

Linguagem de Programação

Módulo 5



Universidade Federal do Rio Grande do Norte – UFRN

Escola Agrícola de Jundiaí – EAJ

Profa. Alessandra Mendes Pacheco

Pensamento Computacional

- Estratégia usada para desenhar soluções e solucionar problemas de maneira eficaz tendo a tecnologia como base.

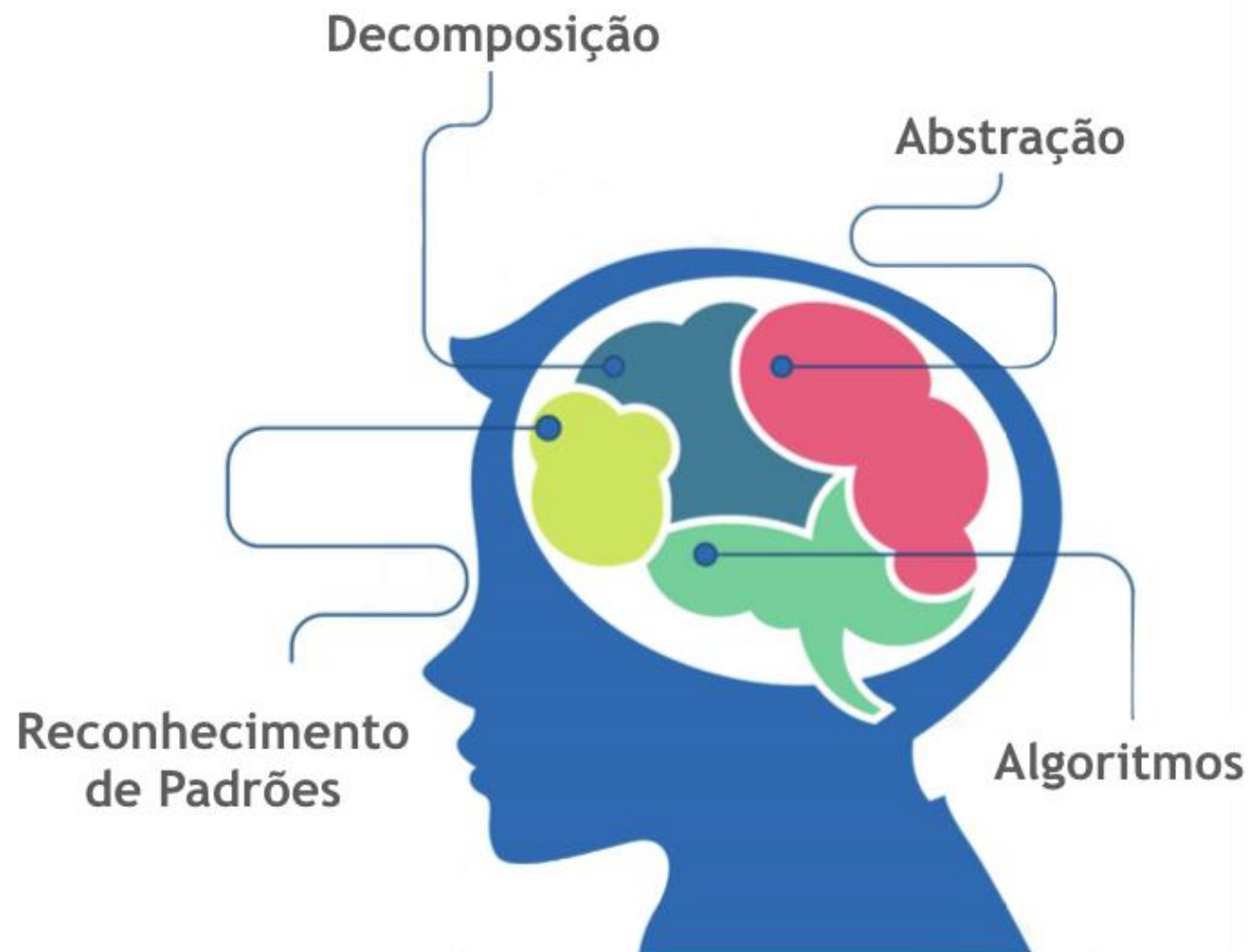
“The mental activity for abstracting problems and formulating solutions that can be automated.”

(Yadav et al., 2014)

- “Distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação nas mais diversas áreas do conhecimento, com a finalidade de **identificar e resolver problemas** colaborativamente através de passos claros de tal forma que uma **pessoa ou uma máquina** possam executá-los eficazmente”. (BRACKMANN, 2017)
- PC no Brasil: Christian Brackmann (professor de algoritmos do IFFar)
 - Atividades desplugadas na educação básica
 - Projeto aplicado em escolas do Brasil, aplicação na BNCC (Base Nacional Comum Curricular)
 - <https://www.youtube.com/watch?v=Bxg8QC93joo>

Pensamento Computacional

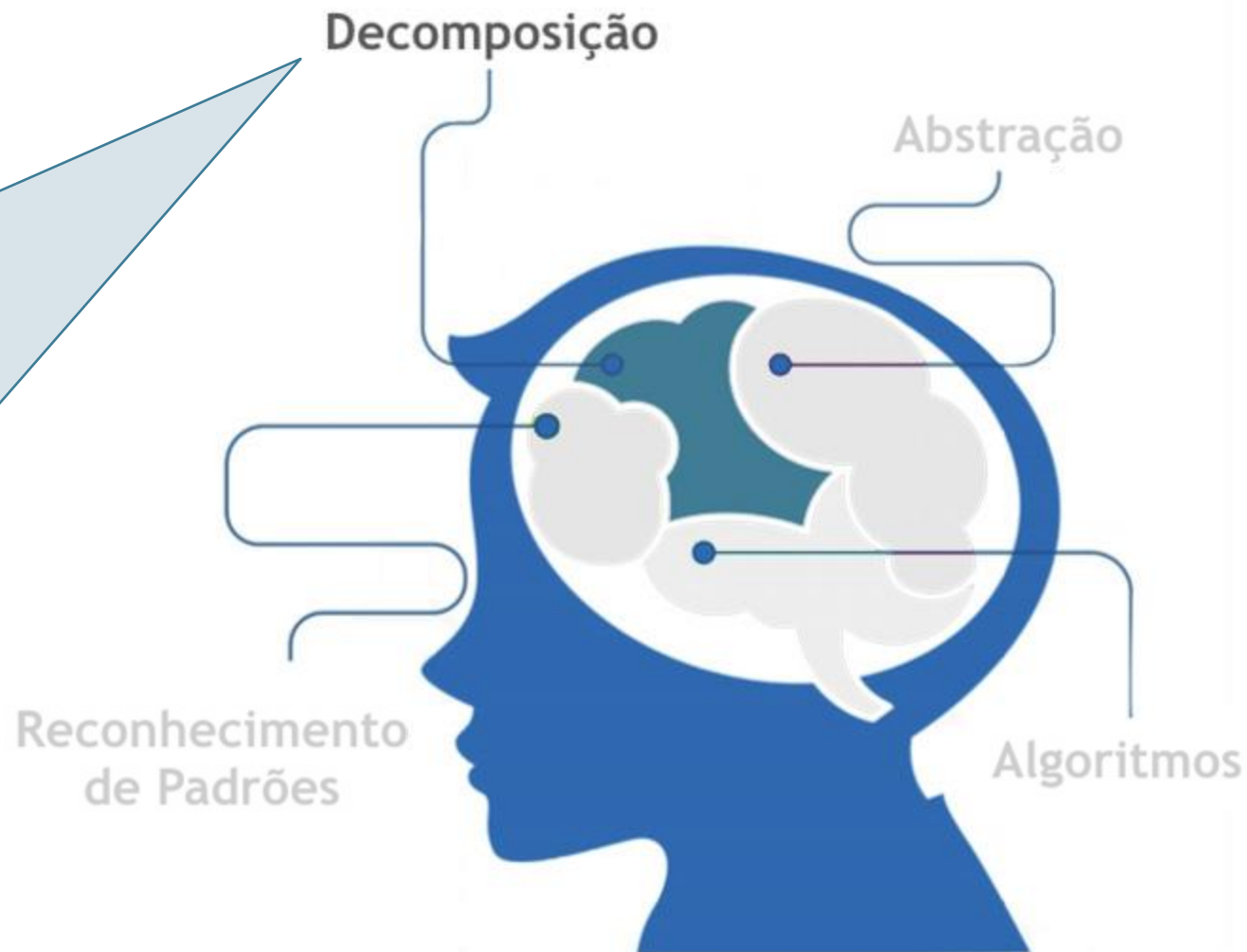
- Pilares:



Pensamento Computacional

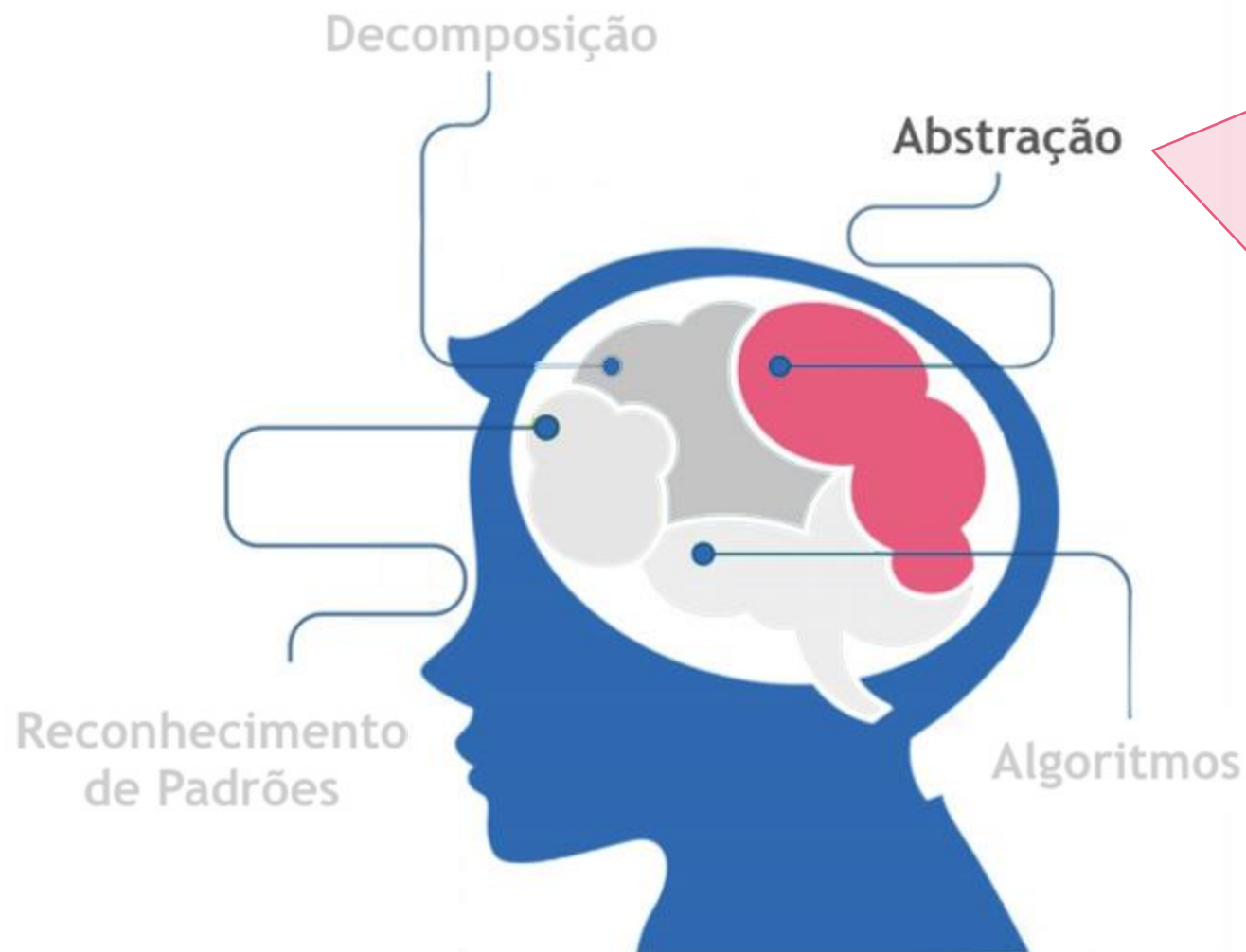
- Pilares:

Habilidade de **dividir um problema complexo em partes menores**, facilitando a solução e permitindo ainda maior atenção a cada etapa.



Pensamento Computacional

- Pilares:



Propõe o **foco em processos relevantes** em vez de priorizar os detalhes, de modo que a solução possa ser válida para outros problemas.

Pensamento Computacional

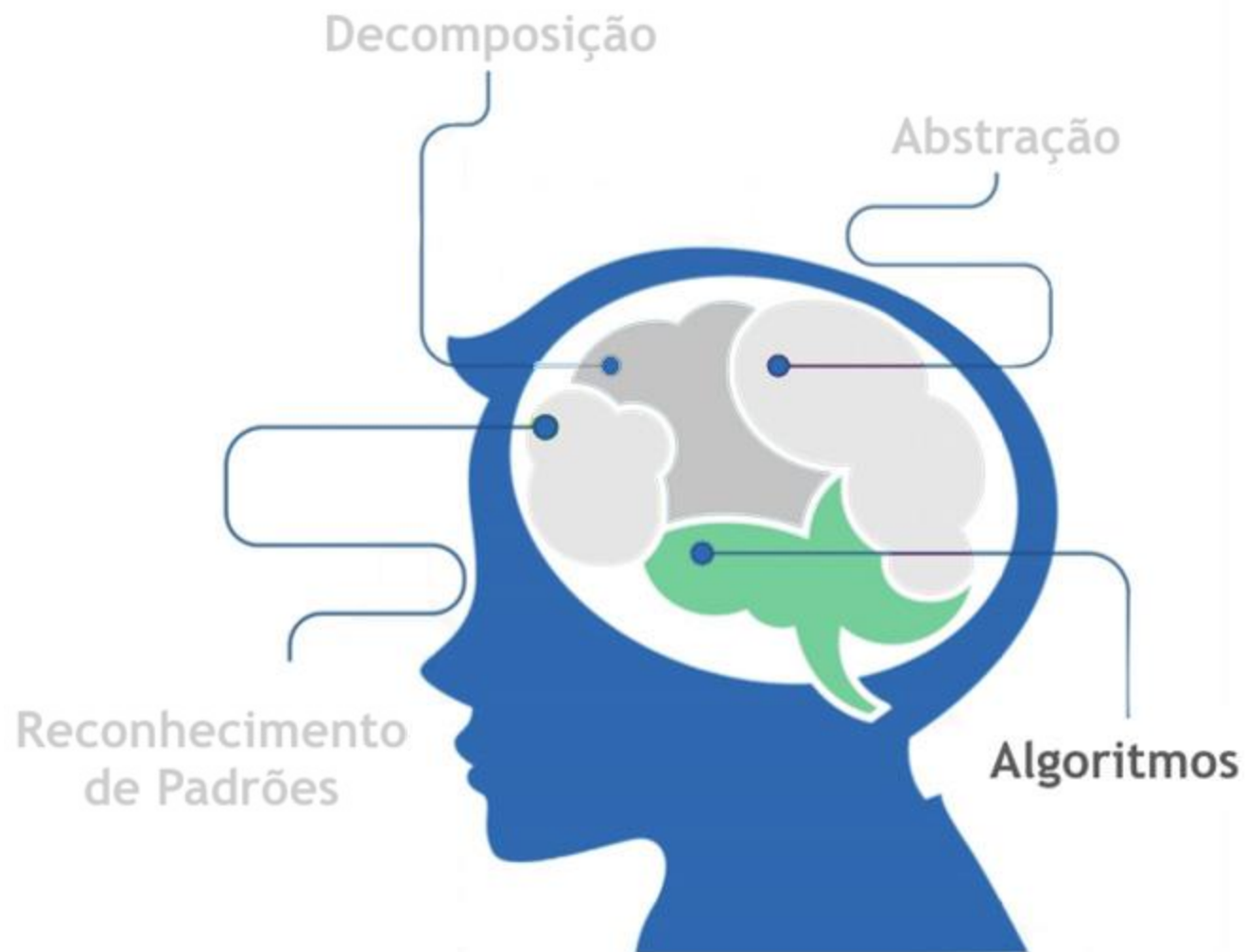
- Pilares:

Identificação de **aspectos comuns** nos processos, reconhecendo padrões e similaridade. Permite a construção de soluções para problemas comuns.



Pensamento Computacional

- Pilares:



Criação de **passos e soluções** para alcançar um **objetivo específico** para qualquer problema, seja de ordem matemática ou não.

Pensamento Computacional

- Pilares:

Planejando um novo videogame

Fundo



Elementos

Elemento 1



Elemento 2



Decomposição

Abstração

Reconhecimento
de Padrões

Algoritmos

Música



Sons



Pensamento Computacional

- Pilares:

Planejando um novo videogame

Fundo



Reconhecimento de Padrões

Decomposição

Abstração

Algoritmos

Elementos

Elemento 1



Elemento 2



Pensamento Computacional

- Pilares:

Planejando um novo videogame



Pensamento Computacional

■ Pilares:

Planejando um novo videogame

Fundo



```
Se fase1() então  
  rotina1()  
Senão  
  rotina2()
```

Decomposição

Abstração

Elemento 1



```
Se tocoubola() então  
  chute()  
Senão  
  aguarde()
```

Reconhecimento
de Padrões

Algoritmos

Pensamento Computacional

- Problema:

Qual a soma de todos os números entre 1 e 200?

$$1+2+3+4+5+6+\dots+200=?$$

- Solução?

$$1+2 = 3$$

$$3+4 = 7$$

$$5+6 = 11$$

$$7+8 = 15 (\dots)$$

Soma os resultados...



Pensamento Computacional

- Problema:

Qual a soma de todos os números entre 1 e 200?

- Decomposição

$$1+200 = 201$$

$$2+199 = 201$$

$$3+198 = 201 (...)$$

$$101+100 = 201$$

Soma por pares



Pensamento Computacional

- Problema:

Qual a soma de todos os números entre 1 e 200?

- Abstração

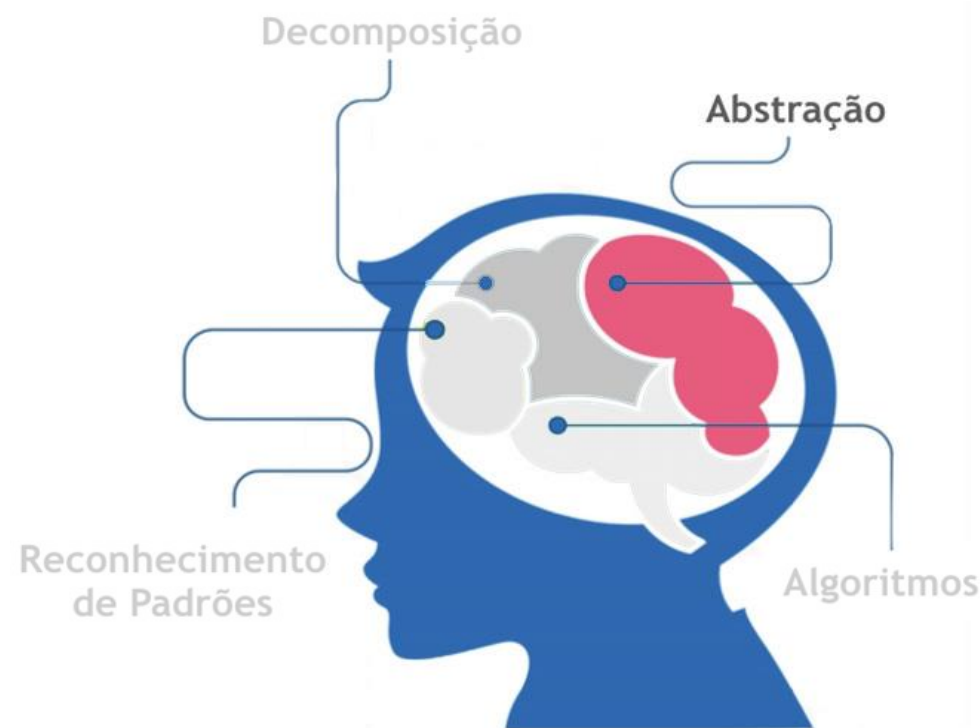
$$1+200 = 201$$

$$2+199 = 201$$

$$3+198 = 201 (...)$$

$$101+100 = 201$$

Soma de cada par = 201



Pensamento Computacional

- Problema:

Qual a soma de todos os números entre 1 e 200?

- Reconhecimento de Padrões

$$1+200 = 201$$

$$2+199 = 201$$

$$3+198 = 201 (...)$$

$$101+100 = 201$$

Repetições: $200/2 = 100$



Pensamento Computacional

- Problema:

Qual a soma de todos os números entre 1 e 200?

- Algoritmos

Passo 1:

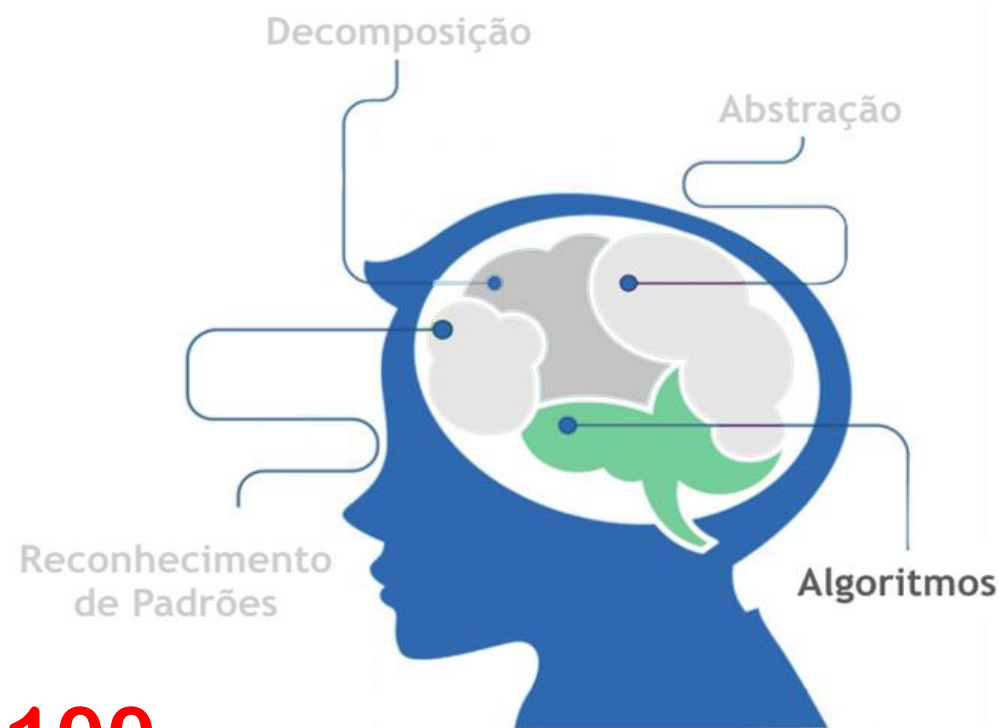
-Soma de cada par (201)

Passo 2:

-Total de pares (100)

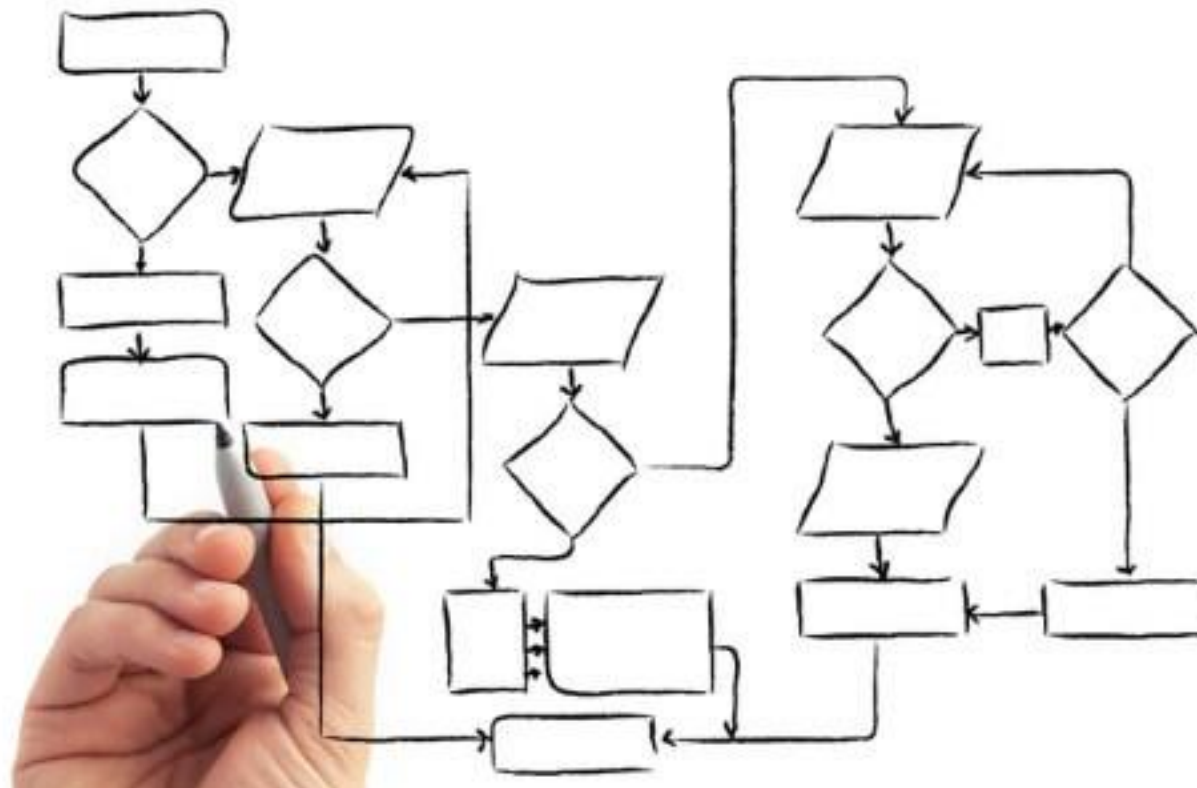
Passo 3:

-Multiplique $(201 * 100) = 20.100$



■ Algoritmos, o que são?

Conjunto de passos sequenciais e finitos para atingir um objetivo específico.



Algoritmos

- Um algoritmo é simplesmente uma "**receita**" para a execução de uma tarefa ou resolução de um problema.
- Como toda receita, um algoritmo também deve ser **finito**.
- Um algoritmo é uma **sequência de raciocínios**, instruções ou operações para alcançar um **objetivo**, ou seja, representa uma solução para um problema.

Compute-it

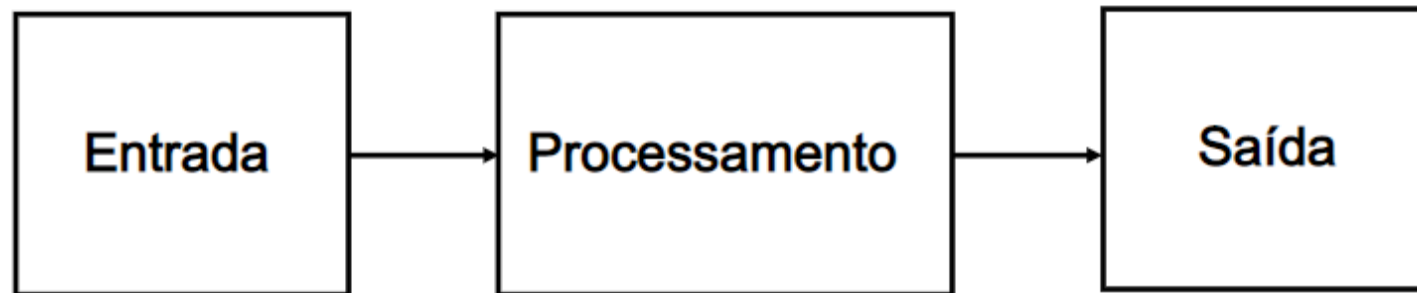
Entendendo um algoritmo...

<http://compute-it.toxicode.fr/>



Algoritmos

- Um algoritmo conta com a **entrada** (input) e **saída** (output) de informações mediadas pelas **instruções**. Assim, todo problema a ser codificado deve ser dividido em 3 fases:
 - **Entrada**: são os dados e informações iniciais do problema
 - **Processamento**: são os procedimentos utilizados para solucionar o problema
 - **Saída**: são os dados e informações resultantes do processamento.



Algoritmos

■ Características de um algoritmo

ENTRADA

→ Ingredientes da receita, dados e informações necessários para o resultado final.

PROCESSAMENTO

→ Modo de fazer, passo-a-passo, conjunto de instruções necessárias para que o objetivo seja atingido.

SAÍDA

→ O produto final, o bolo.

BOLO DE FUBÁ

INGREDIENTES:

- 2 XÍCARAS DE AÇÚCAR
- 3 GEMAS
- 3 CLARAS BATIDAS EM NEVE
- 1 XÍCARA DE ÓLEO
- 2 XÍCARAS DE FARINHA DE TRIGO
- 1 XÍCARA DE FUBÁ
- 1 COPO DE LEITE
- 1 COLHER DE FERMENTO

MODO DE FAZER:

COLOQUE O AÇÚCAR E A GEMA NA BATEDEIRA E BATA ATÉ FICAR CLARO, EM SEGUIDA ACRESCENTE O ÓLEO E DESLIGUE A BATEDEIRA. VÁ COLOCANDO AOS POUCOS A FARINHA E O FUBÁ, MEXENDO COM UMA COLHER. ACRESCENTE O LEITE, AS CLARAS EM NEVE E, POR ÚLTIMO, O FERMENTO. DESPEJE NUMA ASSADEIRA UNTADA E ASSE EM FORNO PRÉ-AQUECIDO.



Algoritmos

■ Formas de Representação

- Um algoritmo pode ser representado como **descrição narrativa, fluxograma ou pseudocódigo**.
- **Descrição narrativa**: utiliza a linguagem natural para especificar os passos para a realização das tarefas.

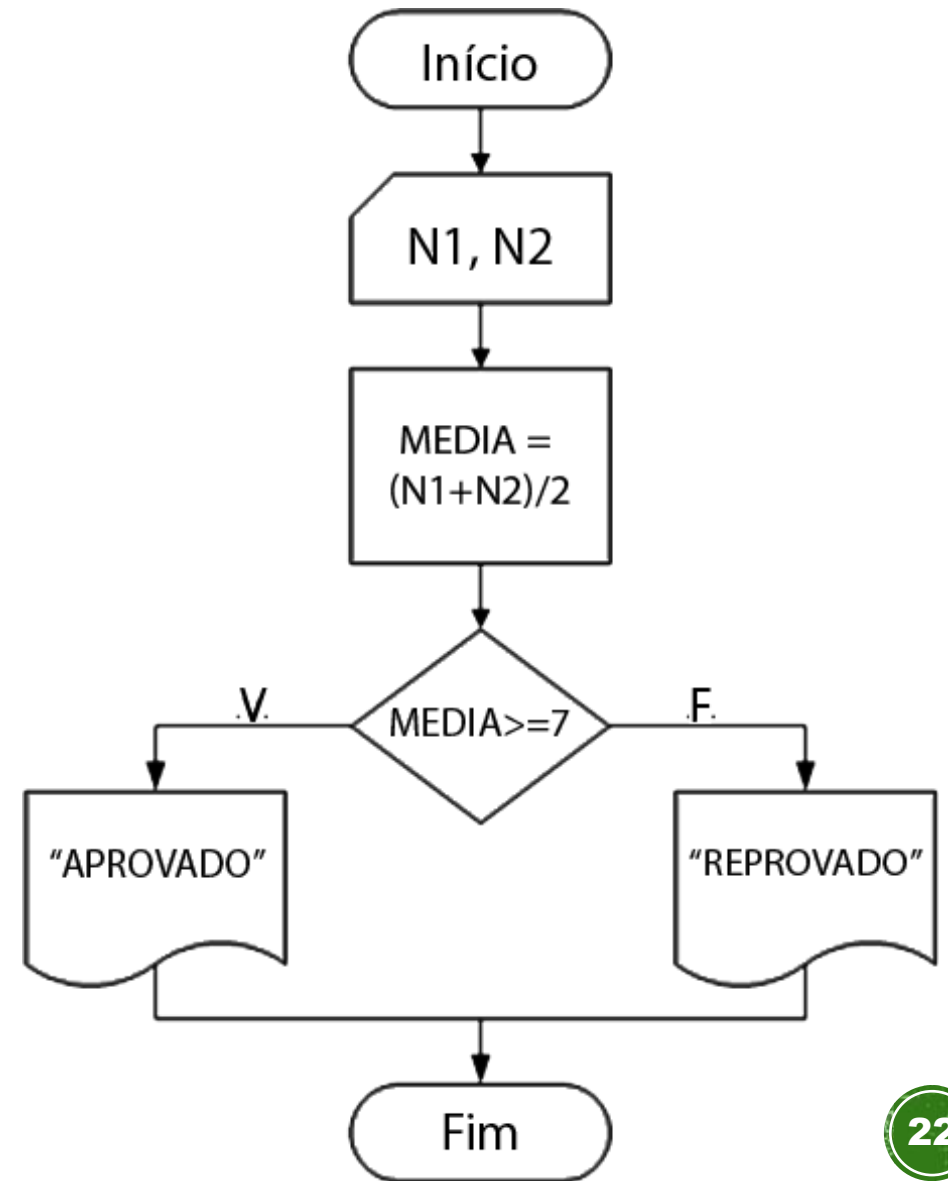
■ Exemplo: Trocar uma lâmpada

- 1: Pegar uma escada
- 2: Posicionar a escada embaixo da lâmpada
- 3: Buscar uma lâmpada nova
- 4: Subir na escada com a lâmpada nova
- 5: Retirar a lâmpada velha
- 6: Colocar a lâmpada nova



Algoritmos

- **Formas de Representação**
- **Fluxogramas:** utiliza-se figuras geométricas para ilustrar os passos a serem seguidos na resolução de problemas.
- Exemplo: Cálculo da média de duas notas
 - Entrada: duas notas (N1 e N2)
 - Processamento: calcular a média aritmética $(N1+N2)/2$ e verificar se a média é maior ou igual a 7.
 - Saída: “Aprovado” (se a condição for verdadeira) ou “Reprovado” (se a condição for falsa).



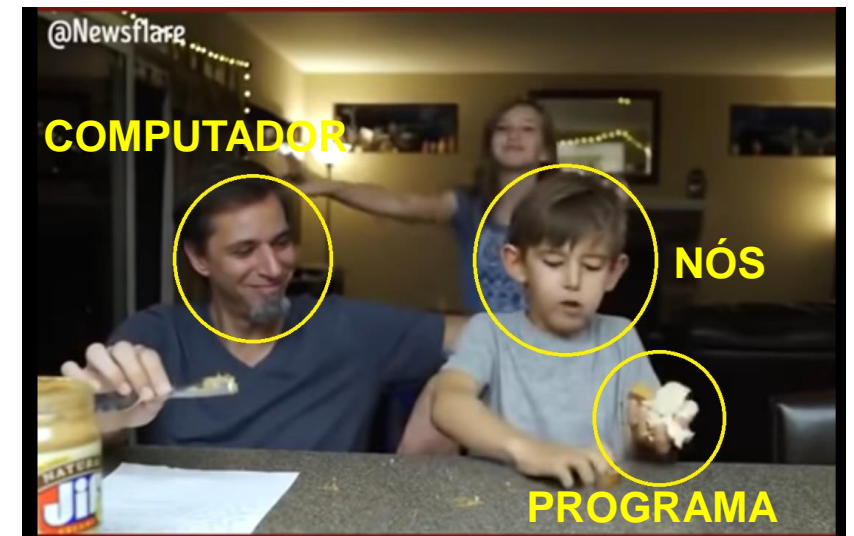
Algoritmos

- **Formas de Representação**
- **Pseudocódigo:** forma de representação de algoritmos rica em detalhes. Utiliza linguagem estruturada e se assemelha, na forma, a um programa escrito na linguagem de programação.
- É um tipo de algoritmo que utiliza uma linguagem flexível intermediária entre a linguagem natural e a linguagem de programação.
 - Pseudocódigo significa “falso código”.

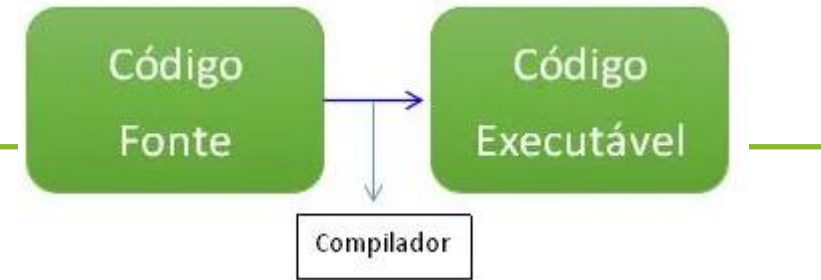
```
Algoritmo Media;  
    Var N1, N2, MEDIA: real;  
Início  
    Leia (N1, N2);  
    MEDIA ← (N1 + N2) / 2;  
    Se MEDIA >= 7 então  
        Escreva "Aprovado"  
    Senão  
        Escreva "Reprovado";  
    Fim_se  
Fim
```

Algoritmos

- **Formas de Representação**
- **Pseudocódigo:** forma de representação de algoritmos rica em detalhes. Utiliza linguagem estruturada e se assemelha, na forma, a um programa escrito na linguagem de programação.
- É um tipo de algoritmo que utiliza uma linguagem flexível intermediária entre a linguagem natural e a linguagem de programação.
 - Pseudocódigo significa “falso código”.
- Vídeo disponível em <https://www.youtube.com/watch?v=pdhqwbUWf4U>



Estrutura Geral



- **O papel da linguagem de programação**
- Um algoritmo escrito em linguagem natural passa a ser chamado de **programa de computador** depois de convertido para uma linguagem aceita por um computador real.
- As linguagens de alto nível são as linguagens de programação que utilizam palavras completas e estruturas de expressão semelhantes ao linguajar humano para informar ao computador quais funções ele deverá realizar.
- Após escrito em uma linguagem de programação, o computador deverá converter os comandos dados em linguagem de alto nível (código-fonte) para linguagem de máquina (executável ou aplicativo).
- Esta tarefa de conversão é feita por um programa especial de computador, chamado Compilador

Estrutura Geral

- **A linguagem de programação C++**

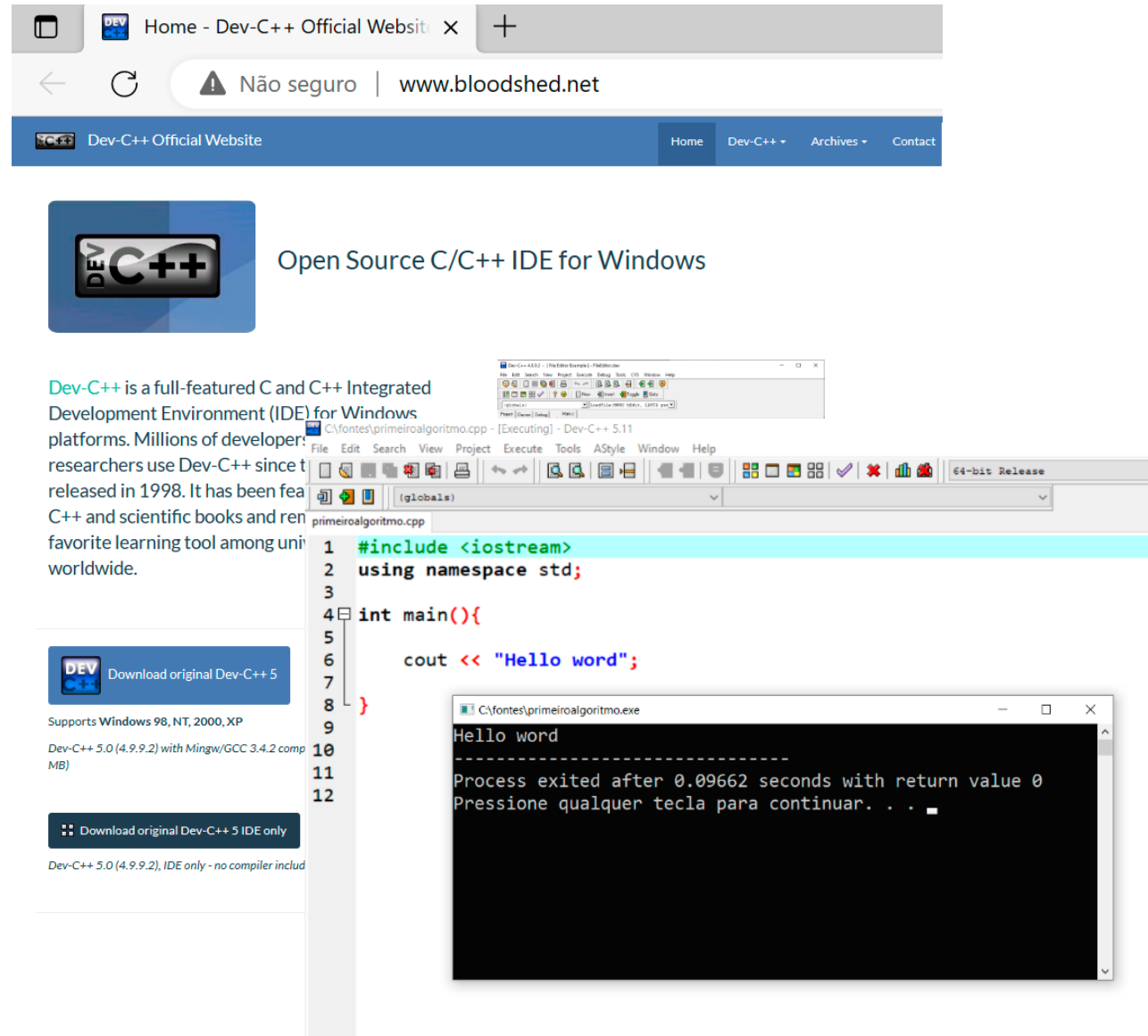
- É uma linguagem de programação compilada de uso geral.
- É uma das linguagens comerciais mais populares, sendo bastante usada também na academia por seu grande desempenho e base de utilizadores.
- Um programa em C++ consiste em um ou mais arquivos.
 - A extensão dos nomes dos arquivos fonte em C++, normalmente, é ".cpp".
- Existem cinco espécies de símbolos em C++: identificadores, palavras-chave, literais, operadores e outros separadores.

- Exemplo de código-fonte em C++:

```
/* Programa Hello World! */  
# include <iostream>  
int main() {                      //função principal  
    cout << "Hello World! \n";  
}
```


Estrutura Geral

- **O ambiente de desenvolvimento**
- Para desenvolver programas usando a linguagem C++ é instalar um compilador.
- Pode-se, adicionalmente, instalar um ambiente de desenvolvimento.
- O ambiente Dev C++ está disponível para download em <http://www.bloodshed.net/devcpp.html>.
 - É gratuito e livre.



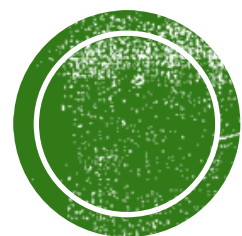
The screenshot displays the official website for Dev-C++, an open-source C/C++ IDE for Windows. The website header includes the Dev-C++ logo and navigation links: Home, Dev-C++ (selected), Archives, and Contact. The main content area features the Dev-C++ logo, the text "Open Source C/C++ IDE for Windows", and a description: "Dev-C++ is a full-featured C and C++ Integrated Development Environment (IDE) for Windows platforms. Millions of developers and researchers use Dev-C++ since it was released in 1998. It has been featured in C++ and scientific books and remains a favorite learning tool among university students worldwide." Below the text is a "Download original Dev-C++ 5" button. A smaller button below it says "Download original Dev-C++ 5 IDE only".

Overlaid on the website is a screenshot of the Dev-C++ IDE interface. The IDE window shows a C++ program named "primeiroalgoritmo.cpp" being executed. The code in the editor is:

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5
6     cout << "Hello word";
7
8 }
9
10
11
12
```

The IDE's output window shows the result of the execution:

```
C:\fontes\primeiroalgoritmo.exe
Hello word
-----
Process exited after 0.09662 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```



Dúvidas?

alemendes@gmail.com