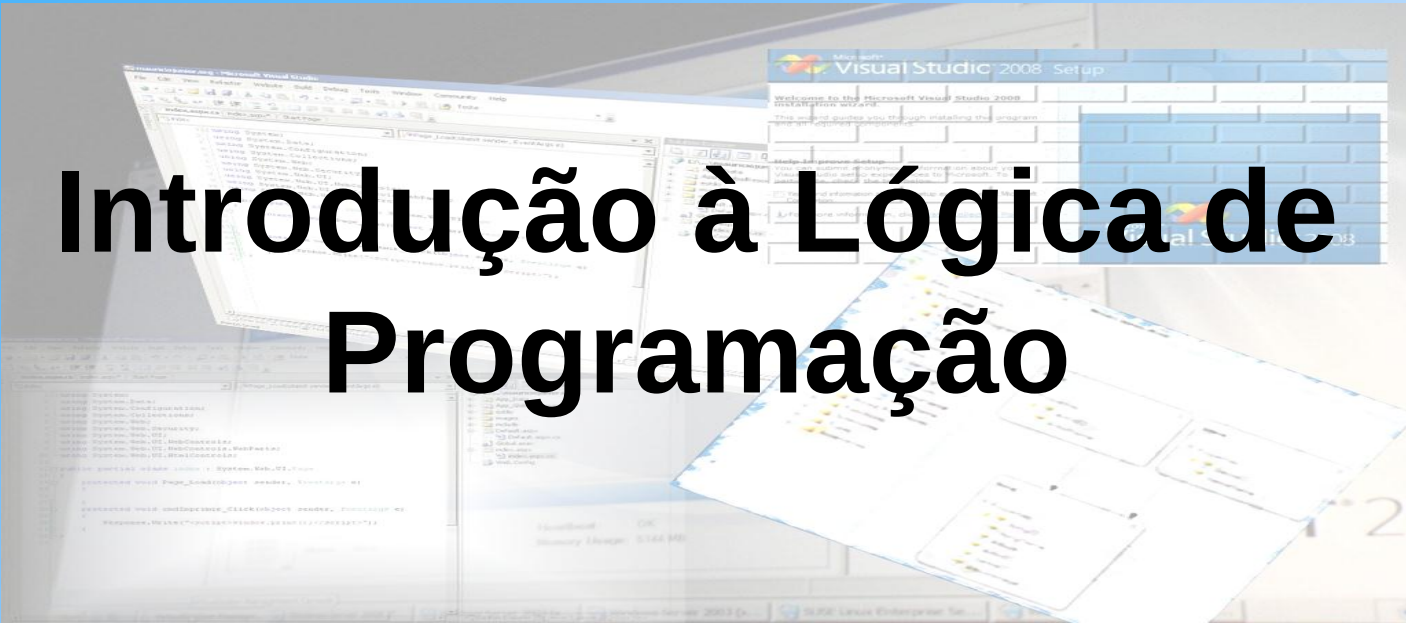




Introdução à Lógica de Programação

"Não estamos aqui para sobreviver e sim para explorar a oportunidade de vencer adquirindo o saber!"

(Autor Desconhecido)

"A lógica é a arte de pensar correctamente e, visto que a forma mais complexa do pensamento é o raciocínio, a lógica estuda ou tem em vista a correcção do raciocínio"

(FORBELONE, 1993)

© Luis Charneca - 2011

Mail: luis.charneca@gmail.com

Programação é o processo de escrita, teste e manutenção de um programa de computador. O programa é escrito numa linguagem de programação(...)

(...)Há várias décadas se debate se a programação é mais semelhante a uma arte (Donald Knuth), a uma ciência, à matemática (Edsger Dijkstra), à engenharia (David Parnas), ou se é um campo completamente novo.

Fonte: Wikipédia

Lógica

- O estudo da Lógica, é o estudo dos métodos e princípios usados para distinguir o raciocínio correcto do incorrecto.

Lógica vem de “Logos” que significa:
palavra;
expressão;
conceito;
pensamento;
discurso;
razão.

Lógica de Programação

- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a sequência lógica para o desenvolvimento.

Então o que é lógica?

Lógica de programação é a técnica de encadear pensamentos para atingir determinado objectivo.

Sequência Lógica

- Estes pensamentos, podem ser descritos como uma sequência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa.
- **Sequência Lógica são os passos executados até atingir um objectivo ou a solução de um problema.**

Instruções 1/3

- Na linguagem comum, podemos definir instruções como: “**um conjunto de regras ou normas definidas para a realização ou emprego de algo**”.
- Em informática, ***instrução*** é a informação que indica a um computador uma acção elementar a executar.

Instruções 2/3

- Convém referir que uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas numa ordem sequencial lógica.
- Por exemplo, se quisermos fazer uma omeleta de batatas, vamos colocar em prática uma série de instruções: descascar as batatas, bater os ovos, fritar as batatas, etc...
- É evidente que essas instruções têm de ser executadas numa ordem adequada – não se podem descascar as batatas depois de fritá-las.

Instruções 3/3

- Dessa maneira, uma instrução executada isoladamente não tem muito sentido.
- Para obtermos o resultado pretendido, devemos colocar em prática o conjunto de todas as instruções, na ordem correcta.

Instruções são um conjunto de regras ou normas definidas para a realização ou execução de algo.

Em informática, é o que indica a um computador uma acção elementar a executar.

Algoritmos 1/5

- A lógica de programação é fundamental para quem estuda programação de computadores.
- Antes de ter contacto com uma linguagem de programação é necessário saber desenvolver a resolução das tarefas propostas.

Algoritmos 2/5

- **Algoritmo** não é a solução de um problema, pois, se assim fosse, cada problema teria um único algoritmo.
- **Algoritmo** é um caminho para a solução de um problema, e em geral, os caminhos que levam a uma solução são muitos.

Algoritmos 3/5

- Quando temos um problema e vamos utilizar um computador para o resolver inevitavelmente temos que passar pelas seguintes etapas:
 - a) Definir o problema.
 - b) Realizar um estudo da situação actual e verificar qual(is) a(s) forma(s) de resolver o problema.
 - c) Terminada a fase de estudo, utilizar uma linguagem de programação para escrever o programa que deverá em princípio, resolver o problema.
 - d) Analisar junto dos utilizadores se o problema foi resolvido.

Se a solução não foi encontrada, deveremos voltar para a fase de estudo afim de descobrir onde está a falha.

Algoritmos 4/5

- O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.
- É importante ressaltar que qualquer tarefa que siga determinado padrão pode ser descrita por um algoritmo, como por exemplo:

COMO FAZER ARROZ DOCE

CALCULAR O SALDO DA CONTA CORRENTE DE UM CLIENTE

Algoritmos 5/5

- Ao criar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais.



ENTRADA: São os dados de entrada do algoritmo

PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final

SAÍDA: São os dados já processados

Exemplo de Algoritmo

- Imagine o seguinte problema: Calcular a média final dos alunos numa disciplina em que realizaram quatro provas: P1, P2, P3 e P4.

- Onde:

$$\text{Média Final} = \frac{P1+P2+P3+P4}{4}$$

- Para montar o algoritmo proposto, faremos três perguntas:
 - a) Quais são os dados de entrada?
 - **R:** Os dados de entrada são P1, P2, P3 e P4
 - b) Qual será o processamento a ser utilizado?
 - **R:** O procedimento será somar todos os dados de entrada e dividi-los por 4 (quatro)
 - c) Quais serão os dados de saída?
 - **R:** O dado de saída será a média final

Programas

- Um programa é um Algoritmo escrito numa linguagem de programação.

Linguagens de Programação

- São Softwares que permitem o desenvolvimento de programas. Possuem um poder de criação ilimitado, desde jogos, editores de texto, sistemas de gestão comercial, sistemas operativos, etc..

Técnicas de Programação

- **Desvio condicional**
- **Ciclos de repetição**
- **Definição de tipos de dados**
- **Operações aritméticas**
- **Operações relacionais**
- **Operações lógicas**
- **Variáveis**
- **Atribuições**

Desvio condicional

- Condições que podemos estabelecer em determinadas etapas do programa para seguir uma ou outra instrução, ou uma sequência delas

Ciclos de repetição

- São utilizados quando é necessário que um parte do programa (uma sequência de instruções) seja repetido várias vezes

Definição de tipos de dados

- Os dados podem ser:
- *Numéricos inteiros* (qualquer número positivo ou negativo, desde que não seja um número fraccionário)
- *Numéricos reais* (qualquer número positivo ou negativo, incluindo números fraccionários)
- *Caracteres* (sequências contendo letras, números e símbolos especiais)
- *Lógicos* (podem assumir 2 valores Verdadeiro ou Falso)

Operações aritméticas

- Adição (+)
- Subtracção (-)
- Multiplicação (*)
- Divisão (/)

Operações relacionais

- Igual (=)
- Diferente (<>)
- Maior (>)
- Menor (<)
- Maior ou igual (>=)
- Menor ou igual (<=)

Operações lógicas

- **E** RETORNA VERDADEIRO SE AMBAS AS PARTES FOREM VERDADEIRAS.
- **OU** BASTA QUE UMA PARTE SEJA VERDADEIRA PARA RETORNAR VERDADEIRO.
- **NÃO** INVERTE O ESTADO, DE VERDADEIRO PASSA PARA FALSO E VICE-VERSA.

Tabela da Verdade

A	B	A E B	A OU B	NÃO (A)
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

Variáveis 1/2

- Um campo de memória que armazena valores que podem ser alterados durante o programa.
- Espaço reservado na memória para guardar informação para ser utilizada durante a execução de um algoritmo.
- Uma variável é caracterizada por um nome e quatro atributos básicos: escopo, tempo de vida, valor e tipo.

Variáveis 2/2

- ESCOPO – Define o espaço, dentro do programa, onde a variável é conhecida e pode ser usada.
- TEMPO DE VIDA – O intervalo de tempo durante o qual uma área da memória está reservada para guardar uma variável e o seu valor.
- VALOR – O conteúdo da variável. É guardado, de forma codificada, na memória reservada À variável.
- TIPO – Especifica o conjunto de valores que podem ser associados à variável, bem como as operações que podem ser exercidas sobre a mesma.

Atribuições

- Utilizadas para alterar o valor de uma variável
- Exemplo:
 - Nome_Variavel = Valor a atribuir

Em Resumo...

O raciocínio da lógica para a programação traz ao individuo um novo mundo cheio de outros mundos.

Dominando a lógica seremos capazes de aprender sem grandes dificuldades qualquer outra linguagem de programação, seja ela estruturada, orientada a objecto, a eventos, etc.