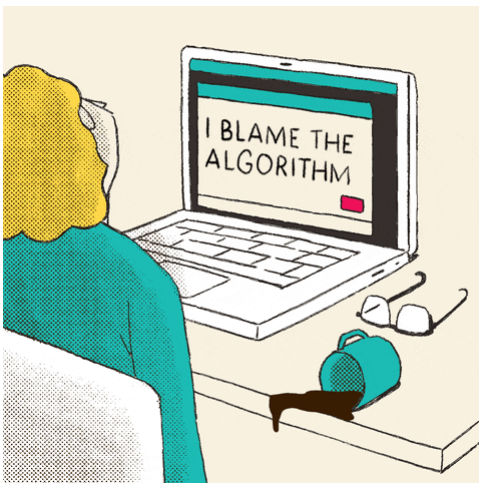


LÓGICA DE PROGRAMAÇÃO ALGORÍTIMA

Curso: Análise e Desenvolvimento de Sistemas
Prof. Me Enoch Menezes de Oliveira Junior

1

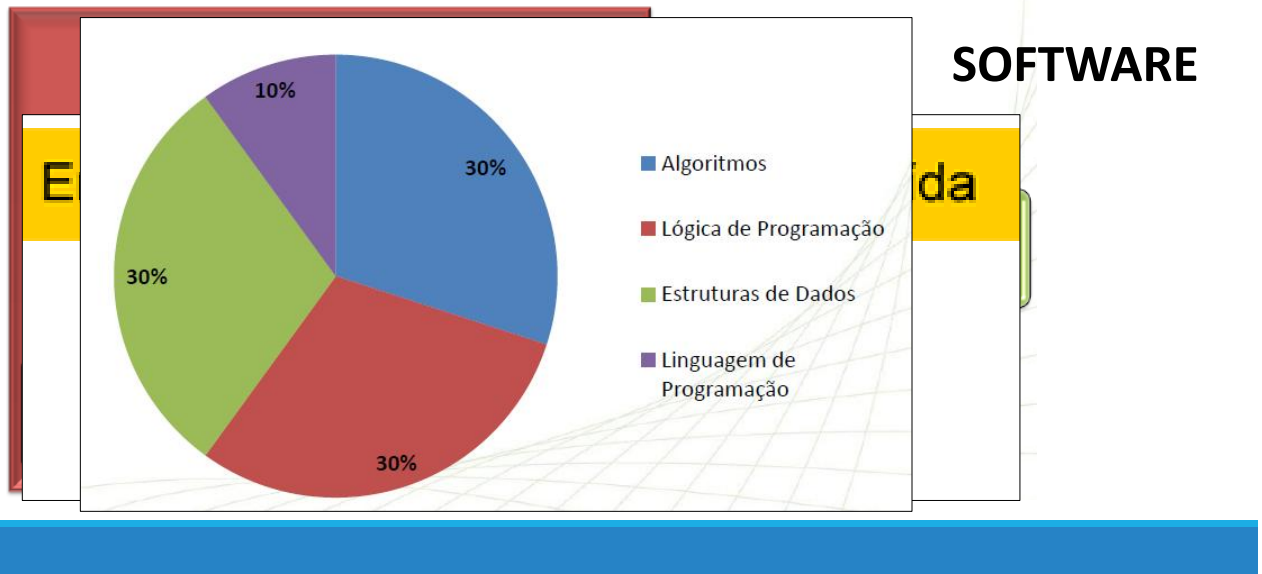
LÓGICA DE PROGRAMAÇÃO ALGORÍTIMA



Curso: Análise e Desenvolvimento de Sistemas
Prof. MSc Enoch Menezes de Oliveira Junior

2

Lógica de Programação Algorítmica



3

Lógica de Programação Algorítmica

Um algoritmo é um conjunto finito de regras que fornece uma sequência precisa de operações para resolver um problema específico.

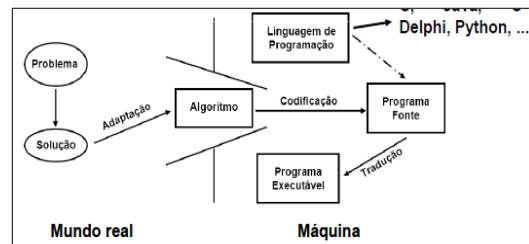
- ✓ **Finitude:** algoritmos devem terminar após um número finito de passos;
- ✓ **Definição:** cada passo deve ser precisamente definido;
- ✓ **Entradas:** devem ter zero ou mais entradas;
- ✓ **Saídas:** devem ter uma ou mais saídas;
- ✓ **Efetividade:** todas as operações devem ser simples de modo que possam ser executadas em um tempo limitado.

4

Lógica de Programação Algorítmica

O **algoritmo**, do ponto de vista computacional, tem um papel fundamental por ser o elo de ligação entre dois mundos (real e computacional).

A atividade de programação começa com a construção do algoritmo.



5

Lógica de Programação



6



Lógica de Programação Algorítmica

Tipos de Algoritmos:

- ✓ Descrição narrativa;
- ✓ Fluxograma;
- ✓ Pseudocódigo.

7

Lógica de Programação Algorítmica

➤ Descrição narrativa:

Analisa o enunciado do problema;

Escreve em linguagem natural os passos a serem seguidos para a resolução do problema;

Vantagem:

Não é necessário aprender novos conceitos.

Desvantagem:

Há espaço para múltiplas interpretações.

8

Lógica de Programação Algorítmica

Algoritmo: trocar lâmpada

Passo 1: pegar a lâmpada nova.

Passo 2: pegar a escada.

Passo 3: posicionar a escada embaixo da lâmpada queimada.

Passo 4: subir na escada com a lâmpada

Passo 5: Retirar a lâmpada queimada.

Passo 6: Colocar a lâmpada nova.

Passo 7: Descer da escada.

Passo 8: Ligar o interruptor.

Passo 9: Guardar a escada.

Passo 10: Jogar a lâmpada velha no lixo.

9

Lógica de Programação Algorítmica

➤ **Fluxograma:**

Analisa o enunciado do problema;

Escrever os passos a serem seguidos para a resolução do problema utilizando símbolos gráficos predefinidos;

Vantagem:

Fácil entendimento devido aos elementos gráficos.

Desvantagem:

É necessário aprender a simbologia dos fluxogramas.

O fluxograma não apresenta muitos detalhes do algoritmo.

10

Lógica de Programação Algorítmica

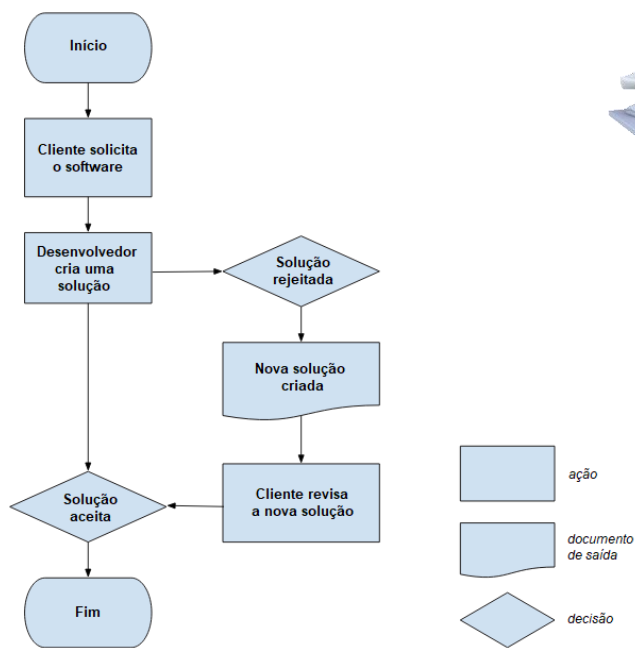


FLUXOGRAMA



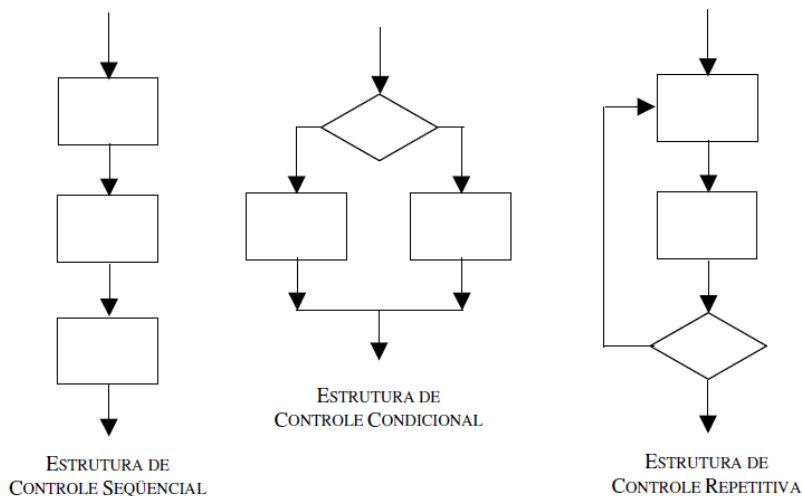
11

FLUXOGRAMA



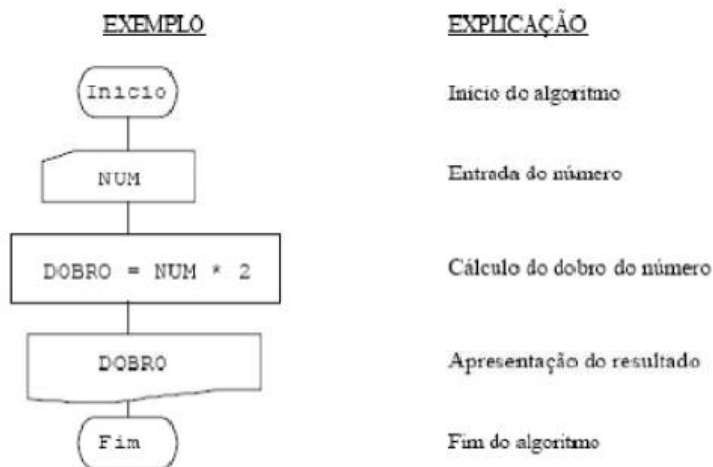
12

Lógica de Programação Algorítmica



13

Lógica de Programação Algorítmica



14

Lógica de Programação Algorítmica

➤ Pseudocódigo (Portugol).

Analisa o enunciado do problema;

Escrever os passos a serem seguidos para a resolução do problema utilizando regras de sintaxe pré-definidas;

Vantagem:

A transcrição do algoritmo para qualquer linguagem de programação é quase imediata, bastando conhecer as palavras reservadas da linguagem de programação de destino.

Desvantagem:

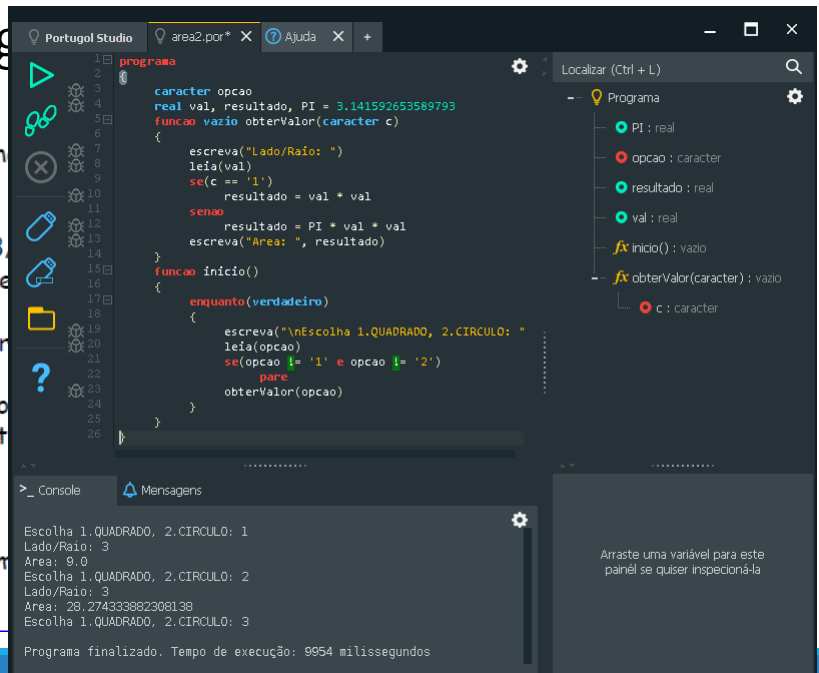
É necessário aprender as regras do pseudocódigo, a sua sintaxe.

15

```

algoritmo "som
// Função :
// Autor :
// Data : 30/3
// Seção de De
var
n1, n2, d: inteir
inicio
// Seção de Co
escreval("Digit
leia(n1)
leia(n2)
dk<-n1+n2
escreva("A som
finalgoritmo

```



```

Portugol Studio
area2.por x Ajuda x +

1 programa
2
3 caracter opcao
4 real val, resultado, PI = 3.141592653589793
5 funcao vazia obterValor(caracter c)
6 {
7   escreva("Lado/Raio: ")
8   leia(val)
9   se(c == '1')
10    resultado = val * val
11   senao
12    resultado = PI * val * val
13   escreva("Area: ", resultado)
14 }
15 funcao inicio()
16 {
17   enquanto(verdadeiro)
18   {
19     escreva("\nEscolha 1.QUADRADO, 2.CIRCULO: ")
20     leia(opcao)
21     se(opcao != '1' e opcao != '2')
22     pare
23     obterValor(opcao)
24   }
25 }
26

> Console Mensagens

Escolha 1.QUADRADO, 2.CIRCULO: 1
Lado/Raio: 3
Area: 9.0
Escolha 1.QUADRADO, 2.CIRCULO: 2
Lado/Raio: 3
Area: 28.274333882308138
Escolha 1.QUADRADO, 2.CIRCULO: 3

Programa finalizado. Tempo de execução: 9954 milissegundos

```

16

Lógica de Programação Algorítmica

Faça um algoritmo para mostrar o resultado da multiplicação de dois números.

Passo
Passo
Passo

ALGORITMO

```
DECLARE N1, N2, M NUMÉRICO
ESCREVA "Digite dois números"
Leia N1, N2
M <- N1 * N2
ESCREVA "Multiplicação = ", M
```

FIM_ALGORITMO

ultiplicados.
ação.

17

Lógica de Programação Algorítmica

Dados primitivos



Dados não primitivos

18

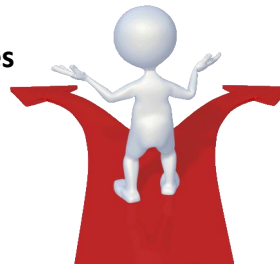
Lógica de Programação Algorítmica

Dados primitivos:

São valores simples que representam dados básicos, como números, caracteres e booleanos. Eles são tratados diretamente pela linguagem e são imutáveis, ou seja, o valor armazenado não pode ser modificado.



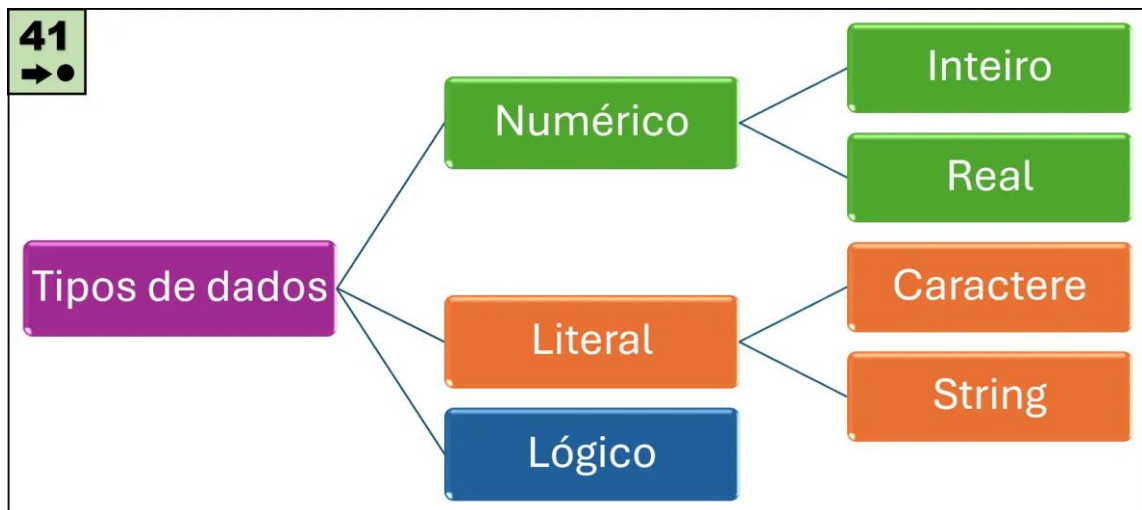
Constantes



Variáveis

19

Lógica de Programação Algorítmica



20

Lógica de Programação Algorítmica

Elementos Básicos para Construção de um Algoritmo:

Constante:

Valores fixos, tais como números. Estes valores não podem ser alterados pelas instruções do algoritmo.

- ✓ **Inteiro:** 10, -23768, ...
- ✓ **Real:** -2.34, 0.149, ...
- ✓ **Caractere:** "k", "computador"
- ✓ **Lógico:** Verdadeiro ou falso

21

Lógica de Programação Algorítmica



Elementos Básicos para Construção de um Algoritmo:

Variável.

Elemento de dado cujo valor pode ser modificado ao longo de sua execução.

Uma variável representa uma posição na memória e pode ter tipo (inteiro, caractere, real), tamanho (16, 32 bits, ...) e nome definidos.

22

Lógica de Programação Algorítmica

Elementos Básicos para Construção de um Algoritmo:

Identificadores.

Nomes utilizados para referenciar variáveis, funções ou vários outros objetos definidos pelo programador.

Exemplo:

letras, dígitos e sublinhado(_);

Não podem começar com dígito;

Não podem ser iguais a uma palavra-chave e nem iguais a um nome de uma função declarada pelo programador ou pelas bibliotecas da linguagem utilizada.

23

Lógica de Programação Algorítmica

Elementos Básicos para Construção de um Algoritmo:

Palavras-reservadas (palavras-chave).

São identificadores predefinidos que possuem significados especiais para o interpretador do algoritmo.

inicio - senao - para - enquanto

var – logico - se - ate

Faca - inteiro - real

24

Lógica de Programação Algorítmica

Dados primitivos:

São valores simples que representam dados básicos, como números, caracteres e booleanos. Eles são tratados diretamente pela linguagem e são imutáveis, ou seja, o valor armazenado não pode ser modificado.



Variáveis locais



Variáveis globais

25

Lógica de Programação Algorítmica

Elementos Básicos para Construção de um Algoritmo:

Declaração de Variáveis

Palavra-reservada: **var** - utilizada para iniciar a seção de declaração de variáveis.

Exemplo:

```
var a: inteiro
    nome_do_aluno: caractere
    sinalizador: logico
    valor1, valor2: real
```

26

Lógica de Programação Algorítmica

Declaração de Variáveis

Regra para criar nomes de variáveis.

Os nomes das variáveis devem representar o que será guardado dentro dela.

O primeiro caractere de um nome deverá ser sempre alfabético.

Não podem ser colocados espaços em branco no nome de variáveis, usar o UNDERSCORE “_”.

A declaração de uma variável é feita no algoritmo informando o seu nome, seguido por : e terminado com o seu tipo.

27

Lógica de Programação Algorítmica

Dados não primitivos:

São estruturas de dados mais complexas, como vetores, matrizes, objetos e enumeradores. A aplicação guarda o endereço em memória do dado, e não o valor em si.



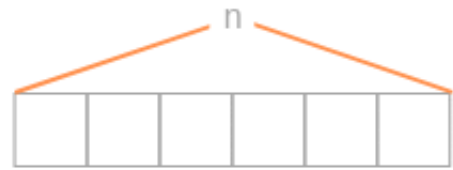
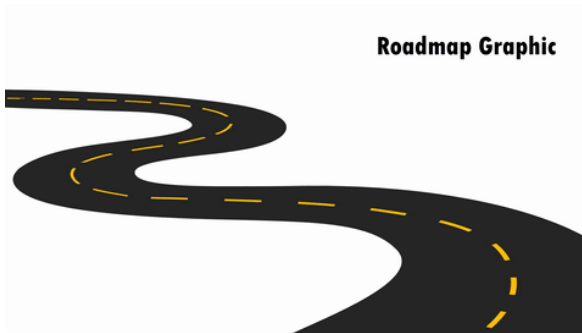
Vetores



Matrizes

28

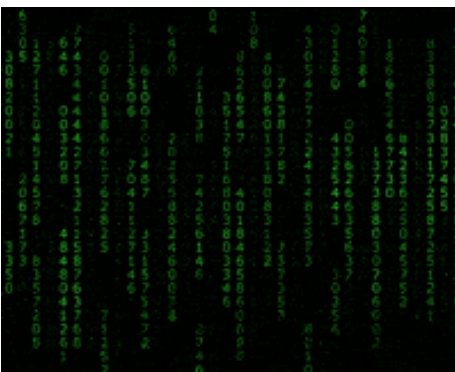
Lógica de Programação Algorítmica



VETORES

29

Lógica de Programação Algorítmica



$$\begin{bmatrix} 5 & 7 \\ 3 & 1 \\ 5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 2 & 2 \\ 8 & 0 & 0 \end{bmatrix}$$

MATRIZES

30

Lógica de Programação Algorítmica

E C A	<-	Atribuição. $x \leftarrow 2$. A variável x recebeu o valor 2. Logo $x = 2$
	+	Adição
	-	Subtração
	*	Multiplicação
	/	Divisão
	$a \backslash b$	Retorna o quociente da divisão inteira de a por b
	$a \% b$	Retorna o resto da divisão inteira de a por b
	a^b	Retorna o valor de a elevado a b
	$a^{1/b}$	Retorna a raiz b de a
	aleatorio (a)	Retorna um número aleatório, em intervalo fechado, entre 0 e a

31

Lógica de Programação Algorítmica



E C R	Operador	Ação
	>	maior que
	>=	maior ou igual a
	<	menor que
	<=	menor ou igual
	=	igual a
	<>	diferente de

32



Lógica de Programação Algorítmica

Elementos Básicos para Construção de um Algoritmo:

Operadores:
Lógicos;

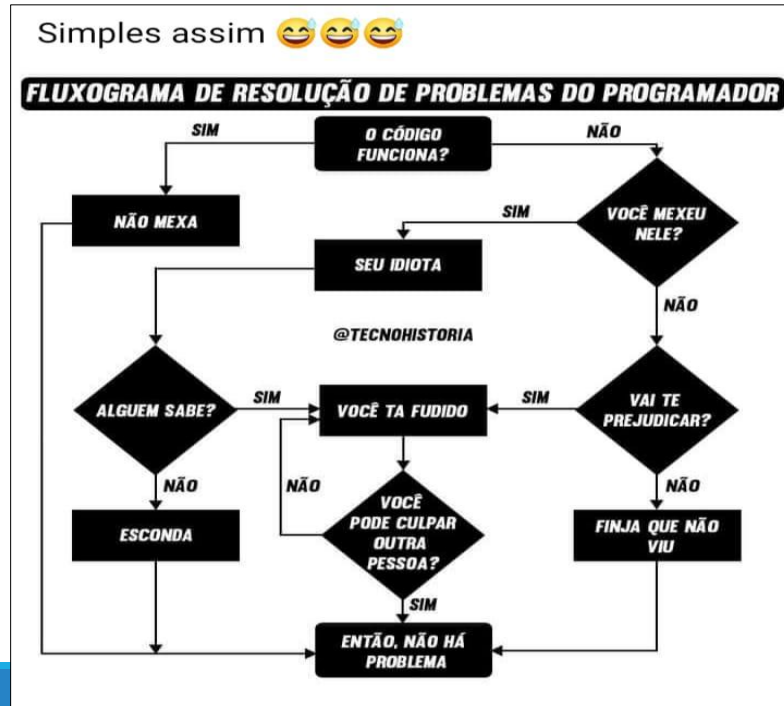
Operador
e
ou
nao

33

Lógica de Programação Algorítmica



34



35

Fluxogramas

- Crie um algoritmo que leia o nome de uma pessoa e mostre uma mensagem de “boas vindas” para ela.
- Crie um algoritmo que leia um número inteiro e mostre o seu antecessor e seu sucessor.
- Crie um algoritmo que leia um número real e mostre na tela o seu dobro e a sua terça parte.
- Crie um algoritmo que calcule a idade de uma pessoa.
- Desenvolva um algoritmo que calcule a média aritmética de dois valores.
- Desenvolva um algoritmo que converta o valor em reais para dólar.
- Desenvolva um algoritmo que leia uma distância em metros e mostre os valores relativos em quilômetros, centímetros e milímetros.
- Crie um algoritmo que calcule a área de uma parede e mostre a quantidade de tinta necessária para o serviço, sabendo que cada litro de tinta pinta uma área de 2 metros quadrados.
- Desenvolva um algoritmo que receba o valor de um produto e mostre o valor com 30% de desconto.
- Desenvolva um algoritmo, que calcule o valor a ser recebido, de acordo com a quantidade de dias trabalhados, levando em consideração, um carga-horária de 08 horas diárias e 25,00 R\$ por hora.

36