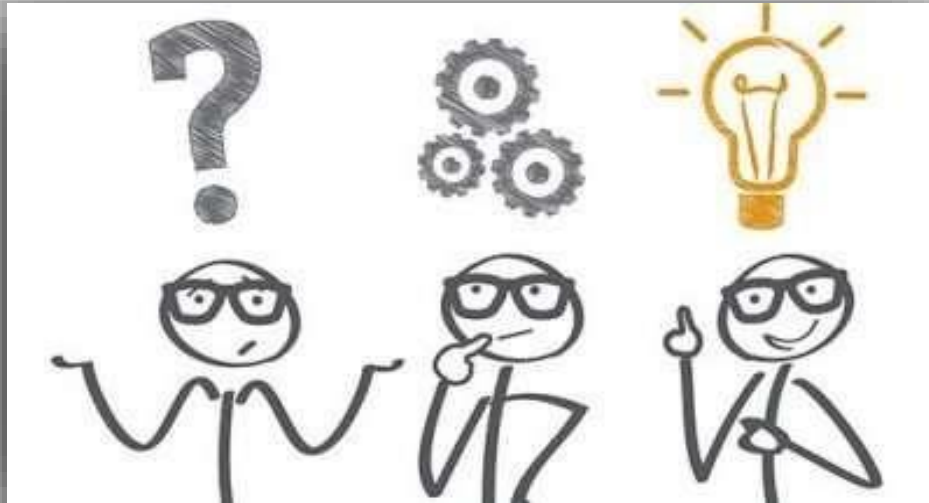


# **PENSAMENTO COMPUTACIONAL**

**Resolução de problemas**

# Definição de Problema

- É um Tema ou Questão, em qualquer área do conhecimento, cuja solução ou resposta requer considerável pesquisa, estudo, reflexão e grande esforço para ser solucionado. (Dicionário Michaelis)



# Problema do Elevador Lento

- Proprietário de um prédio de escritórios estava recebendo reclamações de seus inquilinos - elevador velho e lento - têm que esperar muito tempo
- Possíveis soluções:
  - substituir o elevador
  - instalar um motor mais forte e rápido
  - talvez atualizar o algoritmo que executa o elevador
- Problema de pico de demanda

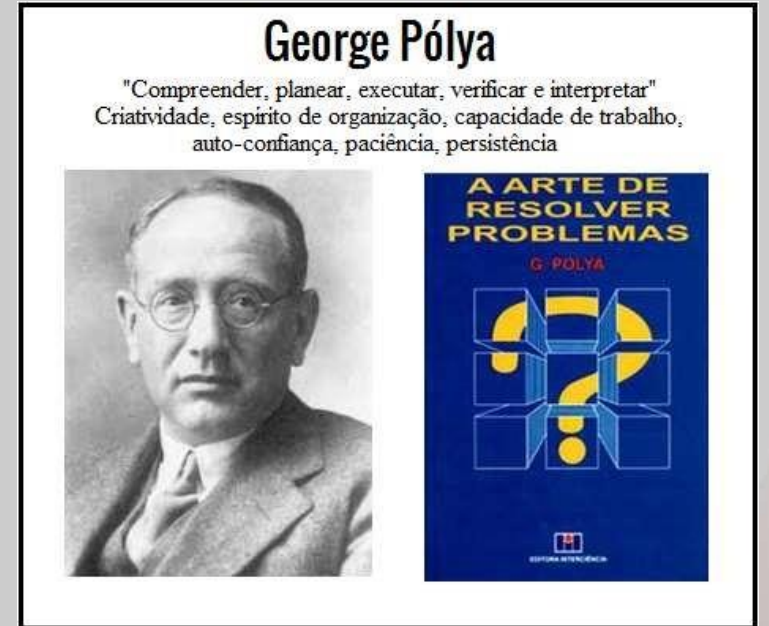
**Solução muito mais elegante: colocar espelhos ao lado do elevador**

Fonte: [hbr.org/2017/01/are-you-solving-the-right-problems](http://hbr.org/2017/01/are-you-solving-the-right-problems)



# Resolução de Problemas

- "Se você não pode resolver um problema, existe um problema mais fácil que pode ser resolvido: encontre-o."
- De acordo com Pólya (1978, p. 65):
  - Resolver problemas é uma habilidade prática, como nadar, esquiar ou tocar piano: você pode aprendê-la por meio de imitação e prática. (...) se você quer aprender a nadar você tem de ir à água e se você quer se tornar um bom 'resolvedor de problemas', tem que resolver problemas.



# Resolução de Problemas

- Uma grande descoberta resolve um grande problema!!!
- Na resolução de qualquer problema há sempre uma pitada de descoberta
- Mesmo modesto, o problema desafia a curiosidade
- Resolver o problema por seus próprios meios, trará a tensão e satisfação do triunfo da descoberta
- Habilidades essenciais para o profissional do presente e principalmente do futuro

# Etapas do Método de Pólya

Baseado no Método Cartesiano de René Descartes - 1673 -  
“Dividir a dificuldade em tantas partes quantas necessárias  
para melhor as resolver” - **Dividir para Conquistar!!!**

1. Compreensão do problema
  - É preciso compreender o problema
2. Planejar sua Resolução
  - Encontre a conexão entre os dados e o desconhecido
3. Execução do Plano
  - Execute o plano
4. Retrospecto
  - Examine a solução obtida

# Compreensão do problema

- Qual a incógnita? O que se desconhece?
- Quais são os dados?
- Qual é a condição?
- É possível satisfazer a condição?
- A condição é suficiente para determinar o desconhecido?
- É suficiente? Redundante? Contraditória?
- Desenhe uma figura representando o problema
- Adote uma notação adequada
- Separe as diversas partes da condicionante



# Planejar sua resolução

- Considerar problemas auxiliares se não puder encontrar uma conexão imediata
- Se não puder resolver o problema, procure antes resolver algum problema correlato
- É possível imaginar um problema correlato mais acessível?
- Um problema mais genérico?
- Um problema mais específico?
- Um problema análogo?
- É possível resolver uma parte do problema?





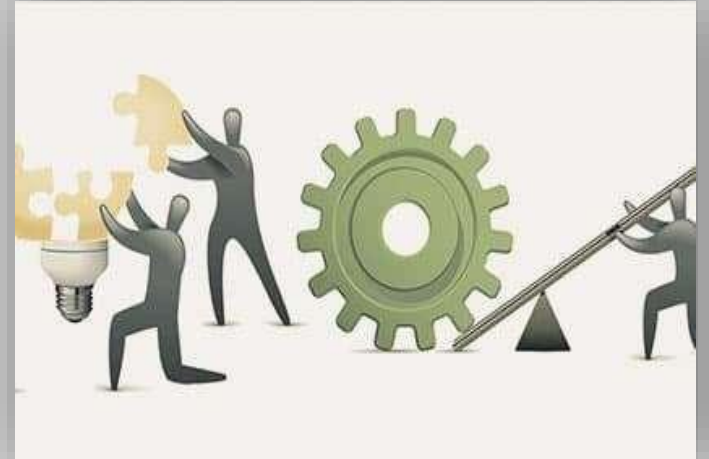
# Planejar sua resolução

- Mantenha apenas parte da condicionante, deixe a outra de lado; até que ponto fica assim determinada a incógnita?
- É possível variar a incógnita, ou os dados, ou todos eles, se necessário, de tal maneira que fiquem mais próximos entre si?
- Utilizou todos os dados?
- Utilizou toda a condicionante?
- Levou em conta todas as noções essenciais implicadas no problema?



# Execução do plano

- Ao executar o seu plano de resolução, verifique cada passo
- É possível verificar claramente que o passo está correto?
- É possível demonstrar que ele está correto?



# Retrospecto

- É possível verificar o resultado?
- É possível verificar o argumento?
- É possível chegar ao resultado por um caminho diferente?
- É possível perceber isto em um relance?
- É possível utilizar o resultado, ou o método, em algum outro problema?



# Solução de problemas segundo Pozo

- **Pozo (1994) - Solução do problema é vista como uma forma de aprender a aprender**
  - **Resolver para aprender e aprender para resolver**
- **Resolução de problemas não pode ser vista apenas como uma técnica a ser ensinada, são feitas considerações a respeito do ensino de Resolução de Problemas como um conteúdo procedimental da Educação Básica**
  - **Aponta como responsabilidade de todos os componentes curriculares o trabalho com a solução de problemas**

# Abstração na Solução de Problemas

- **Abstração é um mecanismo importante no processo de Solução de Problemas**
  - **Permite simplificar a realidade e representar os aspectos mais relevantes de um problema e sua solução**
- **Compreende os seguintes aspectos:**
  - **Dados:** Descrevem as informações envolvidas na solução de um problema (dados de entrada e saída);
  - **Processos:** Abstrações que permitem definir os algoritmos que descrevem a solução de um problema;
  - **Técnicas de Construção de Algoritmos:** Técnicas que permitem obter a solução de problemas complexos de forma mais simples.

# Problema na Computação

- É uma descrição geral de um conjunto de dados de entrada e um conjunto de requisitos que uma solução satisfatória deve apresentar
- Os seguintes passos precisam ser desenvolvidos para solucionar os problemas computacionais:
  - Interpretar o problema proposto
  - Identificar dados disponíveis para a sua resolução
  - Definir seu objetivo. Pergunte-se: “Qual o resultado que se deseja alcançar?”
  - Definir passos necessários para atingir o objetivo

# Pensamento Computacional

- **Abordagem para solução de problemas, que utiliza conceitos básicos da Computação**
- **As atividades desenvolvidas têm como finalidade:**
  - **Construção do pensamento lógico**
  - **Habilidade de reconhecimento de padrões**
  - **Desenvolvimento do raciocínio por meio dos quatro pilares**

# Pensamento Computacional

- Identificar um problema complexo e quebrá-lo em partes menores e mais fáceis de gerenciar (**Decomposição**)
- Problemas menores são analisados individualmente com maior profundidade, identificando problemas parecidos que já foram solucionados anteriormente (**Reconhecimento de Padrões**)
- Focar apenas nos detalhes que são importantes, enquanto informações irrelevantes são ignoradas (**Abstração**)
- Passos ou sequências de instruções são criadas para resolver cada subproblema encontrado (**Algoritmos**)
- Sequências de instruções escritas em uma determinada linguagem de programação (**Programas ou softwares**)



# Abstrações de Dados

- Os dados podem ser muito simples (um número, por exemplo - variável simples), ou complexos (uma pilha de provas de alunos, um mapa, uma ficha de paciente de hospital, etc).
- Para processar uma pilha de provas para, por exemplo, ordenar de alguma forma, precisamos usar uma abstração para esta pilha
- Pilha de provas é composta por unidades menores, que são as provas compostas de uma coleção de informações (nome do aluno, questões, respostas e nota)
- Vários níveis de abstração que podem ser usados para resolver um problema: enxergar a pilha como um todo, ou selecionar uma das provas, ou selecionar uma informação de uma das provas
- O nível de abstração escolhido dependerá do que se quer realizar em cada passo do algoritmo

# Abstrações para descrever Algoritmos

- **Técnicas para descrever as soluções em forma de algoritmos**
- **Um algoritmo é composto por instruções que devem ser executadas de uma forma e na ordem definida para se atingir a solução desejada**
- **Um algoritmo é, portanto, uma combinação de instruções usando operadores de composição, escolha e repetição**
  - **Composição:** permite juntar vários passos na descrição de um algoritmo (sequencial, paralela, etc)
  - **Escolha:** permite definir pontos de escolha, o próximo passo a ser executado depende da situação atual do processo
  - **Repetição:** permite que ações sejam repetidas de forma controlada, utilizando, por exemplo, laços ou recursão

# Técnicas para construir Algoritmos

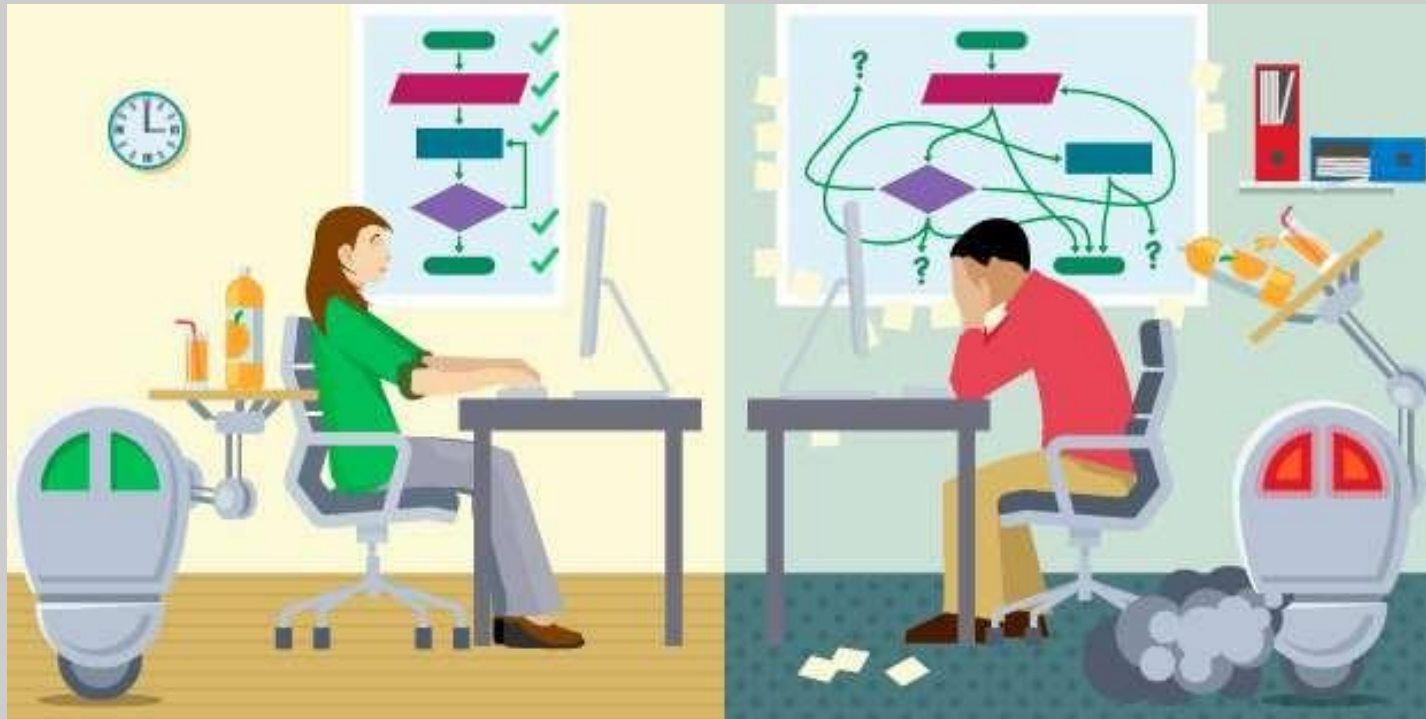
- **Técnicas que permitem desenvolver uma solução para o problema com mais eficiência e facilidade:**
  - **Decomposição:** consiste em decompor o problema em problemas menores
    - **Refinamento e Recursão**
  - **Generalização:** consiste em construir uma solução (algoritmo) mais genérico a partir de outro, permitindo que este novo algoritmo seja utilizado em outros contextos
    - **Reconhecimento de Padrões e Reuso**
  - **Transformação:** consiste em utilizar a solução de um problema para solucionar outro
    - **Reuso**

# Avaliando soluções

- Soluções projetadas devem estar adequadas ao seu objetivo
- Garantir que a solução faça o trabalho para o qual foi projetada e pensar em como ela pode ser melhorada
- Uma vez escrito, um algoritmo deve ser verificado para garantir que:
  - é facilmente entendido - é totalmente decomposto?
  - atende a qualquer critério de projeto?
  - está completo - resolve todos os aspectos do problema?
  - é eficiente - resolve o problema, fazendo o melhor uso dos recursos disponíveis (por exemplo, o mais rápido possível / usando menos espaço)?
  - atende a qualquer critério de projeto?
- Se um algoritmo atender a esses critérios, provavelmente funcionará bem.
- O algoritmo pode então ser programado.

# Avaliando soluções

- Falha na avaliação do algoritmo pode dificultar a criação de um programa
  - A avaliação ajuda a garantir que o menor número de dificuldades possível seja enfrentado ao programar a solução



Fonte: <https://www.bbc.co.uk/bitesize/guides/zssk87h/revision/1>

# **PENSAMENTO COMPUTACIONAL**

**Resolução de problemas**