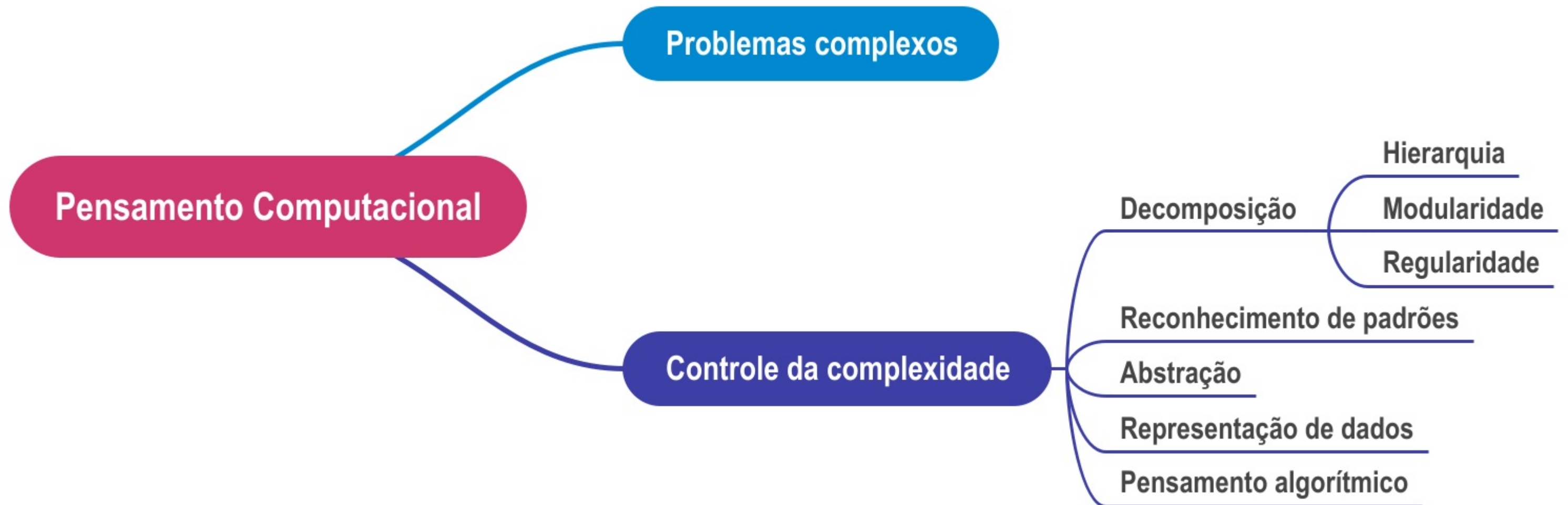


FUNDAMENTOS DA COMPUTAÇÃO



Pensamento Computacional



Pensamento Computacional



Jean-Pol Grandmont, na Wikipedia

(https://en.wikipedia.org/wiki/File:0_Chambre_de_Rapha%C3%ABl_-_%C3%89cole_d%27Ath%C3%A8nes_-_Mus%C3%A9es_du_Vatican.JPG)

Detalhe da pintura de Raffaello Sanzio da Urbino, "A Esola de Atenas" (1509-1511), mostrando Euclides ensinando alguns estudantes.

Euclides (c. 350 - c.250 a.C.)

Proposição 20 do Livro IX de **Os Elementos** (c. 300 a.C): "**Os números primos são mais do que qualquer conjunto atribuído de números primos**".

Sejam A, B, C os números primos atribuídos; Digo que existem mais números primos do que A, B, C .

Pois seja tomado o menor número medido por A, B, C , e seja DE ; deixe a unidade DF ser adicionada a DE .

Então EF é primo ou não.

Primeiro, seja primo; então foram encontrados os números primos A, B, C, EF que são maiores que A, B, C .

A seguir, deixe EF não ser primo; portanto, é medido por algum número primo. [VII. 31] Seja medido pelo número primo G .

Digo que G não é igual a nenhum dos números A, B, C . Pois, se possível, que assim seja. Agora A, B, C medem DE ; portanto G também medirá DE . Mas também mede a EF .

Portanto G , sendo um número, medirá o resto, a unidade DF : o que é um absurdo. Portanto G não é igual a nenhum dos números A, B, C . E por hipótese é primo.

Portanto, foram encontrados os números primos A, B, C, G que são maiores do que o conjunto atribuído de A, B, C .

Euclides

Pensamento Computacional

Problemas complexos

Problemas complexos

Problema a ser resolvido:

Crie um dos foguetes mais potentes de todos os tempos, capaz de levar 64 toneladas de equipamento para o espaço...



Foto: SpaceX (<https://www.spacex.com/vehicles/falcon-heavy/>)

Problemas complexos



Foto: SpaceX (<https://www.spacex.com/vehicles/falcon-heavy/>)

... e que seja capaz de dar ré para pousar os propulsores para reutilização futura.

Problemas complexos



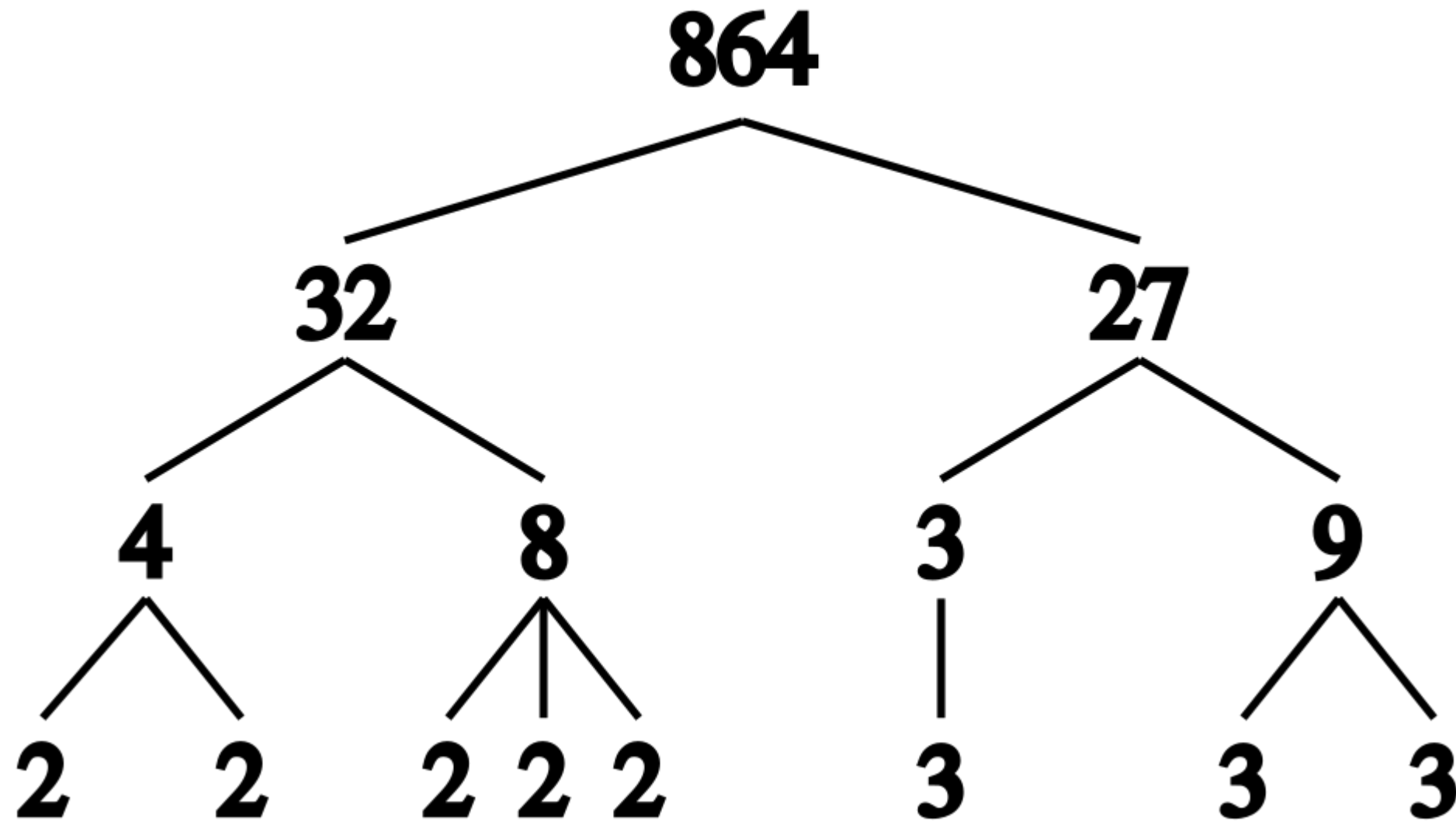
Problema a ser resolvido:

Crie um dispositivo capaz de melhorar a comunicação de pacientes portadores da síndrome do encarceramento.

Jean-Dominique Bauby, em 1996, "ditando" seu livro: O escafandro e a Borboleta.

Foto: Jean-Louis Sieff, no New York Times, 1996 (https://en.wikipedia.org/wiki/File:Jean-Dominique_Bauby.jpg)

Problemas complexos

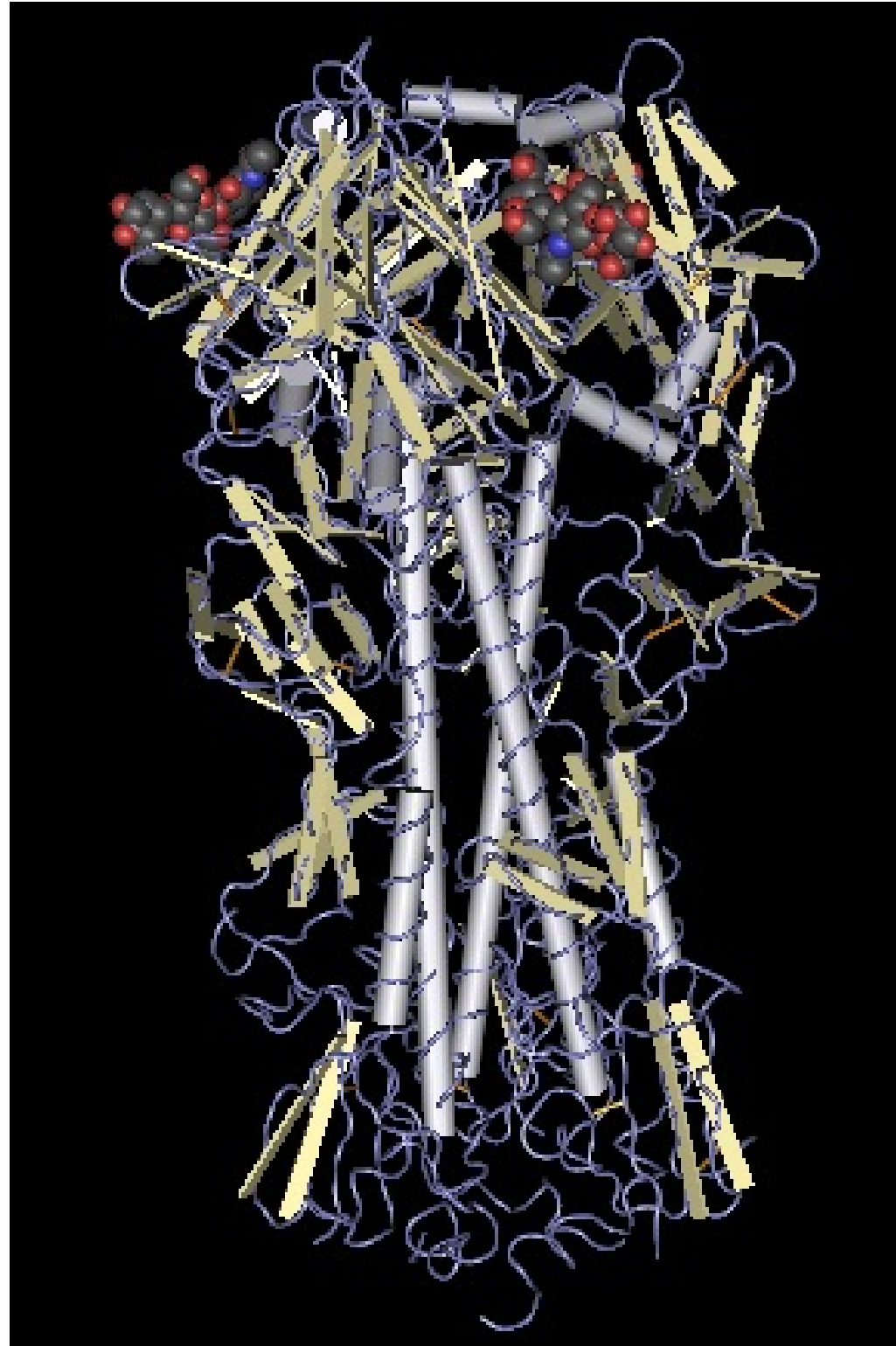


Krishnavedala, na Wikipedia (<https://commons.wikimedia.org/wiki/File:PrimeDecompositionExample.svg>)

Problema a ser resolvido:

Crie um algoritmo rápido para a decomposição em fatores primos de um número inteiro composto.

Problemas complexos



Problema a ser resolvido:

Crie um software capaz de modelar proteínas de vírus, em três dimensões, para a pesquisa de novos remédios contra infecções.

Problemas complexos



Problema a ser resolvido:

Crie validador de CPF que ficará armazenado em um sistema de gerenciamento de bancos de dados para que, no cadastro de algum usuário, ele valide o número e consulte o site da Receita Federal para saber se é um CPF ativo.

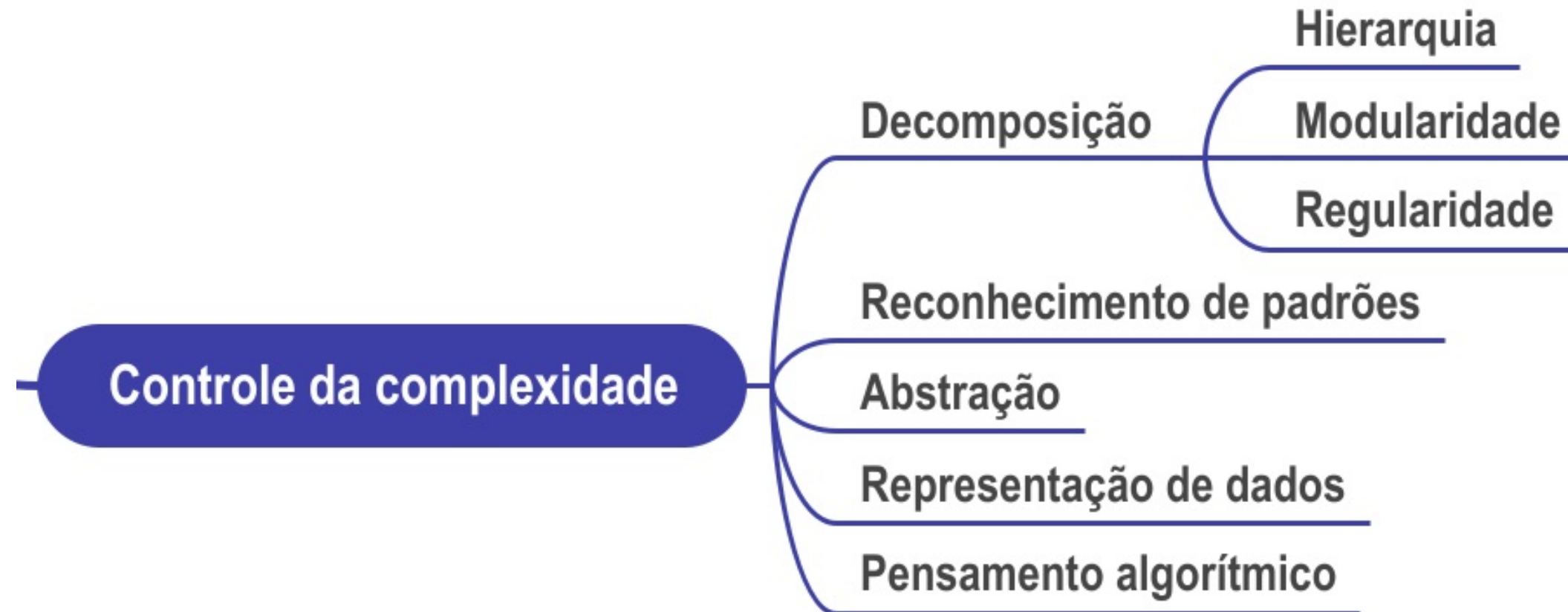
Problemas complexos: como resolver?

O que é comum aos problemas citados?

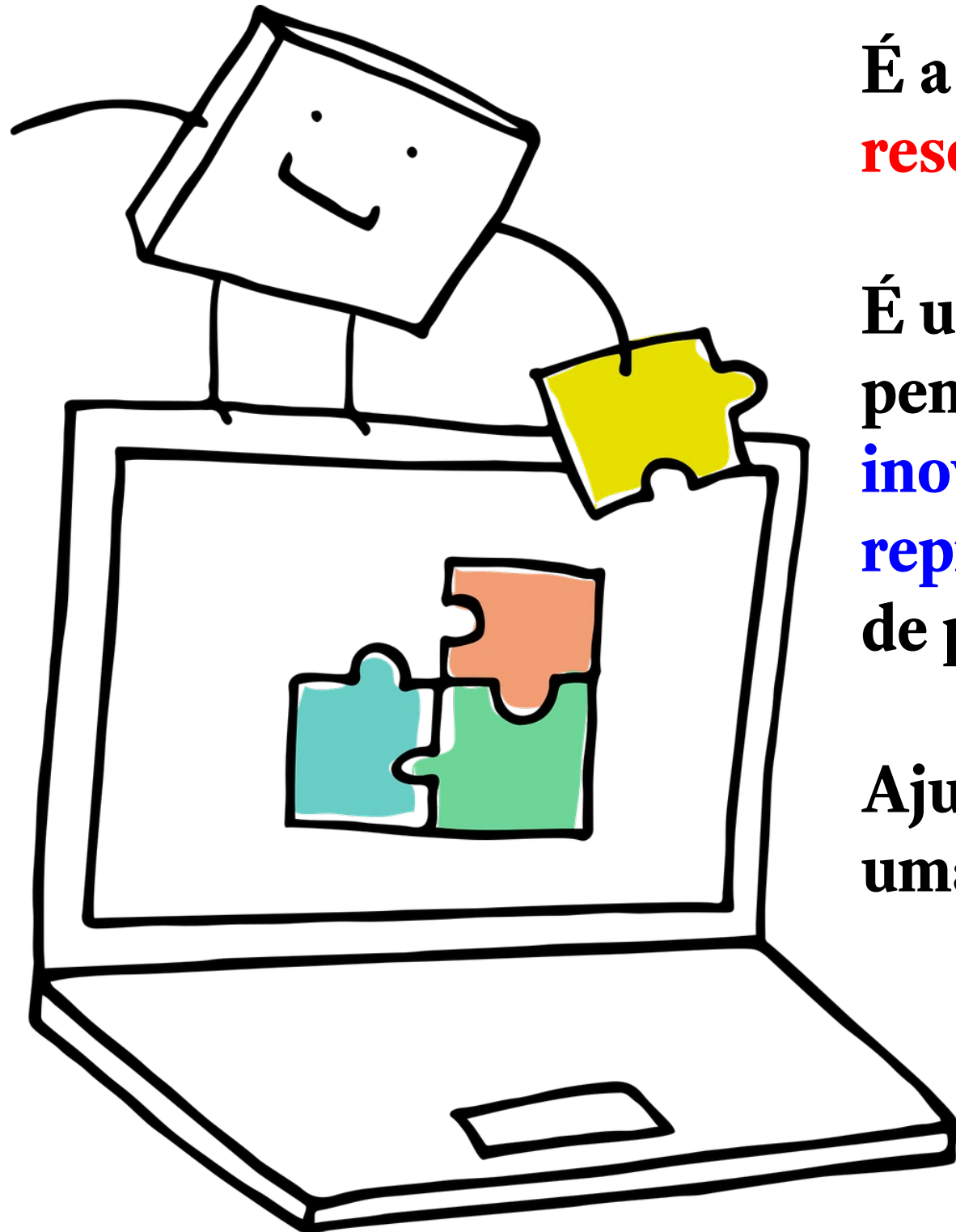
- são complexos
- podem exigir a participação de muitas pessoas
- são mal definidos e pouco compreendidos
- precisam de muitos passos intermediários
- as tarefas precisam ser divididas
- partes feitas separadamente devem ser compatíveis
- cada parte deve ter uma especificação e manual de uso
- o manual de uso deve dizer claramente como usar
- ...

Como **COMEÇAR** a resolver um problema complexo????

Problemas complexos: como resolver?



Pensamento Computacional



É a **maneira como o cientista da computação pensa para resolver problemas complexos**, com ou sem computadores.

É uma mistura de **bom senso**, pensamento **lógico**, pensamento **matemático**, pensamento **abstrato**, **criatividade**, **inovação**, habilidades de **engenharia**, **modelagem** matemática, **representação** e **organização** de dados, métodos de **resolução** de problemas, e, até mesmo, **sorte**.

Ajuda a **compreender o problema**, **projetar** e **implementar** uma solução, através do:

CONTROLE DA COMPLEXIDADE

Pensamento Computacional

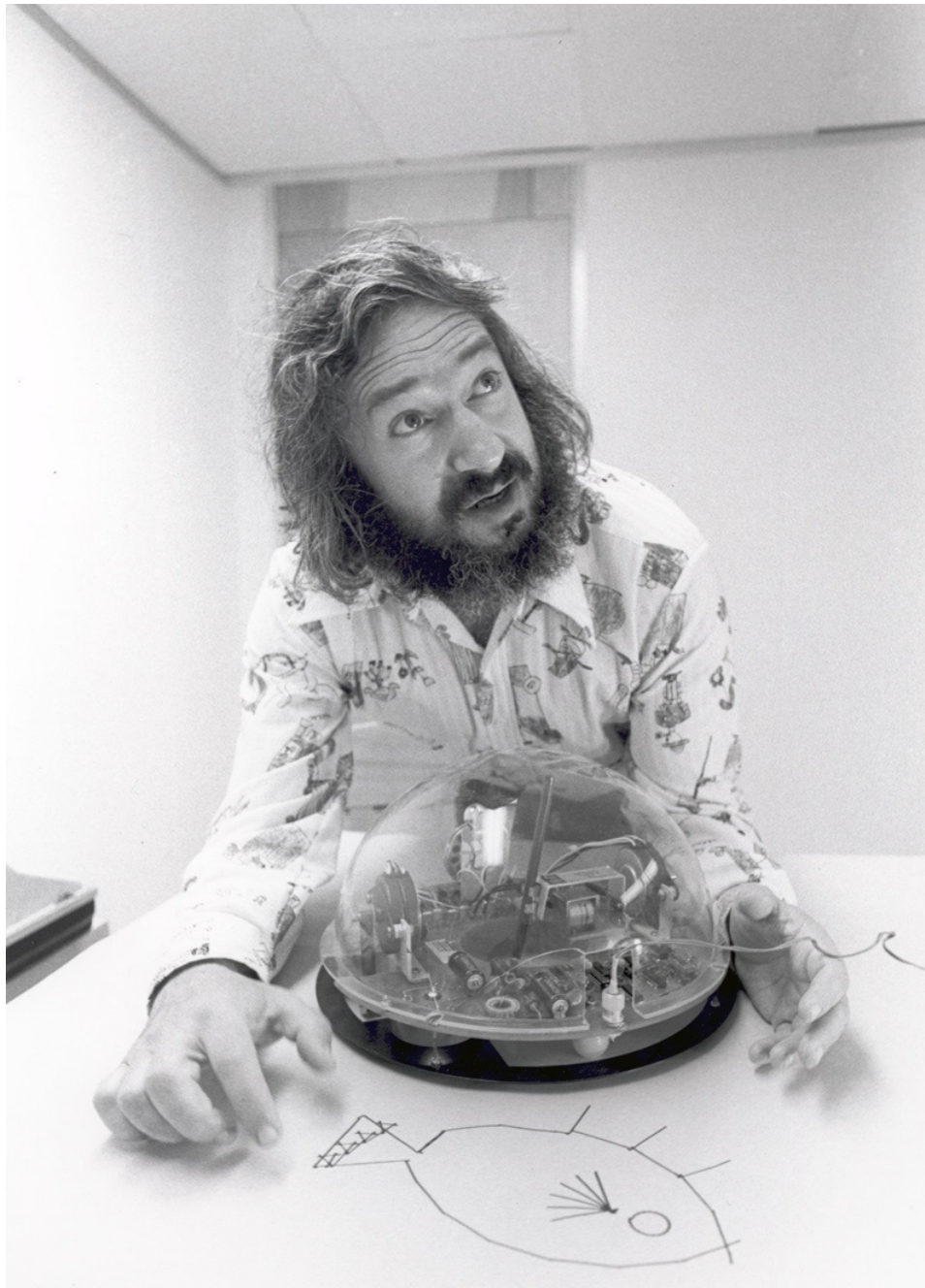
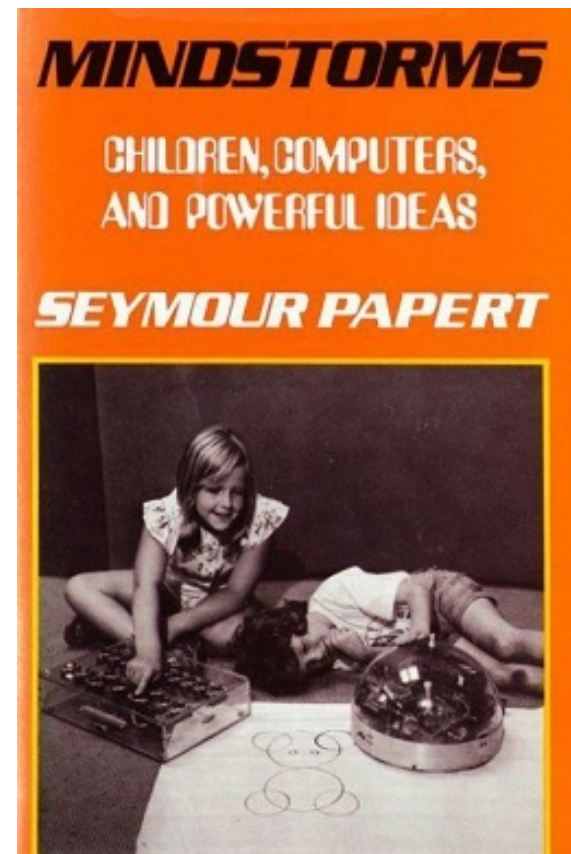


Foto por Matematicamente.it, na Wikipedia
(https://en.wikipedia.org/wiki/File:Seymour_Papert.jpg)

O conceito é antigo, mas o termo é novo. Foi usado pela primeira vez por **Seymour Aubrey Papert**, filho de americanos nascido na África do Sul, em seu livro "Mindstorms: Children, Computers, and Powerful Ideas", publicado em 1980.

- Um dos pioneiros na inteligência artificial
- Um dos pioneiros do construtivismo na educação
- Co-inventor da linguagem Logo



Abebooks, na Wikipedia
([https://en.wikipedia.org/wiki/File:Mindstorms_\(book\).jpg](https://en.wikipedia.org/wiki/File:Mindstorms_(book).jpg))

Pensamento Computacional

Padrões: Getty Images, no Unsplash (https://unsplash.com/photos/colored-hexagonal-digital-background-trendy-futuristic-style-modern-technology-concept-3d-rendering-geometric-art-business-design-SblMo_kHBZo)

Decomposição: Jackson Simmer, no Unsplash (<https://unsplash.com/photos/blue-orange-green-and-yellow-plastic-toy-ZxRHtPacwUY>)

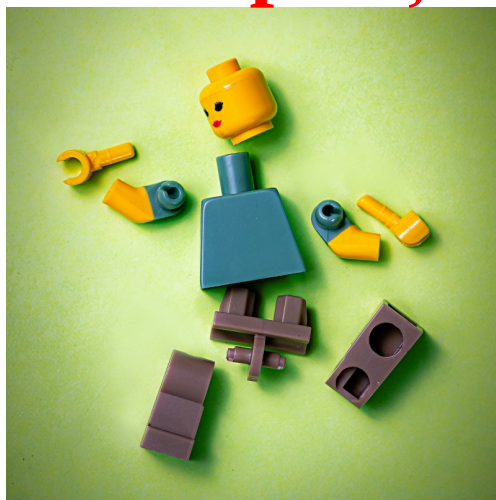
Abstração: OpenClipart-Vectors, no Pixabay (<https://pixabay.com/vectors/cardboard-box-box-cardboard-package-155479/>)

Representação de dados: Jae Rue, no Pixabay (<https://pixabay.com/illustrations/matrix-technology-tech-data-3109378/>)

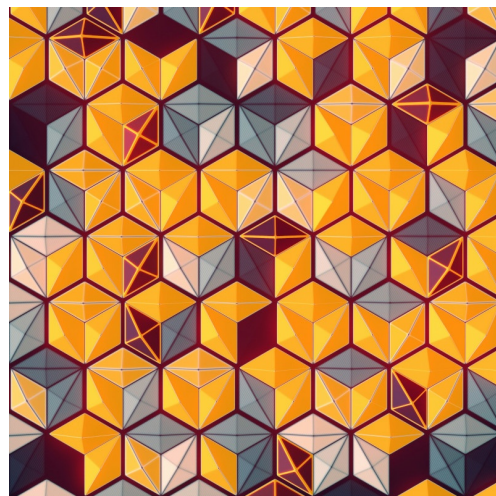
Algoritmo: Wapcaplet, na Wikipedia (<https://commons.wikimedia.org/wiki/File:LampFlowchart.svg>)

Costumamos dividir em 5 grandes componentes:

Decomposição



Reconhecimento de padrões



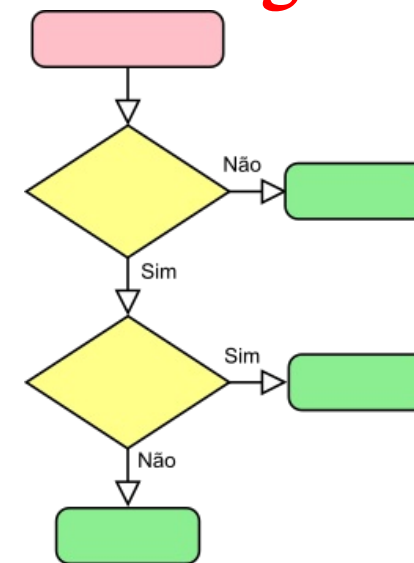
Abstração



Representação de dados



Pensamento Algorítmico



Decomposição



É **quebrar recursivamente um problema ou sistema complexo em partes menores** que são gerenciáveis, fáceis de compreender e resolver.

Interesse específico em:

- **Hierarquia** de problemas/soluções
- **Modularidade**
- **Regularidade**

Decomposição

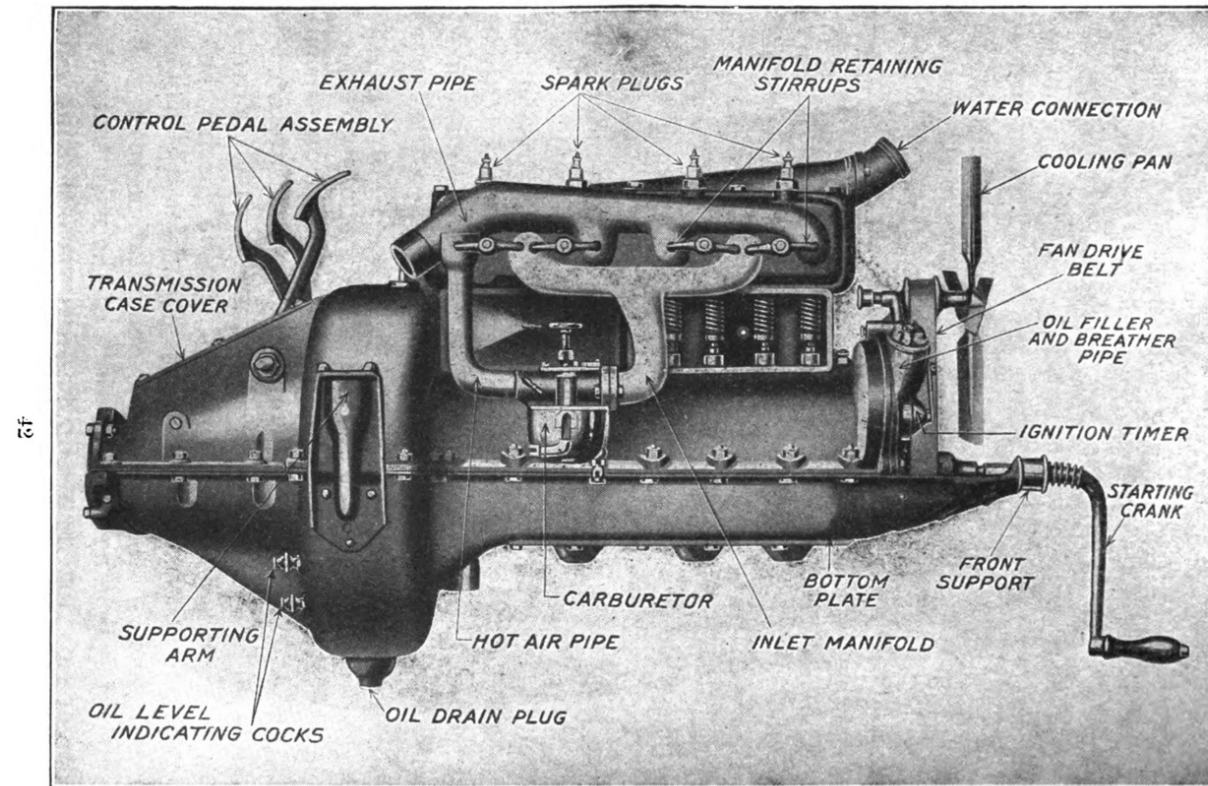
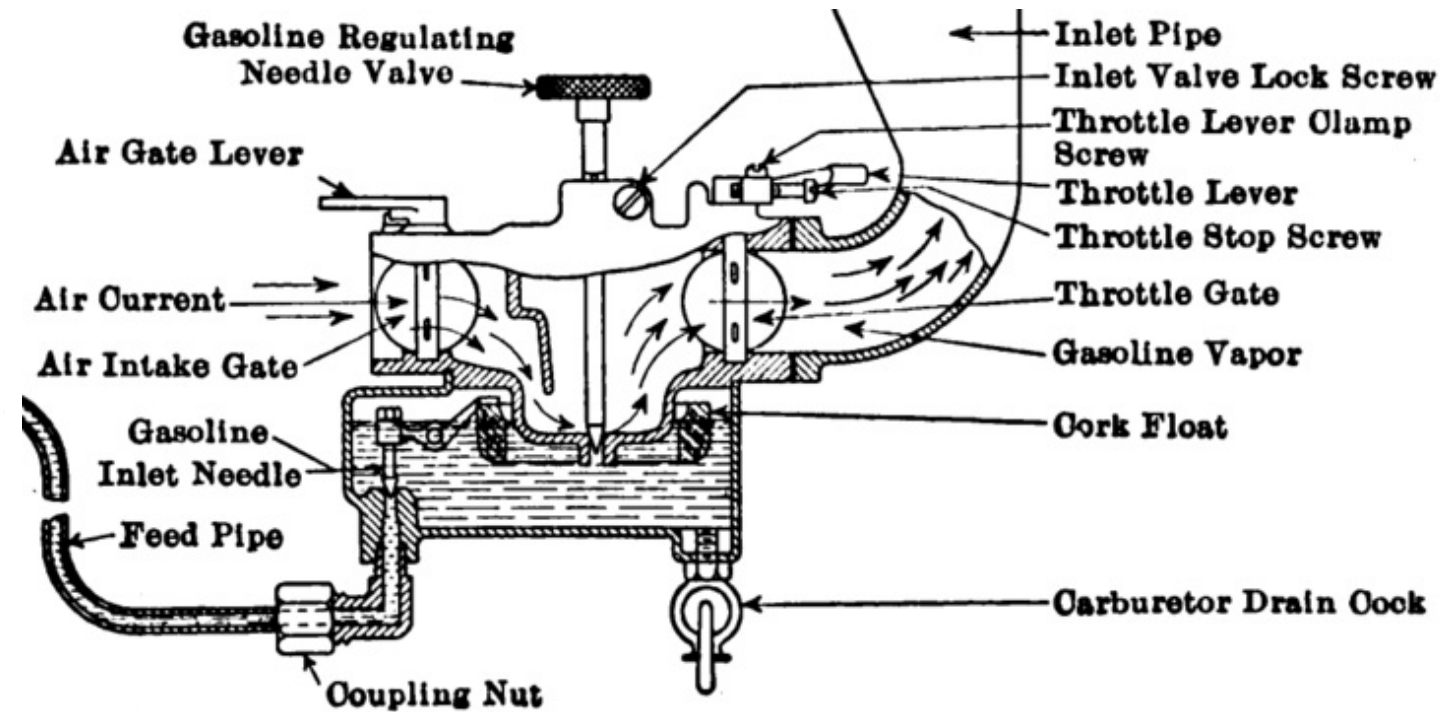


Fig. 8.—Valve Side of the Ford Model T Unit Power Plant Showing Manifolds, Carburetor and Interior of One of the Valve Spring Chambers.

Victor Wilfred Pagé, na Wikipedia

(https://commons.wikimedia.org/wiki/File:Pag%C3%A9_1917_Model_T_Ford_Car_Figure_08.png)



Victor Wilfred Pagé, na Wikipedia

(https://commons.wikimedia.org/wiki/File:Pag%C3%A9_1917_Model_T_Ford_Car_Figure_14.png)

Hierarquia:

quebrar recursivamente em **módulos solucionáveis**

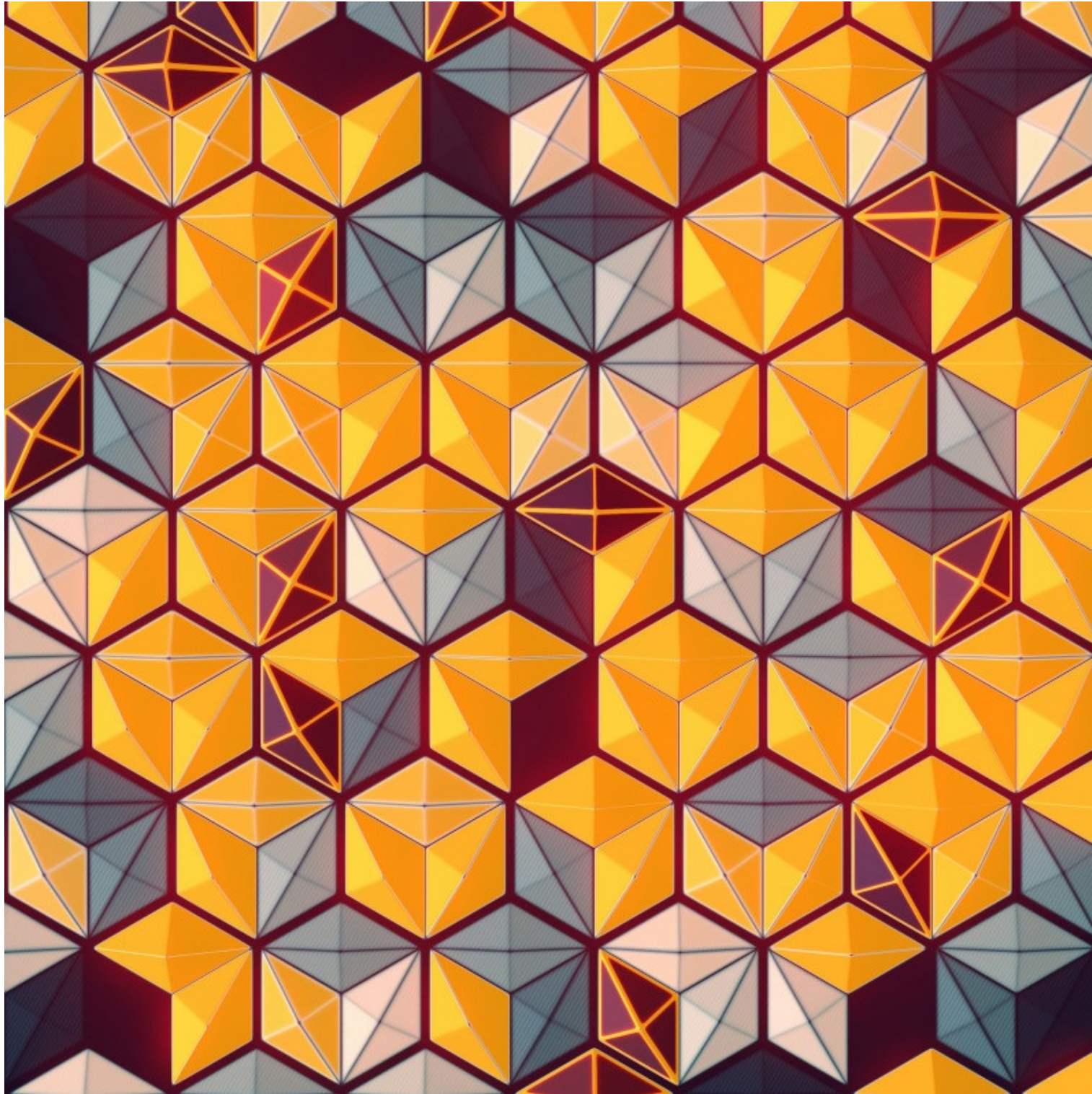
Modularidade:

cada módulo deve ter uma **funcionalidade específica**, com **interface bem definida**, para que eles possam ser conectados sem efeitos colaterais

Regularidade:

uniformidade e **reuso** de módulos intercambiáveis

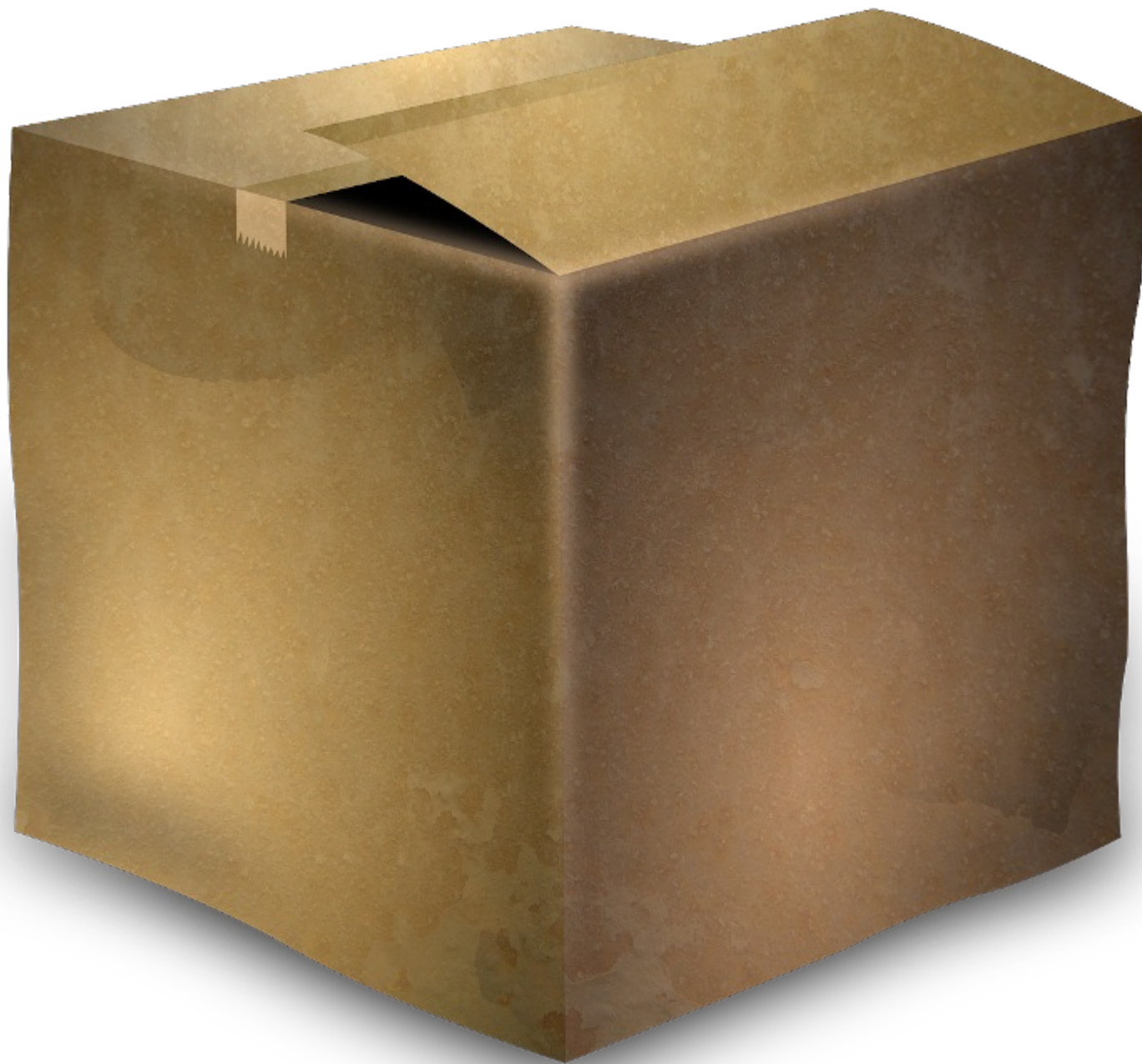
Reconhecimento de padrões



Capacidade de **encontrar padrões ou semelhanças entre os problemas decompostos**, ou entre problemas que **já foram solucionados antes**.

Isso permitirá resolver problemas de modo mais eficaz, sem reinventar a roda ou andar em círculos.

Abstração



Capacidade de **focar no que é essencial**, escondendo ou ignorando os detalhes que não são importantes no momento, e de **generalizar soluções**.

É a chave crítica para o controle da complexidade (e, assim, é a chave da computação!).

(De tão importante, falaremos mais sobre a abstração em outra aula!)

Representação de dados



Encontrar uma **representação adequada para o problema e para a solução** (entrada e saída de um algoritmo) é fundamental. Temos que encontrar a melhor **estrutura de dados** para cada caso.

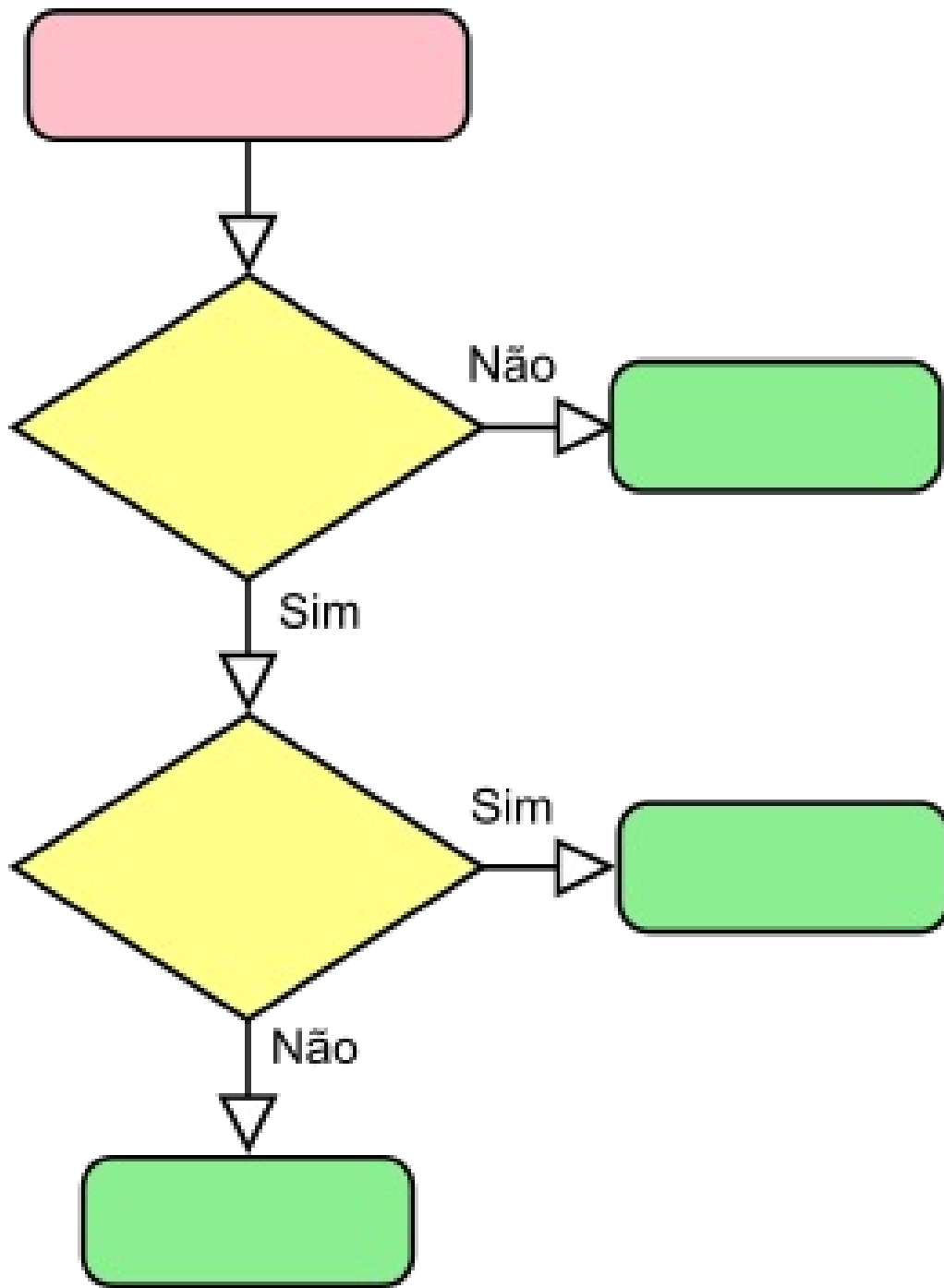
Na grande maioria das vezes, é mais fácil melhorar a representação do que tentar criar um algoritmo mais complexo para uma representação ruim!

Pensamento Algorítmico

Capacidade de **criar um algoritmo**, uma receita passo a passo que indica, de modo correto, como resolver o problema.

Lembre-se de nossa definição formal:

Um algoritmo é uma coleção bem ordenada de operações efetivamente computáveis, definidas e não ambíguas que, quando executada sobre uma entrada produz uma saída e termina em uma quantidade finita de passos e de tempo.



Em resumo:

