

Aulas 02 - Lógica de Programação

O que é lógica

Na área de Informática é fundamental ter um raciocínio lógico bem desenvolvido. É também muito importante aprender as técnicas mais conhecidas, para resolver problemas através do computador.

O indivíduo que conhece lógica é capaz de aprender qualquer linguagem de computação, até mesmo sozinho.

Mas o que é LÓGICA?

LÓGICA DE PROGRAMAÇÃO é a técnica de encadear os pensamentos para se atingir um objetivo. Estes pensamentos podem ser descritos como uma sequência de instruções que deve ser seguida para se cumprir uma determinada tarefa. Chamamos essas sequências de instruções de SEQUÊNCIAS LÓGICAS.

Até mesmo as coisas mais simples, podem ser descritas por SEQUÊNCIAS LÓGICAS. Por exemplo:

“COMER UMA MAÇÃ”.

Para descrever a tarefa “COMER UMA MAÇÃ”, podemos montar uma sequência de ações:

- 1 - Pegar a maçã
- 2 - Lavar a maçã
- 3 - Comer a maçã
- 4 - Jogar as sementes da maçã no lixo

Qualquer pessoa que seguir estes procedimentos vai saber realizar a tarefa “COMER UMA MAÇÃ” porque existe uma SEQUÊNCIA DE INSTRUÇÕES QUE PODE SER CUMPRIDA.

Algoritmos

Nas etapas do desenvolvimento de um programa, primeiro começamos por analisar o problema existente e em seguida procurar escrever no papel uma possível solução para ele. Chamamos de Algoritmo esta solução do problema, ainda escrita no papel e representada de qualquer forma escrita ou através de desenhos, diagramas, etc.

Na realidade, um algoritmo é uma sequência lógica de instruções que são descritas através de frases simples.

Usamos os Algoritmos (Sequências lógicas) em muitas coisas no nosso dia-a-dia:

- Quando explicamos a uma pessoa como chegar a um determinado endereço.
- Quando seguimos uma receita de um livro de culinária.
- Quando montamos nossa agenda diária.

Por exemplo, desejando telefonar para alguém, o algoritmo para a solução deste problema poderá ser:

- 1- Pegar a agenda de telefones.
- 2- Procurar o nome do seu amigo ou amiga.
- 3- Encontrar o número do telefone.
- 4- Pegar o aparelho de telefone.
- 5- Discar o número.
- 6- Falar com seu amigo.
- 7- Desligar o telefone.

A ordem das etapas é importante para se atingir o resultado desejado. O algoritmo também deve ter um fim.

Apesar de parecer perfeito, poderão surgir alguns imprevistos durante a execução deste algoritmo: o telefone do amigo não constar na sua agenda, ou ainda, não encontrar a agenda, a irmã do seu amigo pode estar usando o telefone no exato momento...

Quando elaboramos um algoritmo, ele deve ser o mais detalhado possível para prever todas as situações. Mas você só conseguirá atingir o algoritmo ideal após exaustivos testes.

Como desenvolver algoritmos

Qualquer tarefa que siga determinados padrões pode ser descrita por algoritmos. Assim sendo, da mesma forma como descrevemos a tarefa “comer uma maçã”, podemos descrever também outras tarefas, como por exemplo:

“FAZER O CÁLCULO DO SALÁRIO DE UM FUNCIONÁRIO”;

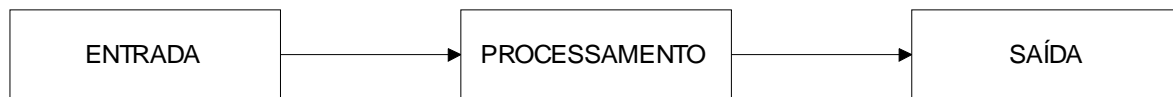
“CALCULAR A MÉDIA FINAL DOS ALUNOS DE LÓGICA”;

“CALCULAR A IDADE DE UMA PESSOA”,

Todas elas são tarefas que podem ser executadas em um computador.

O computador só é capaz de executar uma ordem por vez e não é capaz de interpretar. Assim sendo, quando elaboramos um algoritmo que vai resolver um problema em um computador, precisamos de algumas técnicas especiais.

A primeira preocupação em montar um algoritmo é dividir o problema apresentado nas três fases fundamentais de qualquer resolução lógica.



Digamos, por exemplo, que temos o seguinte problema:

Calcular a média final dos alunos do módulo de LÓGICA, sendo que realizarão três provas parciais (P1, P2, P3) e uma prova final (PF), onde:

$$\text{Média parcial} = \frac{P1 + P2 + P3}{3} \quad \text{e} \quad \text{Média Final} = \frac{\text{Média Parcial} + 2 * PF}{3}$$

Para podermos montar o algoritmo deste problema, faremos três perguntas:

a) Quais são os dados de entrada?

(os dados que necessitamos para realizar o processamento).

Os dados de entrada são: P1, P2, P3 e PF.

b) Qual é o processamento?

(o tratamento que daremos aos dados de entrada, para obtermos os resultados desejados).

O processamento é o cálculo das fórmulas acima.

c) Quais são os dados de saída?

(Os resultados pedidos).

O dado de saída é a Média Final.

Temos Portanto:

ENTRADA	PROCESSAMENTO	SAÍDA
P1, P2 P3, PF	Média Parcial = $\frac{P1 + P2 + P3}{3}$ Média Final = $\frac{\text{Média Parcial} + 2 * PF}{3}$	MÉDIA FINAL

A segunda preocupação é descrever a sequência de instruções de uma maneira simples e objetiva. Para isso, podemos utilizar algumas técnicas:

a) Usar só um verbo por frase.

b) Imaginar que estamos desenvolvendo o processo para uma pessoa que esteja em uma cidade distante, e que não possa nos pedir esclarecimentos.

- c) usar frases curtas e simples.
- d) procurar usar palavras que não tenham sentido duplo.

Exemplo de um algoritmo

Imagine que vamos desenvolver o ALGORITMO para um funcionário de uma empresa, e para facilitar o trabalho deste funcionário, criamos o seguinte formulário:

P1	P2	P3	PF	MP	MF

Algoritmo:

Pergunte: a nota da prova 1.

Escreva: a nota da prova 1 na coluna correspondente (P1).

Pergunte: a nota da prova 2.

Escreva: a nota da prova 2 na coluna correspondente (P2).

Pergunte: a nota da prova 3.

Escreva: a nota da prova 3 na coluna correspondente (P3).

Pergunte: a nota da prova final.

Escreva: a nota da prova final na coluna correspondente (PF).

Some: a nota da prova 1 (P1) com a nota da prova 2 (P2) com a nota da prova 3 (P3).

Escreva: o resultado na coluna MP.

Divida: o conteúdo da coluna MP por 3.

Escreva: o novo resultado na coluna MP, substituindo o anterior.

Multiplique: a prova final (PF) por 2.

Escreva: o resultado na coluna MF.

Some : o conteúdo da coluna MP com o conteúdo da coluna MF.

Escreva: o resultado da coluna MF, substituindo o anterior.

Divida: o conteúdo da coluna MF por 3.

Escreva: o resultado na coluna MF, substituindo o anterior.

Teste de mesa

Depois de desenvolvido, um algoritmo deve ser **sempre** testado. Chamamos este teste de TESTE DE MESA. Realizar um teste de mesa significa seguir as instruções do algoritmo de maneira precisa.

Teste você mesmo o algoritmo anterior, tendo:

Nota da prova 1	- 7.0
Nota da prova 2	- 8.0
Nota da prova 3	- 9.5
Nota da prova final	- 8.5

Utilize para o teste, o formulário a seguir:

P1	P2	P3	PF	MP	MF

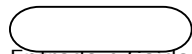
Teoria de fluxos (Fluxogramas)

A segunda forma que utilizaremos aqui para mostrar a construção de algoritmos é a teoria de fluxos, onde, através de um diagrama, chamado “**fluxograma**” mostramos quais são os procedimentos necessários para se chegar à solução do problema. Consegue-se com este método uma visualização global mais rápida que nos outros métodos, além de se escrever menos.

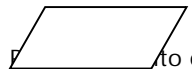
Trata-se de uma sequência de símbolos, com significados próprios, através dos quais se elabora um algoritmo gráfico. Com isso, qualquer pessoa que já conheça o significado dos símbolos e olhe para o “desenho”, tem uma visualização imediata dos procedimentos necessários para o cumprimento de uma tarefa.

O fluxograma utiliza símbolos padronizados para representar as ações mais comuns, como por exemplo:

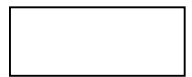
Símbolos Terminais (início / fim)



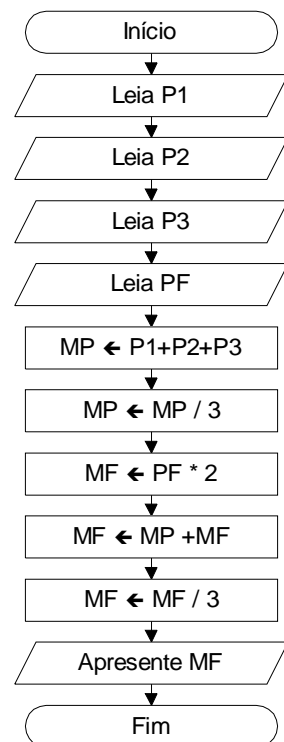
Entrada e Saída de dados



Processamento de dados



Para exemplificar, vamos ver como ficaria o fluxograma para o nosso algoritmo:

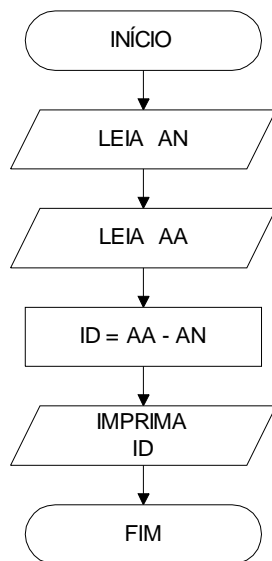


Outro exemplo seria o problema clássico de se calcular a idade de uma pessoa, tendo conhecidas as datas do seu nascimento e a data atual.

Podemos propor o seguinte algoritmo:

- 1- Pergunte qual o ano de nascimento e coloque em NA.
- 2- Pergunte qual o ano atual e coloque a resposta em AA.
- 3- Calcule $AA - AN$ (ano atual - ano de nascimento) e coloque o resultado em ID.
- 4- Mostre o conteúdo de ID.
- 5- Fim

Este problema poderia ser descrito através de um fluxograma como este:



Codificação

Cada etapa de um algoritmo é uma tarefa a ser executada ou uma INSTRUÇÃO.

Ao conjunto de instruções que devem ser executadas para a solução de um problema damos o nome de PROGRAMA. Desta forma, um programa no computador é analogamente, uma sequência de instruções que o ensina a resolver um problema. Agora, resta saber como fazer para que o computador entenda o programa, ou seja, como transformar o algoritmo em uma sequência de instruções entendidas pelo computador.

Teremos antes que reescrevê-la, usando uma linguagem de programação que mais se adapte ao tipo de problema. Chamamos este processo de codificação.

Verifique nosso exemplo que calcula a idade de uma pessoa, já codificado em Linguagem C:

```
#include<stdio.h>
#include<stdlib.h>

int main ()
{
    float ano_nasc,ano_atual,idade;

    printf("Que ano voce nasceu?\n");
    scanf("%f",&ano_nasc);

    printf("Em que ano estamos?\n");
    scanf("%f",&ano_atual);

    idade = ano_atual - ano_nasc;

    printf("\nVoce tem %.1f anos de vida!! Viva!!\n",idade);
    system("pause");
    return 0;
}
```

Regras práticas para a elaboração de programas

Como nosso objetivo é apresentar a construção de algoritmos para iniciantes em lógica, tomamos o cuidado de montar uma metodologia que indicará as regras práticas para a elaboração de programas e que poderá ser seguida como sugestão até se acostumarem com este processo.

Programadores já experientes não mais precisam seguir metodologias como esta, pois já conseguiram “absorver” este mecanismo e já o desenvolvem inconscientemente.

A metodologia proposta está dividida em nove (9) passos:

PASSO 1:

- ◆ Leia cuidadosamente a especificação do problema até o final, quantas vezes forem necessárias até entender bem o que o problema está exigindo.

Se após cuidadosas leituras ainda não entender, procure o responsável pela elaboração do problema e peça explicações mais detalhadas. Sem entender as especificações pedidas, você não será capaz de resolvê-lo corretamente.

PASSO 2:

- ◆ Levantar e analisar todos os dados iniciais exigidos na especificação do problema.

Tentar se possível, definir as variáveis necessárias, responsáveis por receber estes valores de entrada. Às vezes, as entradas podem ser também valores constantes. Procure identificá-los.

PASSO 3:

- ◆ Levantar e analisar todos os resultados necessários e exigidos na especificação do problema.

Aqui se deve perguntar o que se deseja encontrar ao final do algoritmo ou quais as respostas que ele deve encontrar.

PASSO 4:

- ◆ Levantar e analisar todas as transformações necessárias para, dadas as entradas e valores gerados internamente, produzir as saídas especificadas.

PASSO 5:

- ◆ Procurar escrever a solução lógica para esse problema, utilizando uma das técnicas apresentadas, como Fluxograma, ou outra qualquer que preferir.

PASSO 6:

- ◆ Elaborar um mapa de execução do algoritmo.

O mapa consiste em uma tabela contendo todas as variáveis utilizadas no desenvolvimento da solução lógica. A partir de uma massa de dados, vai-se percorrendo o diagrama da solução proposta e validando os valores que as variáveis vão assumindo em função dos passos executados.

Deve-se escolher uma massa de dados que permita primeiramente, prever o resultado final do algoritmo, ou seja, os resultados finais.

A elaboração do mapa de execução é chamada metaforicamente de “teste de mesa”, pois se deve testar a solução lógica independentemente da máquina, na mesa de trabalho.

Persistir nas etapas 4, 5 e 6 até se corrigirem todos os possíveis erros encontrados. Só passe para a etapa seguinte com o algoritmo completamente certificado.

PASSO 7:

- ◆ Proceder a codificação na linguagem escolhida. Desta vez, codificar usando a linguagem C.

PASSO 8:

- ♦ Digitar o programa na máquina e produzir uma sequência de testes via computador para, mais uma vez, validar o programa, corrigindo os erros de sintaxe (erros de codificação ou de digitação) que possa haver.

Inicialmente deve-se testar o programa com a mesma massa de dados utilizada no teste de mesa e comprovar que sejam encontrados os mesmos resultados. Depois diversificar os testes com outra massa de dados diferente.

Se todos os resultados forem satisfatórios, já temos a primeira versão para o programa. Caso sejam insatisfatórios, deve-se voltar ao passo 1 e reavaliar todo o procedimento novamente.

PASSO 9:

- ♦ Gravar e listar o programa. Anexar a essa listagem a solução lógica, os testes de mesa e outros rascunhos e comentários que ajudem a completar a documentação do programa. A documentação auxiliará no futuro quando for necessário efetuar alterações neste programa.

Os nove passos mostrados identificam de uma forma mais detalhada as fases do processo de programação, identificação do problema, desenvolvimento da solução lógica, codificação, testes e documentação.

Utilizando esta metodologia, mais adiante iremos estudar problemas cujas soluções serão apresentadas em nível de exercícios resolvidos, ao final de cada capítulo.

Atividades

1. Descrever as tarefas a seguir de tal maneira que qualquer pessoa que leia as suas instruções, saiba exatamente como agir.

- a) Tomar banho.
- b) Fazer um bolo.
- c) Trocar uma lâmpada.

2. Faça um algoritmo e um fluxograma que calcule a média aritmética de um aluno que realizou três provas.

3. Faça um algoritmo e um fluxograma para calcular a média final de um aluno, sendo que temos uma prova parcial com peso 2 e uma prova final com peso 3.

Faça testes de mesa, propondo dados para o seu problema.

4. Elabore um algoritmo e um fluxograma que some, subtraia e multiplique dois números lidos.

5. Elabore um algoritmo e um fluxograma que calcule o dobro de um número.