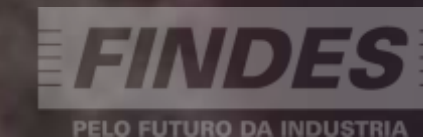




Estruturas Condicionais



Condicional

Adjetivo Que depende de certas condições ou restrições; condicionado: decisão condicional do caso.

Que traz consigo uma condição, exigência, requisito.

Estruturas Condicionais

Quando falamos em lógica de programação, as estruturas condicionais são recursos oferecidos pelas linguagens para que seja possível verificar uma condição e alterar o fluxo de execução do algoritmo. Assim, é possível definir uma ação específica para diferentes cenários e obter exatamente o resultado esperado durante o desenvolvimento de um site ou de uma aplicação.

Em descrição narrativa:

Início

Abrir o site

Se eu tiver 1 milhão

 Compro um pc gamer

Senão

 Fecho o site e choro

FimSe

Fim

Estruturas Condicionais

A Estrutura Condicional pode ser Simples ou Composta.

Estruturas Condicionais Simples

A Estrutura Condicional Simples executa um comando ou vários comandos se a condição for verdadeira. Se a condição for falsa, a estrutura é finalizada sem executar os comandos. O comando

que define a estrutura é representado pela palavra SE
Em descrição narrativa:

Início

Se condição for atingida

faço uma ação

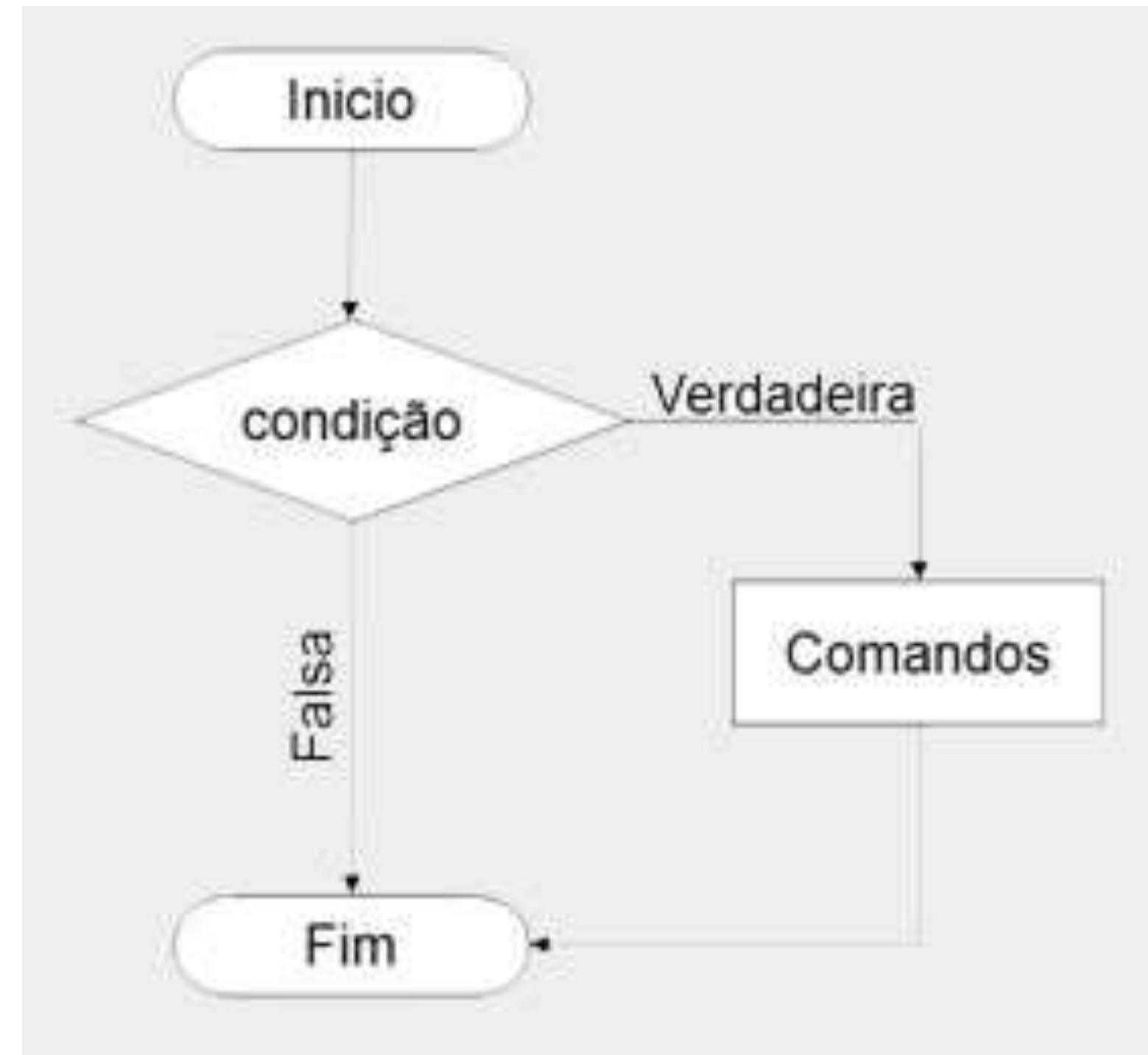
FimSe

Fim

Estruturas Condicionais Simples

Em pseudocódigo:

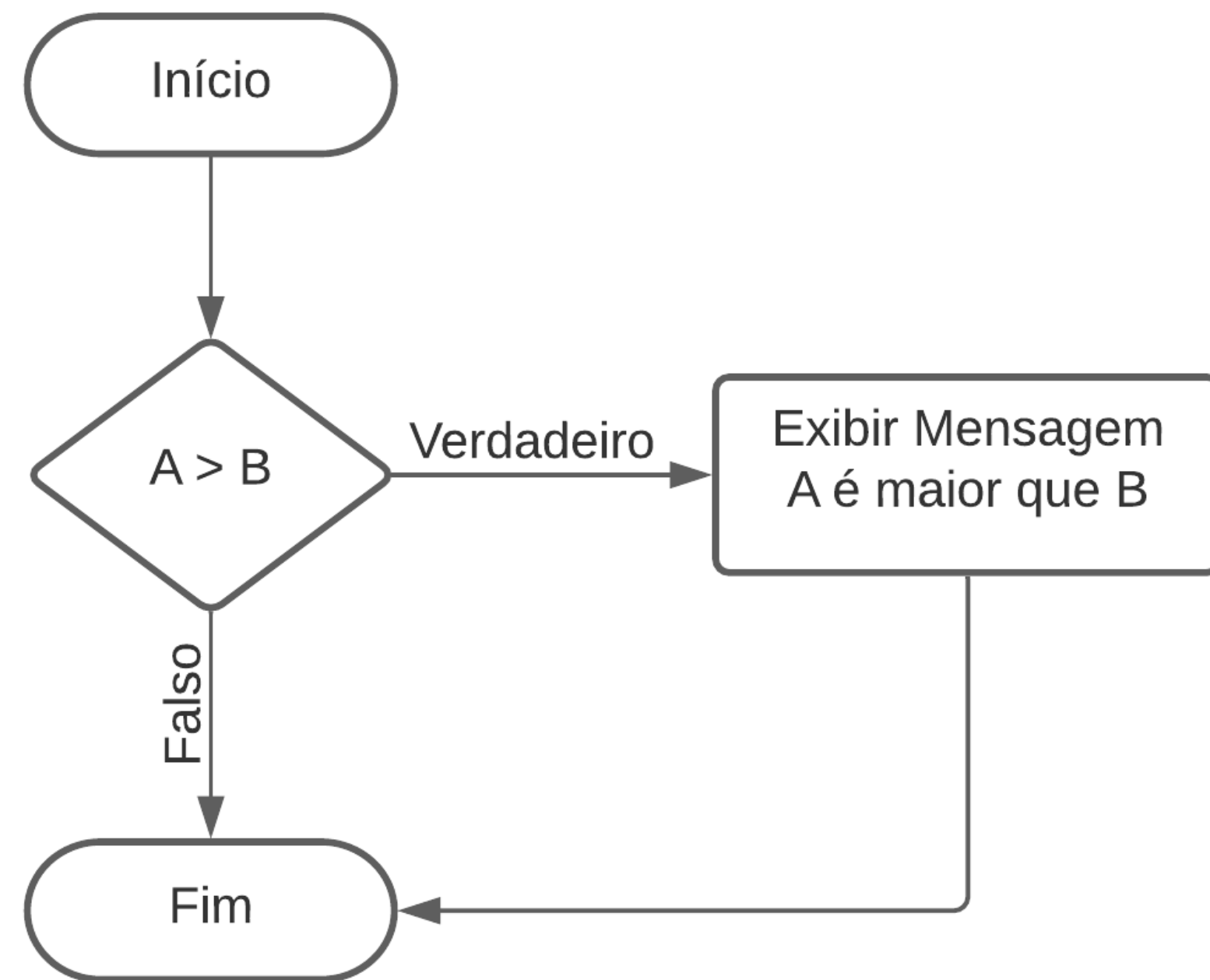
1. INICIO ALGORITMO
2. SE condição ENTAO
3. comando
4. FIM SE
5. FIM ALGORITMO



Estruturas Condicionais Simples

Em pseudocódigo:

1. INICIO ALGORITMO
2. SE (A>B) ENTAO
3. Escreval("A maior que B")
4. FIM SE
5. FIM ALGORITMO



Estruturas Condicionais Simples

- Área dos algoritmos (Edição do código fonte) -> Nome do arquivo: [semnome]

```

1 Algoritmo "semnome"
2 //
3 //
4 // Descrição      : Aqui você descreve o que o programa faz! (função
5 // Autor(a)       : Nome do(a) aluno(a)
6 // Data atual     : 11/05/2023
7 Var
8 // Seção de Declarações das variáveis
9
10 A, B : inteiro
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13
14 A := 10
15 B <- 6
16
17 se (A>B) entao
18     escreval("A é maior que B")
19 fimse
20
21 Fimalgoritmo

```

Áreas das variáveis de memória (Globais e Locais)

[illegible]

- Área de visualização dos resultados

```
Início da execução
A é maior que B

Fim da execução.
```


Estruturas Condicionais

Compostas

A Estrutura Condicional Composta segue o mesmo princípio da Estrutura Condicional Simples, com a diferença de que quando a condição não é satisfeita, será executado o outro comando. O comando que define a estrutura é representado pelas palavras SE e SE NÃO. Em descrição narrativa:

Início

Se condição for atingida

faço uma ação

Senão

faço outra ação

FimSe

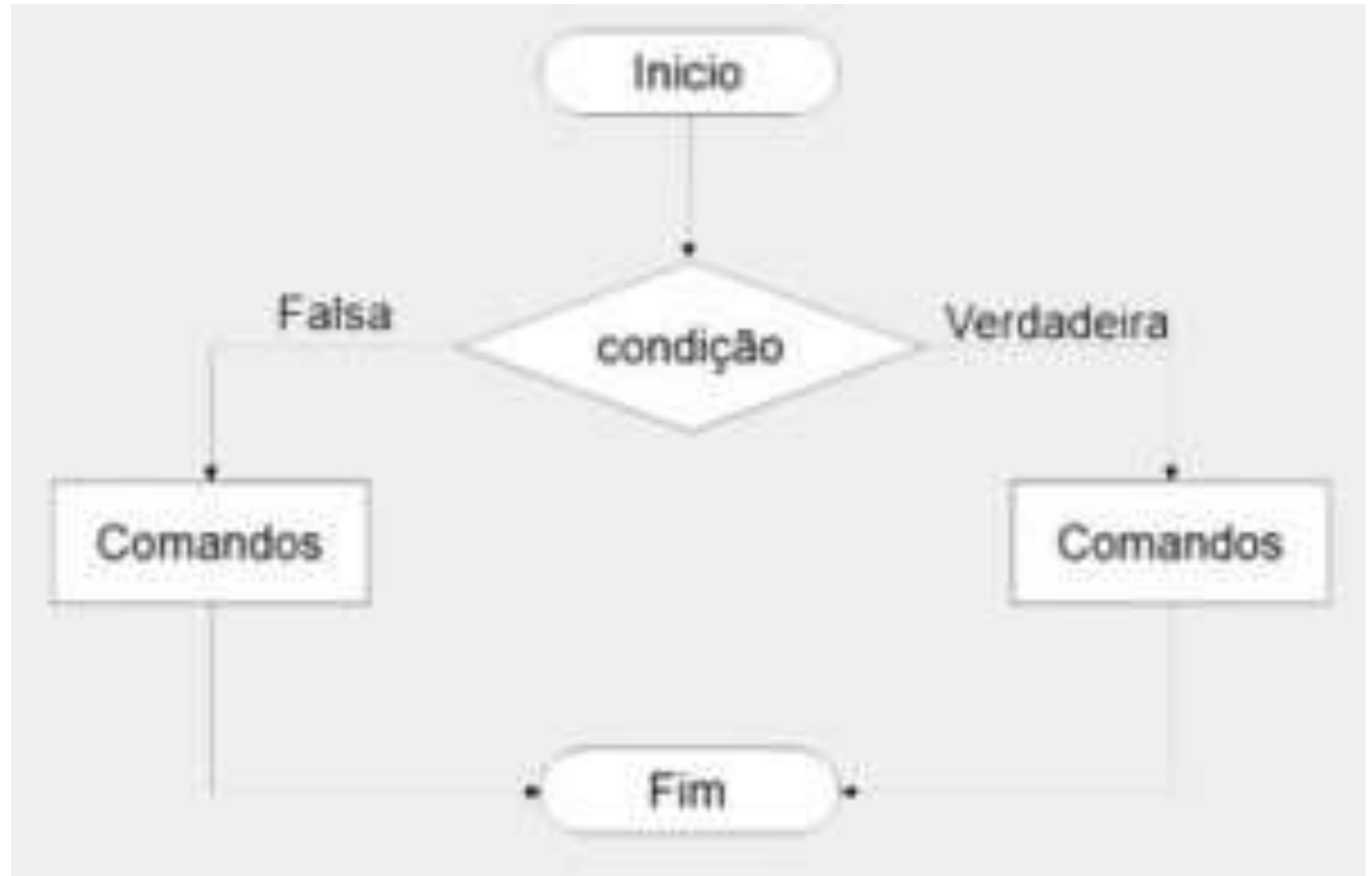
Fim

Estruturas Condicionais

Compostas

Em pseudocódigo:

1. INICIO ALGORITMO
2. SE condição ENTAO
3. comando A
4. SENA0
5. comando B
6. FIM SE
7. FIM ALGORITMO

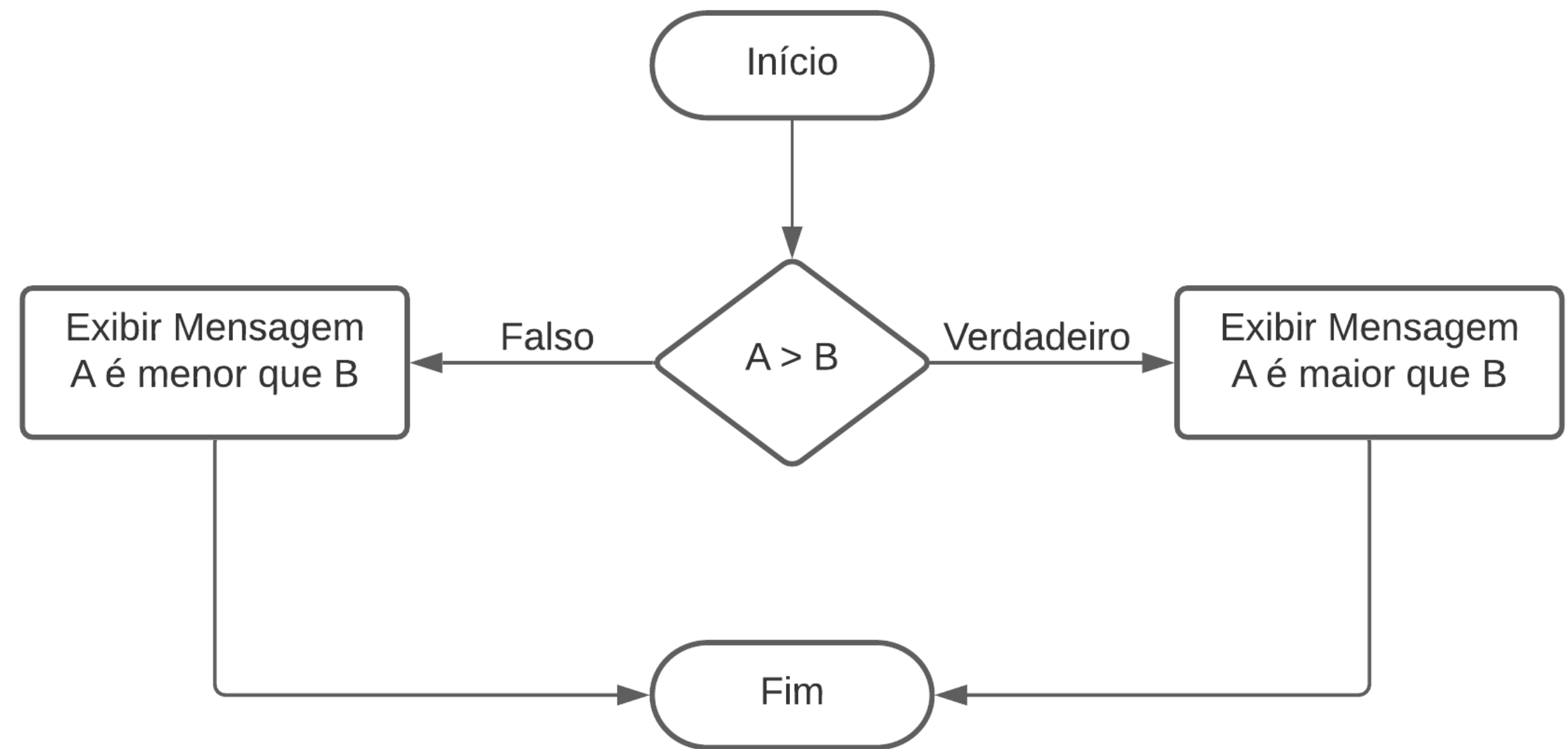


Estruturas Condicionais

Compostas

Em pseudocódigo:

1. INICIO ALGORITMO
2. SE (A>B) ENTAO
3. Escreval("A maior que B")
4. SENA0
5. Escreval("A menor que B")
6. FIM SE
7. FIM ALGORITMO



Estruturas Condicionais

- Área dos algoritmos (Edição do código fonte) -> Nome do arquivo: [semnome]

```

1 Algoritmo "semnome"
2 //
3 //
4 // Descrição      : Aqui você descreve o que o programa faz! (função
5 // Autor(a)       : Nome do(a) aluno(a)
6 // Data atual     : 11/05/2023
7 Var
8 // Seção de Declarações das variáveis
9
10 A, B : inteiro
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13
14 A := 10
15 B <- 16
16
17 se (A>B) entao
18     escreval("A é maior que B")
19 senao
20     escreval("A é menor que B")
21 fimse
22
23 Fimalgoritmo

```

Áreas das variáveis de memória (Globais e Locais)

[illegible]

Área de visualização dos resultados

```
Início da execução
A é menor que B

Fim da execução.
```

Indentação

Ação de indentar, de afastar o texto da sua margem, geralmente inserindo espaços entre a margem e o começo do parágrafo.

[Informática] Em linguagem de programação, digitação dos códigos do programa, afastados por espaço da margem e dispostos hierarquicamente, para facilitar a visualização e percepção do programa.

Indentação

Em ciência da computação, indentação é um termo aplicado ao código fonte de um programa para ressaltar ou definir a estrutura do algoritmo. Na maioria das linguagens de programação, a indentação é empregada com o objetivo de ressaltar a estrutura do algoritmo, aumentando assim a legibilidade do código

Indentação

```
1  <?php
2  if ($nota >= 7) {
3  echo 'Você foi aprovado!';
4  } else { if ($nota > 3) {
5  echo 'Você precisa fazer prova final!';
6  } else {
7  echo 'Você foi reprovado!';
8  }
9  }
```


Indentação

```
1  <?php
2  if ($nota >= 7)
3  {
4      echo 'Você foi aprovado!';
5  }
6  else
7  {
8      if ($nota > 3)
9      {
10         echo 'Você precisa fazer prova final!';
11     }
12     else
13     {
14         echo 'Você foi reprovado!';
15     }
16 }
```


Indentação

Um exemplo do emprego de indentação em C.

Código em C com indentação:

```
if (unlikely(prev->policy == SCHED_RR))
    if (!prev->counter) {
        prev->counter =
NICE_TO_TICKS(prev->nice);
        move_last_runqueue(prev);
    }
switch (prev->state) {
case TASK_INTERRUPTIBLE:
    if (signal_pending(prev)) {
        prev->state = TASK_RUNNING;
        break;
    }
default:
    del_from_runqueue(prev);
}
prev->need_resched = 0;
```

O mesmo código, sem indentação:

```
if (unlikely(prev->policy == SCHED_RR))
if (!prev->counter){
prev->counter = NICE_TO_TICKS(prev->nice);
move_last_runqueue(prev);
}
switch (prev->state){
case TASK_INTERRUPTIBLE:
if (signal_pending(prev)){
prev->state = TASK_RUNNING;
break;
}
default:
del_from_runqueue(prev);
}
prev->need_resched = 0;
```

Indentação

```
if (a > b && a > c)
{
maior = a;
}
else if (b > c)
{
maior = b;
}
else {
maior = c;
}
printf("%d\n", maior);
```

Sem indentação

```
if (a > b && a > c)
{
    maior = a;
}
else if (b > c)
{
    maior = b;
}
else {
    maior = c;
}
printf("%d\n", maior);
```

Com indentação



PELO FUTURO DA INDUSTRIA



PELO FUTURO DO TRABALHO