

01 LÓGICA DE PROGRAMAÇÃO

LÓGICA, ALGORITMOS E PROGRAMAS

OBJETIVOS: Compreender o conceito de lógica e aplicá-la à Programação de Computadores, bem como identificar e aplicar o conceito de Algoritmos.

Considerada uma das mais antigas disciplinas, a Lógica foi criada por Aristóteles (384 a 322 a.C.) e sua essência era a teoria do silogismo, que é a conclusão por inferência a partir de proposições.

Originária do grego *logos*, lógica significa “linguagem racional”. De acordo com o Dicionário Michaelis da Língua Portuguesa (2021, texto digital), lógica é: “Parte da filosofia que se ocupa das formas do pensamento e das operações intelectuais”; “Sequência coerente de ideias”; “Maneira rígida de raciocinar”; ou ainda, “Modelo pelo qual se encadeiam naturalmente as coisas ou os acontecimentos”. E é com base nesta última definição que seguiremos os estudos de todo este componente.

Nesta primeira unidade de estudos, estudaremos sobre lógica, algoritmos e programas.

Lógica dedutiva e indutiva

O estudo da lógica pode ser dividido de acordo com dois métodos: **dedutivo** e **indutivo**. Ambos apresentam argumentos, premissas, ou enunciados, e uma conclusão. A diferença é que, na lógica indutiva, os argumentos não bastam para assegurar a verdade da conclusão, enquanto na dedutiva as premissas verdadeiras sempre resultarão em uma conclusão verdadeira. Veja os exemplos a seguir:

02

- **Lógica dedutiva** (utilizaremos **p** e **q** representando as premissas, e **r** a conclusão):

p: Juca é mais velho do que Álvaro.

q: Álvaro é mais velho do que Simone.

r: Logo, Juca é mais velho do que Simone.

- **Lógica Indutiva** (utilizaremos **p** e **q** representando as premissas, e **r** a conclusão):

p: Ontem o céu estava nublado e choveu.

q: Hoje o céu está nublado.

r: Logo, hoje vai chover.

Perceba que a lógica indutiva parte do **específico para o geral**, e a conclusão é obtida pela comparação da premissa **q**, “Hoje o céu está nublado” (específico), com o fato anterior de que “Ontem o céu estava nublado e choveu” (premissa **p**, geral),

induzindo a um resultado universal e que pode gerar conhecimentos novos, como o conhecimento de que se está nublado é porque vai chover.

Já na lógica dedutiva, que nos interessa neste componente, o raciocínio parte do **geral para o específico**. Analisam-se as premissas em forma de sentenças sem que haja a necessidade de conhecimento prévio de fatos ou situações. A conclusão é obtida em decorrência das premissas. Por exemplo, se Álvaro fosse mais novo do que Simone, não poderíamos afirmar que Juca é mais velho do que Simone, mas poderíamos afirmar que Simone é a pessoa mais jovem dentre os personagens.



Antes de prosseguir com a leitura do material, acesse o Ambiente Virtual e interaja com o objeto de aprendizagem sobre lógica indutiva e dedutiva.

0 uso da lógica aplicada à informática

Na informática, a lógica é utilizada em diversas áreas. Desde a construção de circuitos elétricos que estão presentes em hardwares até os softwares que fazem aqueles funcionarem e permitem seu uso.

Para se construir um circuito de um teclado de computador, por exemplo, utiliza-se o conceito de portas lógicas (físicas), que verifica a passagem ou não de pulsos elétricos entre os componentes, a fim de que estes se comuniquem e, assim, se obtenha uma saída específica ao final do circuito.

No desenvolvimento de software, o programador se utiliza do raciocínio lógico para criar algoritmos que serão transformados em programas de computador a fim de solucionar problemas, dos mais simples aos mais complexos.

Voltando aos personagens citados anteriormente, Juca, Álvaro e Simone, vamos atribuir valores numéricos aos seus nomes, que representarão as suas idades. Em seguida, colocaremos esses valores nas premissas e por fim faremos comparações.

Atribuindo os valores:

Juca \leftarrow 30;

Álvaro \leftarrow 20;

Simone \leftarrow 19;

Comparando as premissas:

p: Juca é mais velho do que Álvaro? $(Juca > Álvaro) = \text{Sim}$.

q: Álvaro é mais velho do que Simone? ($\text{Álvaro} > \text{Simone}$) = Sim.

r: Juca é mais velho do que Simone? ($\text{Juca} > \text{Simone}$) = Sim.

Computacionalmente, as expressões lógicas contidas em **p**, **q** e **r** podem ser escritas da seguinte maneira:

p: Se $\text{Juca} > \text{Álvaro}$ então Verdadeiro;

q: Se $\text{Álvaro} > \text{Simone}$ então Verdadeiro;

r: Se $\text{Juca} > \text{Simone}$ então Verdadeiro.

Note que não é necessário testar a última expressão para saber que Juca é mais velho do que Simone porque já sabemos que ele é mais velho que Álvaro e Álvaro é mais velho que Simone.

Este pensamento lógico, se transformado em pensamento computacional ou em um algoritmo, ficaria assim:

Se ($\text{Juca} > \text{Álvaro}$) e ($\text{Álvaro} > \text{Simone}$) então “Juca é mais velho do que Simone”.

04

Algoritmos e Pensamento Sistêmico

De acordo com Piva Jr (2019), os algoritmos surgiram no século IX e se referiam às regras para a realização de aritmética. Com o passar do tempo, essa expressão passou a ser utilizada para definir todos os procedimentos para resolver problemas ou realizar tarefas. “De forma bastante simples, um algoritmo pode ser definido como ‘um conjunto de instruções para resolver um problema’” (p. 3).

De acordo com o Dicionário Michaelis da Língua Portuguesa (2021, texto digital), a definição de algoritmo é:

- Aritmética: Sistema de notação aritmética com algarismos arábicos.
- Matemática: Processo de cálculo que, por meio de uma sequência finita de regras, raciocínios e operações, aplicada a um número finito de dados, leva à resolução de grupos análogos de problemas.
 - Matemática: Operação ou processo de cálculo; sequência de etapas articuladas que produz a solução de um problema; procedimento sequenciado que leva ao cumprimento de uma tarefa.
- Lógica: Conjunto das regras de operação (conjunto de raciocínios) cuja aplicação permite resolver um problema enunciado por meio de um número finito de operações; pode ser traduzido em um programa executado por um computador, detectável nos mecanismos gramaticais de uma língua ou no sistema de procedimentos racionais finito, utilizado em outras ciências, para resolução de problemas semelhantes.
- Informática: Conjunto de regras e operações e procedimentos, definidos e ordenados usados na solução de um problema, ou de classe de problemas, em um número finito de etapas.

Existem muitas outras definições, mais simples ou mais complexas. O mais importante é entender que:

“Um algoritmo é uma sequência lógica de passos, **com começo, meio e fim**. Cada passo desse algoritmo deve ser expresso de forma clara, utilizando, muitas vezes, um formalismo específico, justamente para que não deixe qualquer dúvida, ou seja, não haja ambiguidade na sua interpretação. Essa sequência de passos tem um objetivo específico, que geralmente é a resolução de um problema. Para tanto, esse algoritmo pode receber dados de entrada, muitas vezes chamados de variáveis, e como visa a resolução de um problema, essa resposta do algoritmo visando a solução do problema gera dados de saída.” (PIVA JR, 2019, p. 6, grifo nosso).

**DICA DE VÍDEO**

Assista ao filme *O Jogo da Imitação* (2014), disponível em serviços de streaming diversos. Sinopse: Em 1939, a recém-criada agência de inteligência britânica MI6 recruta Alan Turing, um aluno da Universidade de Cambridge, para entender códigos nazistas, incluindo o “Enigma”, que criptógrafos acreditavam ser inquebrável. A equipe de Turing, incluindo Joan Clarke, analisa as mensagens de “Enigma”, enquanto ele constrói uma máquina para decifrá-las (TYLDUM, 2015).

**PARA SABER MAIS**

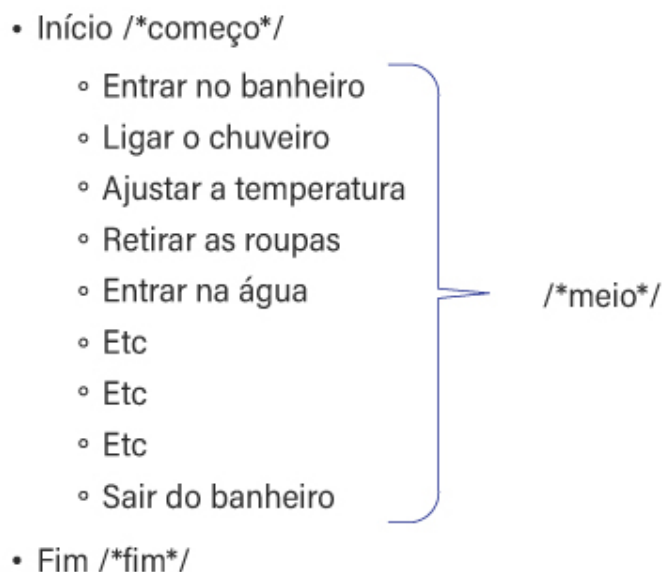
Alan Turing é considerado o pai da ciência computacional e da inteligência artificial. Criou a máquina de Turing, que teve por objetivo materializar a **lógica** humana e solucionar qualquer cálculo representado em formato de **algoritmo** (FRAZÃO, 2021).

Algoritmos do dia a dia

Diariamente, nos deparamos com situações cotidianas em que, inconscientemente, executamos algoritmos para resolvê-las, como tomar banho, por exemplo. Pode parecer algo automático, mas se pensarmos nessa situação ela pode ser dividida em passos simples:

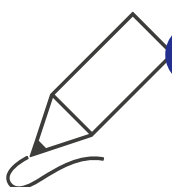
- Início /*começo*/
 - Entrar no chuveiro
 - Ensaboar-se
 - Retirar o sabão
 - Sair do chuveiro
 - Fim /*fim*/
- /*meio*/

Ou dividida em passos mais complexos:



E assim, existem várias situações corriqueiras que exigem um algoritmo, mesmo que mental, para a sua execução correta (arrumar a mesa para o jantar, escovar os dentes, ouvir uma música, preparar um bolo, enviar um e-mail).

07

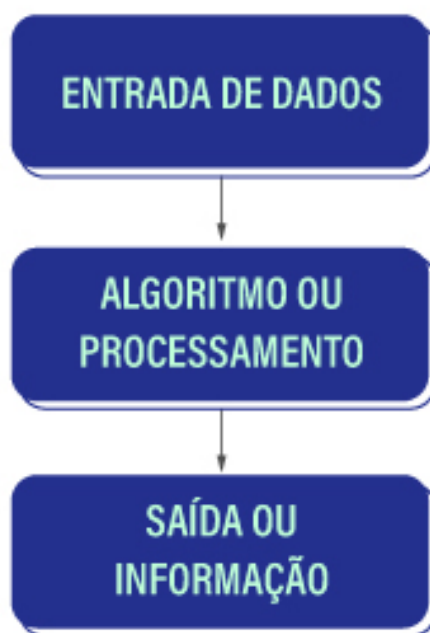


EXERCÍCIO

Agora, acesse o Ambiente Virtual e realize o exercício proposto a fim de verificar os conhecimentos adquiridos até aqui.

Algoritmos computacionais

Um programa de computador pode ser dividido, basicamente, em três partes: Entrada de dados, Algoritmo ou Processamento e Saída ou Informação.



Fonte: Univates (2021).

Como vimos anteriormente, a definição sintetizada de algoritmo é: sequência lógica de passos, com início, meio e fim. Utilizando os conceitos de lógica e descartando os identificadores das premissas (**p**, **q**, **r**), porém mantendo a estrutura onde cada proposição deve ocupar uma linha, vamos construir um algoritmo para somar dois números inteiros.

08

Observando a imagem acima, devemos ter em mente que o primeiro passo é receber os dados de entrada. No nosso exemplo, teremos duas entradas (dois números inteiros) e estas serão armazenadas em variáveis que, aqui, vamos definir como x e y . Após a atribuição dos valores às variáveis, faremos a operação propriamente dita ($x + y$) e armazenaremos o resultado em uma terceira variável s . Por fim, apresentaremos o resultado contido em s .

$$X \leftarrow 5$$
$$Y \leftarrow 3$$
$$S \leftarrow X + Y$$
$$S = 8$$

Observe que o problema foi resolvido parcialmente, pois o algoritmo faz a soma de dois números inteiros, porém, ao atribuirmos valores fixos (5 e 3) para as variáveis X e Y , elas deixam de ser variáveis e passam a ser chamadas de constantes. Sendo assim, toda vez que o algoritmo que soma dois números inteiros for executado, o resultado sempre será 7 (sete).

Para melhorar este algoritmo, vamos utilizar dois comandos (instruções) que permitem que um valor possa ser inserido (comando LEIA) em cada variável e possa ser mostrado (comando ESCREVA).

A sintaxe para o uso desses comandos é: *LEIA(variável)* e *ESCREVA(variável)*, respectivamente. Nosso algoritmo, agora otimizado para receber como entrada qualquer número inteiro, ficará assim:

```
LEIA(X)
LEIA(Y)
S ← X+Y
ESCREVA(S)
```

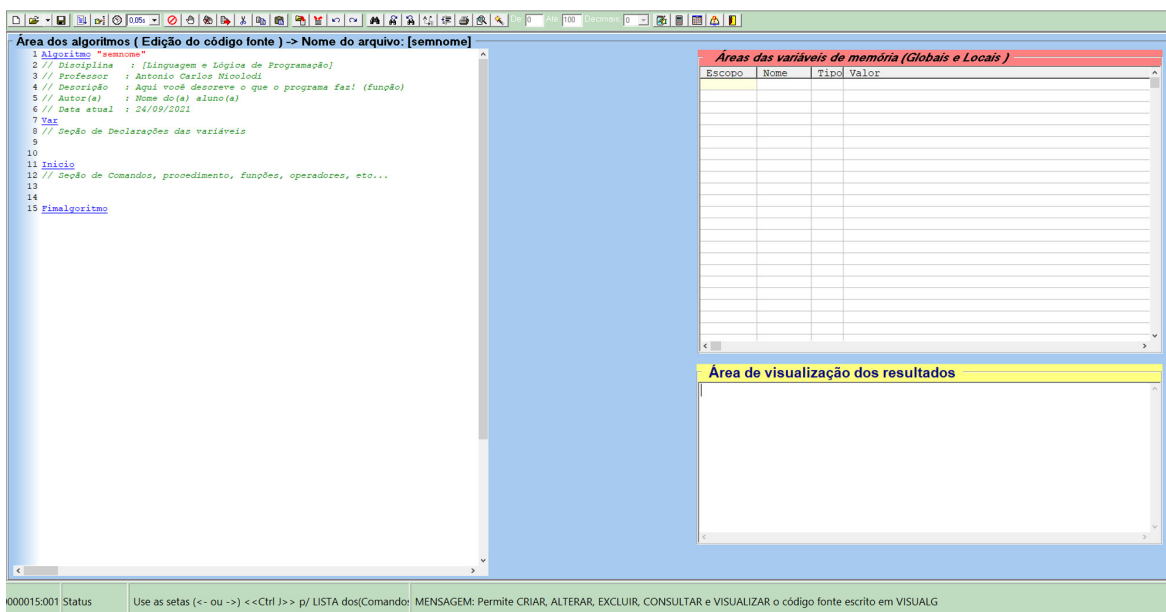
Vamos programar!

Para iniciarmos nossa jornada pelo mundo da programação utilizaremos agora o software Visualg. Ele permite criar e testar algoritmos utilizando a linguagem do Português Estruturado, também conhecida como *Portugol*. Não se preocupe com a linguagem ou comandos, eles serão vistos de forma mais aprofundada no decorrer desta unidade. Agora, faça o download do Visualg no endereço: [Baixe o VisuAlg 3.0.7 – VisuAlg 3.0](#). Após o download, basta descompactar o arquivo em uma pasta qualquer em seu computador e executar o arquivo *visualg30.exe*.

09

Utilizando o Visualg

Ao abrir o Software Visualg, você verá a tela abaixo. Perceba que na área dos algoritmos, que fica no lado esquerdo da tela, já existe um “esqueleto” de pseudocódigo que tem como finalidade mostrar as informações iniciais que qualquer algoritmo requer para ser executado.



Fonte: Univates (2021).

- *Algoritmo* “*nome do algoritmo*” – Este comando define o nome do algoritmo/ programa que deve ser informado entre aspas duplas (“”).
- *Var* - Neste bloco definimos as variáveis que serão utilizadas no algoritmo.
- *Início* – Aqui inicia o algoritmo propriamente dito. É abaixo desta linha que serão inseridos os comandos (passos) que o algoritmo deve executar.
- *Fimalgoritmo* – Esta palavra determina o final do algoritmo, ou seja, nada que estiver abaixo dela será executado ou interpretado pelo Visualg.
- *//* - As barras duplas servem para comentar o código, inserir informações para que a pessoa que ler o algoritmo entenda o que está acontecendo em cada passo. Estes comentários não são interpretados e não aparecem na tela de execução do algoritmo.

10

Vamos construir agora o nosso primeiro algoritmo utilizando o Visualg, e para isso vamos utilizar o nosso exemplo já explicado anteriormente para construirmos um programa que leia e faça a soma de dois números inteiros. O código abaixo mostra como deverá ficar o algoritmo. Copie este código e cole no Visualg. Após, pressione *F9* para a execução e observe o resultado apresentado.

Algoritmo “soma”

```
// Componente : [Lógica de Programação - Univates EAD]
// Descrição : Este programa lê e soma dois números inteiros
// Autor(a) : Univates EAD
// Data atual : 2021
```

Var

```
// Seção de Declarações das variáveis
X : inteiro //variável do tipo inteiro que irá receber o
primeiro número
Y : inteiro //variável do tipo inteiro que irá receber o segundo
número
S : inteiro //variável do tipo inteiro irá receber a soma de X e
Y
```

Início

```
// Seção de Comandos, procedimento, funções, operadores, etc...
```

```
// comando para limpar a tela
```

```
Limpatela
```

```
// comando para escrever uma string (texto) na tela e quebrar a
linha
```

```
EscrevaL (" Soma de dois números inteiros ")
```

```
EscrevaL ("*****")
```

```
EscrevaL ("") // uma linha em branco
```

```
// comando para escrever uma string na tela SEM quebrar a linha
```

```
Escreva ("Digite o primeiro número: ")
```

```
// comando para ler o que será digitado e armazenar em uma
variável
```

```
Leia(X)
```

```
Escreva ("Digite o segundo número: ")
```

```
Leia(Y)
```

```
// comando para atribuir a soma das variáveis X e Y à variável S
```

```
S := X + Y
```

```
Escreva ("O resultado da soma é: ")
```

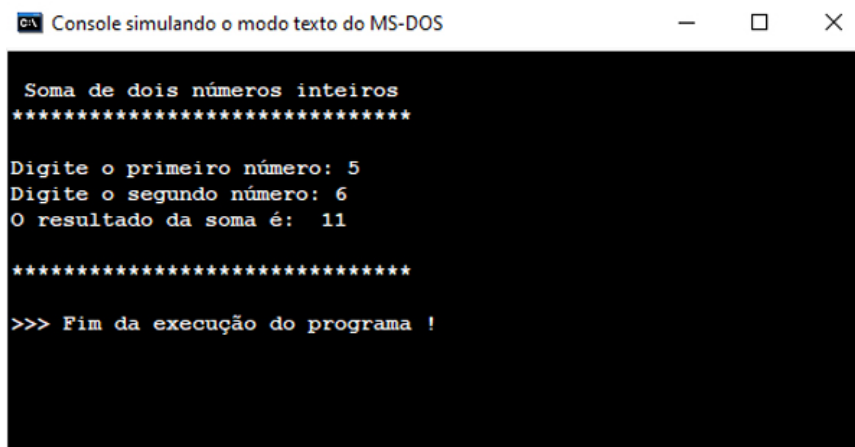
```
EscrevaL (S) // exibindo a variável S e quebrando a linha
```

```
EscrevaL ("")
```

```
EscrevaL ("*****")
```

```
Fimalgoritmo
```

Como resultado, teremos uma saída próxima da tela abaixo:



```
Console simulando o modo texto do MS-DOS

Soma de dois números inteiros
*****

Digite o primeiro número: 5
Digite o segundo número: 6
O resultado da soma é: 11

*****

>>> Fim da execução do programa !
```

Fonte: Univates (2021).

Como se pode ver, o programa solicita que o usuário informe dois números inteiros e apresenta o resultado.

Tipos primitivos de variáveis e operadores

12

O Visualg, assim como todas as linguagens de programação, permite o uso de diversos tipos de variáveis e operadores com a finalidade de armazenar e efetuar diversos cálculos e operações de acordo com a necessidade do programa que está sendo desenvolvido.

Tipos de variáveis

Vamos conhecer agora os tipos de variáveis.

- **Inteiro**

Define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais.

- **Real**

Define variáveis numéricas do tipo real, ou seja, com casas decimais.

- **Caractere**

Define variáveis do tipo *string*, ou seja, cadeia de caracteres.

- **Lógico**

Define variáveis do tipo *booleano*, ou seja, com valor VERDADEIRO ou FALSO.

Operadores aritméticos e de caracteres

Vejamos agora alguns operadores. Clique para ler sobre cada um.

- +, -, *, /

Operadores aritméticos tradicionais de adição, subtração, multiplicação e divisão. Por convenção, * e / têm precedência sobre + e -. Para modificar a ordem de avaliação das operações, é necessário usar parênteses como em qualquer expressão aritmética.

- MOD ou %

Operador de módulo (isto é, resto da divisão inteira). Por exemplo, $8 \text{ MOD } 3 = 2$. Tem a mesma precedência do operador de divisão tradicional.

- ^

Operador de potenciação. Por exemplo, $5^2 = 25$. Tem a maior precedência entre os operadores aritméticos binários (aqueles que têm dois operandos).

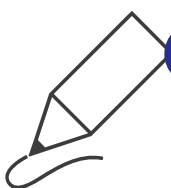
- + (quando utilizado em variáveis do tipo caractere/String)

Operador de concatenação de strings (isto é, cadeias de caracteres), quando usado com dois valores (variáveis ou textos) do tipo “caractere”. Por exemplo: “Rio + Grande do Sul” = “Rio Grande do Sul”.

A lista completa de variáveis, operadores e comandos do Visualg pode ser encontrada no *Help* do software e, ainda, no arquivo *Relação dos comandos do visualg 3.0.txt* que se encontra dentro da pasta do Visualg. Iremos ver mais alguns destes comandos e operadores no decorrer deste componente.

Considerações finais

Nesta Unidade, vimos os conceitos básicos da lógica de programação, a definição e exemplificação de algoritmos e a criação de pequenos programas utilizando pseudocódigo no programa Visualg. Na próxima unidade, seguiremos com a criação de programas mais elaborados, veremos estruturas de controle e de repetição e utilizaremos uma ferramenta do tipo No Code para a criação de programas.



ATIVIDADE

Após realizar a leitura do material e o exercício desta unidade de estudos, chegou o momento de realizarmos uma atividade. Para isso, acesse o Ambiente Virtual.

Referências

FRAZÃO, Dilva. Alan Turing. **E-biografia**, 2021. Disponível em: https://www.ebiografia.com/alan_turing/. Acesso em: 01 dez. 2021.

LÓGICA. *In*: MICHAELIS moderno dicionário da língua portuguesa. São Paulo: Melhoramentos. Disponível em: <https://michaelis.uol.com.br/moderno-portugues/busca/portugues-brasileiro/logica/>. Acesso em: 01 dez. 2021.

PIVA JUNIOR, D. *et al.* **Algoritmos e Programação de Computadores**. 2. ed. Rio de Janeiro: Elsevier, 2019. Disponível em: <http://www.univates.br/biblioteca/e-books-minha-biblioteca?isbn=9788595150508>. Acesso em: 01 dez. 2021.

PUGA, Sandra; RISSETTI, Gerson. **Lógica de programação e estrutura de dados**. 3. Ed. São Paulo: Pearson Education do Brasil, 2016. ISBN 978-85-430-1914-7. Disponível em: <http://www.univates.br/biblioteca/biblioteca-virtual-universitaria?isbn=978-85-430-1914-7>. Acesso em: 01 dez. 2021.

TYLDUM, Morten. O jogo da imitação [sinopse]. **Google**, 2015. Disponível em: https://www.google.com/search?kgmid=/m/0vs_2xp&hl=pt-BR&q=O+Jogo+Da+I-mita%C3%A7%C3%A3o&kgs=6f31a5a401b25752&shndl=0&source=sh/x/kp/1&entrypoint=sh/x/kp. Acesso em: 1 dez. 2021.