

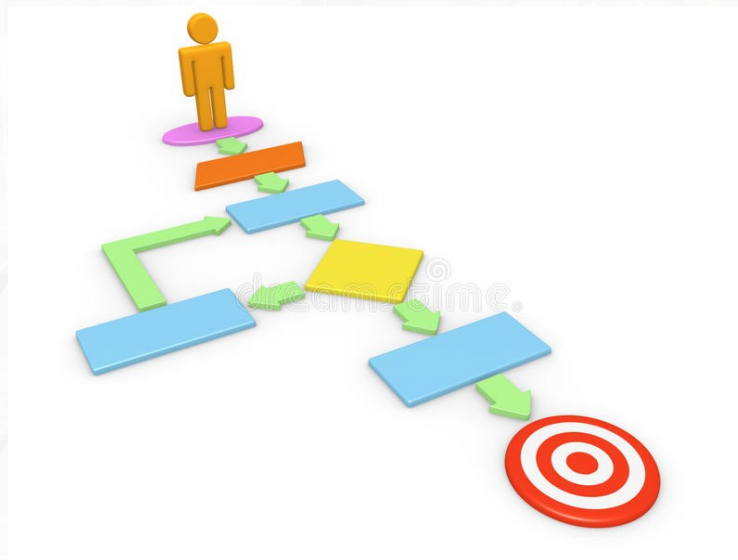
Fundamentos de Programação

Introdução a Algoritmos

Prof. Luiz Fernando Carvalho

luizfcarvalho@utfpr.edu.br

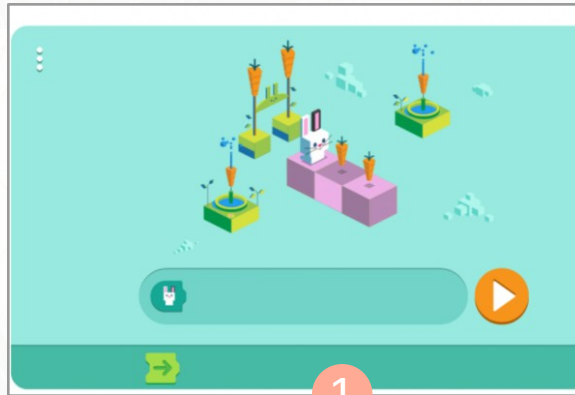
Algoritmos são conjuntos de passos **finitos** e **organizados** que, quando executados, **resolvem um determinado problema**



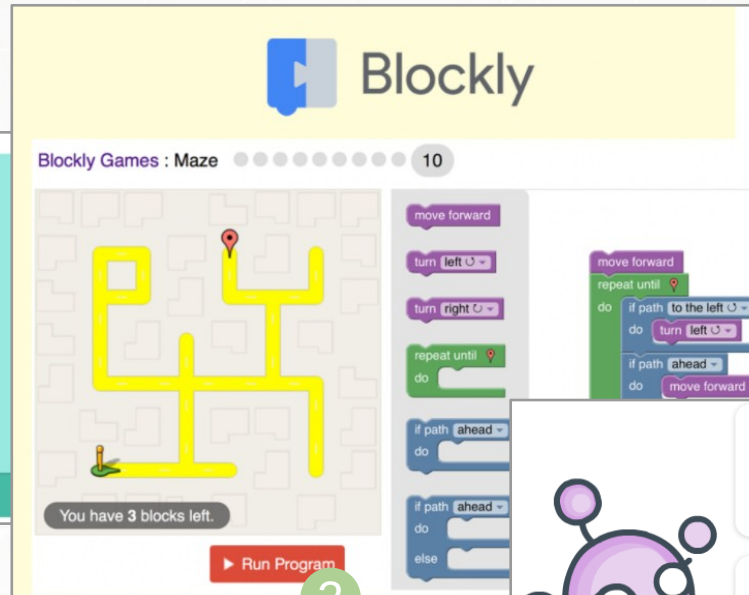
Algoritmos

- Definições:
 - Procedimento passo a passo para a solução de um problema
 - Sequência detalhada de ações a serem executadas para realizar alguma tarefa
 - Conjunto de regras para a solução de um problema
- O termo **algoritmo** não é restrito a computação
- Pode ser aplicado a qualquer circunstância que exija a descrição de uma solução

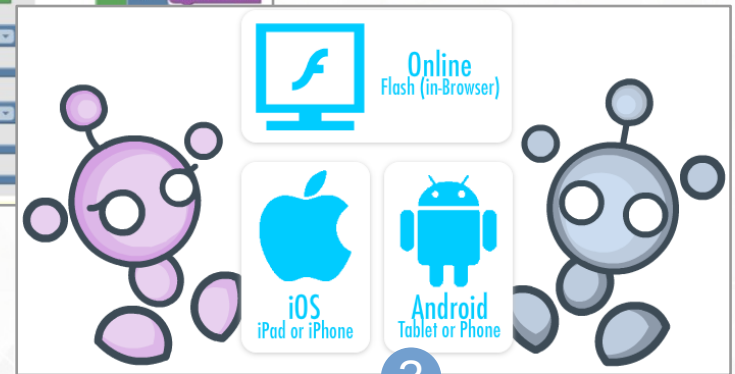
Algoritmos



1



2



3


1 <https://www.google.com/doodles/celebrating-50-years-of-kids-coding?hl=pt-BR>

2 <https://blockly.games>


3 <https://lightbot.com/hour-of-code.html>

Algoritmos


- Um programa de computador é desenvolvido para suprir uma necessidade, ou seja, resolver um problema
 - O problema é conhecido
 - O programador conhece o padrão para resolver o problema
 - Invés de resolver o problema manualmente, desenvolve algoritmos para automatizar essa tarefa




```
Algoritmo AtravessarRua
|   Olhar para a direita
|   Olhar para a esquerda
|   Se estiver vindo carro
|       Não Atravesse
|       Senão
|           Atravesse
|   Fim-Se
Fim-Algoritmo
```



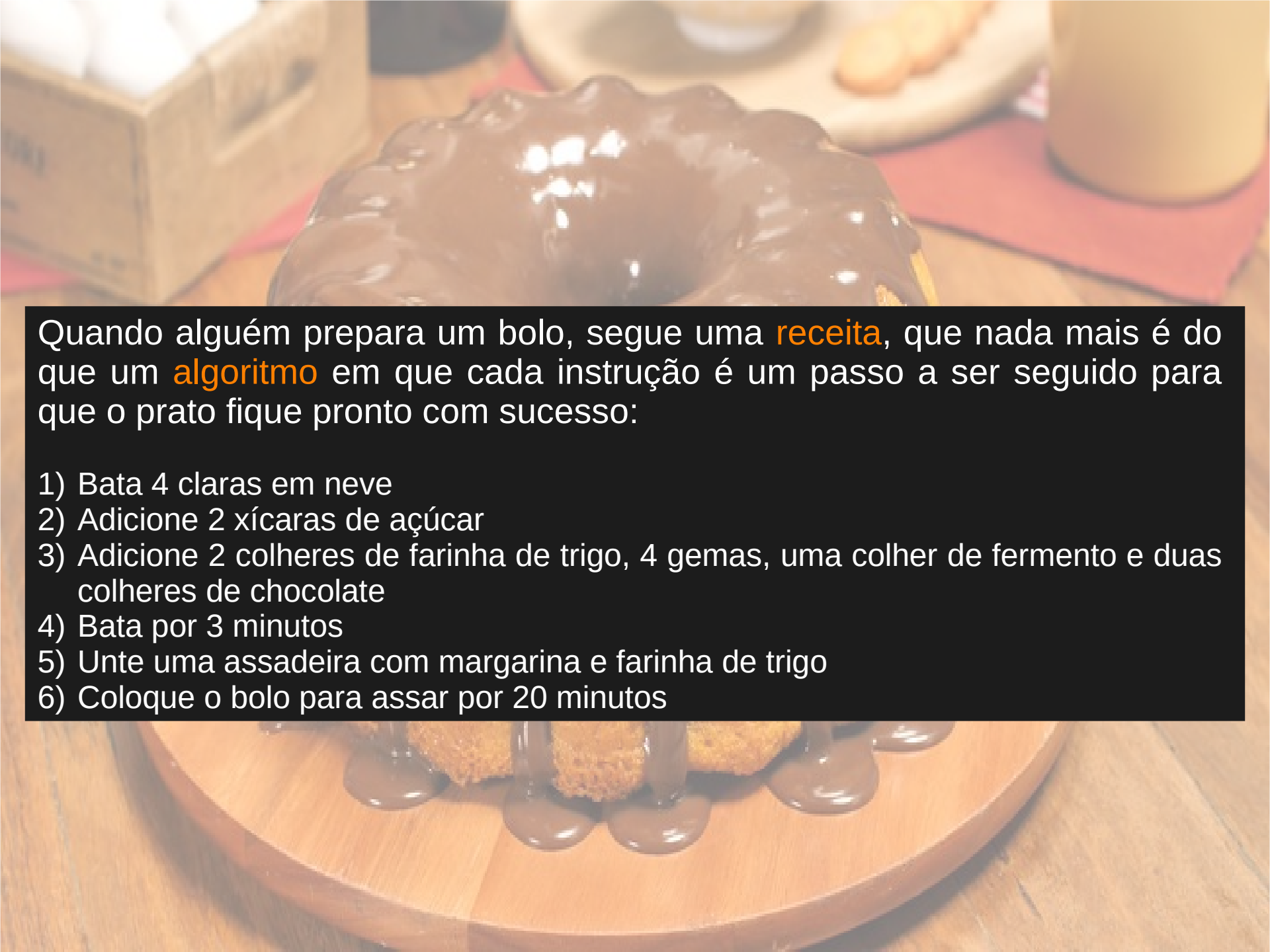
```
Algoritmo AtravessarRua
|   Olhar para a esquerda
|   Olhar para a direita
|   Se não estiver vindo carro
|       Atravesse
|       Senão
|           Não Atravesse
|   Fim-Se
Fim-Algoritmo
```



```
Algoritmo AtravessarRua
|   Olhar para a direita
|   Olhar para a esquerda
|   Se estiver vindo carro
|       Não Atravesse
|   Senão
|       Atravesse
|   Fim-Se
Fim-Algoritmo
```



```
Algoritmo AtravessarRua
|   Atravesse
|   Se estiver vindo carro
|       Olhar para a direita
|   Senão
|       Olhar para a esquerda
|   Fim-Se
|   Não Atravesse
Fim-Algoritmo
```


A close-up photograph of a round chocolate cake sitting on a light-colored wooden plate. The cake is covered in a thick, glossy layer of chocolate. Several dollops of chocolate sauce are drizzled around the base of the cake on the plate. In the background, a cardboard box, a plate with cookies, and a glass of coffee are visible but out of focus.

Quando alguém prepara um bolo, segue uma **receita**, que nada mais é do que um **algoritmo** em que cada instrução é um passo a ser seguido para que o prato fique pronto com sucesso:

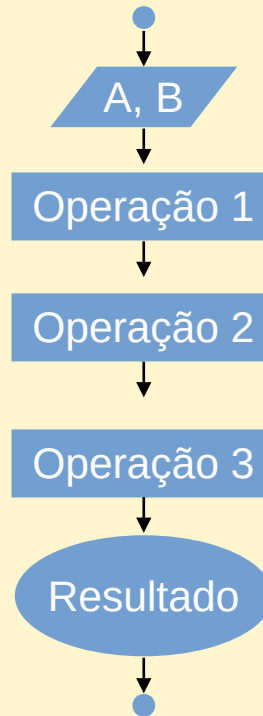
- 1) Bata 4 claras em neve
- 2) Adicione 2 xícaras de açúcar
- 3) Adicione 2 colheres de farinha de trigo, 4 gemas, uma colher de fermento e duas colheres de chocolate
- 4) Bata por 3 minutos
- 5) Unte uma assadeira com margarina e farinha de trigo
- 6) Coloque o bolo para assar por 20 minutos

Tipos de Representação de Algoritmos

- Os três tipos mais utilizados de algoritmos são a descrição narrativa, o fluxograma e o pseudocódigo (portugol)

1. Receber os dados
2. Executar operação 1
3. Executar operação 2
4. Executar operação 3
5. Mostrar o resultado

Narrativa



Fluxograma

```
Algoritmo Operacoes
Início
    Inteiro: A, B, resultado;
    Operacao 1;
    Operacao 2;
    Operacao 3;
    escreva(resultado);
Fim
```

Portugal

Descrição narrativa

- A descrição narrativa consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (por exemplo, a língua portuguesa), os passos que devem ser seguidos para a resolução do problema

Vantagem

Não é necessário aprender nenhum conceito novo, pois a língua natural, o português, já é conhecida.

Desvantagem

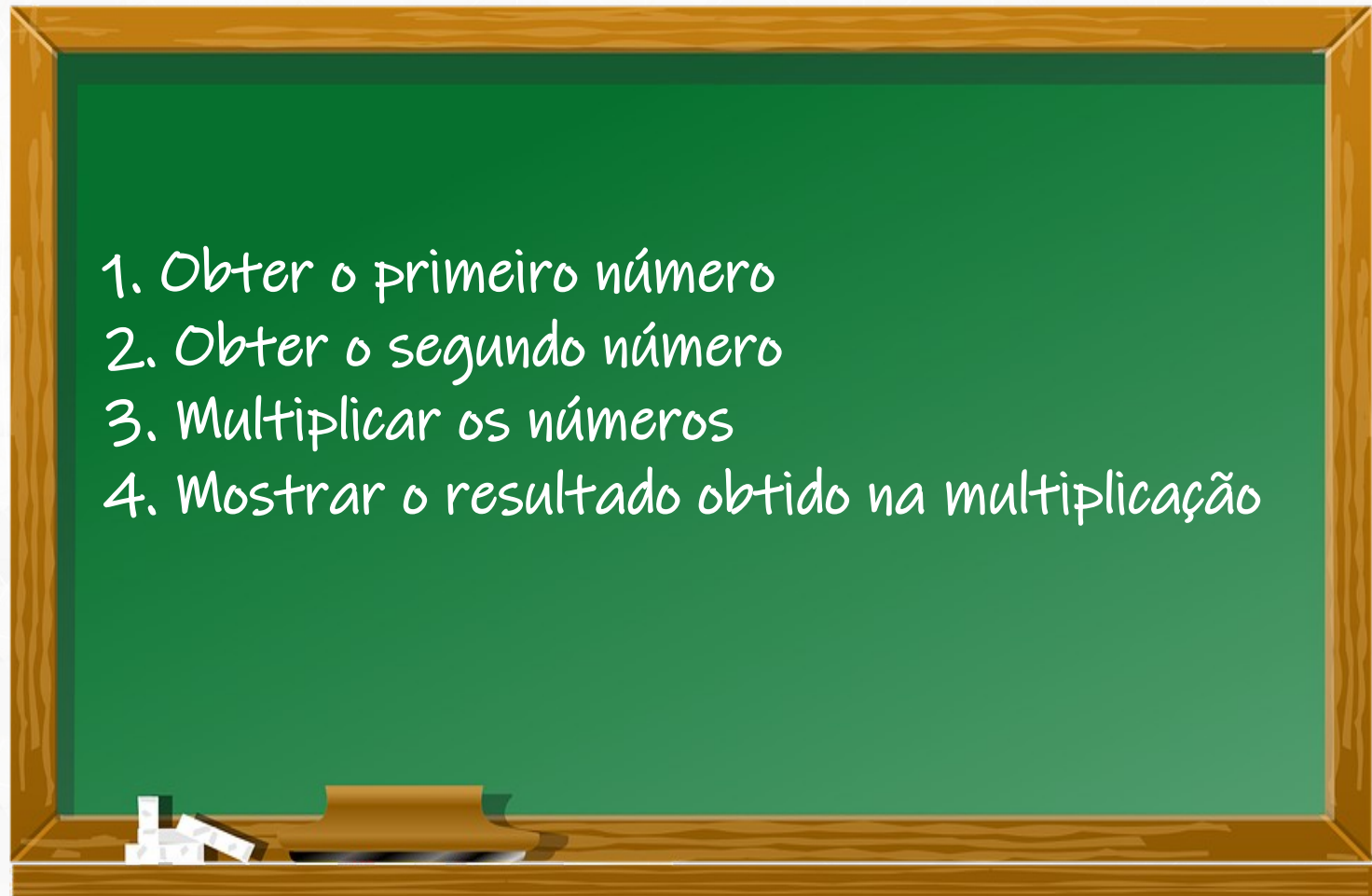
A língua natural abre espaço para diferentes interpretações, ambiguidades, o que posteriormente dificultará a transição deste algoritmo para o programa.

Dicas para construção de algoritmos

- Para a construção de qualquer tipo de algoritmo são necessários os seguintes passos:
 - 1) Definir o processamento, ou seja, quais cálculos serão executados e as suas restrições. O processamento é responsável pela transformação dos dados de entrada em informações de saída
 - 2) Definir os dados de entrada
 - 3) Ler atentamente o enunciado do problema, destacando os pontos mais importantes
 - 4) Definir os dados de saída, ou seja, o que será gerado após o processamento
 - 5) Construir o algoritmo
 - 6) Testar o algoritmo realizando simulações

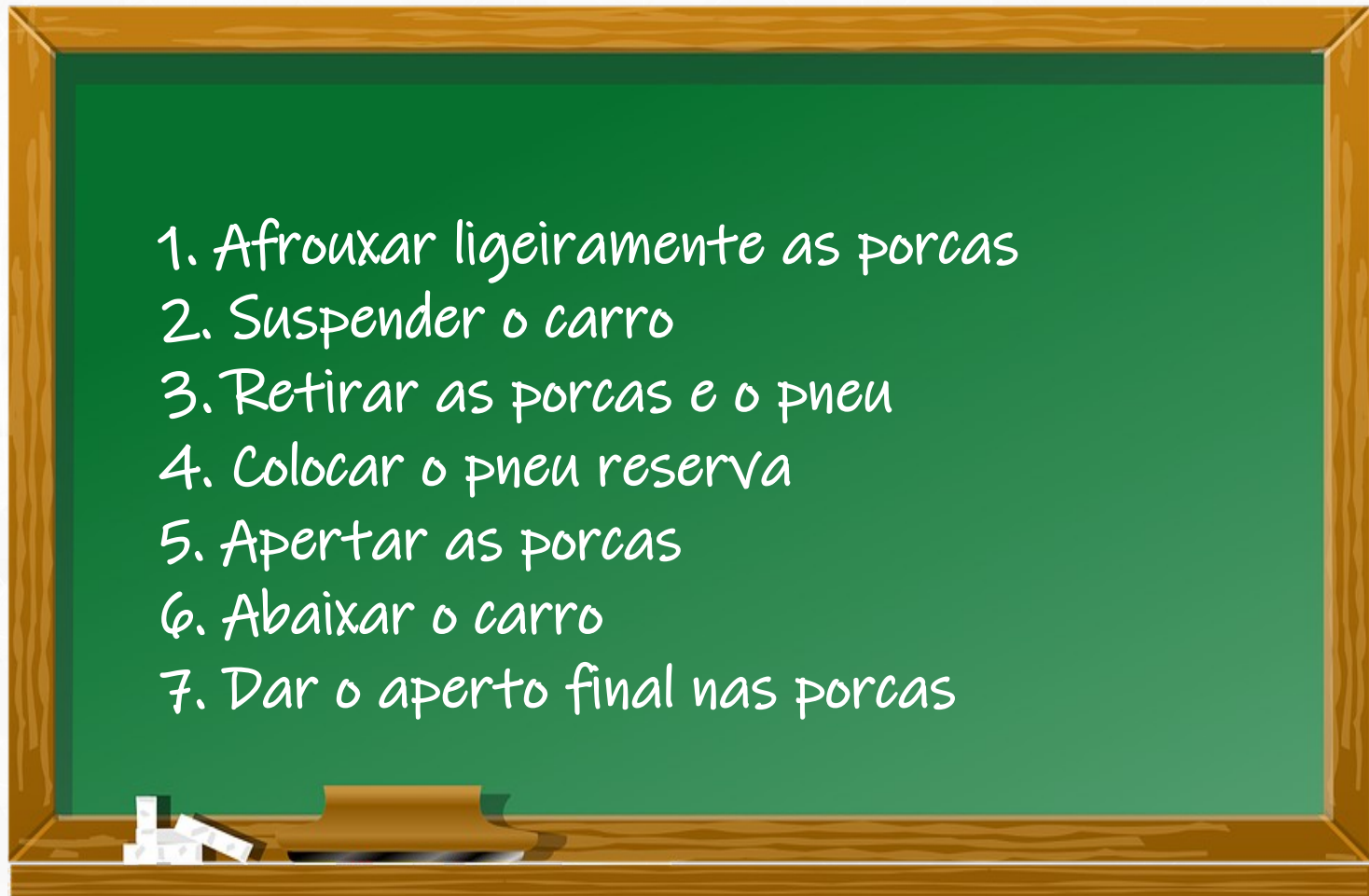
Descrição narrativa

- Exemplo: Algoritmo para multiplicação de dois números



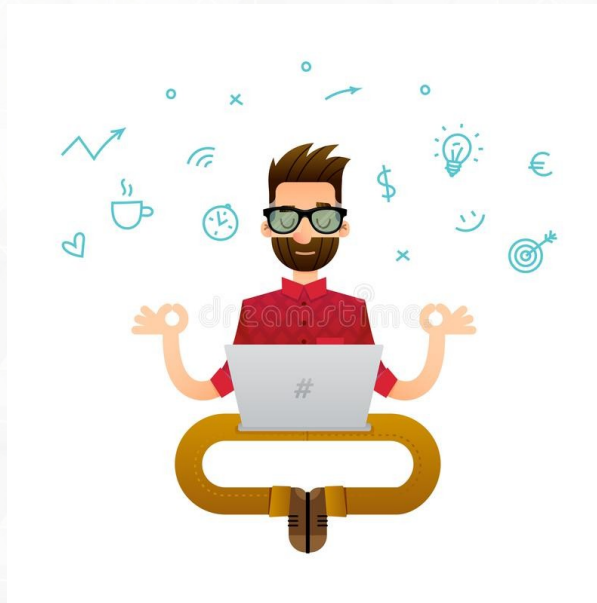
Descrição narrativa

- Exemplo: Algoritmo para troca de pneu de um automóvel



Porque construir algoritmos

- O problema é conhecido
- O programador conhece o padrão para resolver o problema
- Ao invés de resolver o problema manualmente, algoritmos podem ser executados por máquinas para automatizar a resolução da tarefa



Exercícios

- 1) Faça um algoritmo de como fazer café.
- 2) Faça um algoritmo para calcular e apresentar a média final de um aluno, sabendo que a disciplina tem 4 bimestres.
- 3) Faça um algoritmo para verificar se um dado número natural N é par.