

[< VOLTAR](#)

Introdução à Lógica de Programação

Apresentar os conceitos introdutórios sobre a lógica de programação. Serão apresentadas as definições sobre lógica, lógica de programação e algoritmos.

NESTE TÓPICO

- > Introdução
- > Lógica de Programação
- > Algoritmo
- > Fases de um algoritmo

[Marcar
tópico](#)

Introdução

A palavra lógica, de acordo com o dicionário Michaelis, consiste na análise das formas e leis do pensamento, porém sem se preocupar com a produção do pensamento. Isto quer dizer que a lógica não se preocupa com o conteúdo do pensamento, mas sim com a maneira pela qual um pensamento ou ideia são organizados e apresentados.

Dessa forma, podemos relacionar a lógica com a "correção do pensamento", pois uma de suas preocupações é determinar quais operações são válidas e quais não são, fazendo análises das formas e leis do pensamento. Por isso, a lógica estuda e ensina a colocar "ordem no pensamento".

Inconscientemente, sempre que pensamos, a lógica ou a ilógica nos acompanham no dia-a-dia. Isso acontece desde os tempos primitivos, onde o homem utiliza-se do raciocínio lógico para a realização das suas atividades. Quando falamos ou escrevemos, estamos expressando nosso pensamento, logo, precisamos usar a lógica nessas atividades. Podemos, então, citar alguns exemplos relacionados a algumas atividades cotidianas, tais como:

- Locomoção de uma pessoa adulta para o trabalho: primeiramente, a pessoa deve vestir-se, em seguida, pegar uma condução (metrô, trem, ônibus ou automóvel) e, por fim, dirigir-se ao trabalho.
- Retirada de dinheiro no caixa eletrônico: a pessoa que tem conta corrente em um determinado banco precisa, inicialmente, dirigir-se até um caixa eletrônico, em seguida, inserir o cartão, digitar a senha para, então, efetuar



o saque.

A lógica é aplicada a diversas ciências, tais como a informática, a psicologia e a física, entre outras. Na informática e computação, aplica-se todas as suas áreas, para a construção e funcionamento do hardware e do software, sendo este último relacionado à construção de programas para computadores. Um programa de computador é uma sequência de ações (instruções) inseridas na máquina que permite ao computador realizar determinadas tarefas. Com o objetivo de nos auxiliar na resolução de problemas de construção de programas, faremos o uso da lógica de programação.

Lógica de Programação

A Lógica de Programação é a fase de, a partir de um problema, formular soluções que gerem um programa de computador. Ainda significa o uso correto das leis do pensamento, da "ordem da razão" e de processos de raciocínio e simbolização formais na programação de computadores.

Em primeiro lugar, deve-se saber qual é o problema a ser resolvido pelo programa: o seu objetivo. Daí, deve-se extrair todas as informações a respeito desse problema (dados e operações), relacioná-las com o conhecimento atual que se tem sobre o assunto, buscando eventualmente informações de outras fontes. Dessa forma, a lógica de programação representa a modelagem do problema em questão e é resultante de um processo mental de abstração.

Depois, sabendo como resolver o problema, a tarefa consiste em descrever claramente os passos para se chegar a sua solução. Os passos (instruções), por si só, não resolvem o problema; é necessário colocá-los em uma sequência lógica, que, ao ser seguida, de fato o solucionará. Nesse caso, é importante que essa descrição possua algum tipo de convenção para que a sequência de passos seja entendível por qualquer pessoa e, mais adiante, possa ser representada em qualquer uma das inúmeras linguagens de programação. Assim, a essa sequência de passos dá-se o nome de algoritmo.

Algoritmo

Algoritmo consiste em uma sequência lógica e finita de instruções que devem ser seguidas para a resolução de um problema ou a execução de uma tarefa.

Na medida em que precisamos especificar uma sequência de passos, é necessário utilizar ordem, ou seja, "pensar em ordem", portanto precisamos utilizar a lógica.

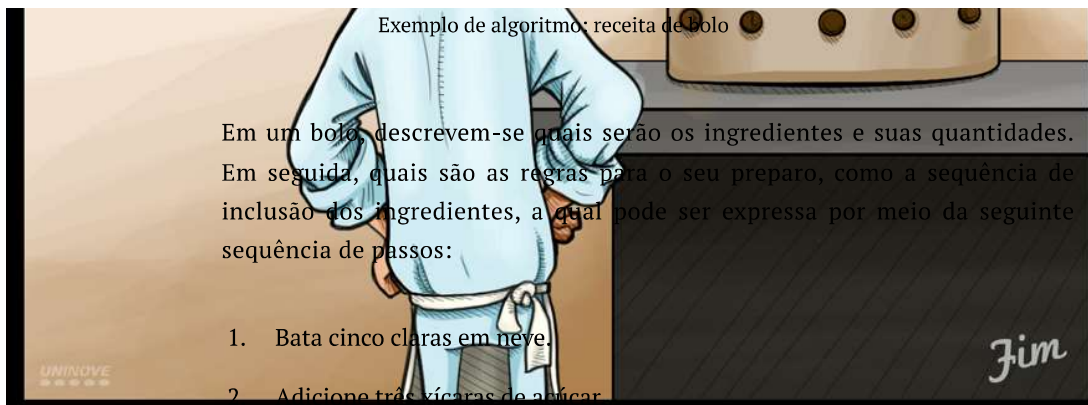
Algoritmos são comuns em nosso cotidiano como, por exemplo, uma receita de bolo (Figura a seguir).





Receita de Bolo





3. Adicione três xícaras de farinha de trigo, cinco gemas, uma colher de fermento e três colheres de chocolate.
4. Bata por cinco minutos.
5. Unte uma assadeira com margarina e farinha de trigo.
6. Coloque o bolo para assar durante vinte e cinco minutos em temperatura média.

As instruções contidas na receita do bolo devem ser corretamente executadas para que haja sucesso na sua execução. No entanto, se essas instruções tiverem sua ordem trocada ou a quantidade dos ingredientes alterada, o resultado vai divergir do original. Da mesma forma, o algoritmo deve especificar, como clareza e de forma correta, as instruções que um programa deverá conter para que, ao ser executado, forneça resultados esperados.

Podemos escrever um algoritmo que descreva o comportamento na resolução de uma determinada atividade como, por exemplo, a retirada da devolução do imposto de renda por uma pessoa em uma instituição bancária. A sequência de passos poderá ser a seguinte:

1. Sair de casa
2. Ir ao banco
3. Dirigir-se ao balcão
4. Já chegou a devolução do imposto de renda?
5. Sim: Vá para o passo 11
6. Não: Voltar para casa
7. Esperar um dia
8. É sábado, domingo ou feriado?
9. Sim: Vá para o passo 7
10. Não: Vá para o passo 1
11. Retirar o dinheiro
12. Voltar para casa



Outro exemplo de algoritmo bastante conhecido é o problema das Torres de Hanói. A proposição do problema é a seguinte: inicialmente têm-se três hastes (A, B e C) e, na haste A, repousam três anéis de diâmetros diferentes, em ordem decrescente por diâmetro (Figura abaixo).



Torre de Hanói com 3 anéis

Fonte:

O objetivo é transferir os três anéis da haste A para C, usando B se necessário. As regras de movimento são as seguintes:

- deve-se mover um único anel por vez;
- um anel de diâmetro maior nunca pode repousar sobre algum outro de diâmetro menor.

Na animação a seguir pode-se observar a sequência de movimentos para mover os 3 anéis da haste A para a C.



00:31



Baseado nos movimentos dos anéis da haste A para a C, a seguinte sequência de passos pode ser considerada:

1. Mover um anel da haste A para a haste C.
2. Mover um anel da haste A para a haste B.
3. Mover um anel da haste C para a haste B.
4. Mover um anel da haste A para a haste C.
5. Mover um anel da haste B para a haste A.
6. Mover um anel da haste B para a haste C.
7. Mover um anel da haste A para a haste C.

O exemplo do problema da travessia dos missionários e canibais também é outro exemplo clássico que pode ser resolvido por meio de um algoritmo. Três missionários e três canibais precisam atravessar um rio. Para tal, dispõem de um barco com capacidade para duas pessoas. Por medida de segurança, não se deve permitir que em alguma margem a quantidade de missionários seja inferior à de canibais. Nesse caso, qual seria a solução para se efetuar a travessia com segurança? Na animação a seguir é possível observar uma possível solução para este problema e a sequência lógica de passos é:

1. Atravessar um missionário e um canibal para a margem B.
2. Voltar o missionário para a margem A.
3. Atravessar dois canibais para a margem B.
4. Voltar um canibal para a margem A.
5. Atravessar dois missionários para a margem B.
6. Voltar um missionário e um canibal para a margem A.



7. Atravessar dois missionários para a margem B.
8. Voltar um canibal para a margem A.
9. Atravessar dois canibais para a margem B.
10. Voltar um canibal para a margem A.
11. Atravessar dois canibais para a margem B.

01:23



Os exemplos de algoritmos apresentados até aqui não têm solução computacional, ou seja, não é possível transformá-los em programas de computador. Por outro lado, existem problemas que podem ser resolvidos por meio de algoritmos os quais têm solução computacional, pois, a partir da sequência de passos, é possível codificá-los em alguma linguagem de programação.

Um exemplo de problema que pode ser resolvido por meio de um algoritmo que tenha solução computacional é a soma de dois números inteiros. Inicialmente, os dois números inteiros devem ser informados para que, em seguida, seja realizada a operação de adição entre eles. Um algoritmo que resolve este problema pode ser descrito como:

1. Informe o primeiro número inteiro.
2. Informe o segundo número inteiro.
3. Some os dois números inteiros.
4. Mostre o resultado da soma.

A partir deste ponto, serão considerados somente problemas que tenham solução computacional.



Fases de um algoritmo

Basicamente, para se construir um algoritmo é necessário dividir o problema apresentado em três fases fundamentais: entrada, processamento e saída de dados. Tais fases são descritas a seguir:

- **Entrada de dados:** todo algoritmo deve possuir uma ou mais entradas de dados. No exemplo da soma de dois números inteiros, a fase de entrada é representada pelos passos 1 e 2.
- **Processamento de dados:** procedimentos utilizados para se chegar ao resultado final a partir da manipulação dos dados de entrada. No exemplo da soma de dois números inteiros, a fase de processamento de dados é representada pelo passo 3.
- **Saída de dados:** todo algoritmo possui uma ou mais saídas, que simbolizam seus resultados. No exemplo da soma de dois números inteiros, a fase de saída de dados é representada pelo passo 4.

Na Figura a seguir, é possível identificar, no algoritmo da soma de dois números inteiros, as fases de um algoritmo.



Fases de um algoritmo

Quiz

Exercício Final



[INICIAR >](#)

Referências

FORBELLONE, A. L. V; EBERSPACHER, H. F. Lógica de Programação: a construção de algoritmos e estruturas de dados (3a. edição). São Paulo: Prentice Hall, 2005.

RISSETTI, G.; PUGA, S. Lógica de Programação e estruturas de dados, com aplicações em Java. São Paulo: Prentice Hall, 2003.

SOUZA, M. A. F. e outros. Algoritmos e lógica de programação: um texto introdutório para Engenharia (2a. edição). São Paulo: Cengage Learning, 2011.



Avalie este tópico



 Índice

Biblioteca
(<https://www.uninove.br/conhec-a-uninove/biblioteca/sobre-a-biblioteca/apresentacao/>)
Portal Uninove
(<http://www.uninove.br>)
Mapa do Site

[Ajuda?](#)
PRÓXIMO
(<https://ava.uninove.br/cursos/>)

Conceitos iniciais: dados, tipos de dados, variáveis e constantes

© Todos os direitos reservados

