Algoritmos e Programação

Ricardo Argenton Ramos

Baseado nos slides do professor Jadsonlee da Silva Sá

Universidade Federal do Vale do São Francisco - UNIVASF
Colegiado de Engenharia de Computação – CECOMP

Ementa

- ✓ Conceito de algoritmo.
- ✓ Lógica de programação e programação estruturada.
- ✓ Linguagem de definição de algoritmos.
- ✓ Estrutura de algoritmo.
- ✓ Constantes.
- ✓ Identificadores.
- ✓ Variáveis.
- ✓ Declaração de variáveis.
- ✓ Operações básicas.
- ✓ Comandos de entrada e saída.
- ✓ Estrutura de controle de fluxo.
- ✓ Conceito e classificação de linguagens de programação.
- ✓ Introdução à uma linguagem de programação de alto nível estruturada – **Linguagem C**.

Ementa

- ✓ Ambiente de programação.
- ✓ Componentes de linguagem de programação:
  - estrutura de um programa;
  - identificadores;
  - palavras reservadas;
  - variáveis;
  - constantes;
  - declaração de variáveis;
  - operações básicas;
  - comandos de entrada e saída;
  - estruturas de controle de fluxo;
  - estruturas de dados homogêneas;
  - modularização.

Objetivos

- ✓ **Objetivo Geral:**
  - Capacitar o aluno a visualizar soluções computacionais para problemas através da aplicação dos conceitos da lógica de programação e dotá-los da capacidade de construção de programas em linguagem de alto nível estruturada **(linguagem C)**.

Objetivos

✓ **Objetivos Específicos:**

- Desenvolver o raciocínio lógico e abstrato do aluno;
- Familiarizar o aluno com o modelo seqüencial de computação;
- Apresentar técnicas e linguagens para representação e construção de algoritmos simples;
- Apresentar conceitos básicos de linguagens de programação;
- Capacitar o aluno no uso da linguagem C;
- Treinar o aluno no processo básico de desenvolvimento de software (concepção, edição, execução e teste de programas de computador).

Metodologia

- ✓ **Aulas expositivo-dialogadas.**
  - Fornecer os componentes teóricos e conceituais.
- ✓ **Aulas práticas ministradas em laboratório.**
  - Experimentação e fixação dos conteúdos.
  - Visualg e DevC++.

Avaliação

- ✓ Duas provas (P1 e P2).

Média = (P1 + P2)/2.

- Se **média** >= 7,0 e freqüência >= 75% → **Aprovado.**
- Se **média** < 4,0 ou freqüência < 75% → **Reprovado.**
- Se 4,0 <= **média** < 7,0 e freqüência > 75% → **Final.**

  - O aluno submetido a prova final (PF) será considerado aprovado se obtiver **nota final >= 5,0**.

Nota final = (Média + PF)/2

Bibliografia

- ✓ **Básica.**
  - ASCENCIO, A.F.G.; CAMPOS, E.A.V. Fundamentos da programação de computadores. 2ª ed. Pearson Prentice Hall.
  - SCHILDT, H. C completo e total. Pearson Prentice Hall, 2006.
- ✓ **Complementar.**
  - CARBONI, I.F. Lógica de programação. Thomson.
  - CORMEN, T.H. et al. Algoritmos, teoria e prática. Campus, 2002.

Introdução

- ✓ **Computador** → É uma máquina capaz de possibilitar variados tipos de tratamento automático de informações ou processamento de dados.

- ✓ O que deve ser feito para que um determinado tratamento automático de informações ocorra?

  - Deve-se instruir o computador para que o mesmo utilizando-se de sua estrutura execute determinada tarefa.

  - Como?

    - Software (programas).

Introdução

- ✓ **Nosso objetivo** → Aprender conceitos básicos para desenvolver programas para computadores.
  - **Exemplos:** sistemas bancários, sistemas de restaurantes, cálculos avançados entre outros.

- ✓ Roteiro para desenvolver programas:

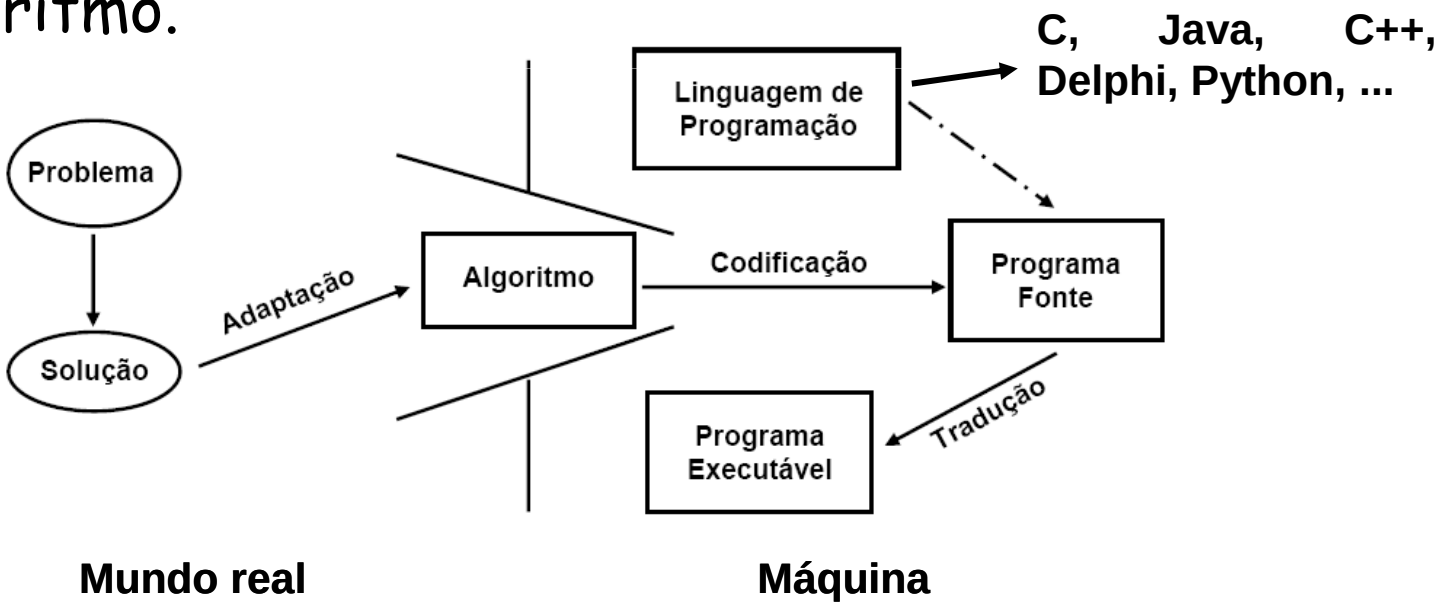
Problema → Solução → **Algoritmo** → **Programa** → Resultado

Algoritmo e Programação

- ✓ **Definições:**
  - **Algoritmo** → Conjunto de regras e operações bem definidas e ordenadas, destinadas à solução de um problema, ou de uma classe de problemas, em um número finito de etapas → **Representação de uma solução para um problema.**
  - **Programa** → Seqüência completa de instruções a serem executadas por um computador → **De acordo com um algoritmo**.

Algoritmo e Programação

- ✓ O **algoritmo**, do ponto de vista computacional, tem um papel fundamental por ser o elo de ligação entre dois mundos (real e computacional).
- ✓ A atividade de programação começa com a construção do algoritmo.



Algoritmo e Programação

- ✓ Exemplos de algoritmos.

Algoritmo: trocar lâmpada

Passo 1: pegar a lâmpada nova.
Passo 2: pegar a escada.
Passo 3: posicionar a escada embaixo da lâmpada queimada.
Passo 4: subir na escada com a lâmpada nova.
Passo 5: Retirar a lâmpada queimada.
Passo 6: Colocar a lâmpada nova.
Passo 7: Descer da escada.
Passo 8: Ligar o interruptor.
Passo 9: Guardar a escada.
Passo 10: Jogar a lâmpada velha no lixo.

Algoritmo: sacar dinheiro

Passo 1: ir até o caixa eletrônico.
Passo 2: colocar o cartão.
Passo 3: digitar a senha.
Passo 4: solicitar o saldo.
Passo 5: se o saldo for maior ou igual à quantia desejada, sacar a quantia desejada; caso contrário sacar o valor do saldo.
Passo 6: retirar dinheiro e cartão.
Passo 7: sair do caixa eletrônico.

Métodos de Representação de Algoritmos

- ✓ Existem duas formas de representação de algoritmos:
  - **Fluxograma** → Representação gráfica.
  - **Pseudocódigo (Português estruturado)** → Representação textual.

Métodos de Representação de Algoritmos

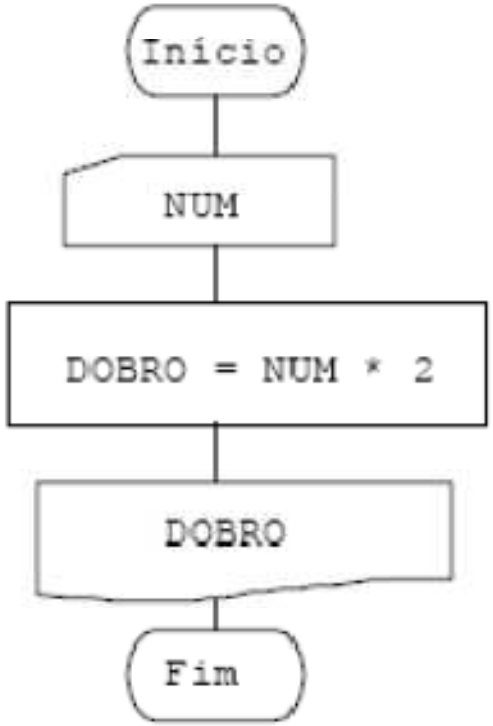
- ✓ **Características.**
  - **Fluxograma.**
    - A representação gráfica é mais concisa que a representação textual.
    - É necessário aprender a simbologia dos fluxogramas.
  - **Pseudocódigo.**
    - A transcrição para qualquer linguagem de programação é quase direta.
    - É necessário aprender as regras do pseudocódigo.

Métodos de Representação de Algoritmos

- ✓ **Fluxograma.**
  - Representação gráfica por meio de símbolos geométricos, da solução algorítmica de um problema.

Métodos de Representação de Algoritmos

✓ Exemplo - Fluxograma.



EXEMPLO	EXPLICAÇÃO
Início	Início do algoritmo
NUM	Entrada do número
DOBRO = NUM * 2	Cálculo do dobro do número
DOBRO	Apresentação do resultado
Fim	Fim do algoritmo

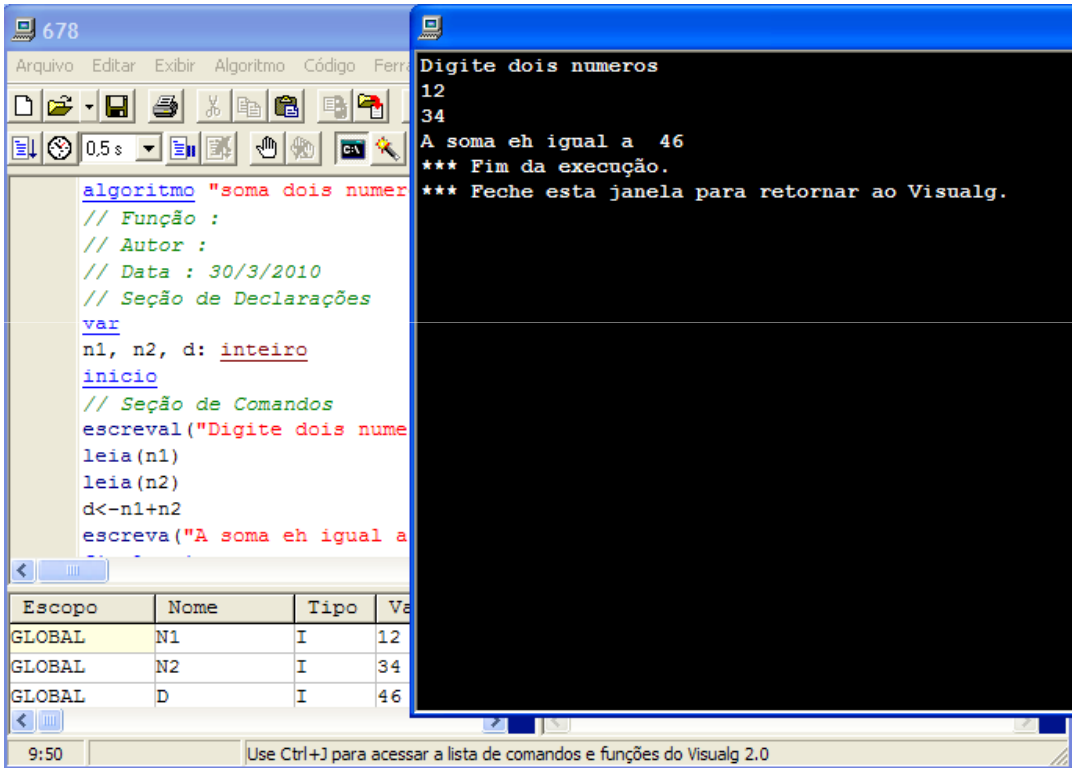
Métodos de Representação de Algoritmos

- ✓ **Pseudocódigo.**
  - Descrição narrativa utilizando nosso idioma para descrever o algoritmo.
  - **Exemplo de uma descrição narrativa.**
    - Soma de dois números.
      1. Receber os dois números.
      2. Efetuar a soma dos dois números.
      3. Mostrar o resultado.

Métodos de Representação de Algoritmos

✓ Exemplo – Descrição narrativa (Visualg).

```
algoritmo "soma dois numeros"
// Função :
// Autor :
// Data : 30/3/2010
// Seção de Declarações
var
n1, n2, d: inteiro
inicio
// Seção de Comandos
escreval("Digite dois numeros")
leia(n1)
leia(n2)
d<-n1+n2
escreva("A soma eh igual a ", d)
fimalgoritmo
```



Métodos de Representação de Algoritmos

- ✓ **Resumindo.**

  - Escrever algoritmos e, por fim, programar, consiste em dividir qualquer problema em vários **passos** menores, usando uma ou mais formas de representação.

  - Esses **passos** que compõem o algoritmo são denominados de **comandos**.

Conceituação de Elementos Básicos para Construção de um Algoritmo

- ✓ **Constante.**

  - Valores fixos, tais como números. Estes valores não podem ser alterados pelas instruções do algoritmo, ou seja, é um espaço de memória cujo valor não deve ser alterado durante a execução do programa.

  - **Exemplo:**

    - Inteiro → 10, -23768, ...
    - Real → -2.34, 0.149, ...
    - Caractere → "k", "computador"

Conceituação de Elementos Básicos para Construção de um Algoritmo

- ✓ **Variável**.
  - Elemento de dado cujo valor pode ser modificado ao longo de sua execução.
  - Uma variável representa uma posição na memória e pode ter tipo (inteiro, caractere, real), tamanho (16, 32 bits, ...) e nome definidos.

Conceituação de Elementos Básicos para Construção de um Algoritmo

- ✓ **Identificadores.**
  - Nomes utilizados para referenciar variáveis, funções ou vários outros objetos definidos pelo programador.
  - **Exemplo:**
    - letras, dígitos e sublinhado(_);
    - Não podem começar com dígito;
    - Não podem ser iguais a uma palavra-chave e nem iguais a um nome de uma função declarada pelo programador ou pelas bibliotecas da linguagem utilizada.

Conceituação de Elementos Básicos para Construção de um Algoritmo

- ✓ **Palavras-reservadas (palavras-chave).**

  - São identificadores predefinidos que possuem significados especiais para o interpretador do algoritmo.

inicio	senao	para	enquanto
var	logico	se	ate
faca	inteiro	real	

Conceituação de Elementos Básicos para Construção de um Algoritmo

- ✓ **Tipos primitivos.**
  - Palavra-reservada: **logico** - define variáveis do tipo booleano, ou seja, com valor VERDADEIRO ou FALSO.
  - Palavra-reservada: **caractere** – define variáveis do tipo string, ou seja, cadeia de caracteres.
  - Palavra-reservada: **inteiro** - define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais.
  - Palavra-reservada: **real** - define variáveis numéricas do tipo real, ou seja, com casas decimais.

Declaração de Variáveis

- ✓ Palavra-reservada: **var** - utilizada para iniciar a seção de declaração de variáveis.

  - Exemplo:

```
var   a: inteiro
      nome_do_aluno: caractere
      sinalizador: logico
      valor1, valor2: real
```

Declaração de Variáveis

- ✓ **Regra para criar nomes de variáveis.**
  - Os nomes das variáveis devem representar o que será guardado dentro dela.
  - O primeiro caractere de um nome deverá ser sempre alfabético.
  - Não podem ser colocados espaços em branco no nome de variáveis, usar o UNDERSCORE "_".
  - A declaração de uma variável é feita no algoritmo informando o seu nome, seguido por : e terminado com o seu tipo.

Operadores e Hierarquia nas Operações

<-	Atribuição. x <- 2. A variável x recebeu o valor 2. Logo x = 2
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
a\b	Retorna o quociente da divisão inteira de a por b
a%b	Retorna o resto da divisão inteira de a por b
a^b	Retorna o valor de a elevado a b
a^1/b	Retorna a raiz b de a
aleatorio (a)	Retorna um número aleatório, em intervalo fechado, entre 0 e a

Hierarquia	Operação
1	Parênteses
2	Função
3	-, + (unários)
4	^
5	*, /, \, %
6	+, -

Exemplos:

3/4+5 = 5.75

3/(4+5) = 0.33333333

3\2*9 = 9

11%3^2 = 2

11%(3^2) = 2

(11%3)^2 = 4

3\2+(65-40)^(1/2) = 6

Operadores Relacionais e Lógicos

Operador	Ação
>	maior que
>=	maior ou igual a
<	menor que
<=	menor ou igual
=	igual a
<>	diferente de

Operador
e
ou
nao
xou

Exemplos:

3>7 = FALSO

“A” = “a” = VERDADEIRO

“a” > “B” = FALSO

(3>=13\4) xou (nao (5%2=0)) = FALSO