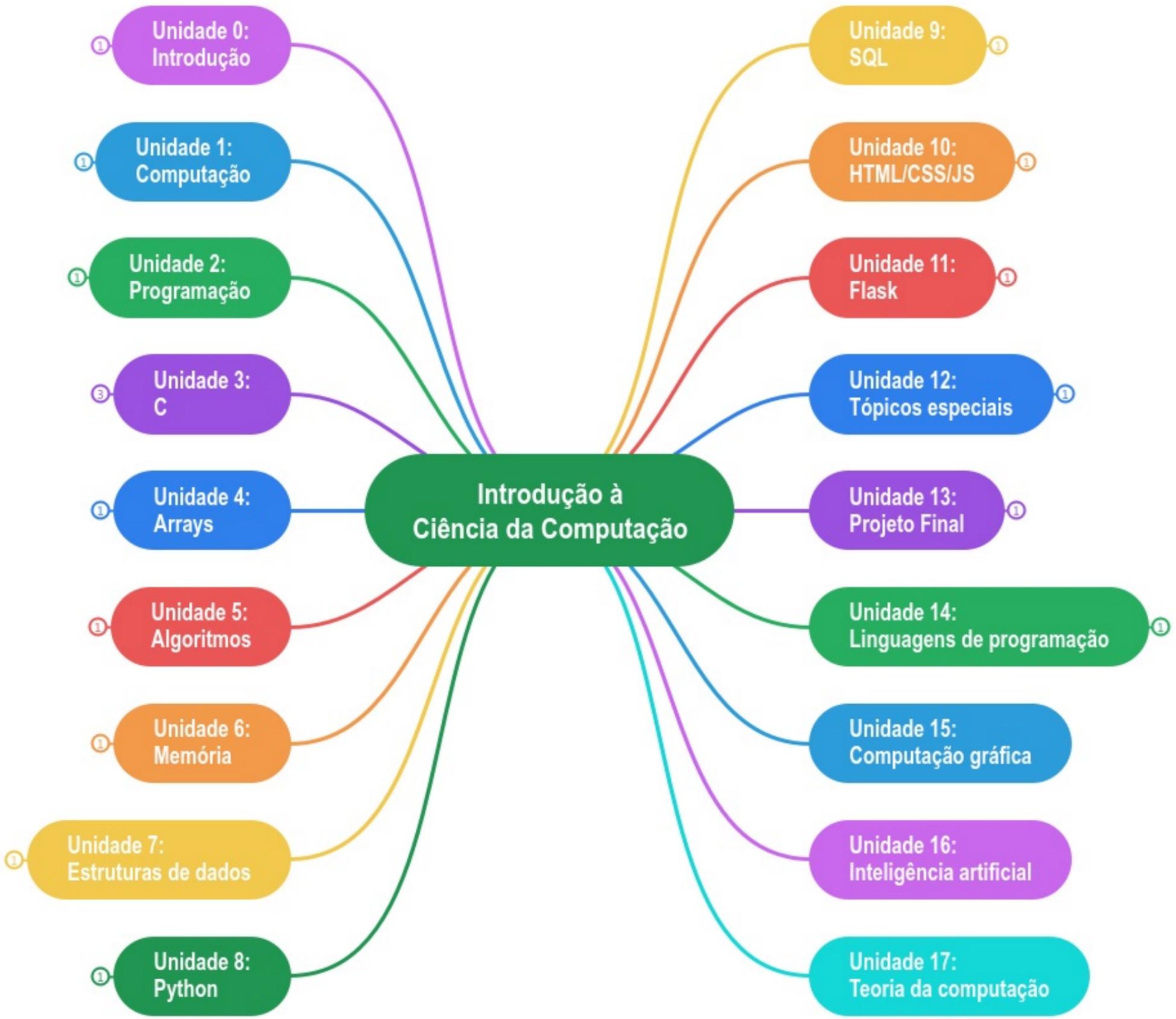


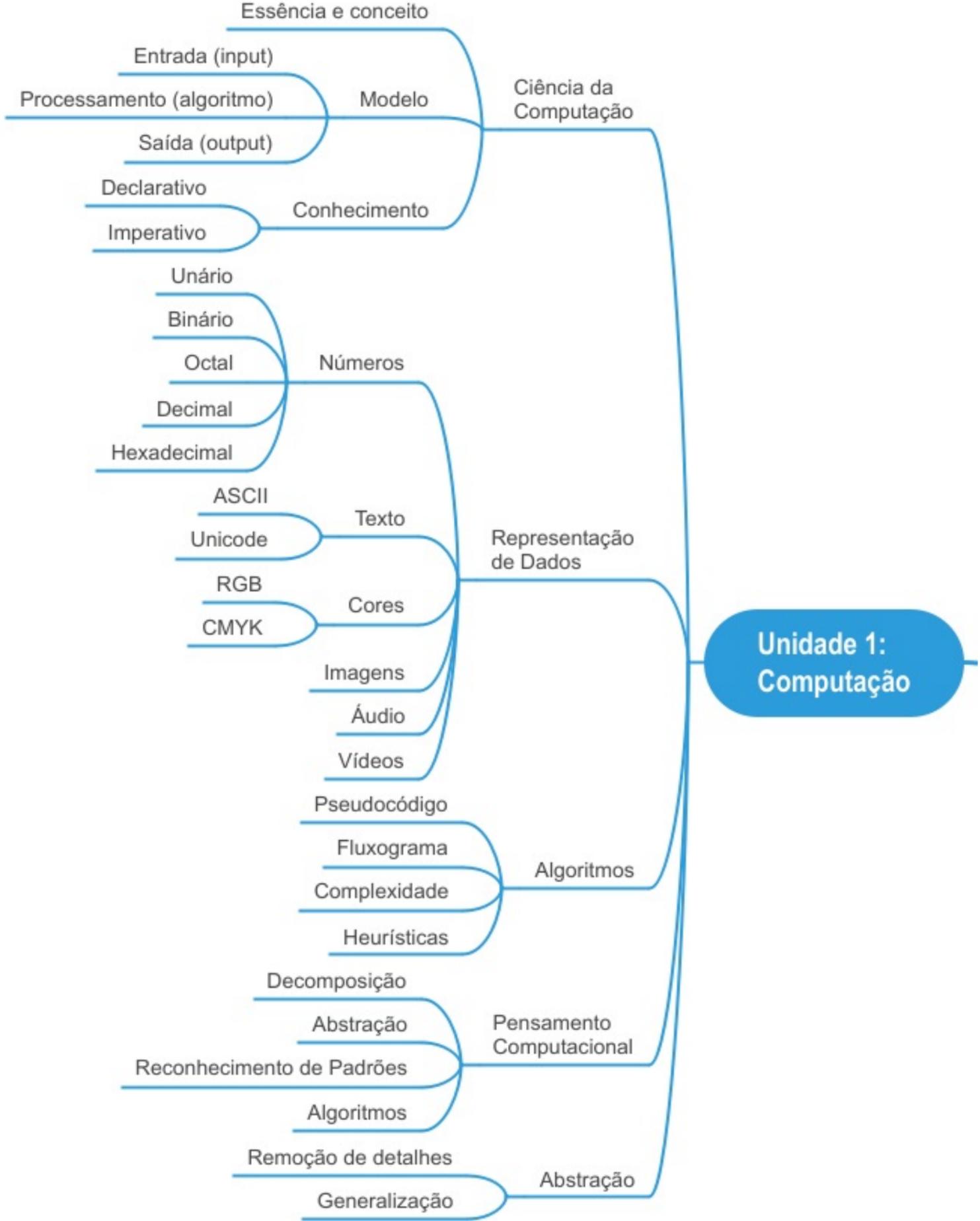
# FUNDAMENTOS DA COMPUTAÇÃO



# Visão geral



# Esta unidade



# O que é ciência da computação?



**Harold Abelson ("Hal Abelson")**

- **Co-fundador: Creative Commons**
- **Co-fundador: Free Soft. Found.**
- **Criador: MIT App Inventor**
- **Logo**
- **Turtle Geometry**
- **SICP e MIT 6.001**
- **BtB**

# O que é ciência da computação?

É sobre programação?

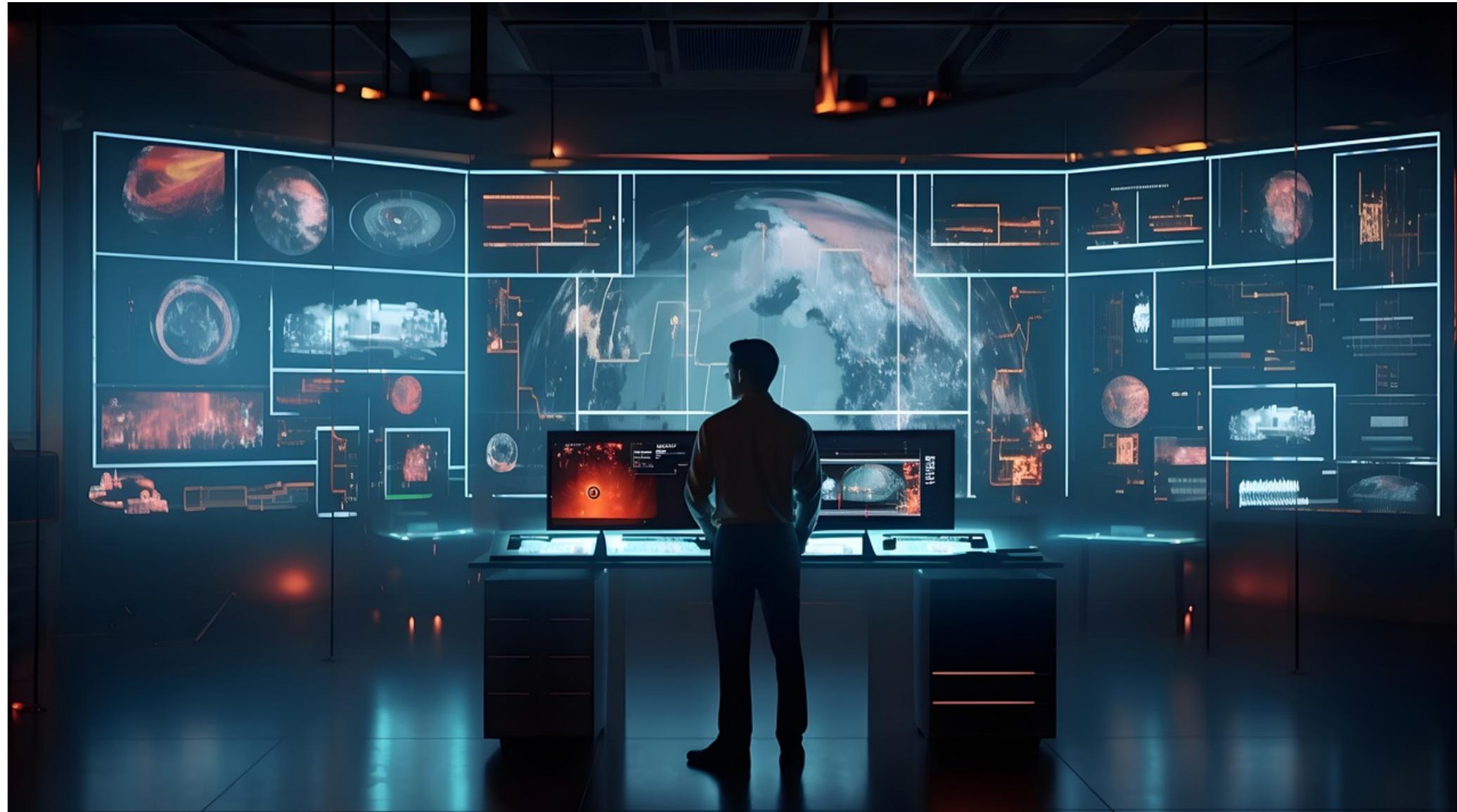
**Não. A programação nos permite expressar idéias e resolver problemas através de computadores, mas a ciência da computação, em essência, não depende de programação.**



# O que é ciência da computação?

É sobre computadores?

**Não. Os computadores são apenas as ferramentas que usamos (programamos) para nos ajudar a resolver problemas de forma mais rápida mas a ciência da computação, em essência, não depende de computadores.**



# O que é ciência da computação?

Se ciência da computação não é sobre computadores, nem sobre programação, então o que é?



Imagem: geralt, no Pixabay (<https://pixabay.com/photos/question-mark-a-notice-duplicate-3585355/>)

Já sabemos que é sobre **resolver problemas**. Como formalizar essa definição?

# O que é ciência da computação?

"[A ciência da computação] Também não se trata muito de computadores. E não se trata de computadores no mesmo sentido em que a física não se trata realmente de aceleradores de partículas, e a biologia não se trata realmente de microscópios e placas de Petri. E não se trata de computadores no mesmo sentido em que a geometria não se trata realmente do uso de instrumentos topográficos.

Na verdade, há muitos pontos em comum entre a ciência da computação e a geometria. A geometria, antes de mais nada, é outra disciplina com um péssimo nome. O nome vem de Gaia, que significa Terra, e metron, que significa medir. Geometria, originalmente, significava medir a Terra ou fazer levantamentos. E a razão para isso foi que, há milhares de anos, o sacerdócio egípcio desenvolveu os rudimentos da geometria para descobrir como restaurar os limites dos campos que foram destruídos pelas cheias anuais do Nilo. E, para os egípcios que fizeram isso, a geometria era realmente o uso de instrumentos topográficos.

Agora, a razão pela qual pensamos que a ciência da computação trata de computadores é praticamente a mesma razão pela qual os egípcios pensavam que a geometria tratava de instrumentos de levantamento. E isto é, quando alguma área está apenas começando e você não a entende muito bem, é muito fácil confundir a essência do que você está fazendo com as ferramentas que você usa. E, de fato, numa escala absoluta de coisas, provavelmente sabemos menos sobre a essência da ciência da computação do que os antigos egípcios realmente sabiam sobre geometria.

Bem, o que eu quero dizer com essência da ciência da computação? O que quero dizer com essência da geometria? Veja, é certamente verdade que esses egípcios usavam instrumentos de topografia, mas quando olhamos para eles depois de alguns milhares de anos, dizemos, caramba!, o que eles estavam fazendo, a coisa importante que estavam fazendo [a essência da geometria], era começar a formalizar noções sobre espaço e tempo, para iniciar uma maneira de falar formalmente sobre verdades matemáticas. Isso levou ao método axiomático. Isso levou a que toda matemática moderna descobrisse uma maneira de falar precisamente sobre o chamado conhecimento declarativo, sobre o que é verdadeiro.

Bem, da mesma forma, acho que no futuro as pessoas olharão para trás e dirão, sim, aqueles primitivos do século 20 estavam brincando com esses dispositivos chamados computadores mas, na verdade, o que eles estavam fazendo era começando a aprender como formalizar intuições sobre processos, sobre como fazer as coisas, começando a desenvolver uma forma de falar precisamente sobre o conhecimento de como fazer alguma coisa [conhecimento imperativo], em oposição à geometria que fala sobre o que é verdadeiro [conhecimento declarativo]."

**Hal Abelson**, vídeo com a gravação da 1ª aula do SICP para funcionários da Hewlett-Packard, em julho de 1986.  
(<https://groups.csail.mit.edu/mac/classes/6.001/abelson-sussman-lectures/>)

# O que é ciência da computação?

É a ciência que **projeta e implementa algoritmos para solucionar problemas.**



**Abū 'Abd Allāh Muḥammad ibn Mūsā al-Khwārizmī**  
(Pai de Abdullah, Mohammed, filho de Moisés, nativo de Khwārizm)

Matemático persa de Khwārizm (c. 780 - c. 850), produziu diversos trabalhos com grande influência na matemática, astronomia e geografia. Escreveu um tratado sobre álgebra onde demonstrou a primeira solução sistemática de equações lineares e quadráticas. Seu nome deu origem aos termos "algarismo" e "agoritmo".

Estátua de al-Khwārizmī em Khiva, Uzbequistão.

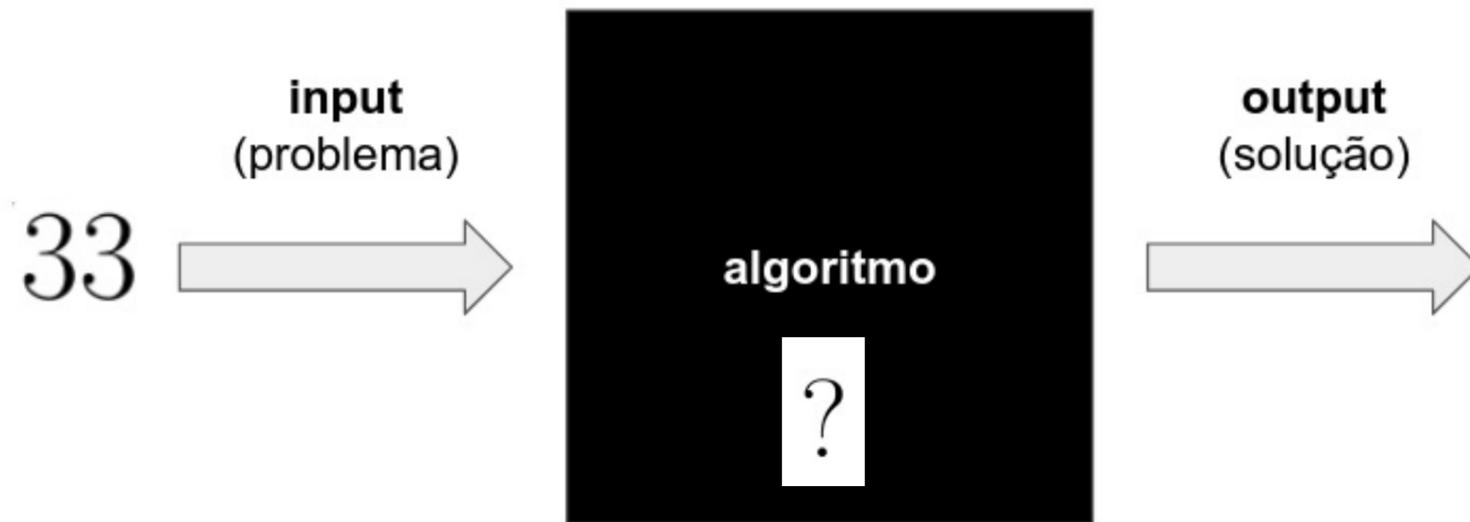
# Modelo da computação: como resolver problemas?



# Conhecimento declarativo x conhecimento imperativo

$$\sqrt{33} = ?$$

$$\sqrt{x} = y \mid y \geq 0 \wedge y^2 = x$$

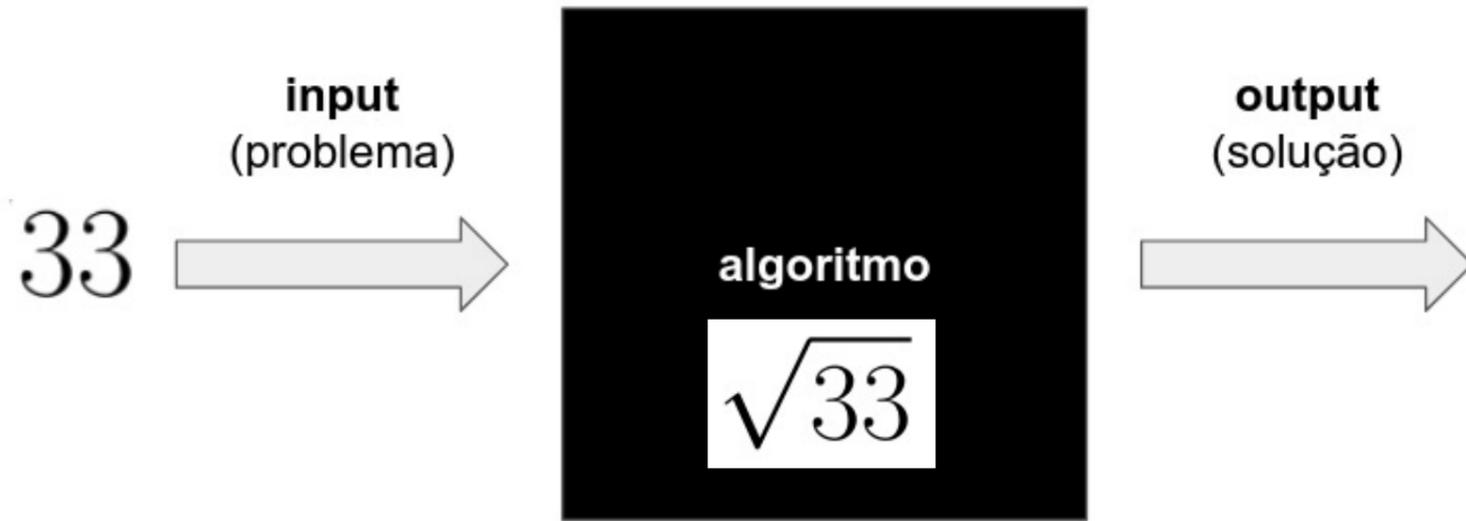


## Conhecimento DECLARATIVO:

- Nos **diz o que É alguma coisa**, define algo, diz o que é verdadeiro.
- Não ajuda a resolver o problema, pelo menos diretamente, ou seja, **não diz COMO resolver**.

# Conhecimento declarativo x conhecimento imperativo

$$\sqrt{x} = y \mid y \geq 0 \wedge y^2 = x$$



## Método de Newton:

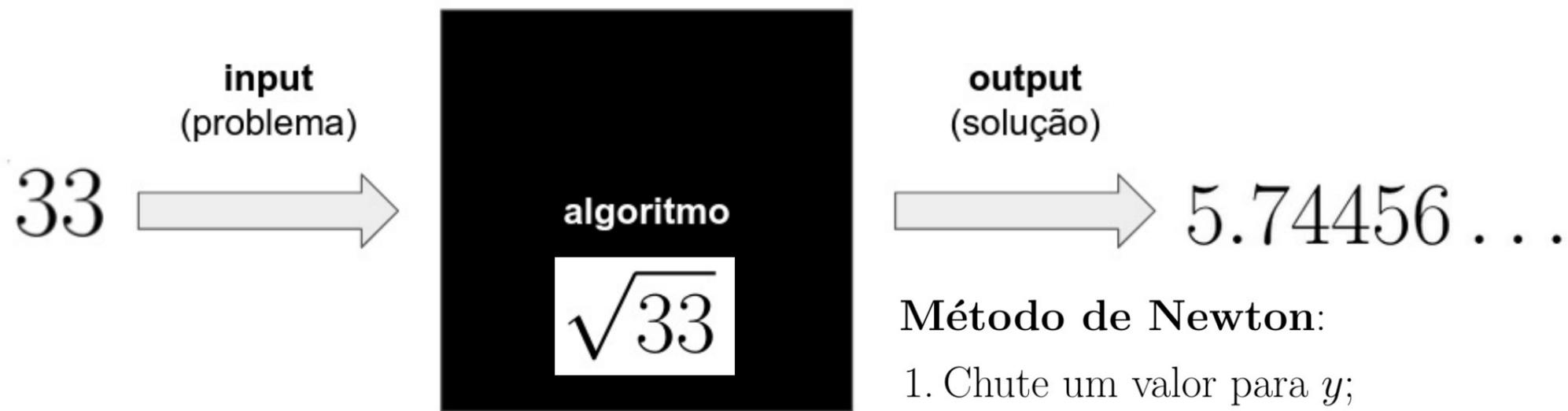
1. Chute um valor para  $y$ ;
2. Calcule o valor de  $y^2$ ;
3. Se  $y^2 = x$  (ou um valor próximo o suficiente), você achou a raiz. Termine o procedimento.
4. Se  $y^2 \neq x$  (ou não está próximo o suficiente), melhore a estimativa de  $y$

fazendo o seguinte cálculo: novo  $y = \frac{y + \frac{x}{y}}{2}$ ;

5. Retorne ao segundo passo até que você encontre a raiz ou uma aproximação suficiente.

# Conhecimento declarativo x conhecimento imperativo

$$\sqrt{x} = y \mid y \geq 0 \wedge y^2 = x$$



Método de Newton:

1. Chute um valor para  $y$ ;
2. Calcule o valor de  $y^2$ ;
3. Se  $y^2 = x$  (ou um valor próximo o suficiente), você achou a raiz. Termine o procedimento.
4. Se  $y^2 \neq x$  (ou não está próximo o suficiente), melhore a estimativa de  $y$

fazendo o seguinte cálculo: novo  $y = \frac{y + \frac{x}{y}}{2}$ ;

5. Retorne ao segundo passo até que você encontre a raiz ou uma aproximação suficiente.

## Conhecimento IMPERATIVO:

- Nos **diz COMO fazer alguma coisa** (mesmo que eu não saiba o que é essa coisa).
- A ciência da computação está interessada nesse tipo de conhecimento pois ele nos permite projetar e implementar soluções (algoritmos) para resolver diversos tipos de problemas.

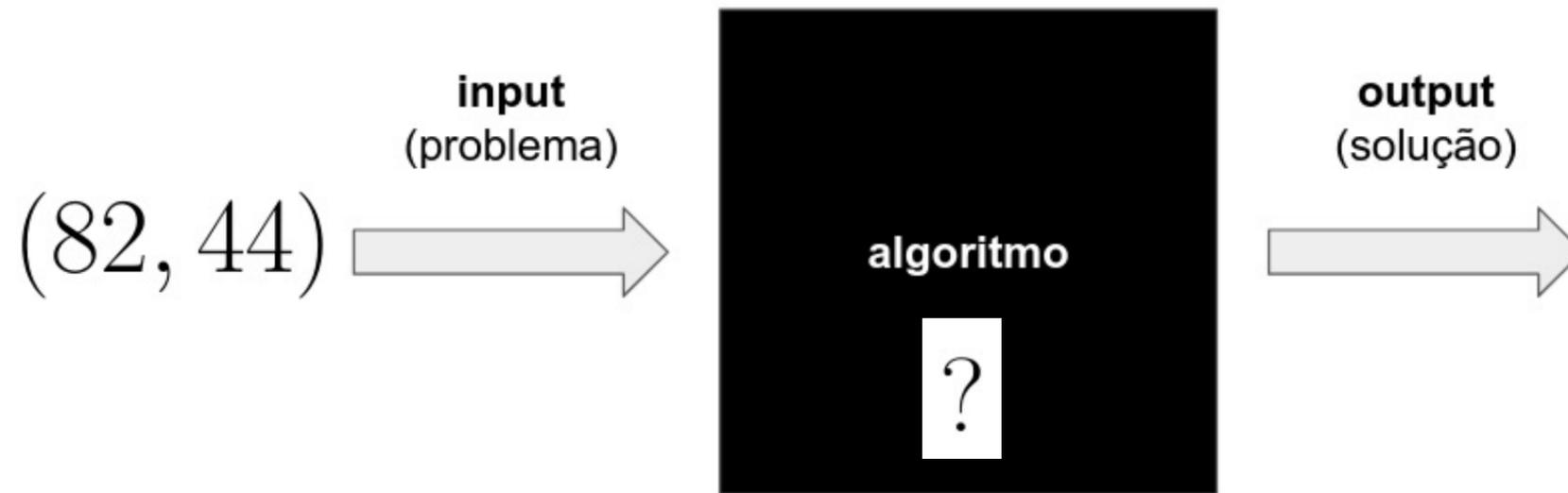
# Conhecimento declarativo x conhecimento imperativo

$$\text{mdc}(82, 44) = ?$$

$\text{mdc}(a, b) = \text{maior } x \mid \text{rem}(a/x) = 0 \wedge \text{rem}(b/x) = 0$ , onde:

$a, b \in \mathbb{Z}$  (ambos não podem ser 0 ao mesmo tempo),  $x \in \mathbb{Z}_+^*$ ,

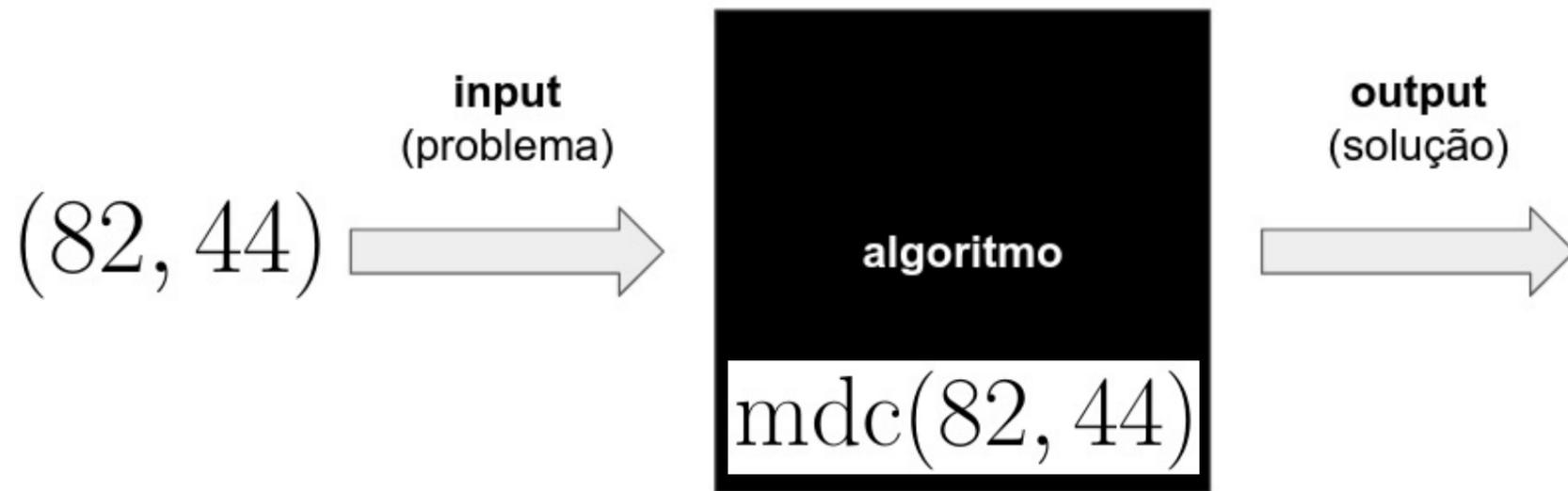
rem é o resto da divisão, e  $\text{mdc}(a, 0) = \text{mdc}(0, a) = |a|$



## Conhecimento DECLARATIVO:

- Nos **diz o que É alguma coisa**, define algo, diz o que é verdadeiro.
- Não ajuda a resolver o problema, pelo menos diretamente, ou seja, **não diz COMO resolver**.

# Conhecimento declarativo x conhecimento imperativo



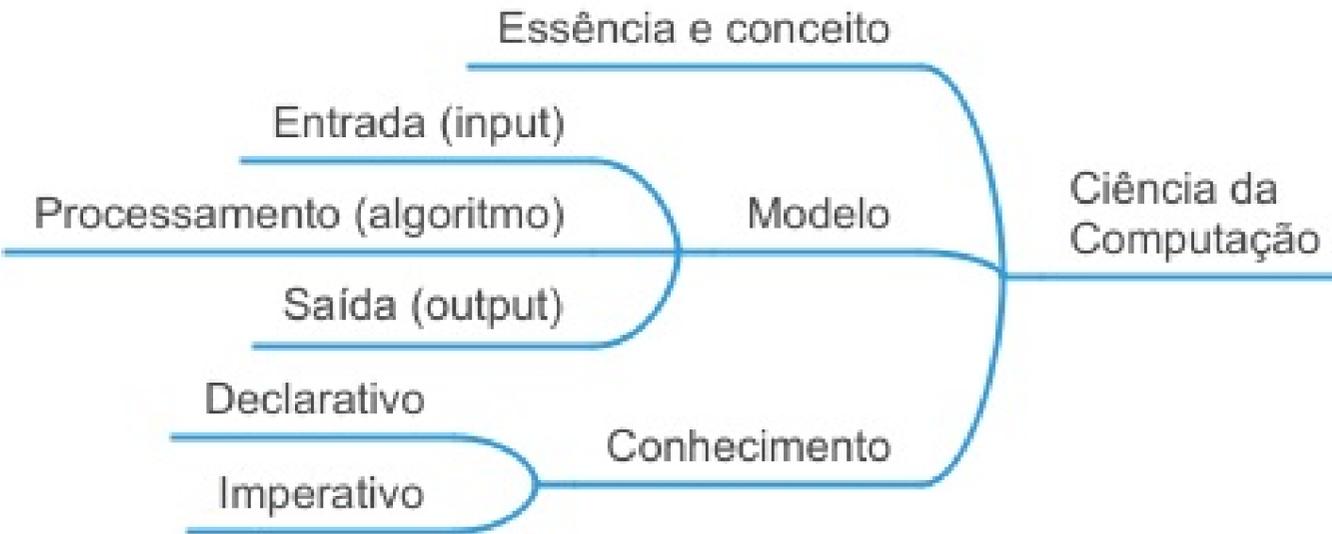
## Método de Euclides:

1. Inicie com  $\text{mdc}(a, b)$ ;
2. Substitua  $\text{mdc}(a, b)$  por  $\text{mdc}(b, \text{rem}(a/b))$ ;
3. Continue substituindo até que apareça  $\text{mdc}(x, 0)$ ;
4. Nesse momento  $\text{mdc}(a, b) = |x|$ .

## Conhecimento IMPERATIVO:

- Nos **diz COMO fazer alguma coisa** (mesmo que eu não saiba o que é essa coisa).
- A ciência da computação está interessada nesse tipo de conhecimento pois ele nos permite projetar e implementar soluções (algoritmos) para resolver diversos tipos de problemas.

# Resumo: 1º grande fundamento da computação: conceito, modelo, conhecimento



# No próximo vídeo: 2º grande fundamento: representação de dados

