



INTERPRETANDO ALGORITMOS





Por onde começar?

Existe uma estrutura padrão a ser seguida para construir um algoritmo. Os principais elementos são:

- O **NOME DO ALGORITMO**, você precisa dar um nome para seu programa.
- Você precisa definir **VARIÁVEIS**, elas servem para armazenar dados que serão fornecidos ou gerados e essas variáveis possuem tipos.
- Seu algoritmo precisa ter um ponto de **INÍCIO**.
- Depois é necessário receber os dados que serão inseridos, ou seja, seu computador vai **LER** estes dados.
- Para que o computador compreenda o que está sendo inserido precisamos colocar uma **LÓGICA**, ou seja, colocar um sentido. Assim, ao ler essa lógica o computador saberá o que fazer.
- E por último, precisamos do resultado do processamento. Seu algoritmos vai **ESCREVER** o resultado.
- Para finalizar todos os processos colocamos um **FIM**.



Estrutura Básica

Algoritmo Nome

Var

Início

Leia

Escreva

Fim



Um algoritmo é uma sequência finita de ações executáveis que visam obter uma solução para um determinado tipo de problema.

(LOPES; GARCIA, 2002).



Há três tipos de algoritmos:

NARRATIVO

Descreve os passos para execução do processo

FLUXOGRAMA

Forma simbólica de descrever os passos.

PSEUDOCÓDIGO

Linguagem nativa restrita e estruturada





TIPOS DE ALGORITMOS

Narrativo

algoritmo "Tomar café"

inicio

partir o pão

passar manteiga no pão

pegar a xícara

colocar café na xícara

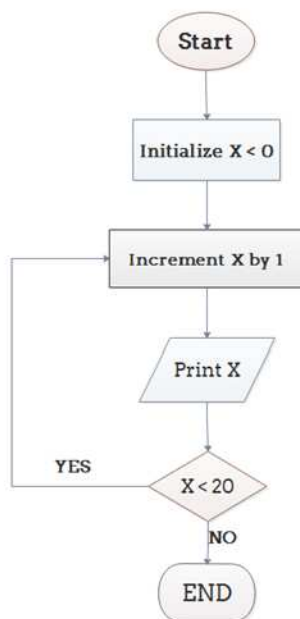
sentar na cadeira

comer o pão

beber o café

fimalgoritmo

Fluxograma



Pseudocódigo

algoritmo "Soma"

var

num1, num2, resultado :
inteiro

inicio

escreva ("Digite o primeiro
número")

leia (num1)

escreva ("Digite o segundo
número")

leia (num2)

resultado <- num1+num2

escreva ("Resultado =",
resultado)

fimalgoritmo

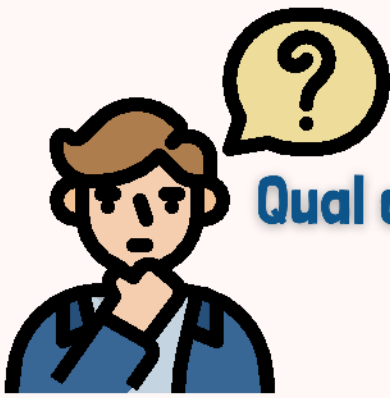
ALGORITMO



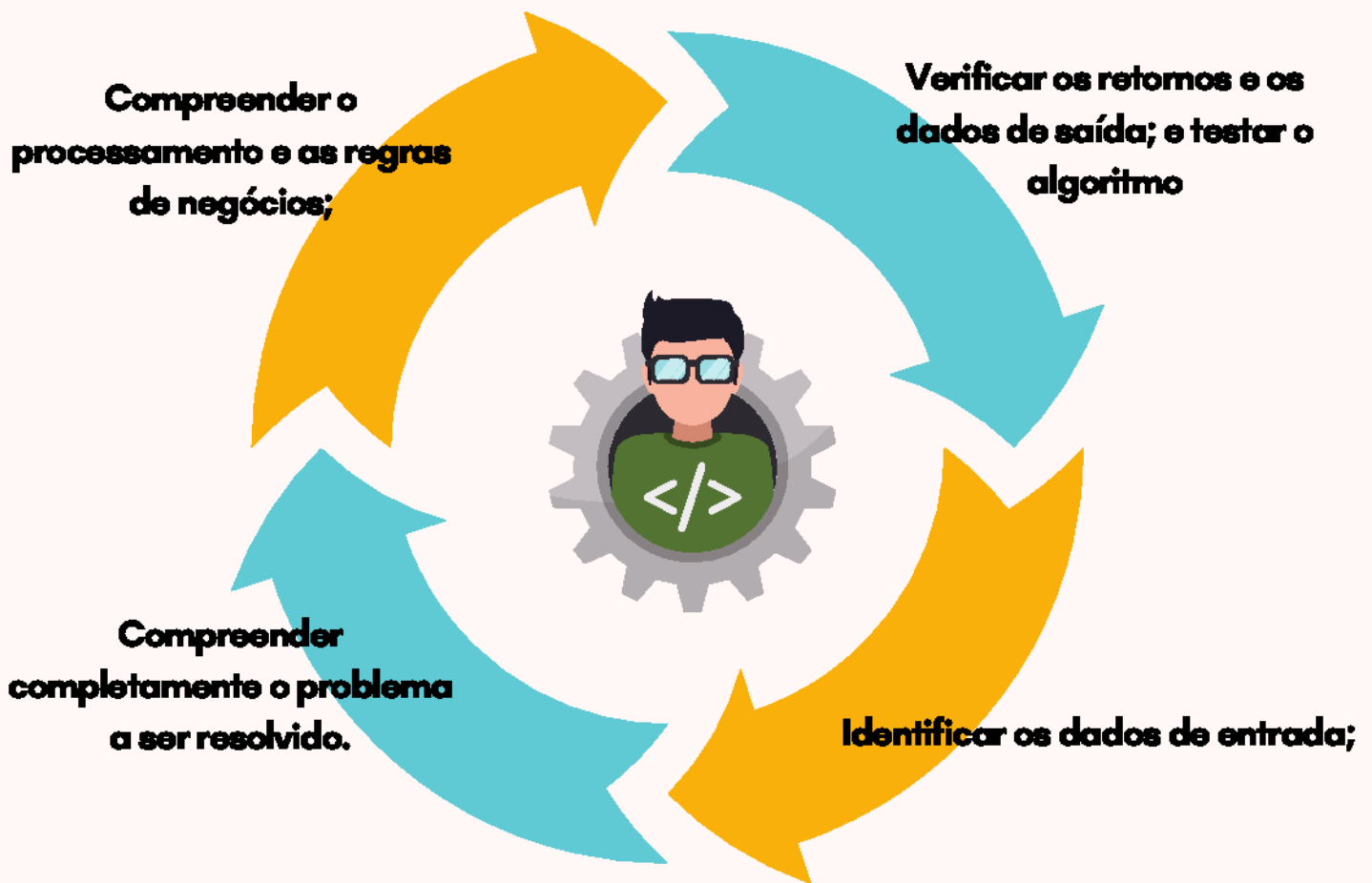
INTERPRETANDO CÓDIGOS

**POR QUE
INTERPRETAR**



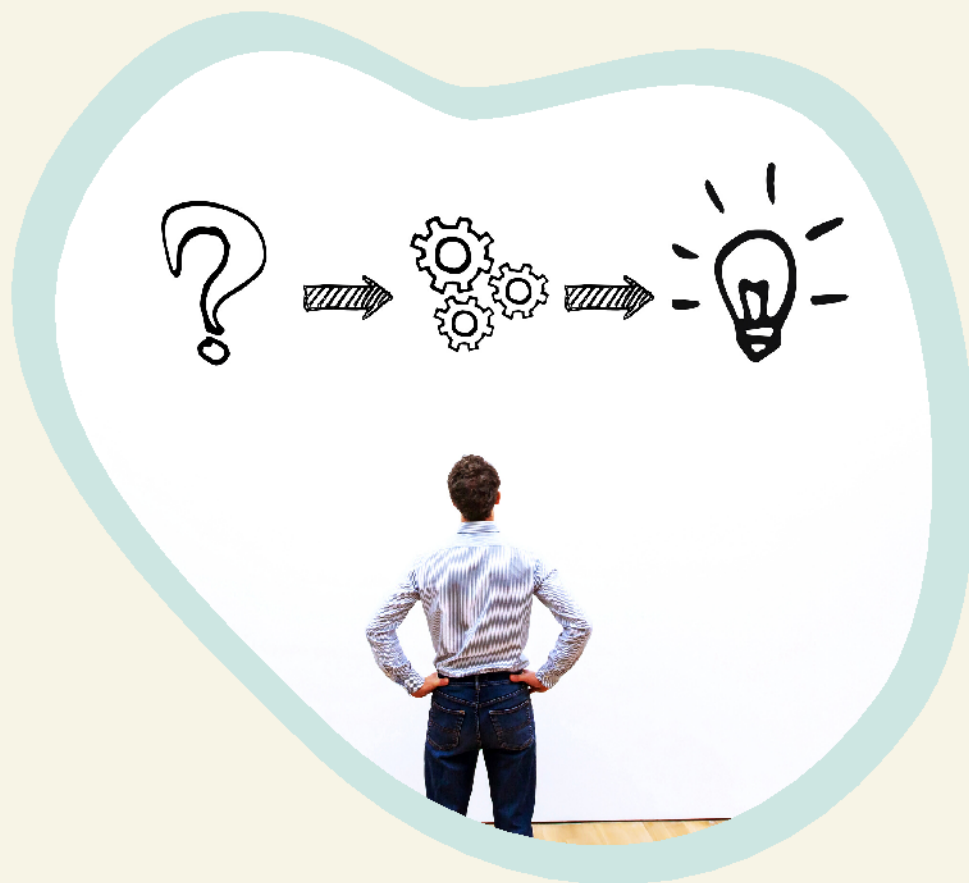


Qual o objetivo de interpretar um Algoritmo?



Ao formular algoritmos, precisamos guardar algumas informações do problema. Além de guardar informações, precisaremos obter dados, mostrar mensagens e resultados de processamento.

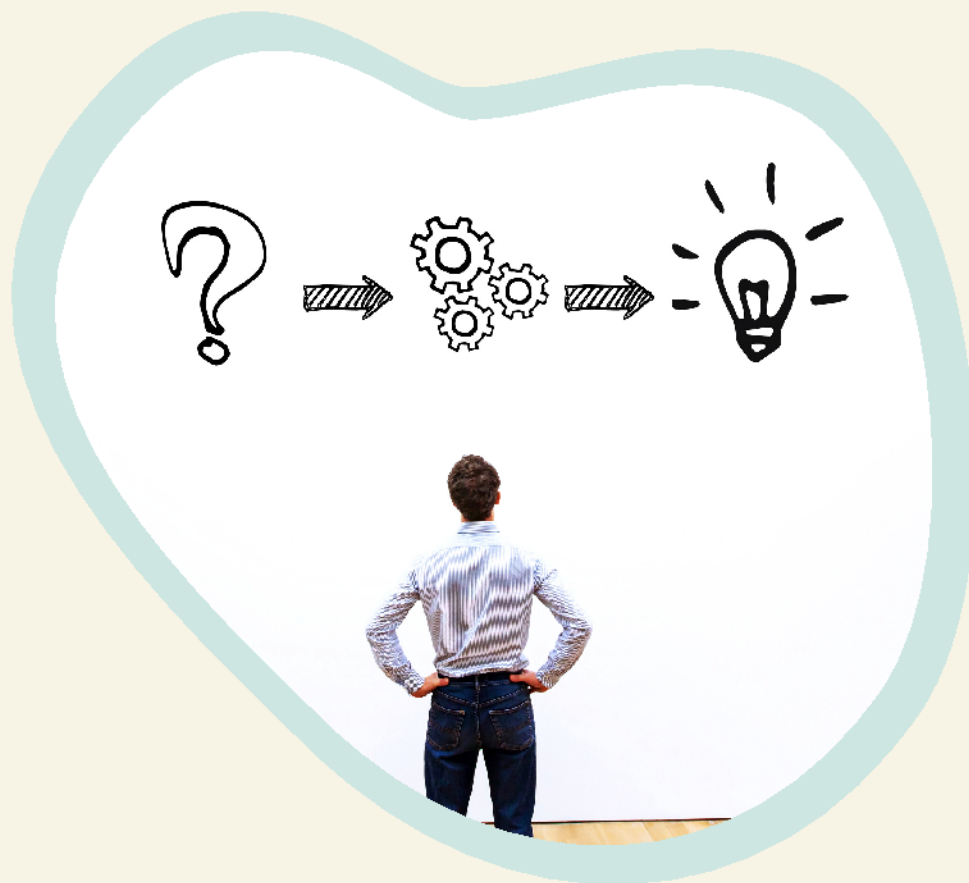
Ao resolver algoritmos, construímos, gradativamente, a nossa própria lógica de programação. Não há receita mágica, e o aprendizado de algoritmos requer prática.



EXEMPLO

Problema

Desenvolva um algoritmo que acumule as vendas de um dia. Para encerrar a leitura, informa o número 0 e ao final imprima o montante da venda do dia.



EXEMPLO

Solução

Algoritmo AcumuleVenda

Var

venda, somaVenda: inteiro

Inicio

somaVenda <- 0

Escreva("Informe o valor de venda diferente de 0 para continuar")

Leia(venda)

enquanto (venda <> 0) faça

somaVenda <- somaVenda + venda

Escreva("Informe o valor de venda diferente de 0 para continuar")

Leia(venda)

Fim_enquanto

Escreva("Venda acumulada do dia = ", somaVenda)

Fim

