

Introdução aos algoritmos e programação por blocos

Durante todo o dia, nos deparamos com algoritmos, desde quando acordamos até quando estamos dormindo. Afinal, um algoritmo é uma sequência finita de ações que tem como objetivo produzir solução para um dado problema. Por exemplo, se lhe perguntar como você faz para escovar seus dentes, você me dará uma lista com um passo a passo que vai desde pegar o tubo do creme dental até enxaguar a boca, fechando a torneira e guardando todos os itens no final do processo.

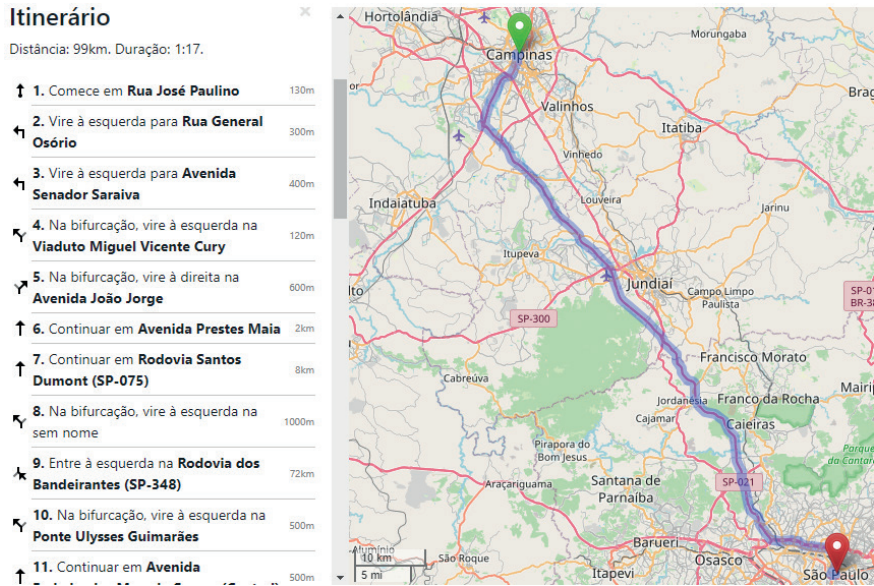
Vamos pensar agora em como fazer um bolo: primeiramente, separamos os ingredientes. Em seguida, quebramos os ovos e separamos as claras e das gemas. Depois, batemos a clara em neve. Misturamos os demais ingredientes. Untamos a forma e despejamos a massa. Assamos em forno preaquecido a 180 °C. Por fim, retiramos a forma com o bolo do forno. Eis uma sequência de instruções. Ela iniciou (separe os ingredientes) e teve um fim (retire o bolo do forno).

A partir desses exemplos, é possível observar que, quando abordamos o conceito de algoritmos, não estamos falando apenas sobre a programação de computadores, mas, sim, sobre uma técnica de pensamento computacional que pode ser utilizada na programação de computadores.

1 Introdução e conceitos de algoritmos

Os algoritmos estão presentes na base de muitas ações que são desempenhadas no dia a dia. Por exemplo, uma pessoa chamada Ana, que mora na cidade de São Paulo, convida seu amigo Betto, que mora na cidade de Campinas, para visitá-la pela primeira vez. Como Betto não sabe como chegar até a residência de Ana, ela necessita desenvolver uma sequência finita de instruções para que ambos se encontrem. A figura 1 a seguir apresenta uma possível rota de Campinas até a casa de Ana em São Paulo.

Figura 1 – Mapa de Campinas até a cidade de São Paulo



Fonte: adaptado de OpenStreetMap ([s. d.]).

Observe na figura 1 que, no lado esquerdo, temos o algoritmo com o passo a passo da rota que Betto deve seguir para chegar até a casa de Ana. Ainda na figura 1, temos no lado direito um algoritmo também, porém, este é gráfico e mostra o mapa com todo o trajeto a ser percorrido. Note, então, que se trata de uma sequência finita de ações que visa produzir um resultado (ou seja, evitar que seu amigo fique perdido e que chegue até a cidade sem erros). Esse é o conceito de algoritmos, uma sequência finita de instruções que visa resolver algum tipo de problema (FORBELLONE; EBERSPACHER, 2005).

1.1 Lógica computacional

Para entender corretamente como construir um algoritmo, primeiramente é necessário compreender a lógica de programação.

Na figura 2, é possível observar que temos uma nova rota, agora destacada em amarelo. Essa segunda rota não é o melhor caminho possível, afinal, é muito pior que o trajeto (algoritmo) apresentado na figura 1. Assim, conclui-se que a lógica utilizada para desenvolver o primeiro algoritmo (figura 1) foi melhor que a lógica utilizada para desenvolver o segundo (figura 2).

Ainda sobre as rotas, Betto logicamente escolherá o primeiro algoritmo, afinal, o tempo que levará para chegar até o destino (bem como a distância a percorrer) é muito menor. Em outras palavras, os algoritmos são conjuntos de instruções lógicas que visam resolver um determinado problema.

Agora, vamos considerar um segundo exemplo, em que é dada a sequência numérica:

1 1 2 3 5 8 13 ?

Como definir qual será o próximo número da sequência? Nesse caso, o próximo valor é o 21. Isso pode ser afirmado em razão do valor atual ser sempre a soma dos dois números anteriores:

- $1 + 1 = 2$
- $1 + 2 = 3$
- $2 + 3 = 5$
- $3 + 5 = 8$
- $5 + 8 = 13$
- $8 + 13 = 21$

Para encontrar o valor da sequência, foi necessário encontrar um padrão lógico e, em seguida, criar uma sequência fixa e finita de instruções. Assim, é possível concluir que um algoritmo deve possuir as seguintes características (FORBELLONE; EBERSPACHER, 2005; FEIJÓ; CLUA; SILVA, 2009):

- Cada passo do algoritmo deve ser uma instrução possível de ser realizada: considere o exemplo da viagem de Campinas até São Paulo. Existe uma sequência lógica de instruções e todas elas precisam ser executadas, se uma dessas instruções não ocorrer, o algoritmo não será executado corretamente.
- A ordem de cada uma das instruções deve ser respeitada: considere o segundo exemplo. Nele, o valor de um número X é obtido somando os dois valores anteriores a ele em uma sequência numérica. Se essa instrução não for obedecida (ou seja, se não somarmos os dois últimos valores para obter o próximo da sequência), o resultado não será conforme o esperado.
- O algoritmo deve ser finito: considere ainda o segundo exemplo. O algoritmo pode ser descrito como “para obter um valor X, devemos somar os dois valores imediatamente anteriores”. Então, apesar de ser possível obter uma sequência infinita de valores, o algoritmo é finito em suas instruções.

Outro exemplo que sintetiza essas três características de um algoritmo é o processo de troca de pneu de um carro (considerando que o estepe está presente e em bom estado):

1. Levantar o carro do lado do pneu que está furado.
2. Desaparafusar cada um dos parafusos da roda.
3. Remover a roda.
4. Deixar a roda no chão.
5. Colocar o estepe no eixo que está sem a roda.
6. Parafusar cada um dos parafusos da roda.
7. Abaixar o carro.

Observe que alguns passos podem parecer óbvios para nós, como no caso das instruções 3 e 4. Para os seres humanos, é natural que a

roda do carro deve ser retirada e colocada no chão para que seja possível colocar o estepe. Isso ocorre porque pessoas possuem inteligência natural e habilidade racional, algo que o computador não é capaz de obter. Logo, é necessário que sejam escritas instruções explícitas (por mais óbvias que pareçam).



NA PRÁTICA

Considere que você é um confeitoiro e está prestes a fazer um bolo. Você pega a receita e observa que a sequência de instrução do modo de preparo é:

1. Leve ao forno por 30 minutos.
2. Separe os ingredientes.
3. Realize o preparo.

Bem, não é a melhor instrução, afinal, como posso levar ao forno algo que não preparei ainda? Logo, podemos concluir que essa instrução é ilógica. Torna-se, então, essencial compreender que a ordem das instruções pode afetar o resultado final do seu algoritmo.

2 Descrição narrativa de um algoritmo e implementação por blocos

Até aqui pudemos compreender o que é um algoritmo, suas características e a necessidade de lógica para elaborá-lo. No exemplo de algoritmo para troca do pneu de um carro, foram descritas as principais instruções, porém, seria possível detalhar ainda mais, indicando, por exemplo, quantos e qual a ordem dos parafusos a serem desaparafusados.

No entanto, como descrever um algoritmo? Existem pelo menos três formas principais para representação de um algoritmo: descrição narrativa, fluxograma e pseudocódigo.

2.1 Descrição narrativa

Um algoritmo pode ser representado em linguagem natural, ou seja, como as pessoas em geral falam (FEIJÓ; CLUA; SILVA, 2009). Por exemplo, a receita de bolo a seguir:

- Bata a clara de dois ovos em neve. Em seguida, coloque as gemas e bata por 1 minuto em velocidade baixa. Adicione os demais ingredientes e bata em velocidade alta por 5 minutos. Enquanto bate, unte uma forma com manteiga e espalhe duas colheres de farinha de trigo. Despeje todo o conteúdo da batedeira na forma e, em seguida, leve ao forno por 30 minutos, o qual deve estar preaquecido a 180 °C.

Neste outro exemplo, trata-se de um algoritmo para troca de lâmpada:

- Primeiramente, localize a lâmpada a ser substituída. Posicione uma escada logo embaixo da lâmpada a ser trocada. Suba os degraus da escada até alcançar a lâmpada. Gire-a no sentido anti-horário até que seja possível retirá-la do soquete. Desça da escada e coloque a lâmpada queimada em um local seguro, como uma mesa. Agora, de posse da lâmpada nova, suba a escada até que consiga alcançar o local de instalação. Posicione a lâmpada nova no soquete e gire no sentido horário até sentir que ela está bem firme. Desça da escada.

Esse tipo de algoritmo é pouco usado na prática, afinal, pode gerar múltiplos sentidos de interpretação. No primeiro exemplo, está escrito que devemos “colocar” as gemas, porém, não é dito onde. Poderíamos dizer: “coloque na batedeira”, porém, ainda estaria errado, pois a batedeira é um aparelho formado por um motor, uma tigela e um suporte. O ovo deve ser colocado e batido na tigela da batedeira.

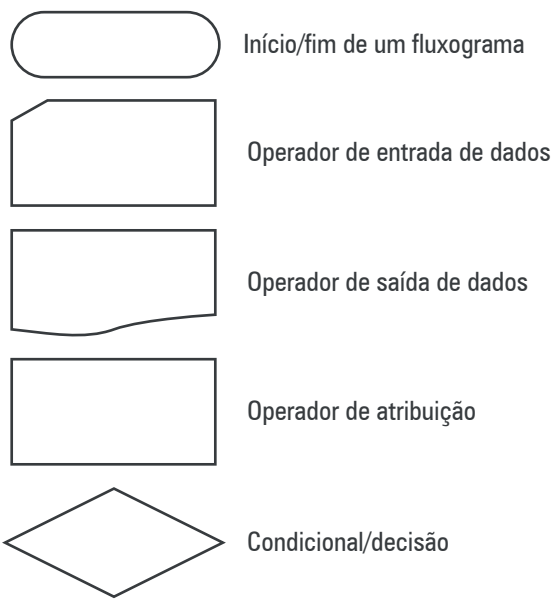
Como dito anteriormente, as instruções devem ser colocadas de modo explícito, sem dar abertura para mais de um sentido de interpretação. Podemos resolver esse tipo de questão utilizando fluxogramas.

Já no exemplo da troca de lâmpada, é dito para posicionar a escada “logo embaixo da lâmpada a ser trocada”, entretanto, não é dito para apoiar ou para abrir a escada. Logo, dependendo de quem ouvir essa instrução, pode acarretar algum mal-entendido.

2.2 Fluxograma

O fluxograma é um conjunto de estados que é utilizado para representar graficamente um algoritmo (FORBELLONE; EBERSPACHER, 2005). Cada um desses estados (instruções e comandos) é representado por uma forma geométrica específica, conforme ilustrado na figura 3.

Figura 3 – Formas geométricas utilizadas em um fluxograma



Na figura 3, é possível observar que os principais elementos que compõem um fluxograma apresentam formas geométricas diferentes. Isso auxilia a interpretar suas diferentes ações em um algoritmo.

Para poder desenhar um fluxograma, é possível utilizar ferramentas digitais, como Microsoft PowerPoint, Google Presentation ou qualquer outro editor de slides, que, em sua grande maioria, possuem um conjunto de formas específicas para isso. Além disso, é possível realizar o desenho manual ou, então, utilizar ferramentas digitais como o Flowgorithm, draw.io ou Diagrams, que são sistemas gratuitos para elaboração de fluxogramas.

Para obter o Flowgorithm, basta pesquisar pelo seu nome em qualquer site de busca e, em seguida, escolher o site do proprietário. O processo de download e instalação é detalhado diretamente na página.

Por outro lado, há também a possibilidade de utilizar ferramentas on-lines gratuitas. Uma das principais e mais simples para se trabalhar é o draw.io. Para acessá-la, digite em qualquer site de busca o nome da ferramenta (draw.io) e, em seguida, acesse o site. O modo de operação para criar o seu primeiro fluxo é apresentado na tela do próprio desenvolvedor da ferramenta.

Uma das propriedades ou características de um algoritmo é sua finitude, ou seja, todo algoritmo é finito. Na figura 4 a seguir, é possível observar que, quando criamos um fluxograma no Flowgorithm, ele já nos traz um nó inicial (principal) e um nó final (fim). Então, torna-se possível adicionar elementos entre esses dois nós de modo a desenhar o algoritmo.



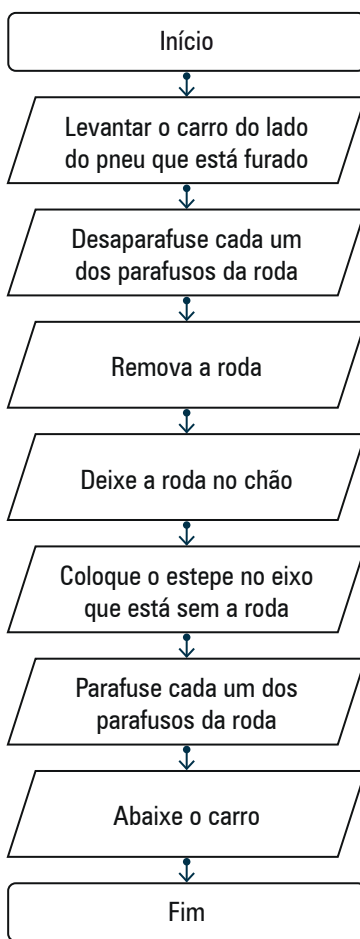
NA PRÁTICA

Ao utilizar o Flowgorithm, é possível observar que o programa já nos traz, de modo fixo, os dois nós de início e fim do algoritmo. Para adicionar um novo nó, basta clicar sobre a linha que separa as duas formas geométricas e escolher a operação desejada.

2.2.1 Algoritmo em uma representação por blocos (fluxograma)

Para que seja possível compreender a ideia das formas geométricas ou nós apresentados na figura 3, desenvolvemos no software Flowgorithm um fluxograma (ou representação por blocos) do algoritmo de troca de pneus, representado graficamente na figura 4 a seguir.

Figura 4 – Fluxograma de um algoritmo para troca de pneus de um carro



Na figura 4, é descrito o algoritmo para o processo de troca de pneu que considera o fato de o estepe estar cheio (se o estepe estiver cheio, execute toda essa sequência do algoritmo). Observe que a sequência lógica está mantida, porém, agora não no formato descritivo, mas, sim, utilizando um fluxograma.

2.3 Pseudocódigo

Observe que, até o momento, o conceito de algoritmo não foi relacionado à programação, mas, sim, a um procedimento para resolver problemas utilizando uma sequência lógica de instruções.

Além da representação narrativa ou textual de um algoritmo, uma outra maneira de representá-lo é em pseudocódigo, o que não vem a ser um código de programação (ainda), porém, contém instruções inteligíveis por humanos e que se assemelha a códigos para computadores.

Trata-se, portanto, de uma linguagem intermediária entre a linguagem de máquina e a linguagem de computador. O pseudocódigo a seguir ilustra o algoritmo para troca de pneu do fluxograma da figura 4.

Quadro 1 – Pseudocódigo do algoritmo para troca de pneu

início
se estepe == "cheio"
escreva ("levantar o carro do lado que está furado")
escreva ("desaparafuse cada um dos parafusos")
escreva ("remova a roda")
escreva ("coloque o estepe no eixo que está sem a roda")
escreva ("parafuse cada um dos parafusos da roda")
escreva ("abaixe o carro")

(cont.)

senão
escreva("levar o carro ao borracheiro")
fim-se
fim

Observe que o pseudocódigo apresenta uma sequência de instruções estruturadas de como o algoritmo deve se comportar. Um pseudocódigo é muito útil em uma fase intermediária de aprendizado, principalmente para traduzir um algoritmo em uma linguagem de programação específica. Também é muito prático para quando tivermos uma ideia do que deve ser feito em um programa, porém, ainda não conhecemos os comandos de uma determinada linguagem.



NA PRÁTICA

Quando desenvolvemos um pseudocódigo, não precisamos nos preocupar com a sintaxe, ou seja, com a forma de escrita das palavras. Observe no exemplo que foram utilizadas algumas palavras como “início”, “se”, “escreva”, “senão”, “fim-se” e “fim”. Estas palavras são encontradas na maioria das linguagens de programação. Logo, compreender o uso delas facilitará no processo de desenvolver códigos de computadores.

Considerações finais

Neste capítulo, compreendemos que os algoritmos são uma sequência lógica e finita de instruções, com início e fim bem definidos, e que esses algoritmos podem ser representados de três modos diferentes: declaração narrativa, fluxograma ou pseudocódigo.

Compreendemos também que, em uma declaração narrativa, o algoritmo é descrito textualmente, como uma receita de bolo ou como um guia explicativo de como montar um móvel. Além disso, vimos que os

fluxogramas são representações gráficas de um algoritmo, que pode ser descrito com o uso de nós ou formas geométricas, as quais cada uma possui uma determinada característica ou aplicação.

Em seguida, tivemos contato com um pseudocódigo, uma declaração em tópicos de um algoritmo, ou seja, uma representação intermediária entre uma descrição narrativa e uma linguagem de programação.

Por fim, estudamos um caso prático, que é o algoritmo para a troca de pneu, e desenvolvemos um fluxograma representando esse algoritmo.

Referências

FEIJÓ, Bruno; CLUA, Esteban; SILVA, Flávio S. C. da. **Introdução à ciência da computação com jogos**: aprendendo a programar com entretenimento. Rio de Janeiro: Campus, 2009.

FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. **Lógica de programação**: a construção de algoritmos e estruturas de dados. 3. ed. São Paulo: Pearson, 2005.

OPENSTREETMAP. Home. **OpenStreetMap**, [s. d.]. Disponível em: <https://www.openstreetmap.org/#map=4/-15.13/-53.19>. Acesso em: 8 nov. 2021.