de programa: Word.
- Software vs Hardware.

O que é uma linguagem de programação?

- Computador só entende bits (0 e 1).
- Para representar letras, números, símbolos, etc., utilizamos um conjunto de 8 bits, chamado byte.
- a: 01100001.
- Seria inviável programar dessa forma.
- Com uma linguagem de programação e um compilador essa tarefa é facilitada.

Exemplos de Comandos

- JavaScript:
 - `console.log("Hello World!");`
- Java:
 - `System.out.print("Hello World!");`
- Python:
 - `print("Hello World!")`

O que são algoritmos?

- Um algoritmo é uma sequência de passos que devem ser executados para atingir determinado objetivo.
- Exemplo: receita de bolo.
- Um algoritmo não necessariamente é um programa de computador, ele pode ser executado por uma pessoa.
- Uma mesma tarefa pode ser realizada por diferentes algoritmos.

Exemplo de algoritmo



Conclusão

- Programa de computador.
- Linguagem de programação.
- Algoritmos.



Fundamentos de Programação e Dados

1 – Introdução

Dev & Data

Prof. Guilherme Assis

1.5. Introdução ao JavaScript

Nesta aula

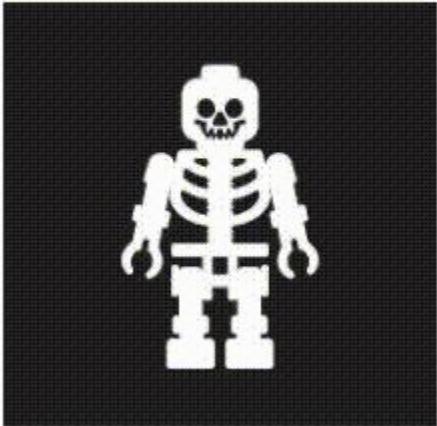
- O que é o JavaScript?
- HTML, CSS e JavaScript.

O que é JavaScript

- É uma linguagem de programação criada em 1995.
- É uma das mais populares da atualidade.
- Apesar do nome, não tem relação com a linguagem de programação Java.
- Inicialmente foi criada para ser utilizada dentro dos navegadores.
- Recentemente ela também vem sendo adotada fora dos navegadores.

Estrutura de uma página web

HTML
structure



CSS
presentation/appearance



JavaScript
dynamism/action



Conclusão

- O que é JavaScript.
- Estrutura de uma página web.



Fundamentos de Programação e Dados

3 – Variáveis, operadores e funções
Dev & Data
Prof. Guilherme Assis

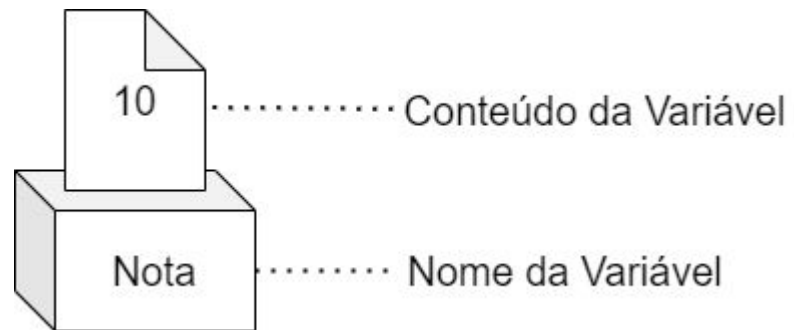
3.1. Variáveis

Variáveis

- Local que podemos armazenar um valor.
- Fica armazenada na memória.
- Um algoritmo pode ter diversas variáveis.
- Permite que o valor seja alterado.
- As variáveis podem ter tipos diferentes.
- Texto, número etc.

Variáveis

- Cada variável tem um identificador (nome) para que possamos utilizá-la no código.
- Exemplo: calculadora.



Conclusão

- Variáveis.



Fundamentos de Programação e Dados

3 – Variáveis, operadores e funções
Dev & Data
Prof. Guilherme Assis

3.2. Tipos de Variáveis

Variáveis

- string:
 - Representado entre aspas duplas ou simples.
 - “Teste de Caractere” ou ‘Teste de Caractere’.
- number:
 - Inteiro: 100.
 - Número com casas decimais: 100.95.
- boolean:
 - true ou false.

Conclusão

- Tipos de Variáveis.



Fundamentos de Programação e Dados

3 – Variáveis, operadores e funções
Dev & Data
Prof. Guilherme Assis

3.3. Operadores

Operadores

- Soma: +
- Subtração: -
- Multiplicação: *
- Divisão: /
- Resto: %
- Precedência: ()
 - $10 + 80 / 8 + 2 = 22$
 - $(10 + 80) / (8 + 2) = 9$
- Incremento: $i = i + 1$
- Decremento: $i = i - 1$

Conclusão

- Operadores.



Fundamentos de Programação e Dados

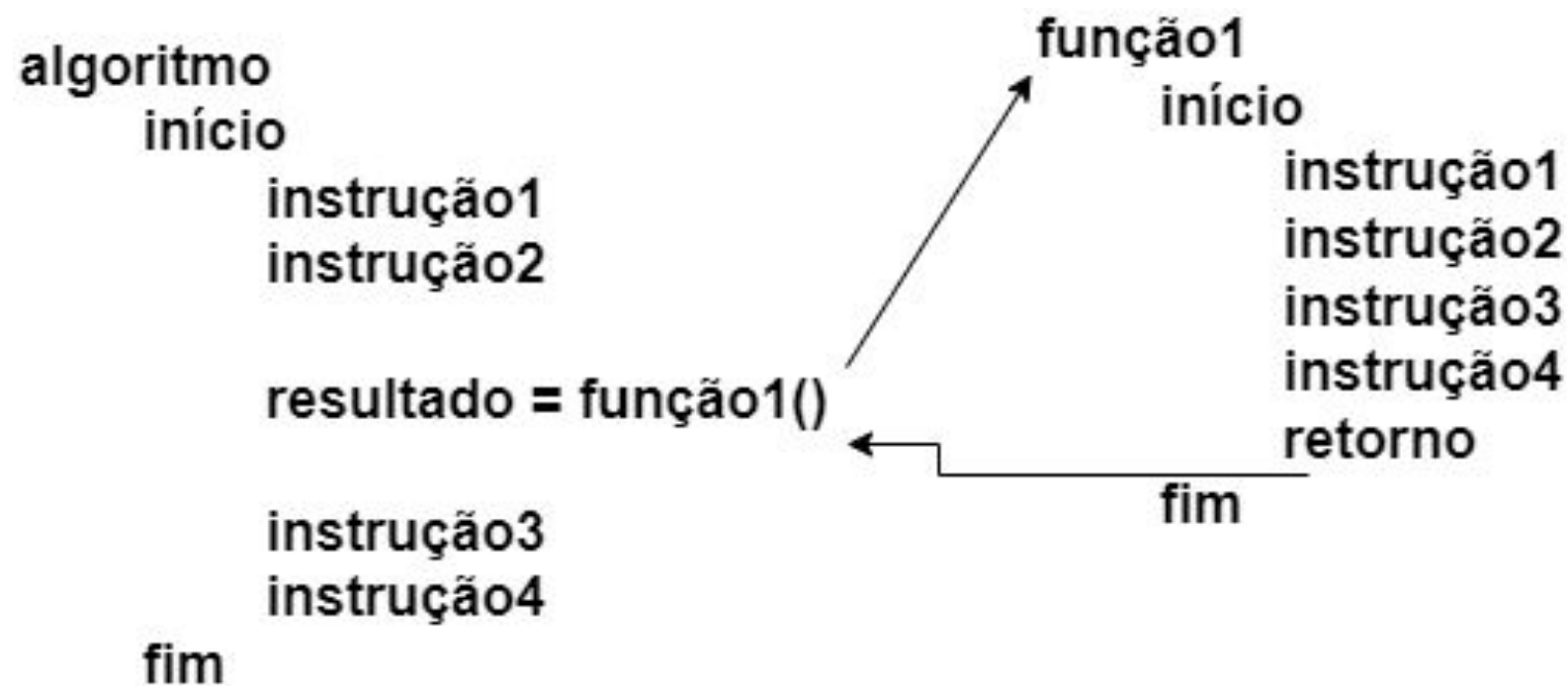
3 – Variáveis, operadores e funções
Dev & Data
Prof. Guilherme Assis

3.4. Funções

Funções

- Subprograma dentro do programa principal.
- Auxilia a realização de tarefas.
- Precisa ser chamado para que tenha seu código executado.
- Desvio provisório no fluxo de execução.
- Podemos chamar o mesmo subprograma várias vezes.

Funções



Funções

- Podem receber parâmetros.
- Podem ou não retornar valores.
- Enxergam variáveis criadas no programa principal (variáveis globais).
- Podem criar variáveis locais.

Conclusão

- Funções.



Fundamentos de Programação e Dados

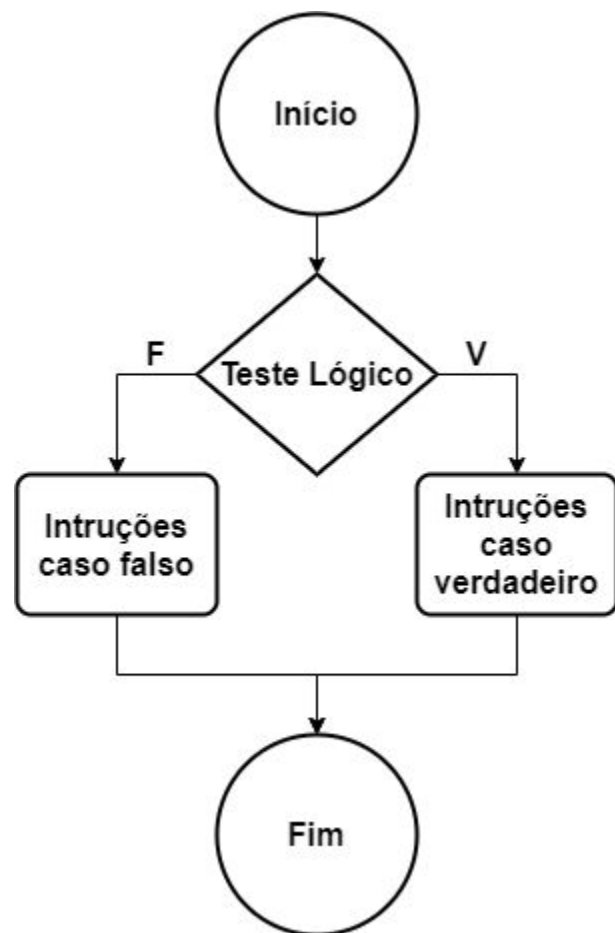
4 – Estruturas de decisão e lógica booleana
Dev & Data
Prof. Guilherme Assis

4.1. Estruturas de decisão

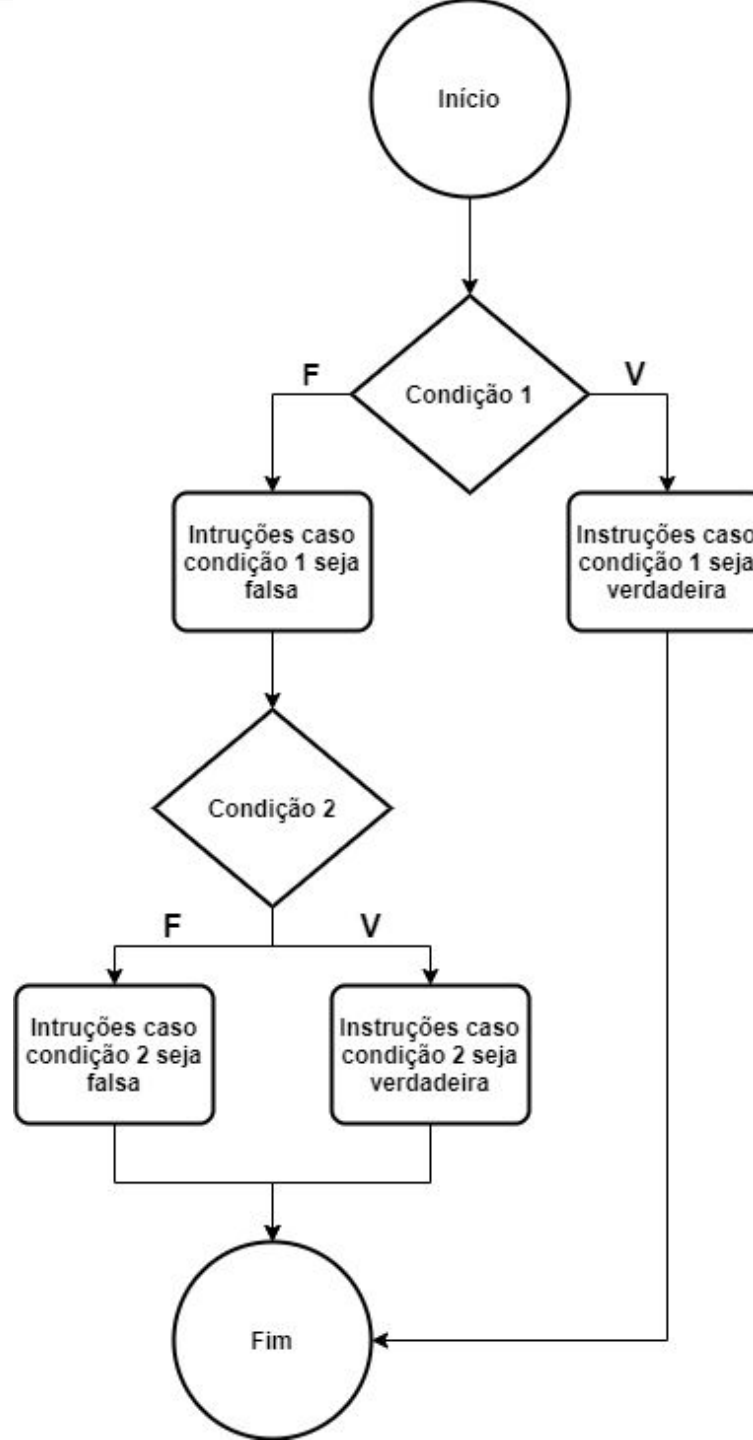
Estruturas de decisão

- Também chamadas de estruturas condicionais.
- Desviar o fluxo de execução do algoritmo de acordo com determinadas condições.
- Programador deve especificar uma ou mais condições a serem testadas pelo algoritmo.
- É possível encadear estruturas de decisão.

if/else



if/else if/else



Operadores de comparação

- Igual a:
 - $3 = 3$
- Menor que:
 - $5 < 7$
- Maior que:
 - $7 > 2$
- Menor ou igual a:
 - $4 \leq 5$
- Maior ou igual a:
 - $6 \geq 6$
- Diferente:
 - $7 \neq 4$

Conclusão

- Estruturas de decisão.



Fundamentos de Programação e Dados

4 – Estruturas de decisão e lógica
booleana
Dev & Data
Prof. Guilherme Assis

4.2. Lógica booleana

Lógica booleana

- Também chamada de álgebra booleana.
- Representar a lógica em expressões.
- Valores lógicos: VERDADEIRO ou FALSO.
- Operadores E, OU e NÃO.

AND (&&)

- Retorna verdadeiro somente se os dois lados da expressão forem verdadeiros:
 - $(A > B) \ \&\& \ (C > D)$.

Valor 1	Valor 2	AND
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso
Falso	Verdadeiro	Falso
Falso	Falso	Falso

OR

- Retorna verdadeiro se pelo menos um dos lados da expressão forem verdadeiros:
 - $(A > B) \parallel (C > D)$.

Valor 1	Valor 2	OR
Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Falso	Falso	Falso

NOT

- Inverte o resultado da expressão:
 - `if (!A){ }`

Valor	NOT
Verdadeiro	Falso
Falso	Verdadeiro

Conclusão

- Lógica booleana.



Fundamentos de Programação e Dados

5 – Listas e estruturas de repetição
Dev & Data
Prof. Guilherme Assis

5.1. Listas

Listas

- Também chamada de vetores ou arrays.
- Utilizada para armazenar diversas informações de um mesmo tipo.
- Exemplo: armazenar nomes dos alunos de uma escola. Seria inviável criar uma variável para cada.
- Elementos acessados a partir do índice.

0	1	2	3	4
João	Maria	José	Ana	Jorge

Conclusão

- Listas.



Fundamentos de Programação e Dados

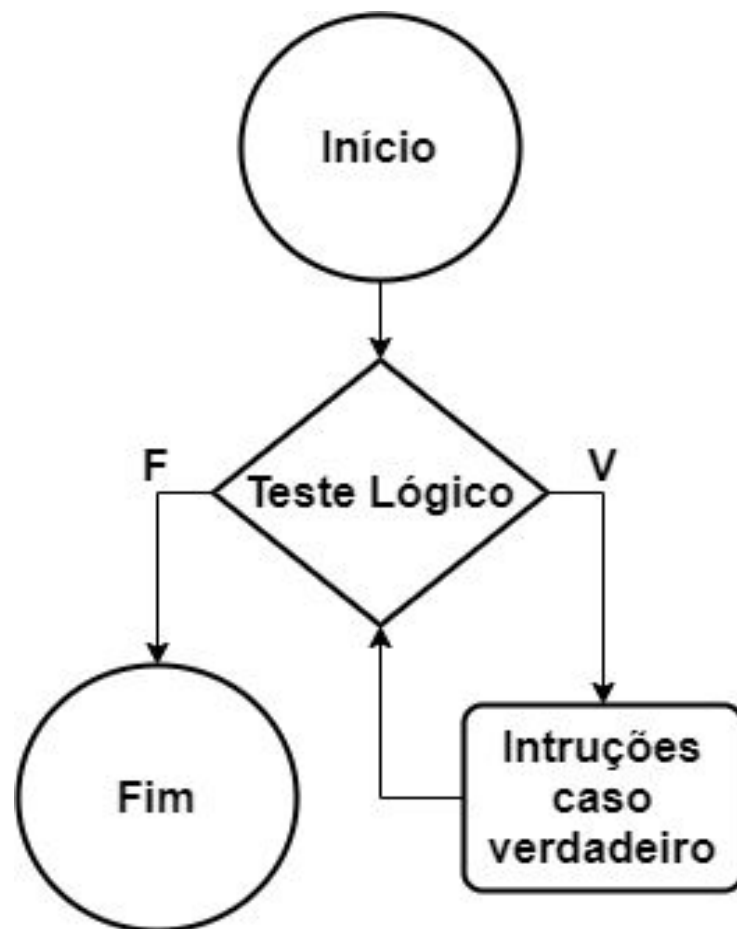
5 – Listas e estruturas de repetição
Dev & Data
Prof. Guilherme Assis

5.2. Estruturas de repetição

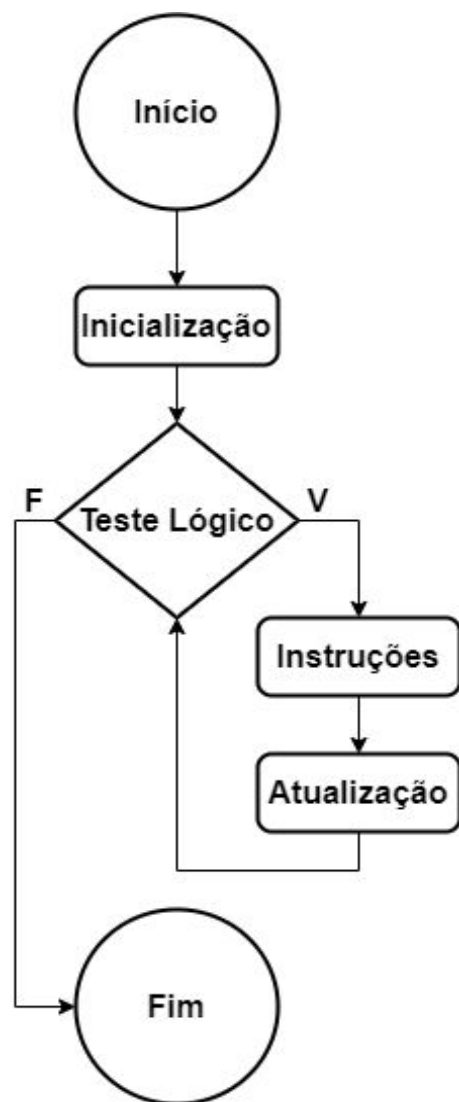
Estruturas de repetição

- Utilizadas para executar uma ação várias vezes.
- Definir quando aquela repetição deve encerrar a execução.
- Muito utilizada em conjunto com os vetores, para que possa realizar uma ação em todos os elementos.
- While e For.

While



For



Conclusão

- Estruturas de repetição.