



**FG**  
Faculdade dos Guararapes  
LAUREATE INTERNATIONAL UNIVERSITIES



**LAUREATE**  
INTERNATIONAL  
UNIVERSITIES

# Lógica de Programação

## Capítulo 1

Introdução

# Introdução

- ▶ Nesta aula
  - ▶ Introdução à Lógica de Programação
  - ▶ Algoritmizando a Lógica
  - ▶ Conceitos e Exemplos de Algoritmos
  - ▶ Noções de Fluxo de Controle

# Noções de Lógica

- ▶ A lógica de programação é importante para aquelas pessoas que desejam desenvolver programas, aplicativos e sistemas. Ela permite que o programador defina uma sequência lógica a seguir para o desenvolvimento de uma aplicação em uma determinada linguagem.
- ▶ A lógica de programação permite que uma pessoa organize os pensamentos que ela tem, para transformá-lo em um objetivo, alcançado através da programação.

# Sequência Lógica

- ▶ Os pensamentos que você tem, como por exemplo, incluir um espaço adicional de publicidade em seu blog, precisam de uma sequência lógica a ser seguida para ser construído, portanto conceituamos que sequência lógica são os passos que devemos seguir para até encontrarmos a solução para aquilo que queremos construir ou até mesmo para solucionar um problema encontrado no código,

# Algoritmo

- ▶ É o pensamento descrito como uma seqüência de passos que visam atingir um objetivo
- ▶ Algoritmos no dia-a-dia: Receita de bolo, orientação para se chegar em algum endereço
- ▶ Qual sua importância na programação?
  - ▶ Representar o raciocínio, independentemente de detalhes computacionais, que podem ser acrescentados mais tarde
  - ▶ Focalizar primeiro na resolução algorítmica do problema, possibilitando depois codificá-la em qualquer linguagem

# Exemplos

- ▶ Trocar uma lâmpada
  - ▶ Seqüenciação

## Algoritmo 1.1:

pegar uma escada;  
posicionar a escada embaixo da lâmpada;  
buscar uma lâmpada nova;  
subir na escada;  
retirar lâmpada velha;  
colocar lâmpada nova.

# Exemplos

- ▶ Trocar uma lâmpada SE estiver queimada
  - ▶ Seleção (Decisão)

## Algoritmo 1.2:

pegar uma escada;  
posicionar a escada embaixo da lâmpada;  
buscar uma lâmpada nova;  
acionar o interruptor;  
se a lâmpada não acender, então  
  subir na escada;  
  retirar lâmpada queimada;  
  colocar lâmpada nova.

# Exemplos

- ▶ Trocar uma lâmpada SE estiver queimada (v. 2)
  - ▶ Seleção (Decisão)

## Algoritmo 1.3: Evita buscar a escada e lâmpada

acionar o interruptor;  
se a lâmpada não acender, então  
pegar uma escada;  
posicionar a escada embaixo da lâmpada;  
buscar uma lâmpada nova;  
acionar o interruptor;  
subir na escada;  
retirar lâmpada queimada;  
colocar lâmpada nova.



# Exemplos

- ▶ Trocar uma lâmpada SE estiver queimada (v. 3)
  - ▶ Seleção (Decisão)

## Algoritmo 1.4: Re-teste depois da troca

```
acionar o interruptor;  
  se a lâmpada não acender, então  
    pegar uma escada;  
    posicionar a escada embaixo da lâmpada;  
    buscar uma lâmpada nova;  
    acionar o interruptor;  
    subir na escada;  
    retirar lâmpada queimada;  
    colocar lâmpada nova;  
  se a lâmpada não acender, então  
    retirar lâmpada queimada;  
    colocar lâmpada nova;  
  se a lâmpada não acender, então
```

...

# Exemplos

- ▶ Trocar uma lâmpada SE estiver queimada (v. 4)
  - ▶ Repetição

## Algoritmo 1.5: Re-teste depois da troca (por repetição)

```
acionar o interruptor;  
  se a lâmpada não acender, então  
    pegar uma escada;  
    posicionar a escada embaixo da lâmpada;  
    buscar uma lâmpada nova;  
    acionar o interruptor;  
    subir na escada;  
    retirar lâmpada queimada;  
    colocar lâmpada nova;  
  enquanto a lâmpada não acender, faça  
    retirar lâmpada queimada;  
    colocar lâmpada nova;
```

# Exemplos

- ▶ Trocar **10** lâmpadas SE estiverem queimadas
  - ▶ Repetição

## Algoritmo 1.6: Escrever 10 vezes

acionar o interruptor do **primeiro** soquete;  
se a lâmpada não acender, então  
pegar uma escada;  
posicionar a escada embaixo da lâmpada;  
buscar uma lâmpada nova;  
acionar o interruptor;  
subir na escada;  
retirar lâmpada queimada;  
colocar lâmpada nova;  
enquanto a lâmpada não acender, faça  
retirar lâmpada queimada;  
colocar lâmpada nova;  
acionar o interruptor do **segundo** soquete;  
...

# Exemplos

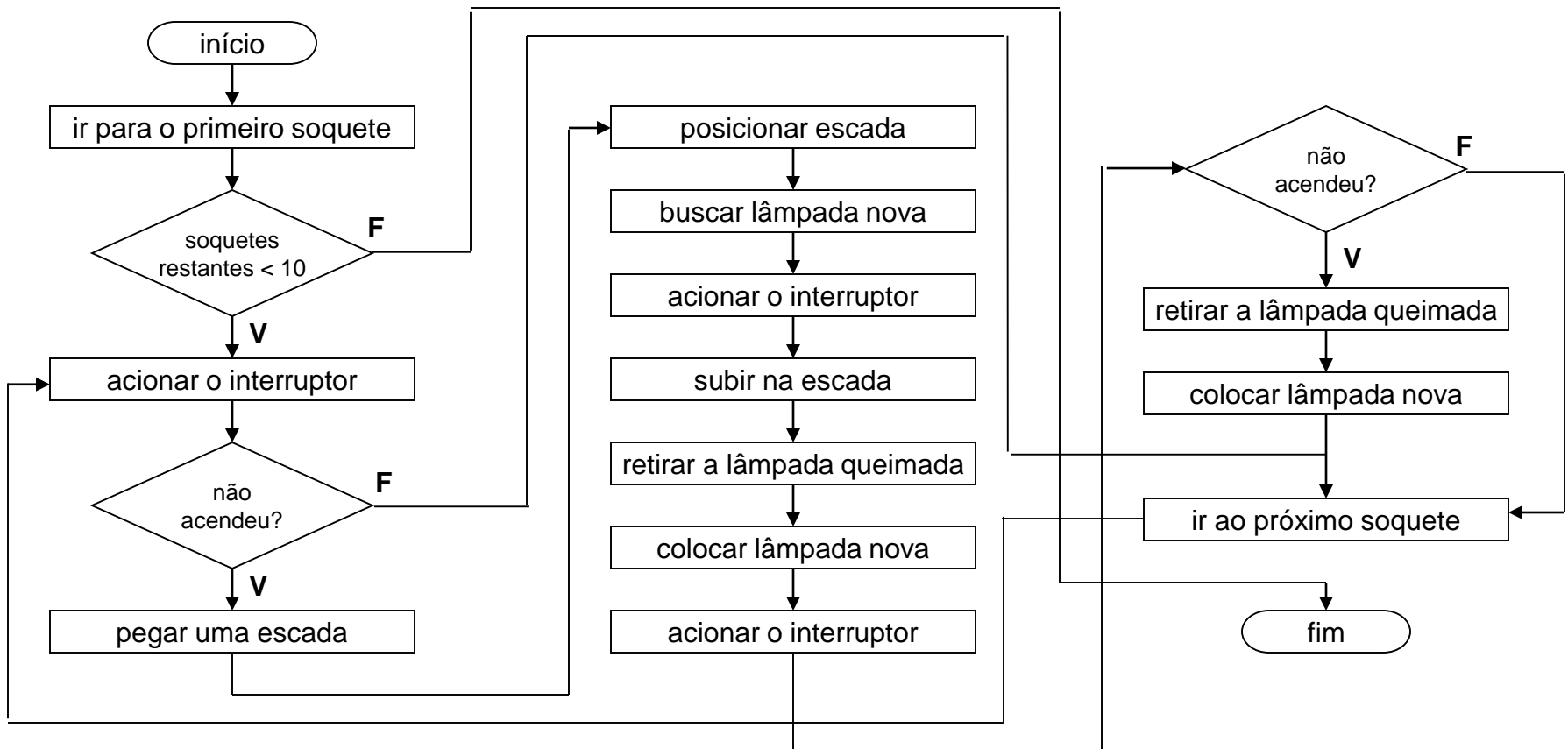
- ▶ Trocar **10** lâmpadas SE estiverem queimadas (v. 2)
  - ▶ Repetição

## Algoritmo 1.7: Contagem de trocas

ir até o interruptor do **primeiro** soquete;  
enquanto a quantidade de soquetes testados for menor que 10, faça  
acionar o interruptor;  
se a lâmpada não acender, então  
pegar uma escada;  
posicionar a escada embaixo da lâmpada;  
buscar uma lâmpada nova;  
acionar o interruptor;  
subir na escada;  
retirar lâmpada queimada;  
colocar lâmpada nova;  
enquanto a lâmpada não acender, faça  
retirar lâmpada queimada;  
colocar lâmpada nova;  
ir até o interruptor do **próximo** soquete;

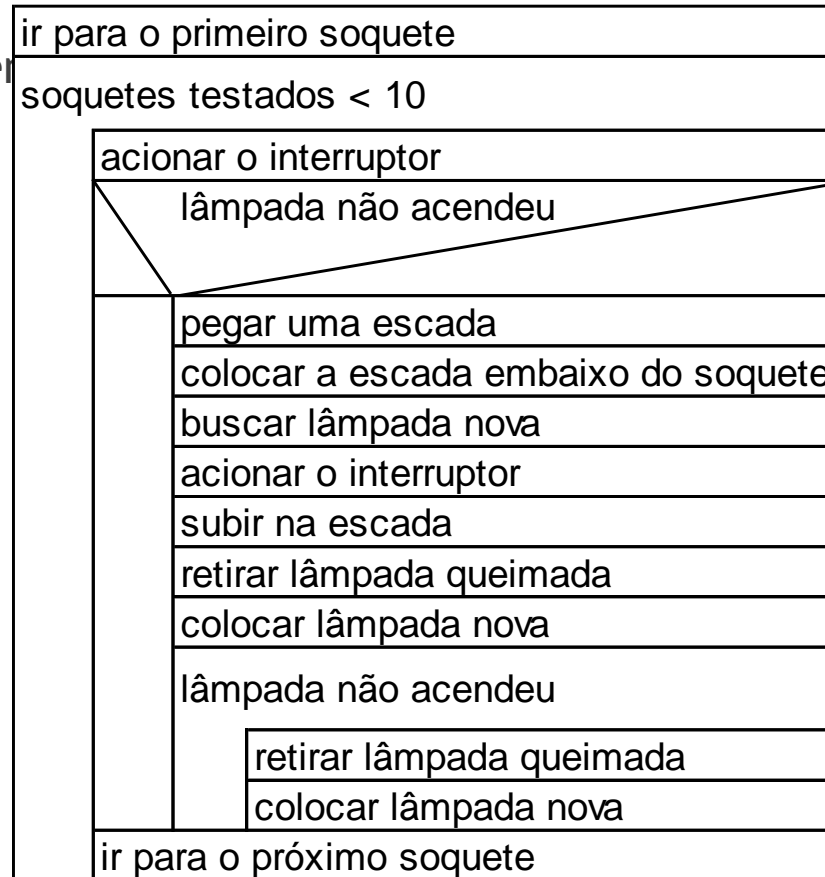
# Formas de Representação

## ► Algoritmo 1.7 em Fluxograma



# Formas de Representação

► Algoritmo 1.7 e



# Formas de Representação

## ▶ Gráficas (Fluxograma e Chapin)

### ▶ Vantagens

- ▶ Maior clareza no fluxo de execução
- ▶ Linguagem visual

### ▶ Desvantagens

- ▶ Requer conhecimento de convenções gráficas
- ▶ Mais trabalhoso em decorrência de seus desenhos
- ▶ Dificuldade para fazer correções

## ▶ Textuais (Português Estruturado)

### ▶ Apresenta mais vantagens, desde que se tomem alguns cuidados:

- ▶ Riqueza gramatical de nossa língua pode levar a ambigüidades
- ▶ A frase “O pregador foi grampeado durante o concerto” tem 8 sentidos diferentes quando pronunciada
- ▶ Para resolver, utilizaremos um conjunto restrito de regras, conhecido como Português Estruturado