

3 ALGORITMOS

3.1 O QUE É?

Um algoritmo é uma linguagem intermediária entre a linguagem humana e a linguagem de programação. É utilizado para representar a solução de um problema. Deve descrever as instruções que serão executadas pelos computadores.

Podemos definir algoritmos como:

*A especificação de uma **sequência ordenada de instruções, finitas e não ambíguas**, que deve ser seguida para a solução de um determinado problema **garantindo a sua repetibilidade**.*

No dia a dia usamos o conceito de algoritmo constantemente sempre que estabelecemos uma rotina, ou quando criamos um plano para executar algumas tarefas. Esse conceito de algoritmo está intrínseco no ser humano e por vezes nem percebemos que possuímos essa habilidade, mas é essa habilidade que vai definir a qualidade dos códigos a serem executadas.



Exemplos de algoritmos no dia a dia

- Receitas culinárias;
- Manuais de instrução;
- Roteiros realização de tarefas específicas.

Um dos vários exemplos do uso de algoritmos no nosso dia a dia são as receitas culinárias, pois estas possuem um conjunto de passos que devem ser seguidos para obter o resultado esperado. Vamos analisar a receita do ponto de vista de um programador:

Receita de Brigadeiro

1. Separar os ingredientes:

1 lata de leite condensado
 1 colher de sopa de manteiga
 4 colheres de sopa de chocolate em pó
2. Colocar todos os ingredientes em uma panela;
3. Misturar os ingredientes;
4. Cozinhar a mistura em fogo médio até começar a soltar do fundo da panela.
5. Desligar o fogo;
6. Colocar o brigadeiro em refratário de vidro;
7. Esperar o brigadeiro esfriar;
8. Enrolar o brigadeiro em formato esférico;
9. Passar o brigadeiro enrolado no granulado;
10. Colocar o brigadeiro na forminha de papel.

- (1). É uma lista ordenada, possui 10 passos para ser executado.
- (2). As instruções são finitas, ou seja, tem começo e fim para ser executada
- (3). Não são ambíguas, ou seja, não dúvidas do que deve ser feito e em qual momento deve ser realizado.

3.2 PARA QUE SERVE UM ALGORITMO?

Podemos dividir nos seguintes tópicos:

- O algoritmo é uma sequência de passos lógicos e finitos que permite solucionar problemas;
- O objetivo de aprender a criar algoritmos é que este é a base de conhecimentos para as linguagens de programação;
- Em geral, existem muitas maneiras de resolver o mesmo problema. Ou seja, podem ser criados vários algoritmos diferentes para resolver o mesmo problema;

- Assim, ao criarmos um algoritmo, indicamos uma dentre várias possíveis sequências de passos para solucionar o problema.

3.3 REPRESENTAÇÃO DE ALGORITMOS

Os algoritmos podem ser representados de várias formas diferentes, as mais conhecidas são:

Descrição Narrativa: são expressos em linguagem natural, ou seja, a sequência de passos é descrita em nossa língua nativa. (Língua Portuguesa)

Abaixo temos um exemplo de uso da Descrição Narrativa e

Exemplo:

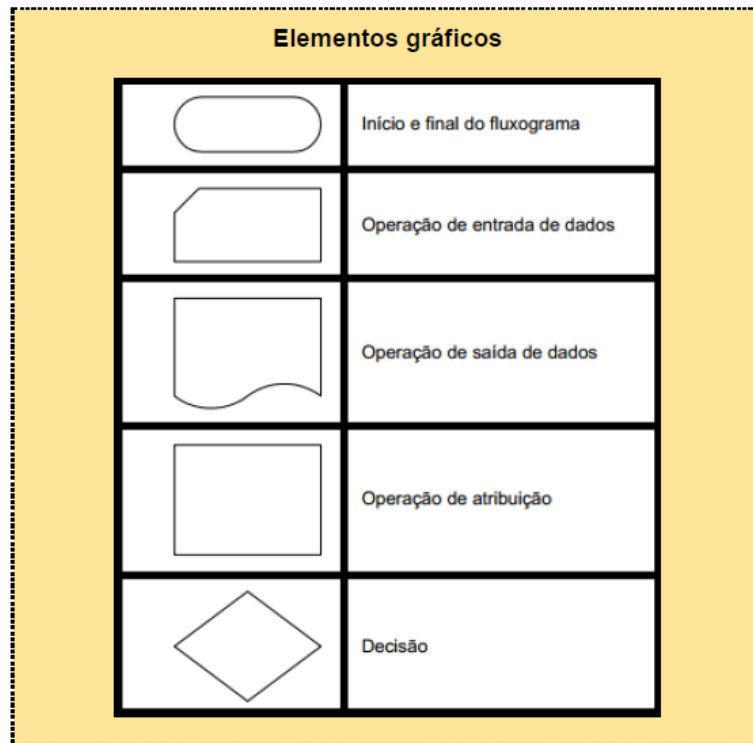
- Cálculo da média de um aluno:

- Obter as suas 2 notas de provas;
- Calcular a média aritmética;
- Se a média for maior ou igual a 7, o aluno foi aprovado;
- Senão o aluno foi reprovado.

Aspecto positivo	Aspecto negativo
Não é necessário aprender novos conceitos, pois a língua natural já é bem conhecida.	A língua natural dá oportunidade para várias interpretações e ambiguidades, dificultando a transcrição desse algoritmo para programa.

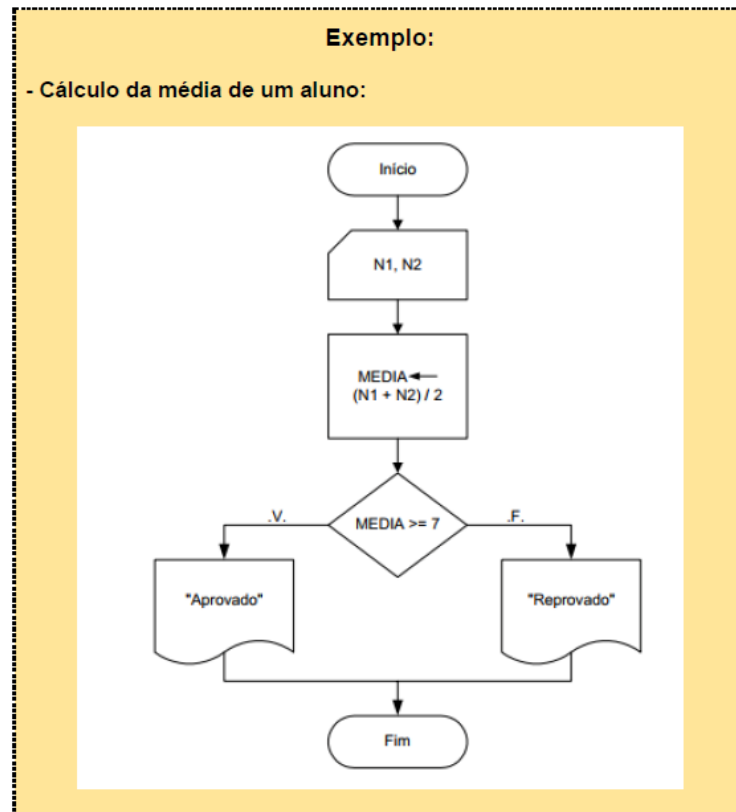
Fluxograma: É uma representação gráfica em que formas geométricas diferentes implicam ações (instruções, comandos) distintas. É mais precisa que a Descrição Narrativa, porém não se preocupa com detalhes de implementação do programa, como o tipo das variáveis utilizadas.

A seguir temos alguns exemplos de símbolos que podem ser utilizados na montagem de fluxogramas que ajudem a criar o programa em questão:



O fluxograma utiliza símbolos específicos para a representação gráfica dos algoritmos. Os símbolos podem sofrer algumas variações de acordo com o autor ou ferramenta em uso. Não existe uma norma ou regra geral para uso dos fluxogramas, mas isso deve ser levado em consideração, principalmente se você estiver trabalhando em equipe dentro de um projeto ou empresa.

Vamos voltar novamente ao nosso exemplo de cálculo da média escolar.



Aspecto positivo	Aspecto negativo
O entendimento de elementos gráficos é mais simples que o entendimento de textos.	Os fluxogramas devem ser entendidos e o algoritmo resultante não é detalhado, dificultando sua transcrição para um programa.

Pseudocódigo (Linguagem estruturada ou Portugal): É rico em detalhes, como a definição dos tipos das variáveis usadas no algoritmo. Costuma já estar próximo da linguagem a ser programada.

Exemplo:

- Cálculo da média de um aluno:

Algoritmo Calculo_Media

Var Nota1, Nota2, MEDIA: real;

Início

Leia Nota1, Nota2;

 MEDIA \leftarrow (Nota1 + Nota2) / 2;

Se MEDIA \geq 7 **então**

Escreva "Aprovado";

Senão

Escreva "Reprovado";

Fim_se

Fim

Aspecto positivo	Aspecto negativo
Representação clara sem as especificações de linguagem de programação.	As regras do pseudocódigo devem ser aprendidas.

3.4 EXEMPLOS PRÁTICOS

3.4.1 PSEUDOCÓDIGO 1 – CALCULAR A ÁREA

Pseudocódigo 1 - Calcular a Área de Um Paralelograma

Nota: Um paralelogramo é um quadrilátero com lados opostos paralelos (e portanto ângulos opostos iguais). Um quadrilátero com lados iguais é chamado de losango e um paralelogramo cujos ângulos são todos ângulos retos é chamado de retângulo.

Inicie

```
Exiba "Bem-vindo ao Calculador de Área de Paralelogramo"
Peça para o usuário inserir o comprimento da base
Armazene o comprimento da base em uma variável
Peça para o usuário inserir a altura
Armazene a altura em uma variável
Calcule a área do paralelogramo: base * altura
Armazene o resultado em uma variável
Exiba o resultado
```

Fim

3.4.2 PSEUDOCÓDIGO 2 – CRIAR UMA CALCULADORA SIMPLES

Pseudocódigo 2 - Calculadora Simples

Inicie

```
Exiba "Bem-vindo à Calculadora"
Peça para o usuário inserir o primeiro número
Armazene o primeiro número em uma variável
Peça para o usuário inserir o segundo número
Armazene o segundo número em uma variável
Peça para o usuário selecionar uma operação (+, -, *, /)
Armazene a operação em uma variável
Utilize a operação selecionada e os números armazenados para realizar o cálculo
Exiba o resultado
```

Fim

3.4.3 PSEUDOCÓDIGO 3 – ALGORITMO BUBBLE SORT

Pseudocódigo 3 - Algoritmo Bubble Sort

Bubble Sort é um algoritmo de ordenação simples que funciona comparando cada elemento com o próximo, e trocando-os de lugar se eles estiverem em ordem incorreta. O algoritmo repete esse processo várias vezes, até que todos os elementos estejam ordenados. A cada passagem, o maior elemento "flutua" para o final do array, como uma bolha, dando origem ao nome do algoritmo.

Inicie

```
Para cada elemento i no array de tamanho n
  Para cada elemento j no array de tamanho n - 1
    Se elemento i for maior que elemento j
      Troque os elementos i e j
  Exiba o array ordenado
```

Fim

3.5 EXERCÍCIOS – LISTA 1

Faça uma cópia do arquivo (OUT2023T – Lista 1) e comece a responder pelo notebook. Ao final, salve o arquivo em PDF conforme orientado no item 2.7.10 e envie o mesmo na atividade correspondente do Google Classroom.