

Aula 1

Lógica de Programação e Algoritmos

Prof. Sandro de Araujo

Conversa Inicial

- A aula apresenta-se com a seguinte estrutura de conteúdo:
 1. Introdução à lógica
 2. Introdução aos algoritmos
 3. Formas de representação de algoritmos
 4. Construção de algoritmos

- O objetivo da aula é introduzir os principais conceitos e temas das abordagens sobre lógica de programação e algoritmos para a resolução de diferentes problemas, bem como entender os tipos de algoritmos a serem utilizados nessa disciplina

Introdução à lógica

- Antes de começarmos nossos estudos sobre lógica de programação faremos uma reflexão sobre o significado da palavra lógica:
 - Afinal, o que é lógica?



Arte de pensar

Colocar em ordem os pensamentos



- A lógica baseia-se em argumentos compostos por premissas e conclusões. Exemplo:
 - A seguir temos em p e q as premissas e em r a conclusão:
 - a. p: Sei que a camisa está no guarda-roupas
 - q: Sei que o guarda-roupas está fechado
 - r: Logo, concluo que tenho de abrir o guarda-roupas para pegar a camisa

- Embora o uso do raciocínio lógico pareça simples, é uma tarefa bastante complexa, que exige muita prática
- Faça uma reflexão e análise de todos os passos envolvidos na troca de um pneu. Percebeu que essa tarefa não se resume a tirar um pneu e colocar outro?

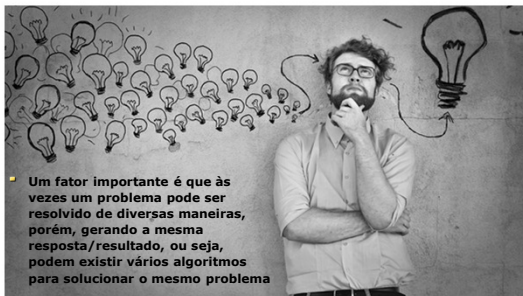


- Se detalharmos um pouquinho os passos para a troca do pneu, teremos algo como:

Passo 1 – O estepe está cheio o suficiente?	Passo 9 – Parafusar todos os parafusos
Passo 2 – Se o estepe estiver cheio	Passo 10 – Abaixar o macaco
Passo 3 – Pegar o macaco no porta-malas	Passo 11 – Guardar o macaco no porta-malas
Passo 4 – Posicionar o macaco no lugar adequado	Passo 12 – Guardar o pneu furado no porta-malas
Passo 5 – Suspender o carro	Passo 13 – Fechar o porta-malas
Passo 6 – Retirar todos os parafusos	Passo 14 – Entrar no carro
Passo 7 – Retirar o pneu	Passo 15 – Ligar o carro
Passo 8 – Colocar o estepe	Passo 16 – Dirigir até encontrar uma borracharia

Introdução aos algoritmos

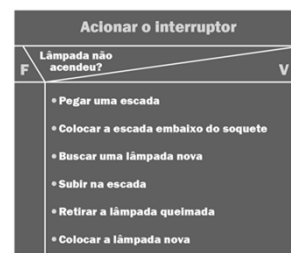
- **Algoritmo é uma sequência lógica de passos que levam a um determinado objetivo**
- **Sabemos o que é e como construir algoritmos**
 - **Se não fosse verdade, ninguém decidiria qual o melhor caminho para chegar a um lugar**



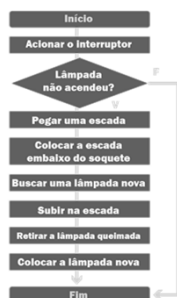
IR À ESCOLA		
Ônibus (mais barato)	Caminhando (mais saudável)	Carro (mais confortável)
INÍCIO	INÍCIO	INÍCIO
1. Caminhar até o ponto de ônibus	1. Colocar um calçado adequado para caminhada	1. Entrar no carro
2. Esperar o ônibus	2. Mapear os trajetos até a escola	2. Mapear os trajetos até a escola
3. Entrar no ônibus	3. Escolher o trajeto	3. Escolher o trajeto
4. Descer no ponto mais próximo da escola	4. Caminhar até a escola	4. Dirigir até a escola
5. Caminhar até a escola	5. Chegar à escola	5. Sair do carro
6. Chegar à escola		6. Chegar à escola
FIM	FIM	FIM

Formas de representação de algoritmos

Diagrama Nassi-Shneiderman



Fluxograma



Pseudocódigo

- **Início**
 - 1. Acionar o interruptor
 - ✓ 2. Se a lâmpada não acender:
 - ✓ 2.1. Pegue uma escada
 - (...)

Pseudocódigo

- (...)
- ✓ 2.2. Coloque a escada embaixo do soquete
- ✓ 2.3. Busque uma lâmpada nova
- ✓ 2.4. Suba na escada
- ✓ 2.5. Retire a lâmpada queimada
- ✓ 2.6. Coloque a lâmpada nova
- Fim**

Construção de algoritmos

- Conforme conceituado anteriormente, um algoritmo é uma sequência finita de passos que levam à execução de uma tarefa
- Podemos pensar em algoritmo como uma receita de uma pizza *marguerita*. Essa sequência finita de instruções em forma de receita dá cabo de uma meta específica, que é o preparo de uma deliciosa pizza marguerita

Para construir um algoritmo é preciso seguir os seguintes passos:

1. Entender o problema a ser resolvido, destacar os pontos mais importantes e os objetivos que o compõem
 2. Definir os dados de entrada, ou seja, quais dados serão fornecidos e quais objetivos fazem parte do cenário do problema
- (...)

Para construir um algoritmo é preciso seguir os seguintes passos:

(...)

3. Definir o processamento, ou seja, quais operações serão executadas e quais as restrições para essas operações; o processamento deve transformar os dados de entrada em dados de saída e também verificar quais objetos são responsáveis pelas atividades
4. Definir os dados de saída, ou seja, quais dados gerados depois do processo

(...)

Para construir um algoritmo é preciso seguir os seguintes passos:

(...)

5. Construir o algoritmo utilizando um dos três tipos apresentados anteriormente (diagrama N-S, fluxograma ou pseudocódigo)
6. Testar o algoritmo realizando simulações
7. Corrigir possíveis erros e voltar ao item 5

Estrutura de um algoritmo em pseudocódigo

- algoritmo "nome" /* Tem como objetivo identificar o algoritmo; deve-se utilizar um nome o mais significativo possível, para facilitar a identificação */
- var /* Seção de declarações – Neste ponto são informadas quais variáveis e seus respectivos tipos que serão utilizados no algoritmo */

(...)

Estrutura de um algoritmo em pseudocódigo

(...)

- inicio /* Seção de comandos – Aqui será escrita a sequência de comandos que deve ser executada para solucionar o problema em questão */
- finalgoritmo // Marca o final do algoritmo

- Observa-se agora um pseudocódigo que recebeu um valor inteiro, fornecido pelo usuário, e o retornou no monitor
- algoritmo "exemplo 1"

var x: inteiro

inicio

leia (x)

escreva (x)

fimalgoritmo

- Nesse exemplo, o algoritmo que recebe um valor inteiro acresce duas unidades a este, e exibe o resultado desta manipulação

algoritmo "exemplo 2"

var n: inteiro

inicio

escreva ("Digite um número inteiro: ")

leia (n) n <- n+2

escreval ("Resultado (número + 2): ", n)

fimalgoritmo

■ Nesse exemplo o algoritmo que recebe um valor inteiro, acresce duas unidades a este, e exibe o resultado desta manipulação

algoritmo "exemplo 3"

var n1, n2, res: inteiro

inicio

 escreva ("Digite o multiplicando inteiro: ")

 leia (n1)

 escreva ("Digite o multiplicador inteiro: ")

 leia (n2) res <- n1 * n2

 escreva ("Resultado da multiplicação: ", res)

fimalgoritmo

