



# Algoritmo

---

- **Algoritmo** é a forma organizada de expressar uma seqüência de passos que visam atingir um objetivo definido. **Algoritmo** é a lógica necessária para o desenvolvimento de um programa.
- Apesar do nome estranho, os algoritmos são muito comuns no nosso cotidiano, como por exemplo, em uma receita de bolo. Nela estão escritos os ingredientes necessários e a seqüências de passos ou ações a serem cumpridos para que se consiga fazer um determinado tipo de bolo.



# Algoritmo

---

- Descrevemos então uma atividade bem cotidiana: trocar uma lâmpada. Apesar de parecer óbvia demais, muitas vezes fazemos este tipo de atividade inconscientemente, sem percebermos os pequenos detalhes.
- Vejamos como seria descrevê-la passo a passo:
  - pegar uma escada;
  - posicionar a escada embaixo da lâmpada;
  - buscar uma lâmpada nova;
  - subir na escada;
  - retirar a lâmpada velha;
  - colocar a lâmpada nova.



# Algoritmo

---

- Para se trocar a lâmpada, é seguida uma determinada seqüência de ações, representadas através desse algoritmo. Como isso pode ser seguido por qualquer pessoa, estabelece-se aí um padrão de comportamento. O sequenciamento tem por objetivo reger o fluxo de execução, determinando qual ação vem a seguir.
- O algoritmo anterior tem um objetivo bem específico: trocar uma lâmpada. E se a lâmpada não estiver queimada? O algoritmo faz com ela seja trocada do mesmo modo, não prevendo essa situação. Para solucionar este problema, podemos efetuar um teste **seletivo**, verificando se a lâmpada está ou não queimada:



# Algoritmo

---

- pegar uma escada;
- posicionar embaixo da lâmpada;
- buscar uma lâmpada nova;
- ligar o interruptor;
- **se a lâmpada não acender, então:**
  - subir na escada;
  - retirar a lâmpada velha;
  - colocar a lâmpada nova.



# Algoritmo

---

- Dessa forma, algumas ações estão ligadas à condição (lâmpada não acender). No caso da lâmpada acender, as três linhas:
  - subir na escada;
  - retirar a lâmpada velha;
  - colocar a lâmpada nova.

**não** serão executadas.

- Diante do conhecimento de um teste seletivo como reescrever o algoritmo para otimizá-lo?



# Algoritmo

---

- É possível otimizar o algoritmo anterior, uma vez que buscamos a escada e a lâmpada sem saber se serão necessárias:
  - ligar o interruptor;
  - **se a lâmpada não acender, então:**
    - pegar uma escada;
    - posicionar a escada embaixo da lâmpada;
    - buscar uma lâmpada nova;
    - subir na escada;
    - retirar a lâmpada velha;
    - colocar a lâmpada nova.



# Algoritmo

---

- Podemos considerar ainda que a lâmpada nova pode **não** funcionar. Nesse caso devemos trocá-la novamente, quantas vezes for necessário, até que a lâmpada acenda:

# Algoritmo



---

- ligar o interruptor;
- **se** a lâmpada não acender, então:
  - pegar uma escada;
  - posicionar a escada embaixo da lâmpada;
  - buscar uma lâmpada nova;
  - subir na escada;
  - retirar a lâmpada velha;
  - colocar a lâmpada nova;
  - **se** a nova lâmpada não acender, então:
    - retirar a lâmpada;
    - colocar outra lâmpada;
    - **se** a nova lâmpada não acender, então:
    - ...





# Algoritmo

---

- Observamos que o teste da lâmpada nova é efetuado por um conjunto de ações:
  - **se a nova lâmpada não acender** então:
    - retire a lâmpada
    - coloque outra lâmpada
- Em vez de escrevermos várias vezes este conjunto de ações, podemos alterar o fluxo sequencial de execução do programa, de forma que, após executar a ação “coloque outra lâmpada”, voltemos a executar a ação - -  
- **se a nova lâmpada não acender.**



# Algoritmo

---

- Precisa-se então determinar um limite para tal repetição, para garantir que ela cesse quando a lâmpada finalmente acender:
  - **enquanto** a lâmpada não acender, faça:
    - retire a lâmpada
    - coloque outra lâmpada



# Algoritmo

---

- Uma versão final do algoritmo, que repete ações até alcançar o seu objetivo: trocar a lâmpada queimada por uma que funcione, é apresentada abaixo.
  - ligar o interruptor;
  - **se** a lâmpada não acender, então:
    - pegar uma escada;
    - posicionar a escada embaixo da lâmpada;
    - buscar uma lâmpada nova;
    - subir na escada;
    - retirar a lâmpada velha;
    - colocar a lâmpada nova;
    - **enquanto** a lâmpada não acender, faça:
      - retirar a lâmpada;
      - colocar outra lâmpada.

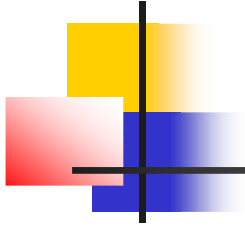


# Algoritmo

---

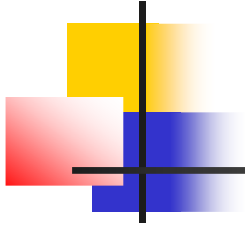
- Até agora, estamos efetuando a troca de uma única lâmpada. Todo o procedimento poderia ser repetido 10 vezes, por exemplo, no caso de quisermos trocar 10 lâmpadas.
- Inicialmente, tínhamos um pequeno conjunto de ações que deveriam ser executadas uma a uma (**estrutura sequencial**).
- Através de uma condição, incluímos uma **estrutura de seleção**.
- Na necessidade de repetir um determinado trecho do algoritmo, construiu-se no final uma **estrutura de repetição**.

# EXEMPLO DE ALGORITMO. soma de dois números



- Informar os dois números a serem somados
- Somar os dois números
- Informar ao interessado a soma dos dois números

# EXEMPLO DE ALGORITMO. soma de dois números (cont.)



- Leia os dois números (N1 e N2)
- Some N1 com N2 e guarde o resultado em S
- Escreva o resultado (S)



# Algoritmos

---

- Ações devem ser executadas numa seqüência
- Tipos de ações que podem estar presentes:
  - Testes seletivos (se <condição> então ...)
  - Ações de repetição (enquanto <condição> ...)
  - Ações de leitura (leia ...)
  - Ações de escrita (escreva ...)
  - Instruções de processamento