

LÓGICA DE PROGRAMAÇÃO

Esp. Marcelo Takashi Uemura

INICIAR

introdução

Introdução

Nesta unidade, apresentaremos os conceitos relacionados à Lógica de Programação, iniciando com o conceito de algoritmos, que representa o coração de uma base sólida para a construção de um programa. Serão abordados os tipos de algoritmos, como a linguagem natural, o fluxograma e o pseudocódigo, bem como comandos de entrada e saída.

Na sequência, estudaremos os tipos de processamento, com o uso de variáveis, constantes, expressões aritméticas e lógicas. Também serão abordados os operadores matemáticos e lógicos, com o uso de funções matemáticas e tabela da verdade e a apresentação de uma estrutura sequencial.

Apresentaremos, também, um breve relato da história da linguagem de programação, complementando com os tipos de dados, constantes e variáveis, os comandos de entrada e saída e a estrutura sequencial, com uma aplicabilidade voltada para a construção de algoritmos.

Por fim, abordaremos a estrutura de seleção, apresentando a sintaxe dessa estrutura, também conhecida como estrutura de desvio, classificada como seleção simples ou composta.

Introdução à Lógica de Programação

Neste conteúdo, serão apresentados os conceitos iniciais da lógica de programação, visando entender como construir um algoritmo e os tipos de algoritmos, como a linguagem natural, o fluxograma e o pseudocódigo.

Conceitos Iniciais

O entendimento de algoritmos e lógica de programação é essencial para a construção de um programa (*software*), que nada mais é do que um conjunto de instruções definido por um ser humano e que cumpre um objetivo ao ser executado em uma máquina (*hardware*) (SOUZA et al., 2011).

A lógica de programação pode ser concebida por meio de inúmeras linguagens de programação, que são formas padronizadas de representar o raciocínio utilizado (lógica) para a cooperação de desenvolvimento de soluções, logicamente, válidas e coerentes, por meio da construção de algoritmos (FORBELLONE; EBESPÄCHER, 2005).

Segundo Forbellone e Ebspächer (2005), um algoritmo pode ser definido como uma sequência de passos para a consecução de um objetivo. Para tanto, é importante o uso de uma lógica para ordenar esses passos definidos pelo algoritmo. Por exemplo, em uma receita de bolo, é importante a ordem da sequência das instruções em relação ao manuseio dos ingredientes, para que o resultado final seja atingido.

Na esfera da computação, o algoritmo está associado a um conjunto de regras e operações que está ordenado para a solução de um problema, ou classe de problemas, com um número finitos de passos (MANZANO; OLIVEIRA, 2010).

Construção de um algoritmo

A construção de um algoritmo é importante, devido à necessidade de especificar, de forma clara e precisa, o raciocínio lógico empregado para uma sequência de ações, desde um estado inicial até um estado final previsível e bem definido, em que o resultado esperado é alcançado (FORBELLONE; EBERSPÄCHER, 2005).

Os algoritmos devem representar um programa, detalhando os dados que serão por ele processados e as instruções que serão utilizadas. A formalização dos algoritmos deve seguir alguma convenção, para que todos possam entendê-los da mesma forma (SOUZA et al., 2011). Para tanto, algumas regras devem ser seguidas com relação à sintaxe e à semântica.

Segundo Souza et al. (2011), as regras utilizadas por algoritmos, no que tange à sintaxe, estabelecem os tipos de comandos que podem ser utilizados e as suas expressões adotadas. Essas expressões realizam algum tipo de operação com os dados e podem ser classificadas em três tipos:

- estruturas sequenciais – a execução das instruções são diretas e imperativas, não havendo condições ou desvios em seu caminho.

- estruturas de decisão – a execução da sequência de instruções depende de condições atribuídas a decisões ao longo do caminho.
- estruturas de repetição – a execução das instruções se repete enquanto não for atingida uma condição.

Vejamos, a seguir, alguns exemplos.

1) Estrutura sequencial:

Início

1. *pegar uma pilha*
2. *colocar a pilha no rádio*
3. *ligar o rádio*
4. *escutar a estação*

Fim

2) Estrutura de decisão:

Início

1. *pegar uma pilha*
2. *colocar pilha no rádio*
3. *ligar o rádio*
4. *escutar a estação*
5. *se está passando o jogo, então*
 - a. *continuar escutando*
6. *senão*
 - a. *desligar o rádio*

Fim

3) Estrutura de repetição:

Início

1. *pegar uma pilha*
2. *colocar pilha no rádio*
3. *ligar o rádio*
4. *escutar a estação*
5. *enquanto estiver tocando algo interessante*
 - a. *continuar escutando o rádio*

6. desligar o rádio

Fim

saiba mais
Saiba mais

Saiba mais sobre o real impacto do algoritmo na vida das pessoas, em um contexto atual. Assista ao vídeo Algoritmos: você está no controle?

ASSISTIR

As sentenças utilizadas em um algoritmo são, normalmente, objetivas, curtas e simples, para que não haja dubiedade na sua interpretação (semântica). A seguir, veremos os tipos de algoritmo em relação à forma de representação.

Tipos de algoritmo

Os algoritmos devem apresentar uma sequência lógica que permita três elementos: entrada de dados, processamento de dados e saída de dados. A entrada e a saída de dados são caracterizadas por **comandos de entrada e saída**. Esses comandos permitem que dados possam ser inseridos para o processamento a ser dado pelo algoritmo e resultados possam ser colocados na saída de dados.

Exemplos:

Os tipos de algoritmos podem ser classificados em relação à forma de representação deles, utilizando os três elementos mencionados. Podemos ter algoritmos representados por linguagem natural, fluxograma e pseudocódigo.

Linguagem natural

A linguagem natural representa uma descrição narrativa do algoritmo, conforme apresentado nos exemplos do tópico anterior. Algumas palavras-chave podem ser

utilizadas na linguagem natural, como “se...então...senão”, para representar uma decisão (MEDINA; FERTIG, 2006, p.19).

Exemplo:

1. Início
2. Inserir primeiro dado
3. Inserir segundo dado
4. Somar os dois dados
5. Apresentar o resultado.
6. Fim

Esse formato é próximo ao entendimento do ser humano, por isso é chamado de linguagem natural. A seguir, será apresentado outro formato de algoritmo, também muito utilizado, o fluxograma.

Fluxograma

O fluxograma, ou diagrama de blocos, é uma forma padronizada de representar algoritmos com o uso de símbolos gráficos (FLUXOGRAMAS..., on-line). Vejamos um exemplo:

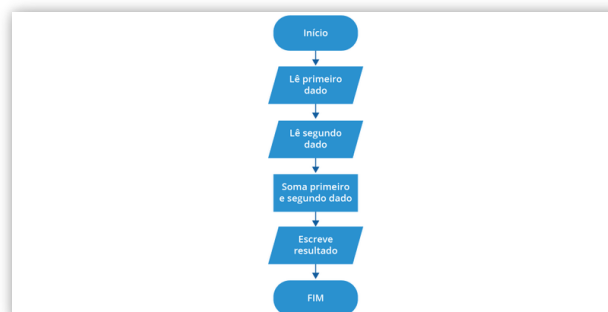


Figura 1.1: Fluxograma
Fonte: Elaborada pelo autor.

O fluxograma da Figura 1.1 permite apresentar, de forma gráfica, as sentenças, a entrada de dados, a saída de dados, os desvios e as repetições. Isso facilita visualmente o entendimento do algoritmo, em especial, dos algoritmos complexos.

Outro formato utilizado para representação de algoritmos é o pseudocódigo, que está mais próximo do formato utilizado pelas linguagens de programação.

Pseudocódigo

O pseudocódigo é uma forma de representação de algoritmo, similar a uma linguagem de programação, por meio de comandos básicos padronizados (LÓGICA, *on-line*). O pseudocódigo pode ser reaproveitado para as linguagens de programação conhecidas, adequando-se de acordo com o paradigma a ser adotado (ex.: adequação para linguagem orientada a objetos). Vejamos um exemplo:

1. início
2. inteiro a, b, resultado
3. ler (a)
4. ler (b)
5. resultado = a+b
6. escrever (resultado)
7. fim

O pseudocódigo do exemplo anterior apresenta a leitura de dois dados (a e b) e a escrita de um resultado que corresponde a sua soma.

atividade

Atividade

Os algoritmos são utilizados para apresentar a lógica de programação de forma sequencial e ordenada, por meio de comandos que realizam operações sobre dados (SOUZA et al., 2011). A respeito das estruturas utilizadas por algoritmos, assinale a alternativa que representa a estrutura baseada em condição que define uma decisão sobre o caminho a ser seguido na sequência.

SOUZA, M. A. F. et al. **Algoritmos e Lógica de Programação**. São Paulo: Cengage Learning, 2011.

- ☐ a) Sequência.
- ☐ b) Decisão.
- ☐ c) Repetição.
- ☐ d) Salto incondicional.
- ☐ e) Encadeamento.

Introdução à Lógica de Programação

Neste tópico, serão explorados os princípios teóricos referente ao desenvolvimento de algoritmos, como o processamento de dados por meio do uso de variáveis e constantes, bem serão descritas as ferramentas básicas para a construção de algoritmos, como a tabela da verdade, o emprego de operadores matemáticos e lógicos.

Tipos de Processamento

Os dados que devem ser processados pelos programas de um computador devem estar armazenados em áreas que foram reservadas na memória principal. Esses dados podem ser constantes ou variáveis.

Variáveis

Os dados variáveis são a representação simbólica de elementos que podem variar ao longo da execução de um algoritmo. Toda variável é identificada por um identificador, que é formado por um ou mais caracteres, sendo que o primeiro caractere deve ser, obrigatoriamente, uma letra, e os caracteres seguintes podem ser números ou letras, não sendo permitido o uso de símbolos especiais. As variáveis podem ser classificadas como numéricas, lógicas ou literais (FARRER et al., 2013).

Vejamos um exemplo:

1. início
2. var a = 1

3. var b = 10
4. enquanto b > 0 faça
 - a. a = a +1
 - b. b = b-1
5. fim

Nesse exemplo, a e b são variáveis, cujo conteúdo é modificado dentro do laço de repetição “enquanto”.

Constantes

O dado constante é um valor fixo que não se altera ao longo da execução de um algoritmo, podendo ser um número (constante numérica), um valor lógico (constante lógica) ou uma sequência de caracteres (constante literal) (FARRER et al., 2013, p. 29).

Vejamos um exemplo:

1. início
2. pi = 3.14
3. area = pi*raio*raio/2
4. fim

Nesse exemplo, pi é uma constante cujo valor é 3,14, que é utilizada para o cálculo da área, e seu valor não é alterado ao longo da execução do algoritmo.

Expressões aritméticas e lógicas

As expressões aritméticas são utilizadas para o processamento matemático e são realizadas a partir do relacionamento entre operandos (variáveis e constantes numéricas inteiras ou de ponto flutuante), por meio de operadores matemáticos (MANZANO; OLIVEIRA, 2010).

As expressões lógicas são expressões em que o resultado só pode ser verdadeiro ou falso, por meio de operadores lógicos e operandos, que podem ser relações, constantes e/ou variáveis do tipo lógico (FARRER et al., 2013).

Operadores Matemáticos

Segundo Farrer et al. (2013), algumas operações matemáticas básicas utilizadas em expressões aritméticas são: **adição (+)**, **subtração (-)**, **multiplicação (mul)**, **divisão (div)**, **potenciação (pot)** e **radiciação (raiz)**. Dentre essas operações, existem

prioridades que devem ser obedecidas, como potenciação e radiciação, na sequência, multiplicação e divisão e, por fim, adição e subtração.

Além das operações básicas, podem-se usar, em expressões aritméticas, **funções matemáticas**. No Quadro 1.1, temos as principais funções matemáticas utilizadas.

Nome	Resultado
LOG(EA)	logaritmo na base 10 de EA
LN(EA)	logaritmo neperiano de EA
EXP(EA)	o número “e” (Neper) elevado a EA
ABS(EA)	valor absoluto de EA
TRUNCA(EA)	a parte inteira de um número fracionário
ARREDONDA(EA)	transforma, por arredondamento, um número fracionário em um número inteiro
SINAL(EA)	fornece o valor -1, +1 ou zero, conforme o valor de EA seja negativo, positivo ou igual a zero
QUOCIENTE(EAx, EAy)	quociente inteiro da divisão EAx por EAy
RESTO(EAx, EAy)	resto da divisão de EAx por EAy

Quadro 1.1 - Funções de EA, EAx, EAy

Fonte: Farrer et al. (2013, p. 33).

A função deve atuar sobre um argumento numérico, sendo o resultado de uma avaliação de expressão aritmética entre parênteses (FARRER et al., 2013).

Operadores Lógicos

Segundo Farrer et al. (2013, p. 36), os operadores lógicos são representados por conectivos, como conjunção (e, representado, simbolicamente, por ^), disjunção (ou, representado, simbolicamente, por v) e negação (não).

Uma forma de representar as expressões lógicas com os operadores lógicos é pela tabela da verdade.

Tabela-verdade

A tabela da verdade é o conjunto de todas as possibilidades combinatórias entre os valores de diversas variáveis lógicas, baseados em situações de verdadeiro (V) ou falso (F), e os operadores lógicos (FOBERLLONE; EBERSPÄCHER, 2005).

Veja, a seguir, alguns exemplos.

A	não A
F	V
V	F

Quadro 1.2 - Operação não

Fonte: Foberllone e Eberspächer (2005, p. 23).

A	B	A e B ($A \wedge B$)
F	F	F
F	V	F
V	F	F
V	V	V

Quadro 1.3 - Operação e

Fonte: Foberllone e Eberspächer (2005, p. 23).

A	B	A ou B (A v B)
F	F	F
F	V	V
V	F	V
V	V	V

Quadro 1.4 - Operação ou

Fonte: Foberllone e Eberspächer (2005, p. 23).

Também existe precedência entre os operadores lógicos. A prioridade é a operação lógica não, seguida da e e, posteriormente, da ou.

Estrutura Sequencial

De acordo com Forbellone e Eberspächer (2005, p. 30), uma estrutura sequencial de um algoritmo está voltada a um conjunto de ações primitivas que será executado em uma sequência linear de cima para baixo e da esquerda para a direita, na mesma ordem em que foi escrito. As ações serão seguidas por ponto e vírgula (;), para separar uma ação de outra. Vejamos um exemplo:

1. início
2. // declaração de variáveis
3. ação 1;
4. ação 2 ;
5. ação 3;
6. fim

As estruturas sequenciais representam os comandos (ações) que são executados sem nenhum desvio no fluxo, tendo como exemplos, cálculos, execuções de funções e procedimentos (SOUZA et al., 2011, p.126).

atividade

Atividade

Os algoritmos fazem uso de expressões aritméticas e lógicas, as quais são representadas por comandos que contêm operandos e operadores. No caso de expressões lógicas, assinale a alternativa que indica o que podemos considerar um exemplo de operador.

- ☐ **a)** Adição.
- ☐ **b)** Negação.
- ☐ **c)** Potenciação.
- ☐ **d)** Subtração.
- ☐ **e)** Multiplicação.

Introdução a Lógica de Programação

Neste tópico, será apresentado um breve histórico das linguagens de programação, bem como serão identificados os princípios básicos das formas de construção de programas em uma linguagem de programação, como a declaração de variáveis, o uso de comandos de entrada e saída e a elaboração de uma estrutura sequencial.

História da Linguagem de Programação

As linguagens de programação não possuem, historicamente, uma data exata de início, mas a evolução começou a partir da década de 30, com os computadores elétricos. Segundo Pacievitch (*on-line*), Konrad Zuse publicou a linguagem de programação Plankalkül, em 1948, sem ter tido muito sucesso. Antes da programação ser passada para o computador, fazia-se uso de cartões de papel perfurados, que representavam o código dos programas.

Na década de 50, surgiram as primeiras linguagens modernas, como FORTRAN, LISP, COBOL e ALGOL 60, que ficaram conhecidas como linguagens de uso generalizado (FARRER et al., 2013). Nas décadas de 60 e 70, foram criados os paradigmas de programação de forma estruturada, por meio das linguagens Simula (Nygaard e Dahl), C (Dennis Ritchie e Ken Thompson), Prolog e Pascal (PACIEVITCH, *on-line*).

Na década de 80, surgiram a linguagem C++ (compatível com C e orientada a objetos) e Perl (para trabalhos de sobrecarga grandes). Nos anos 90, com o crescimento da

internet, surgiram as linguagens Java e Javascript, bem como PHP, Visual Basic e Object Pascal (PACIEVITCH, *on-line*).

Atualmente, existe muita procura pelo uso de linguagens de programação baseadas em plataformas como .NET, bem com linguagens voltadas para programação Web, utilizando sistemas distribuídos, como APIs via Web Services.

saiba mais
Saiba mais

Para saber mais sobre as linguagens de programação mais utilizadas em 2018 e 2019, leia o artigo a seguir, de Jonathan Soares.

ACESSAR

Tipos de Dados, Constantes e Variáveis

A construção dos algoritmos é fundamentada nos dados. O dado é uma representação no estado primitivo, que retrata uma ideia ou uma situação, cujo processamento trará como resultado uma informação. Como os dados são armazenados na memória do computador, é importante a definição do seu tipo, que pode ser um inteiro, real, lógico, caractere ou cadeia de caracteres (VILARIM, 2004).

Um inteiro representa um número no conjunto dos números inteiros, tanto positivos como negativos, não contendo partes fracionárias. O real representa números dos conjuntos dos números reais, que permitem frações com a adoção de ponto decimal. Os dados lógicos representam valores booleanos, como verdadeiro ou falso. O caractere representa uma letra, um dígito ou um símbolo, enquanto a cadeia de caracteres representa um conjunto de caracteres armazenado na memória (VILARIM, 2004).

Os valores de dados utilizados em um algoritmo podem ser aplicados com constantes ou variáveis. Para que as constantes e as variáveis sejam reconhecidas pelos algoritmos,

devem ser declaradas, para que o processador possa alocar adequadamente o espaço a ser utilizado na memória (FORBELLONE; EBERSPÄCHER, 2005). Vejamos alguns exemplos.

inteiro:x;

caractere: nome;

real: dólar

lógico: resposta;

x, nome, dólar e resposta são os identificadores das variáveis, enquanto inteiro, caractere, real e lógico são os tipos de dados declarados para essas variáveis.

Comandos de Entrada e Saída

Os comandos de entrada e saída são utilizados para uma interação do programa, permitindo a entrada e a saída de dados, em operações conhecidas como leitura e escrita de dados.

Nos comandos de entrada, são realizadas instruções de leitura de dados, as quais podemos representar nos algoritmos com a palavra “ler” ou “leia”. Uma sintaxe que pode ser adotada é:

leia (identificador da variável);

Para comandos de saída, as instruções têm o propósito de realizar uma escrita de dados, sendo representada por palavras como “escreve” ou “escreva”.

escreva (identificador da variável);

A seguir, veremos como utilizar as declarações de variáveis e os comandos de entrada e saída em uma estrutura sequencial de um algoritmo para uma linguagem de programação.

Estrutura sequencial

A estrutura sequencial estabelece um conjunto de ações primitivas, comandos que serão executados de forma sequencial, sem desvios. O formato para a construção de um programa com algoritmos deve se basear na declaração de variáveis antes do corpo de ações do algoritmo (FORBELLONE; EBELSPÄCHER, 2005).

1. *início // início do bloco do algoritmo*
2. *// declaração de variáveis*
3. *// corpo do algoritmo*
4. *ação 1;*
5. *ação 2;*
6. *ação 3;*
7. *ação N;*
8. *fim // fim do bloco do algoritmo*

Exemplo: construir um algoritmo com estrutura sequencial que calcule a média aritmética de 4 notas bimestrais.

1. início
2. *// declaração de variáveis*
3. real: N1, N2, N3, N4;
4. real: media;
5. *// corpo do algoritmo*
6. *leia (N1, N2, N3, N4); // comando de entrada de dados*
7. *media = (N1 + N2 + N3 + N4) div 4; // expressão aritmética para média*
8. *escreva (media); // comando de saída de dados*
9. fim

O exemplo apresentado mostra um algoritmo com declaração de variáveis de dados do tipo real, com identificadores de variáveis N1, N2, N3, N4 e media, comandos de entrada e saída, como leia e escreva, e o uso de expressões aritméticas com operadores de adição e divisão em uma estrutura sequencial.

atividade

Atividade

Os tipos de dados auxiliam na alocação de memória para as variáveis e constantes que são criadas ao longo de um algoritmo. Nesse sentido, o tipo de dado que representa apenas uma letra é conhecido como:

- ☐ **a)** inteiro.
- ☐ **b)** real.
- ☐ **c)** lógico.
- ☐ **d)** caractere.
- ☐ **e)** cadeia de caracteres.

Estrutura de Seleção

Uma estrutura de seleção permite que um conjunto de ações possa ser selecionado em relação a sua execução, baseada em condições representadas por expressões lógicas ou relacionais (FORBELLONE; EBERSPÄCHER, 2005, p.33). As estruturas de seleção também são conhecidas como estruturas de desvio.

Sintaxe da Estrutura de Sequência: Seleção Simples e Seleção Composta

As estruturas de seleção simples são utilizadas quando uma ação é tomada com base no teste de uma condição. Vejamos, a seguir, um exemplo.

1. se <condição>
 - a. então
 - i. ação;
2. fimse

Nesse exemplo, se a condição for verdadeira, a ação sob a cláusula então será executada. Caso a condição seja falsa, será encerrada (fimse). Nas estruturas de seleção composta, temos algumas ações que podem ser executadas caso a condição seja verdadeira (então), e outras caso seja falsa (senão). Observe, na sequência, um exemplo.

1. se <condição>
 - a. então

- i. ação 1;
- b. senão
 - i. ação 2;
- 2. fimse

Nesse exemplo, a ação 1 será executada se a condição for verdadeira, e a ação 2 será executada no caso de a condição ser falsa.

atividade

Atividade

Analise o algoritmo a seguir.

1. *se* ($a > 5$)
2. *então*
 - a. *escreva* "positivo";
 - b. *apaga a luz*;
3. *senão*
 - a. *escreva* "negativo"
 - b. *mantenha a luz acesa*
4. *fimse*.

Agora, assinale a alternativa que indique quais ações serão executadas no caso de a variável a ser igual a 4.

- ☐ **a)** Escrever "negativo" e apagar a luz.
- ☐ **b)** Escrever "positivo" e apagar a luz.
- ☐ **c)** Escrever "negativo" e manter a luz acesa.
- ☐ **d)** Escrever "positivo" e manter a luz acesa.
- ☐ **e)** Escrever "positivo" e, depois, "negativo".

indicações

Material Complementar



FILME

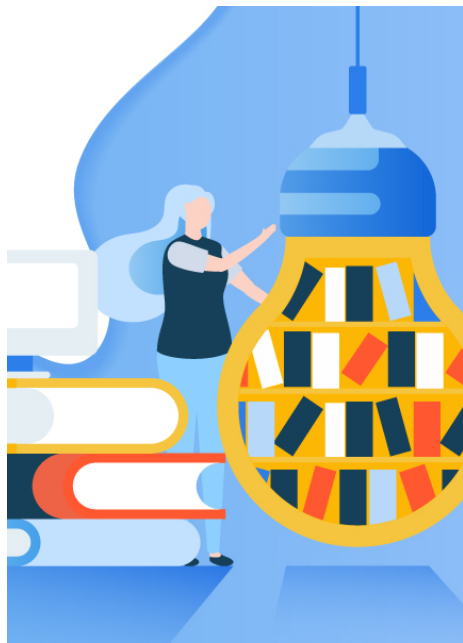
O prazer da lógica

Ano: 2013

Comentário: 2013

Comentário: Esse interessante documentário retrata a lógica e sua importância para nossas vidas, em especial para a ciência, utilizando a lógica *booleana* e fazendo associações com situações do cotidiano.

TRAILER



LIVRO

Lógica de Programação - A construção de algoritmos e estruturas de dados

André Luiz Villar Forbellone e Henri Frederico Eberspächer

Editora: Pearson

ISBN: 978-85-7605-024-7

Comentário: Esse livro explica, de forma objetiva, os algoritmos, tópico mencionado nesta unidade, com a apresentação de diversos exemplos e o uso de uma pseudolinguagem próxima às linguagens de programação, que facilita a fixação do conteúdo.

conclusão

Conclusão

Abordamos, nesta unidade, os conceitos relacionados à lógica de programação, em especial os algoritmos, que são ferramentas que permitem representar, de forma ordenada, a lógica das ações que serão utilizadas para a resolução de um problema.

Para a construção de um algoritmo, é importante conhecer os tipos de dados que serão utilizados, as expressões aritméticas e lógicas, os comandos de entrada e saída, bem como uma sintaxe e semântica que permitam uma maior clareza e padronização, além das estruturas de controle, como as sequenciais, de seleção e repetição.

Por fim, destacamos que é possível a construção de algoritmos que permitam elaborar uma lógica, que, além de resolver uma situação proposta, venha a facilitar a confecção de um programa pelo uso de uma linguagem de programação.

referências

Referências Bibliográficas

CINTRA, C. Crianças aprendem programação ainda na educação infantil. **Correio Braziliense**, 27 set. 2018. Disponível em: <<https://www.correiobraziliense.com.br/app/noticia/escolhaaescola/2018/2018/09/27/noticias/escolhaaescola2018,708752/tecnologia-no-ensino.shtml>>. Acesso em: 12 abr. 2019.

FARRER, H. et al. **Algoritmos Estruturados**. Rio de Janeiro: LTC, 2013.

FLUXOGRAMAS, diagrama de blocos e de Chapin no desenvolvimento de algoritmos. **DEVMEDIA**. Disponível em: <<https://www.devmedia.com.br/fluxogramas-diagrama-de-blocos-e-de-chapin-no-desenvolvimento-de-algoritmos/28550>>. Acesso em: 07 abr. 2019.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F. **Lógica de Programação** - a construção de algoritmos e estruturas de dados. São Paulo: Prentice Hall, 2005.

LÓGICA de programação: introdução a algoritmos e pseudocódigo. **DEVMEDIA**. Disponível em: <<https://www.devmedia.com.br/logica-de-programacao-introducao-a-algoritmos-e-pseudocodigo/37918>>. Acesso em: 07 abr. 2019.

MANZANO, J. A. N. G.; OLIVEIRA, J. F. **Algoritmos** - Lógica para Desenvolvimento de Programação e Computadores. São Paulo: Erica, 2010.

MEDINA, M.; FERTIG, C. **Algoritmos e Programação** - Teoria e Prática. São Paulo: Novatec Editora, 2006.

PACIEVITCH, Y. História da programação. **Infoescola**. Disponível em: <<https://www.infoescola.com/informatica/historia-da-programacao/>>. Acesso em: 09 abr. 2019.

SOARES, J. Confira a lista das linguagens de programação mais utilizadas em 2018-2019. **Codigosimples.net**, 09 mar. 2019. Disponível em: <<https://codigosimples.net/2019/03/09/confira-a-lista-das-linguagens-de-programacao-mais-utilizadas-em-2018-2019/>>. Acesso em: 10 abr. 2019.

SOUZA, M. A. F. et al. **Algoritmos e Lógica de Programação**. São Paulo: Cengage Learning, 2011.

VILARIM, G. **Algoritmos** - programação para iniciantes. Rio de Janeiro: Editora Ciência Moderna, 2004.

IMPRIMIR