

LÓGICA DE PROGRAMAÇÃO

Aula 1 Algoritmos



AGENDA

Algoritmos e Lógica de Programação

- Introdução à Lógica de Programação
- Algoritmos e Técnicas de Representação
- Representação x Implementação

Introdução à Lógica de Programação

- **Linguagem de Máquina:** composta por bits, que representam presença e ausência de corrente elétrica atravessando o componente eletrônico.
- **Linguagem de Programação:** idioma intermediário, que traduz instruções lógicas (comandos) escritas por humanos em instruções binárias entendíveis pela máquina.
- **Linguagem Humana:** utilizada para comunicação entre pessoas.

A lógica de programação é a técnica que nos permite escrever, em linguagem de programação, as instruções que vão comandar a etapa do processamento.

LÓGICA DE PROGRAMAÇÃO

Estrutura utilizada por todas as linguagens de programação para escrever instruções lógicas
Representada através de algoritmos

LINGUAGEM DE PROGRAMAÇÃO

Idioma em que as instruções lógicas podem ser implementadas e reconhecidas pela máquina
Implementada através de códigos-fonte (C, Java, Python, Javascript, PHP, Ruby...)

Algoritmos

Veja os exemplos abaixo:

- Receitas culinárias
- Manuais de instrução
- Bulas de medicamentos

O que estes elementos têm em comum?

Eles estabelecem uma **sequência finita e ordenada de tarefas** a serem executadas a fim de atingir determinado objetivo.

Na programação de sistemas, chamamos de *algoritmo* o **conjunto de instruções lógicas sequenciais necessárias para realizar um processamento**. Essas instruções, também chamadas de comandos, têm como objetivo extrair valor dos dados, gerando, assim, informação.

Softwares são algoritmos agrupados para executar uma tarefa específica. Eles são escritos na forma de comandos em uma linguagem de programação e armazenados em um *código-fonte*. O programador é a pessoa que escreve o código-fonte, enquanto o computador traduz e o executa esse mesmo código.

Técnicas de Representação de Algoritmos

DESCRIÇÃO NARRATIVA

Escrever em linguagem natural os passos a serem seguidos para a execução de uma tarefa (receita de bolo).

- Ponto positivo: não é necessário aprender novos conceitos, pois a língua natural já é conhecida.
- Ponto negativo: a linguagem natural abre espaço para interpretações ambíguas.

DIAGRAMA DE BLOCOS

Utilizar símbolos gráficos para representar a sequência dos passos e o tipo de tarefa a ser realizada.

- Ponto positivo: a visualização facilita o entendimento do fluxo de operações.
- Ponto negativo: abstrai elementos importantes para a implementação em uma linguagem de programação.

PSEUDOCÓDIGO

Escrever, por meio de regras predefinidas, os passos a serem seguidos para a execução de uma tarefa.

- Ponto positivo: a representação é clara e objetiva, utilizando palavras já existentes na língua natural.
- Ponto negativo: as regras do pseudocódigo para o uso de determinadas palavras podem ser diferentes do seu uso cotidiano em linguagem natural.

DESCRIÇÃO NARRATIVA: EXEMPLOS

FAZER UM SANDUÍCHE

1. Pegar o pão, o molho e o recheio
2. Cortar o pão ao meio na vertical
3. Colocar um pouco de molho dentro do pão fatiado
4. Colocar o recheio
5. Fechar o pão
6. Degustar a refeição

ESTACIONAR ENTRE DOIS VEÍCULOS

1. Emparelhar seu veículo com o veículo da frente
2. Girar o volante para a esquerda e mover o veículo para trás até o meio-fio
3. Girar o volante para a direita e mover o veículo para trás até que se encaixe
4. Uma vez encaixado, mova o veículo para a frente até estar equidistante dos dois outros veículos

DIAGRAMA DE BLOCOS: REGRAS

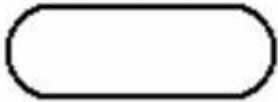
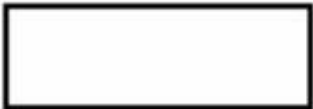
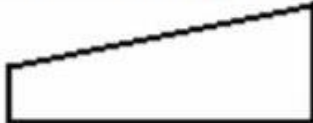


Símbolo	Função
 TERMINAL	Indica o INICIO ou FIM de um fluxo de dados
 PROCESSAMENTO	Indica o processamento de informações Exemplo: $SOMA = A + B$
 ENTRADA	Indica a entrada de dados através do teclado
 SAÍDA	Mostra mensagens ao usuário: informações, solicitações ou resultados.
 CONECTOR	Indica o sentido do fluxo de dados e conecta símbolos e/ou blocos.

DIAGRAMA DE BLOCOS: EXEMPLO

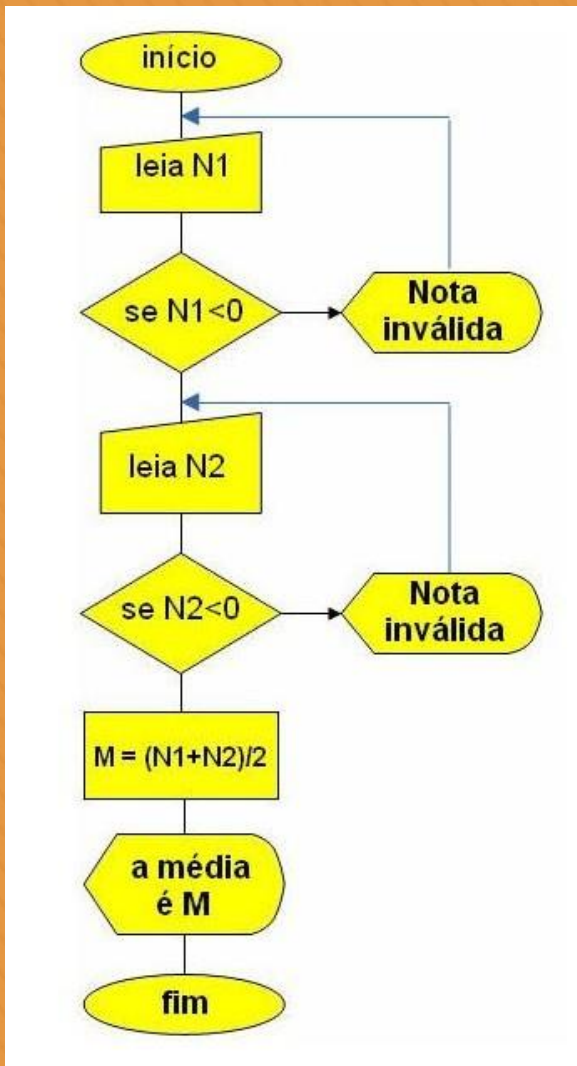
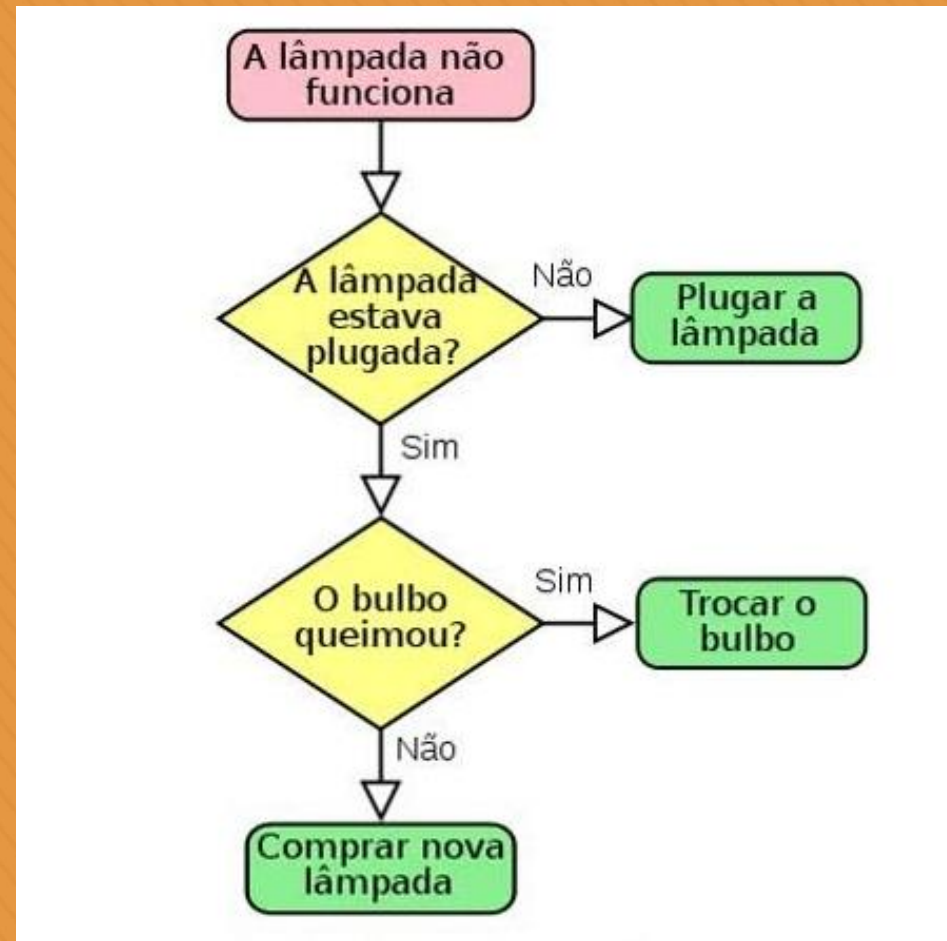


DIAGRAMA DE BLOCOS: EXEMPLO



Pseudocódigo

O pseudocódigo é um código de escrita em que se utilizam termos convencionais para indicar as instruções do programa. Esses termos são geralmente uma mistura de palavras da nossa linguagem natural com palavras e notações típicas das linguagens de programação.

A utilização de pseudocódigo permite ao programador estruturar o sem se preocupar com a sintaxe. Para isso, são utilizados comandos genéricos que podem ser facilmente traduzidos para uma linguagem de programação específica.

```
algoritmo "semnome" |  
// Função :  
// Autor :  
// Data : 24/02/2019  
// Seção de Declarações  
var  
  
inicio  
// Seção de Comandos  
fimalgoritmo
```

PSEUDOCÓDIGO: BÁSICO

```
algoritmo "semnome" |  
// Função :  
// Autor :  
// Data : 24/02/2019  
// Seção de Declarações  
var  
  
inicio  
// Seção de Comandos  
fimalgoritmo
```


PSEUDOCÓDIGO: EXEMPLO

```
algoritmo "semnome"  
var  
    N1, N2, N3, N4 : Inteiro  
    M : Real  
inicio  
  
    Escreva("Digite o 1º número: ")  
    Leia(N1)  
    Escreva("Digite o 2º número: ")  
    Leia(N2)  
    Escreva("Digite o 3º número: ")  
    Leia(N3)  
    Escreva("Digite o 4º número: ")  
    Leia(N4)  
    M <- (N1+N2+N3+N4)/4  
    Escreva("Média aritmética = ", M)  
fimalgoritmo
```

Representação x Implementação

O objetivo das técnicas de escrita de algoritmos é estimular o Pensamento Lógico-Computacional, pois programar é, antes de tudo, raciocinar. O Algoritmo é, portanto, a **REPRESENTAÇÃO** do raciocínio do programador.

O ato de traduzir um Algoritmo representado pelo Diagrama de Blocos, Pseudocódigo ou Descrição Narrativa para uma linguagem de programação específica é chamado de **IMPLEMENTAÇÃO**.

Perceba que **a REPRESENTAÇÃO do algoritmo independe da IMPLEMENTAÇÃO**, pois do mesmo modo que uma receita funciona em qualquer que seja o tamanho da cozinha, da forma ou do forno, a sequência de passos para executar um processamento sempre será a mesma, não importa qual seja a linguagem escolhida pelo programador.

Por isso o domínio das técnicas de representação de algoritmos são tão importantes: eles permitem que o programador fique independente de qualquer ferramenta.

danielle.carvalho@go.senac.br

Serviço Nacional de
Aprendizagem
Comercial - Goiás

go.senac.br
senacgoias
senacgoias

