

Curso Técnico Superior Profissional

Algoritmos e Fundamentos de Programação

Introdução ao Algoritmos

2018-2019

José A. Sena Pereira
Patrícia Dinis

❑ Objetivos

- Introdução aos algoritmos e à programação.
 - ✓ Pensar algoritmicamente;
 - ✓ Codificar um algoritmo em pseudocódigo;
 - ✓ Compreender alguns tipos de algoritmos;
 - ✓ Comparar algoritmos;
 - ✓ Conhecer algumas estruturas de dados;
 - ✓ Utilizar estruturas genéricas para a resolução de problemas.

□ Definições

■ Algoritmo

- ✓ Representa uma sequência finita e não ambígua de comandos bem definidos para a resolução de um determinado problema.

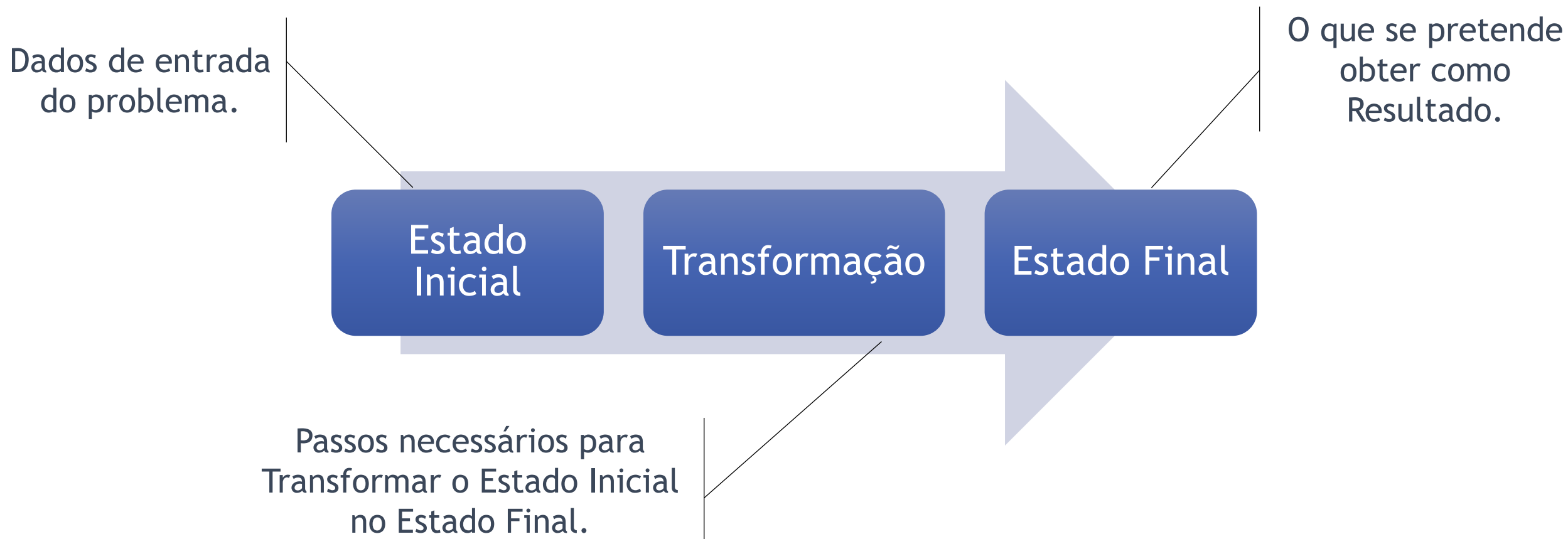
Um algoritmo deve representar uma solução única para um problema, no entanto isso não impede que um problema tenha múltiplas soluções.

■ Programa

- ✓ Um programa é a concretização de um algoritmo, para um determinado ambiente, utilizando uma Linguagem de Programação.

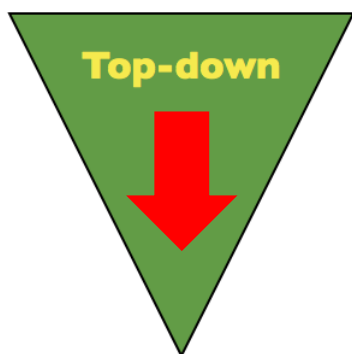
❑ Algoritmo

- Quando se faz a análise de um problema é importante considerar três componentes:



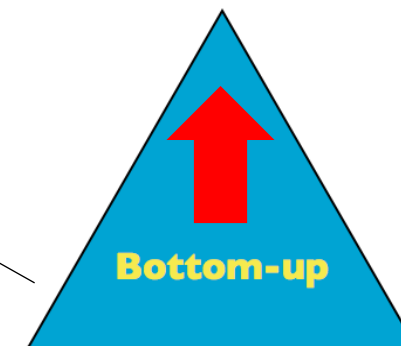
❑ Abordagens utilizadas

- Top-down e Bottom-up são estratégias de processamento de informação e ordenação do conhecimento, utilizado em diversos campos, incluindo software, humanística e teorias científicas (Sistemia), e gestão e organização.



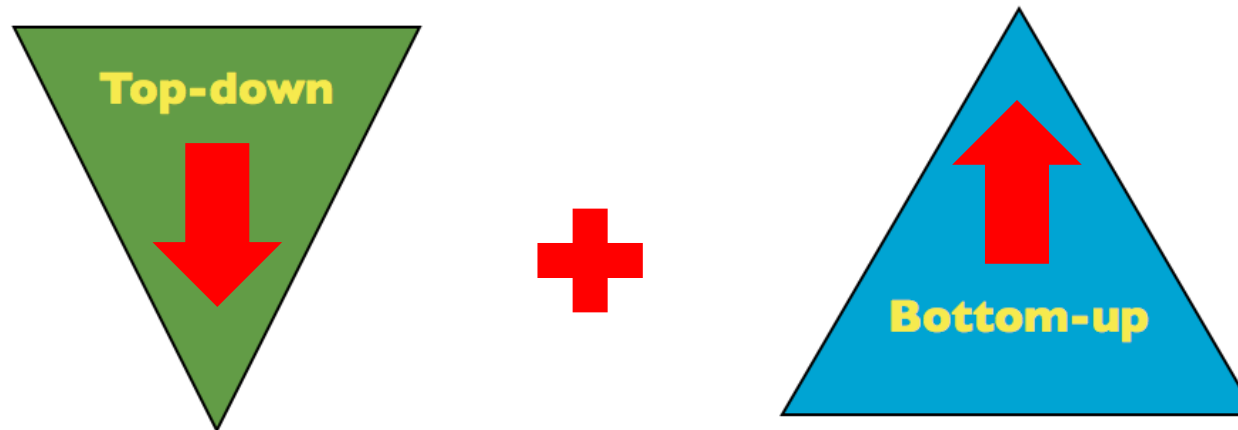
Partir de um problema geral e procurar chegar aos detalhes.

Partir de problemas pequenos, visualizar a solução, como algo mais amplo e genérico.



❑ Abordagens utilizadas

- De uma maneira geral a utilização das duas abordagens (i.e., Top-down e Bottom-up) representam um papel importante no processo de desenvolvimento de software.



□ Técnicas utilizadas




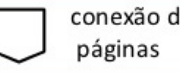


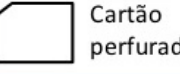




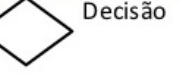


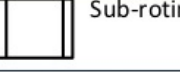



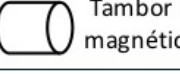
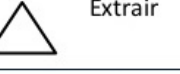



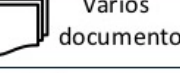


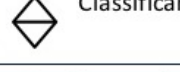
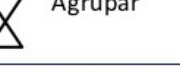


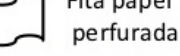
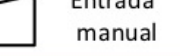
- Linguagem natural
 - ✓ Os algoritmos são expressos diretamente em linguagem natural
- Pseudocódigo
 - ✓ É um código de escrita em que se utilizam termos convencionais para indicar as instruções do programa.

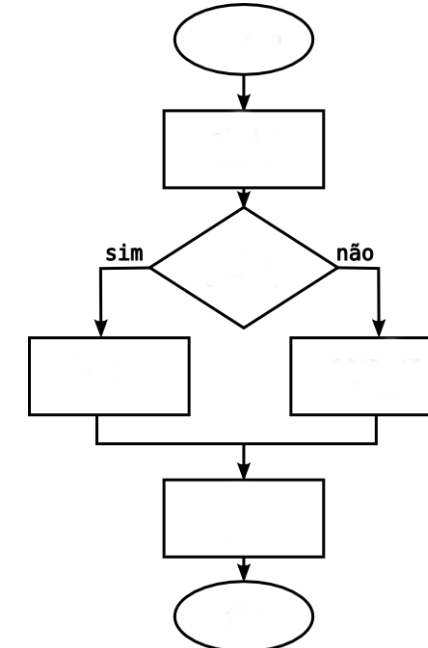
```
nome: título { variáveis de entrada, variáveis de saída}
variáveis de entrada: lista de variáveis de entrada e sua descrição
variáveis de saída: lista de variáveis de saída e sua descrição
variáveis internas: lista de variáveis internas e sua descrição
INICIO
  decomposição da operação numa sequencia simples de instruções
FIM
```

☐ Técnicas utilizadas

▪ Fluxograma

- ✓ É um diagrama representativo do fluxo de ações de um programa, através de símbolos, que representam os diferentes tipos de ações e o seu encadeamento na sequência do programa.

 Operação	 Decisão	 Input Output	 conexão de páginas
 Inspeção	 Preparação	 Cartão perfurado	 Preparação
 Demora	 Terminal	 Memória principal	 Decisão
 Transporte	 Junção	 Sub-rotina	 Display
 Armazenamento	 "Ou"	 Tambor magnético	 Extrair
 Ações combinadas	 Disco magnético	 Conector	 Vários documentos
 Processo	 Fita magnética	 Classificar	 Agrupar
 Operação Manual	 Documento	 Fita papel perfurada	 Entrada manual



❑ Exemplo de um Algoritmo

- Abordagem Top-down
 - ✓ Especificar o algoritmo de uma forma simples e direta em linguagem natural, e ir detalhando cada um dos seus passos.
- Exemplo: Mudar um pneu
 1. Preparar material;
 2. Retirar pneu furado;
 3. Colocar pneu suplente;
 4. Guardar material;

❑ Exemplo de um Algoritmo

1. Preparar material;
 1. Retirar pneu suplente do porta bagagens;
 2. Retirar macaco e ferramentas da caixa;

2. Retirar pneu furado;
 1. Desenroscar os parafusos;
 2. Içar o carro com o macaco;
 3. Remover os parafusos;
 4. Retirar o pneu;

❑ Exemplo de um Algoritmo

3. Colocar pneu suplente;

1. Colocar o pneu;
2. Colocar parafusos;
3. Baixar macaco;
4. Apertar parafusos;

4. Guardar material;

1. Guardar pneu furado no porta-bagagens;
2. Guardar macaco e ferramentas;

❑ Exemplo de um Algoritmo

- Utilização da técnica do pseudocódigo
 - ✓ Algoritmo que recebe um número, calcula e mostra o seu dobro.

```

Nome: calcular_Dobro {num, dobro}
Variáveis de entrada: num: número inteiro
Variáveis internas: dobro: número inteiro
INÍCIO
  num = dobro = 0
  LER num
  dobro = 2 * num
  ESCREVER dobro
FIM
  
```

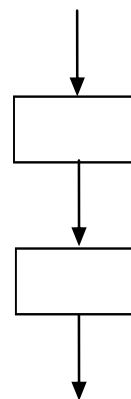
❑ Técnica do Pseudocódigo

- Sequência de Instruções

Pseudocódigo

```
INICIO
    op1
    op2
    . . . .
FIM
```

Fluxograma



Linguagem C

```
{
    op1;
    op2;
}
```

❑ Técnica do Pseudocódigo

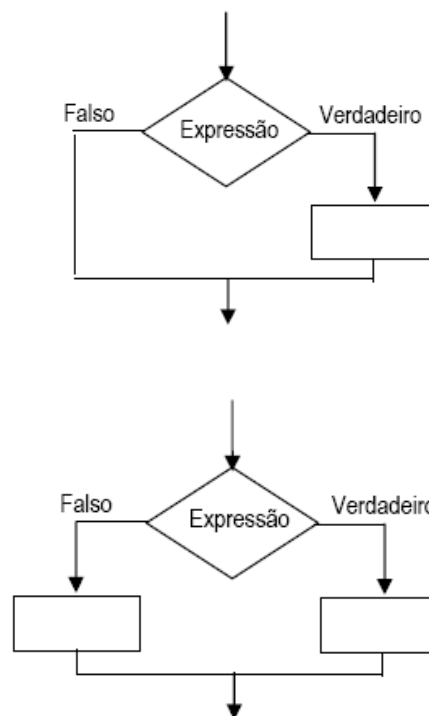
▪ Instrução condicional - Seleção

Pseudocódigo

```
SE (condição) ENTÃO
    instrução
FIMSE
```

```
SE (condição) ENTÃO
    instrução1
SENÃO
    instrução2
FIMSE
```

Fluxograma



Linguagem C

```
if (condição)
    instrução;
```

```
if (condição)
    instrução1;
else
    Instrução2;
```

❑ Técnica do Pseudocódigo

▪ Instrução de Repetição

Pseudocódigo

REPETIR

Sequencia de instruções

ENQUANTO condição

REPETIR ENQUANTO condição

Sequencia de instruções

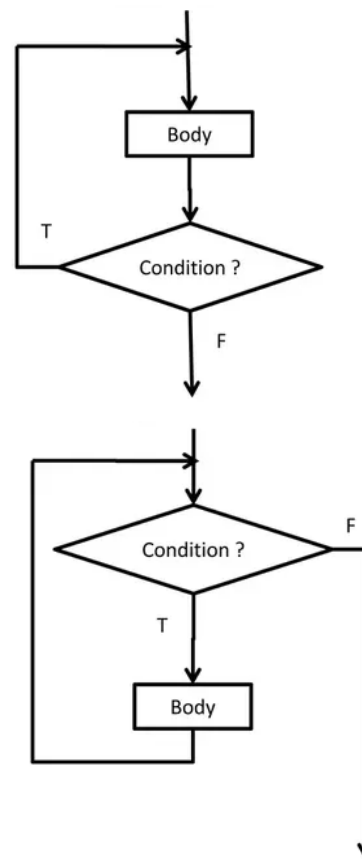
FIMREPETIR

REPETIR X Vezes

Sequencia de instruções

FIMREPETIR

Fluxograma



Linguagem C

```
do{
    op1;
    op2;
} while (cond);
```

```
while (cond)
{
    op1;
    op2;
}
```

```
for(int i=0 ;i<X;i++)
{
    op1;
    op2;
}
```

❑ Técnica do Pseudocódigo

■ VANTAGENS:

- ✓ Usa o português como base;
- ✓ Pode-se definir quais e como os dados vão estar estruturados;
- ✓ Passagem quase imediata do algoritmo para uma linguagem de programação qualquer.

■ DESVANTAGENS:

- ✓ Exige a definição de uma linguagem não real para trabalho;
- ✓ Não padronizado;

❑ Técnica do pseudocódigo

▪ Exemplo / Exercício

- ✓ Elabore um algoritmo que receba quatro números inteiros, calcule e mostre a soma desses números.



❑ Técnica do pseudocódigo

- Algoritmo para somar os quatro números.

Nome: `somar {n1, n2, n3, n4, soma}`

Variáveis de entrada: `n1, n2, n3, n4`: número inteiro

Variáveis internas: `soma`: número inteiro

INÍCIO

`n1 = n2 = n3 = n4 = soma = 0`

`LER n1, n2, n3, n4`

`soma = n1 + n2 + n3 + n4`

`ESCREVER "Soma igual a:" + soma`

FIM

☐ Técnica do Pseudocódigo

▪ Exemplo / Exercício

- ✓ Elaborar um algoritmo em pseudocódigo que resolve a equação $ax + b = 0$ e que mostre o seu resultado.



❑ Técnica do Pseudocódigo

■ Resolução.

Nome: `resolveEquação` {a, b, x }

Variáveis internas:

a: número real que representa o valor de **a** na equação.

b: número real que representa o valor de **b** na equação.

X: número real que representa o valor a calcular na equação.

INICIO

ESCREVER ("Introduza o valor de a: ")

LER (a)

ESCREVER ("Introduza o valor de b: ")

LER(b)

SE $a \neq 0$ ENTÃO

$x = -b/a$

ESCREVER (X)

SENÃO

ESCREVER("Não tem solução")

FIMSE

FIM

❑ Técnica do Pseudocódigo

■ Exemplo / Exercício

- ✓ Elaborar um algoritmo em que um utilizador introduz um número e outro utilizador vai tentar adivinhar esse número, introduzindo números até acertar. Só parará de introduzir números quando acertar.



❑ Técnica do Pseudocódigo

▪ Resolução

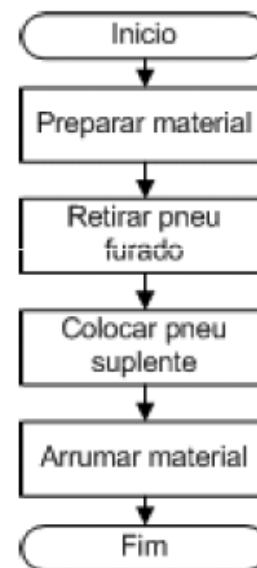
```

Nome: adivinha{numero, palpite}
variáveis internas: numero, palpite: número inteiro
    numero - guarda o valor a adivinhar.
    palpite - valor introduzido pelo jogador
INICIO
    ESCREVER("jogador1, Introduza um numero: ")
    LER(numero)
    REPETIR
        ESCREVER("jogador2, Introduza um numero: ")
        LER(palpite)
    ENQUANTO (numero <> palpite)
        ESCREVER("Acertou")
FIM

```

❑ Técnica do Fluxograma

- Exemplo: Mudar um pneu.



- Exemplo / Exercício

✓Elabore um algoritmo que consiste em decidir o que fazer mediante o estado do tempo. Se estiver a chover fico em a casa a ver televisão senão vou jogar golf.



❑ Técnica do Fluxograma

- Algoritmo que consiste em decidir o que fazer mediante o estado do tempo.



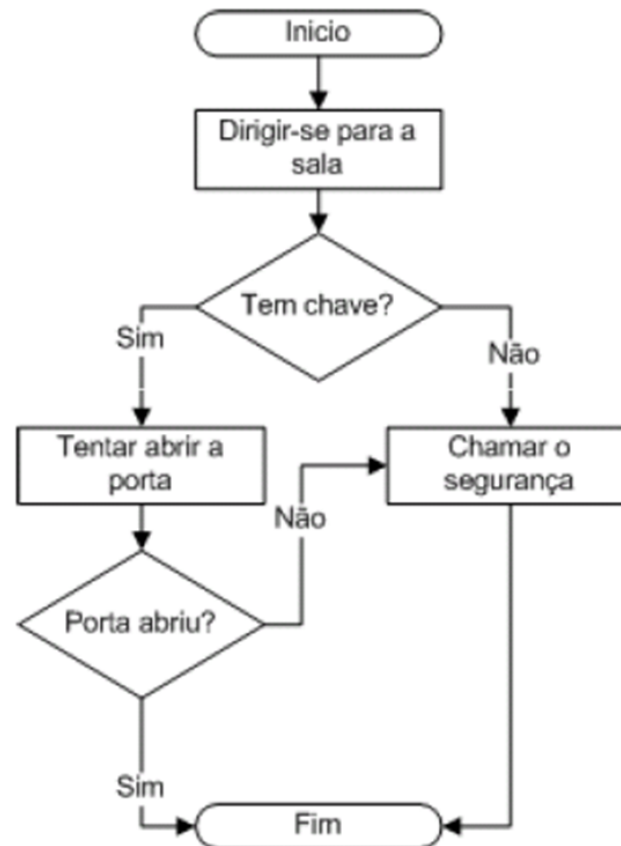
❑ Técnica do Fluxograma

- Representar num fluxograma o algoritmo para uma possível solução do seguinte problema:
 - Entrada na sala de aula por parte do professor para iniciar uma aula:
 - ✓ Abrir a sala caso tenha chave;
 - ✓ Se não tiver chave é necessário chamar o segurança, que abrirá a porta;
 - ✓ A chave pode não funcionar na porta, e nesse caso é também necessário chamar o segurança para a abrir;



❑ Técnica do Fluxograma

- Possível solução do problema de entrada na sala de aula por parte do professor para iniciar uma aula.



❑ Técnica do Fluxograma

■ VANTAGENS:

- ✓ Uma das ferramentas mais conhecidas;
- ✓ Figuras dizem muito mais que palavras;
- ✓ Padrão mundial.

■ DESVANTAGENS:

- ✓ Pouca atenção aos dados, não oferecendo recursos para descrevê-los ou representá-los;
- ✓ Complica-se à medida que o algoritmo cresce.