

Introdução

Algoritmos I 2014/1

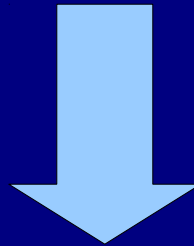
Profa: Daniela Scherer dos Santos

daniela.santos37@ulbra.edu.br

www.danielascherer.com.br

Noções de Lógica

Algumas vezes cometemos enganos no pensamento. Tiramos conclusões precipitadas, fazemos suposições sem respaldo, generalizamos.



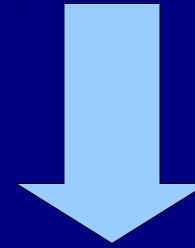
LÓGICA

Estuda a correção do raciocínio

Noções de Lógica



Lógica



estuda e ensina a colocar ordem no
pensamento

Exemplos de pensamentos lógicos:

Todo mamífero é um animal.
Todo cavalo é um mamífero.
Portanto, todo o cavalo é um animal.

Todos os gaúchos são brasileiros.
Paulo é gaúcho.
Portanto, Paulo é brasileiro.



Exemplos de pensamentos ILÓGICOS:

Todas as peças deste caminhão são leves.
Logo, este caminhão é leve.

Vou passar na prova de hoje porque hoje é terça-feira
e até agora nunca reprovei em provas nas terças-feiras.



Existe Lógica no dia a dia?

- ◆ Sempre que falamos ou escrevemos estamos expressando nosso pensamento → precisamos usar a lógica nessas atividades;

A gaveta está fechada.
A caneta está dentro da gaveta.
Precisamos primeiro abrir a gaveta
para depois pegar a caneta.

Exemplos do uso de lógica na
prática:

Todos sabemos, desde pequenos,
que para comer uma bala
precisamos antes tirá-la da
embalagem.

E a Lógica de Programação?

- ◆ Significa o uso correto das leis do pensamento e de processos de raciocínio e simbolização formais na programação de computadores;
- ◆ objetivo: desenvolvimento de técnicas que cooperem para a produção de soluções logicamente válidas e coerentes, que resolvam com qualidade os problemas que se deseja programar.

E a Lógica de Programação?

O objetivo principal do estudo da Lógica da Programação é a construção de ALGORITMOS coerentes e válidos.

E a Lógica de Programação?

O objetivo principal do estudo da Lógica da Programação é a construção de **ALGORITMOS** coerentes e válidos.

Mas o que é um ALGORITMO?



O que é um Algoritmo?



BRIGADEIRO

INGREDIENTES:

- 1 lata de leite condensado;
- 1 colher de sopa de margarina sem sal;
- 7 colheres rasas das de sopa de Nescau ou 4 colheres de sopa de chocolate em pó
- chocolate granulado para fazer bolinhas.

MODO DE PREPARAR:

- 1) Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó;
- 2) Cozinhe em fogo médio e mexa sem parar com uma colher de pau;
- 3) Cozinhe até que o brigadeiro comece a desgrudar da panela;
- 4) Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.

O que é um Algoritmo?

Um algoritmo pode ser definido como **uma sequência de passos que visam atingir um objetivo bem definido**

O que é um Algoritmo?



BRIGADEIRO

INGREDIENTES:

- 1 lata de leite condensado;
- 1 colher de sopa de margarina sem sal;
- 7 colheres rasas de sopa de Nescau ou 4 colheres de sopa de chocolate em pó
- chocolate granulado para fazer bolinhas.

MODO DE PREPARAR:

- 1) Coloque em uma panela funda o leite condensado, a margarina e o chocolate em pó;
- 2) Cozinhe em fogo médio e mexa sem parar com uma colher de pau;
- 3) Cozinhe até que o brigadeiro comece a desgrudar da panela;
- 4) Deixe esfriar bem, então unte as mãos com margarina, faça as bolinhas e envolva-as em chocolate granulado.

O que é um Algoritmo?

Parte de um TUTORIAL sobre
como instalar Office 2013

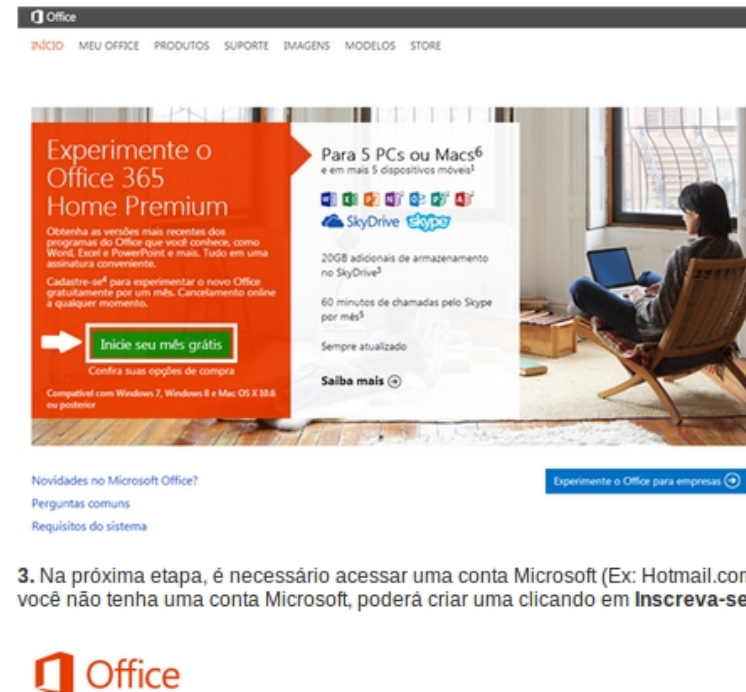


Abaixo explicaremos o procedimento para efetuarmos a instalação do seu Office 2013 Versão de avaliação, válida por 30 dias.

Este procedimento lhe concederá o direito de utilizar o Office 2013 **#gratuitamente** por um período de 30 dias. Após este período, se não for feito o cancelamento, será cobrada uma mensalidade de R\$18,00.

1. Acesse [este link](#) para iniciarmos o procedimento.

2. Clique em **Inicie seu Mês grátis**



3. Na próxima etapa, é necessário acessar uma conta Microsoft (Ex: Hotmail.com, Xbox Live, etc.). Caso você não tenha uma conta Microsoft, poderá criar uma clicando em **Inscrição-se Já** mais abaixo.

O que é um Algoritmo?

- ◆ Quando elaboramos um algoritmo devemos:
 - ◆ especificar ações claras e precisas, que a partir de um estado inicial, produzem um estado final previsível e bem definido;
- ◆ Um algoritmo determina uma norma de execução a ser trilhada, para alcançar, como resultado final, a solução de um problema:
 - ◆ exemplos:
 - ◆ somar os números de 1 até 100;
 - ◆ calcular a tabuada do 9.

Por que é importante construir um algoritmo?

Uma vez concebida uma solução algorítmica para um problema, esta pode ser traduzida para qualquer linguagem de programação.

Como criar um algoritmo?

- ♦ Interpretar o problema;
- ♦ Definir os dados de entrada:
 - ♦ Valores que serão usados para chegar à solução do problema;
 - ♦ São, normalmente, fornecidos pelo usuário → teclado, mouse, leitor de código de barras;

Quais dados são necessários para começar o algoritmo?

Que dados ele vai precisar para executar?

Como criar um algoritmo?

- ◆ Definir o processamento:
 - ◆ procedimento usado para chegar o resultado final;
 - ◆ transforma dados de entrada em dados de saída;
- ◆ Definir os dados de saída;

Quais dados deverão ser mostrados para o usuário?

Como criar um algoritmo?

- Escrever o algoritmo;
- Testar o algoritmo (teste de mesa):
 - efetuar simulações para controlar e verificar o fluxo de execução do algoritmo.

Representação de Algoritmos

- Descrição narrativa;
- Fluxograma;
- Português estruturado.

Representação de Algoritmos

- ◆ Descrição narrativa:
 - ◆ Os passos do algoritmo são descritos em linguagem natural
 - ◆ Exemplo: somar 2 valores
 - ◆ Passo 1: obter o 1º valor
 - ◆ Passo 2: obter o 2º valor
 - ◆ Passo 3: somar os dois valores
 - ◆ Passo 4: mostrar o resultado obtido

Algoritmo 1.1 - Trocar Lâmpada

- 1.pegar uma escada;
- 2.posicionar a escada embaixo da lâmpada;
- 3.buscar uma lâmpada nova;
- 4.subir na escada;
- 5.retirar a lâmpada velha;
- 6.colocar a lâmpada nova.



Algoritmo 1.1 - Trocar Lâmpada

- 1.pegar uma escada;
- 2.posicionar a escada próximo da lâmpada;
- 3.buscar uma lâmpada nova;
- 4.subir na escada;
- 5.remover a lâmpada velha;
- 6.colocar a lâmpada nova.

E se a lâmpada não estivesse queimada?



Algoritmos Naturais - Exemplos

Algoritmo 1.2 - Trocar Lâmpada

- 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 a lâmpada queimada;
 - colocar a lâmpada nova.



Algoritmos Naturais - Exemplos

Algoritmo 1.2 - Trocar Lâmpada

- 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 a lâmpada queimada;
 - colocar a lâmpada nova.

Como podemos melhorar esse algoritmo?



Algoritmos Naturais - Exemplos

Algoritmo 1.3 - Trocar Lâmpada

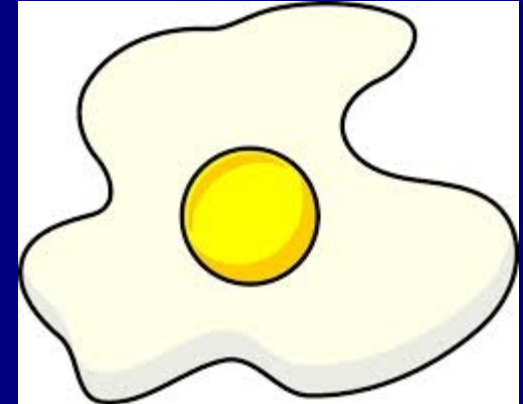
- acionar o interruptor;
- se a lâmpada não acender, então:
 - pegar uma escada;
 - posicionar a escada embaixo da lâmpada;
 - buscar uma lâmpada nova;
 - subir na escada;
 - retirar a lâmpada queimada;
 - colocar a lâmpada nova.



Algoritmos Naturais - Exemplos

Algoritmo 1.4 - Fritar um Ovo

- colocar a frigideira no fogo;
- colocar óleo na frigideira;
- retirar o ovo da geladeira;
- esperar o óleo ficar quente;
- quebrar o ovo separando a casca;
- colocar o conteúdo do ovo na frigideira;
- esperar o ovo ficar frito;
- retirar o ovo da frigideira;
- apagar o fogo.



Exercícios:

- 1) Tendo como exemplo os algoritmos desenvolvidos para solucionar o problema da troca de lâmpadas, elabore um algoritmo que mostre os passos necessários para trocar um pneu furado. Considere o seguinte conjunto de situações:
 - a) trocar o pneu traseiro esquerdo;
 - b) trocar o pneu traseiro esquerdo e, antes, verificar se o pneu reserva está em condições de uso.

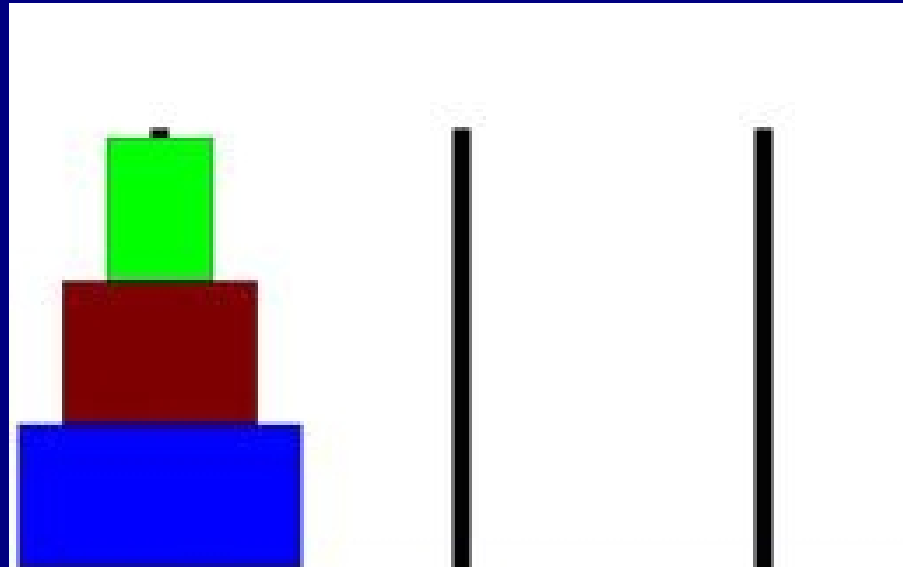
2) Três jesuítas e três canibais precisam atravessar um rio; para tal dispõem de um barco com capacidade para duas pessoas. Por medidas de segurança, não se deve permitir que em alguma margem a quantidade de jesuítas seja inferior à de canibais. Qual a solução para efetuar a travessia com segurança? Elabore um algoritmo mostrando a resposta, indicando as ações que concretizam a solução deste problema.

Exercícios:

3) Elabore um algoritmo que mova 3 discos de uma Torre de Hanói, que consiste em três hastes (a – b – c), uma das quais serve de suporte para três discos de tamanhos diferentes (1 – 2 – 3), os menores sobre os maiores. Pode-se mover um disco de cada vez para qualquer haste, contanto que nunca seja colocado um disco maior sobre um menor. O objetivo é transferir os três discos para outra haste.

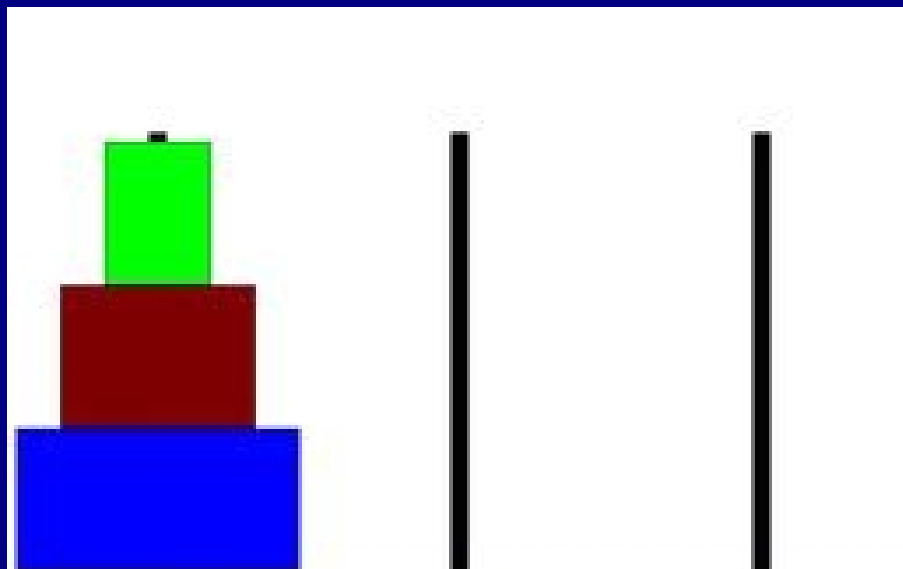
Exercícios:

Torre de Hanói



Exercícios:

Torre de Hanói



Representação de Algoritmos

- ◆ Descrição Narrativa:
 - ◆ Vantagens:
 - ◆ Todos conhecemos bem o português e sabemos fazer uma descrição, uma lista de atividades;




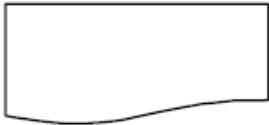
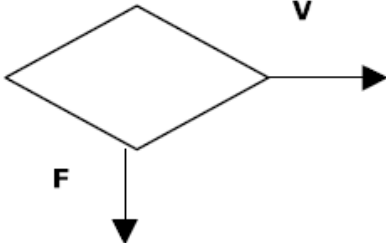
Representação de Algoritmos

- ◆ Descrição Narrativa:
 - ◆ Desvantagens:
 - ◆ Não há um padrão: cada pessoa escreve como quiser;
 - ◆ É impreciso: cada pessoa pode interpretar de uma forma. Exemplo: "leve ao forno até assar";
 - ◆ A escrita pode ficar confusa se a pessoa não souber se expressar bem.

- ◆ Fluxograma:
 - ◆ representação gráfica onde diferentes formas geométricas representam ações (instruções, comandos) distintas que devem ser executadas.

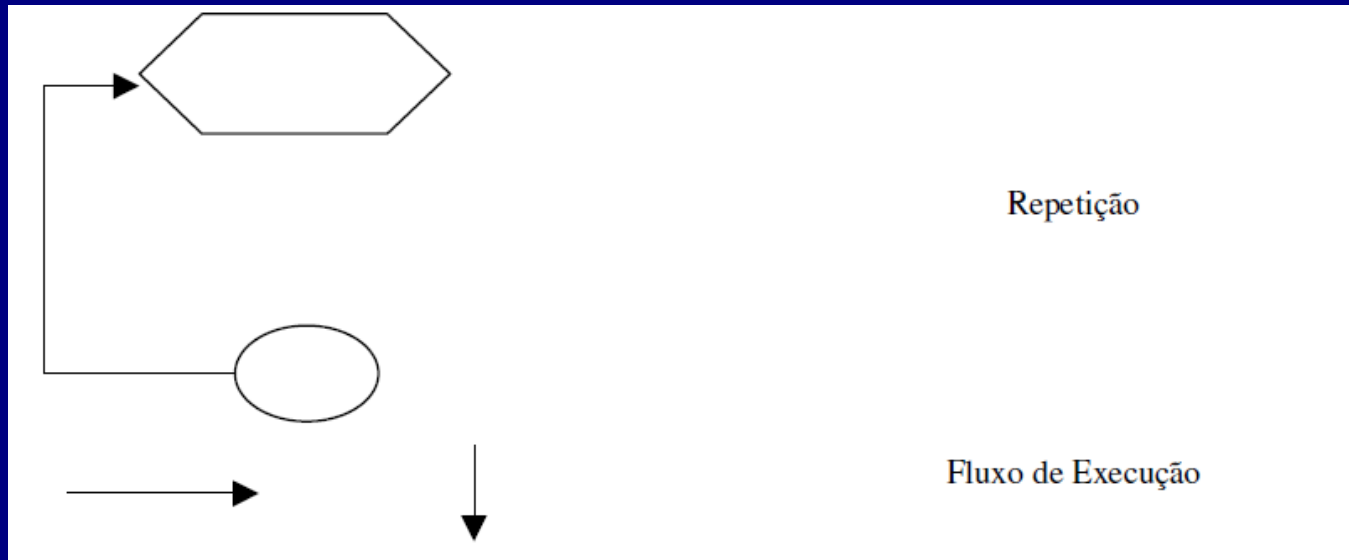
Representação de Algoritmos

Fluxograma:

Símbolo	Descrição/Função
	Início e fim do fluxograma
	Entrada de dados (Leitura)
	Processamento (execução)
	Saída de Dados (Impressão)
	Desvio ou Seleção (Tomada de Decisão)

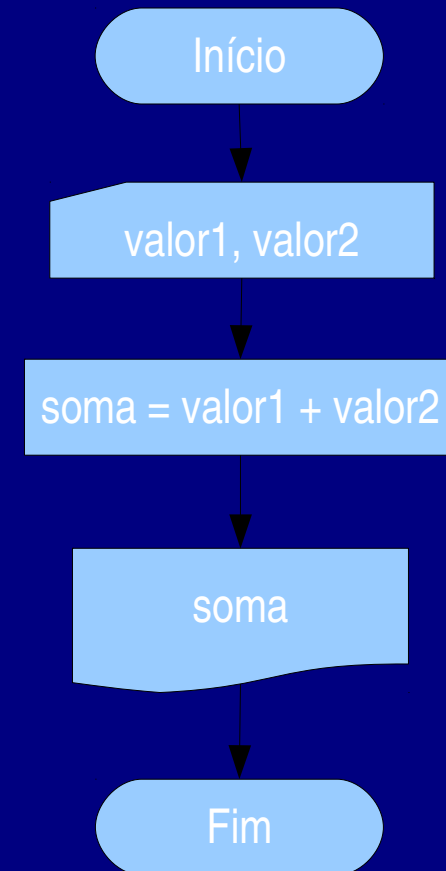
Representação de Algoritmos

- Fluxograma:



Representação de Algoritmos

- ◆ Fluxograma:
 - ◆ Exemplo: Somar dois valores



Representação de Algoritmos

- ◆ Fluxograma:
 - ◆ Vantagens:
 - ◆ ferramenta conhecida em outras áreas;
 - ◆ figuras dizem muito mais que palavras, podem facilitar o entendimento do que deve ser feito no algoritmo;

Representação de Algoritmos

- ◆ Fluxograma:
 - ◆ Vantagens (cont.):
 - ◆ tem um padrão mundial de símbolos bem definidos, logo, poderá ser compreendido em qualquer lugar.

- ◆ Fluxograma:
 - ◆ Desvantagens:
 - ◆ dá pouca atenção aos dados, não oferecendo recursos para descrevê-los ou representá-los. Quando estiver programando, você vai precisar definir o tipo (número inteiro, caractere, etc) de cada variável.

- ◆ Português estruturado:
 - ◆ conjunto de regras que visam restringir e estruturar o uso do português na representação dos algoritmos;
 - ◆ se aproxima da maneira pela qual o fazem as linguagens de programação (Pascal, C, C++, etc), facilitando a futura codificação dos algoritmos.

Representação de Algoritmos

- Estrutura básica de um algoritmo em português estruturado:

```
algoritmo nomeDoAlgoritmo
```

```
var
```

```
    <lista de variáveis>
```

```
inicio
```

```
// Seção de Comandos
```

```
fim
```

Representação de Algoritmos

◆ Português Estruturado → Exemplo:

```
algoritmo somaValores
```

cabeçalho do algoritmo

```
var
```

declaração de variáveis

```
    inteiro valor1, valor2, soma;
```

```
inicio
```

```
    Ler (valor1, valor2);
```

```
    soma = valor1 + valor2;
```

corpo do algoritmo

```
    Escrever (soma);
```

```
fim
```

Representação de Algoritmos



- ◆ Português estruturado:
 - ◆ Vantagens:
 - ◆ usa o português como base → facilidade para quem está aprendendo;
 - ◆ a passagem para uma linguagem de programação qualquer é quase imediata;



Representação de Algoritmos



- ♦ Português estruturado:
 - ♦ Desvantagens:
 - ♦ Não é padronizada, podem existir várias versões da mesma.

FORBELLONE, A. L. V.; EBERSPÄCHER, H. F.. Lógica de Programação: a construção de algoritmos e estruturas de dados. São Paulo: Makron Books, 2000.