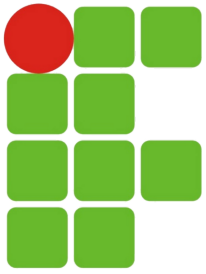


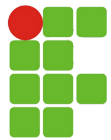
Algoritmos

Diego Silveira Costa Nascimento

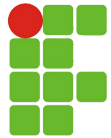
Instituto Federal do Rio Grande do Norte
diego.nascimento@ifrn.edu.br

12 de setembro de 2015

Ementa do Curso

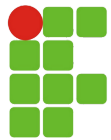


Ementa do Curso



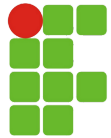
Objetivos da Disciplina

- Fazer uma introdução aos conceitos de Algoritmo; e
- E apresentar algumas metodologias de desenvolvimento de Algoritmos; e
- Aprender sobre as características e os conceitos envolvidos na escrita de um programa de computador.



Motivações em Estudar Algoritmos

- Atualmente temos um conjunto vasto de linguagens de programação disponíveis para se desenvolver sistemas, sejam elas: Java, C/C++, Python, Pascal, Fortran, Cobol, entre outras;
- Quase tudo ao nosso redor possui sistemas embarcados (celular, televisor, ar-condicionado, carro, entre outros); e
- Essa disciplina é de fundamental importância para que o aluno possa se aprofundar em qualquer linguagem de programação que venha a utilizar no futuro.



Algoritmos

Definição

Um algoritmo é uma sequência finita de instruções ou passos bem definidos e não ambíguos, cada uma dos quais pode ser executada mecanicamente num período de tempo finito e com uma quantidade de esforço finito.

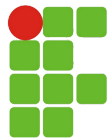
Observações

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

História

- A palavra algoritmo vem do nome do matemático iraniano Abu Abdullah Mohammad Ibn Musa **al-Khawarizmi**, nascido em Khawarizm (Kheva), ao sul do mar Aral, que viveu no século XVII; e

- O termo algoritmo também é utilizado em outras áreas, como engenharia, administração, entre outras.

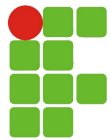


Exemplos de Algoritmos no Nosso Cotidiano

- Trocar o pneu de um carro;
- Fazer um bolo a partir de uma receita;
- Substituir uma lâmpada queimada;
- Desmontar e montar uma bicicleta;
- Ligar um televisor e escolher um canal;
- Sair para o trabalho; e
- Sacar dinheiro no caixa eletrônico.

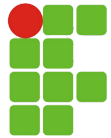
O que é um Programa de Computador?

- Um programa nada mais é que um algoritmo;
- Sua particularidade é que suas operações são específicas para o computador e restritas ao conjunto de instruções que o processador pode executar;
- Podemos considerar esse conjunto de instruções como a primeira linguagem de programação do computador (linguagem de máquina);
- Já as linguagens de programação são classificadas segundo sua proximidade com a linguagem de máquina, que podem ser:
  - Baixo nível (Mais próxima da linguagem do computador); ou
  - De alto nível (Mais próxima da linguagem humana).



Formas de Representação de Algoritmos

- Existem diversas formas de representação de algoritmos, mas não há um consenso com relação à melhor delas;
- Algumas formas de representação de algoritmos tratam dos problemas apenas em nível lógico, abstraindo-se de detalhes de implementação muitas vezes relacionados com alguma linguagem de programação específica;
- Dentre as formas de representação de algoritmos mais conhecidas, sobressaltam:
  - Descrição Narrativa;
  - Fluxogramas; e
  - Pseudocódigo.



Descrição Narrativa

A descrição narrativa consiste em analisar o enunciado do problema e escrever, utilizando uma linguagem natural (por exemplo, a língua portuguesa), os passos a serem seguidos para a sua resolução.

Vantagem

Não é necessário aprender nenhum conceito novo, pois uma linguagem natural, neste ponto, já é bem conhecida.

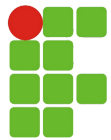
Desvantagem

A linguagem natural abre espaço para várias interpretações, o que posteriormente dificultará a transcrição desse algoritmo para o programa.

Descrição Narrativa: Verificar Aprovação do Aluno

Exemplo

Passo 1: Receber duas notas.
Passo 2: Calcular a média aritmética (Primeira nota + Segunda nota, e dividir por dois).
Passo 3: Exibir o valor obtido.
Passo 4: Analisar o valor obtido, e se o valor da média for maior ou igual a sete, exibir a mensagem "Aprovado", caso contrário, exibir a mensagem "Reprovado".



Fluxograma

O fluxograma consiste em analisar o enunciado do problema e escrever, utilizando símbolos gráficos predefinidos, os passos a serem seguidos para sua resolução.

Vantagem

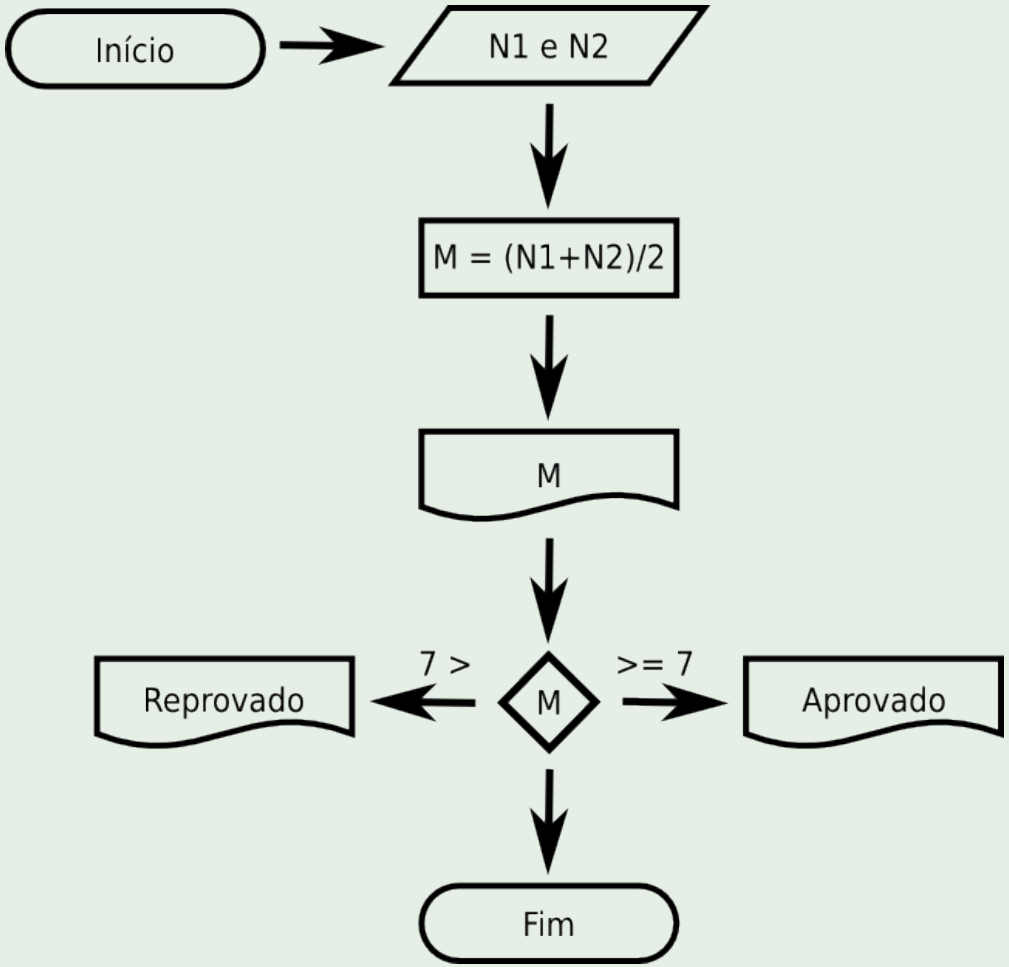
O entendimento de elementos gráficos é mais simples que o entendimento de textos.

Desvantagem

É necessário aprender a simbologia dos fluxogramas e, além disso, o algoritmo resultante não apresenta muitos detalhes, dificultando sua transcrição para um programa.

Fluxograma: Verificar Aprovação do Aluno

Exemplo



Pseudocódigo

O pseudocódigo consiste em analisar o enunciado do problema e escrever, por meio de regras predefinidas, os passos a serem seguidos para sua resolução.

Vantagem

A passagem do algoritmo para qualquer linguagem de programação é quase que imediata, bastando conhecer as palavras reservadas dessa linguagem que serão utilizadas.

Desvantagem

É necessário aprender as regras do pseudocódigo.

Pseudocódigo: Verificar Aprovação do Aluno

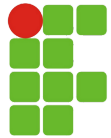
Exemplo

Algoritmo: Verificar_Aprovação_Aluno

Variáveis: M, N1, N2

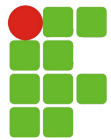
Início
    Leia N1
    Leia N2
    M $\leftarrow$ (N1 + N2) / 2
    Escreva M
    Se M $\geq$ 7 **Então**
        Escreva "Aprovado"
    Senão
        Escreva "Reprovado"
Fim.

Ementa do Curso



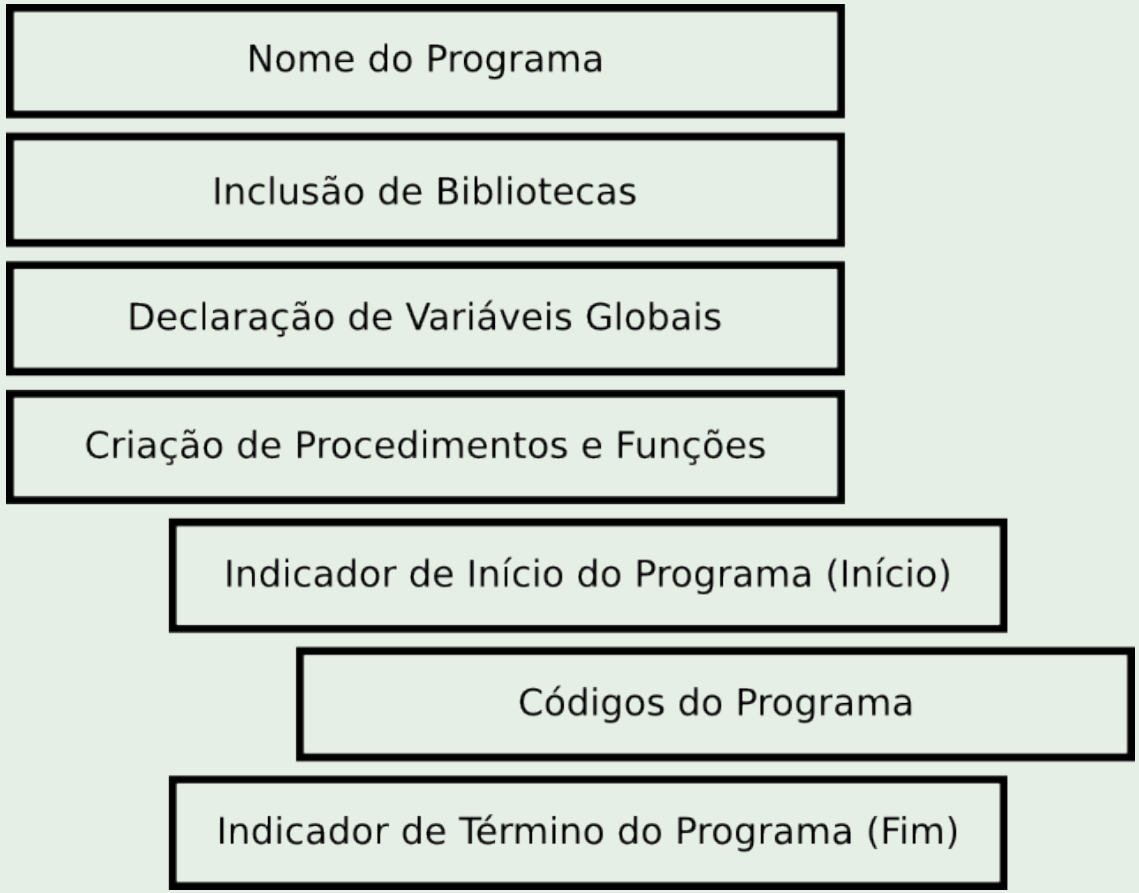
Programação Estruturada

- A programação estruturada (Top-Down) estabelece uma disciplina de desenvolvimento de algoritmos que facilita a compreensão de programas através do número restrito de mecanismos de controle da execução de programas;
- Qualquer algoritmo, independentemente da área de aplicação, de sua complexidade e da linguagem de programação na qual será codificado, pode ser descrito através destes mecanismos básicos;
- O princípio básico de programação estruturada é que um programa é composto por blocos elementares de código que se interligam através de três mecanismos básicos, que são:
  - Sequência;
  - Seleção; e
  - Iteração;
- Cada uma destas construções tem um ponto de início (o topo do bloco) e um ponto de término (o fim do bloco) de execução.



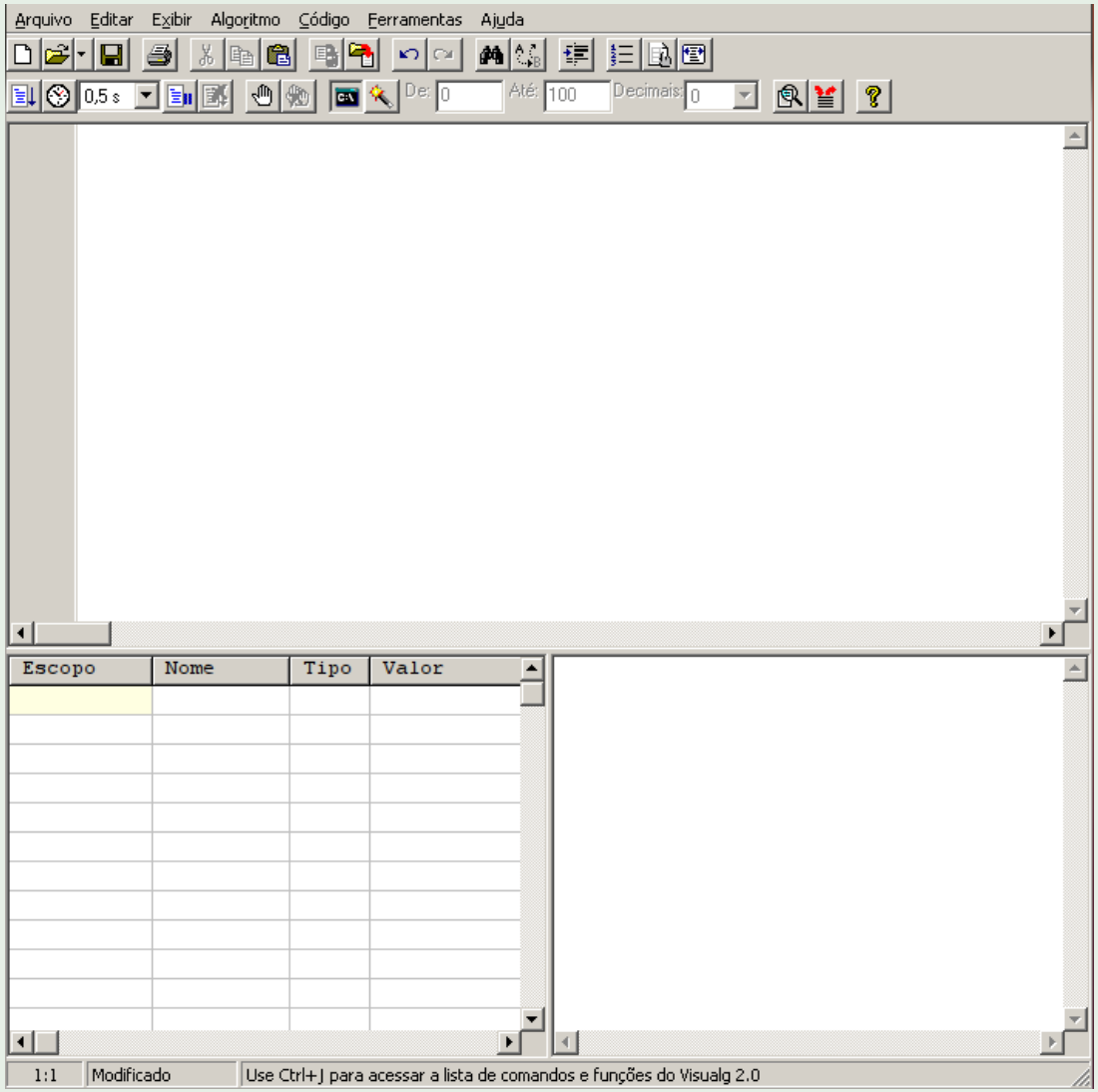
Estrutura de um Algoritmo

Exemplo



Ambiente de Desenvolvimento do Visualg

Exemplo

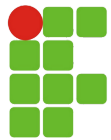


Instrução de Saída de um Algoritmo

Definição

A instrução de saída de dados é a instrução através da qual o computador se comunica com usuário durante a execução do algoritmo. Isso é feito, geralmente, através da exibição de alguma informação na tela.

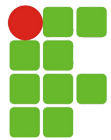
- O visualg permite saída em texto na tela para usuário a partir de dois comandos definidos:
  - *escreva*; e
  - *escreval*.
- O primeiro comando escreve o resultado de saída em forma sequencial, no qual, os valores são escritos sempre a direita, já o segundo, escreve os valores em linha, um abaixo do outro.



Algoritmo no Visualg

Exemplo

```
algoritmo "Exemplo"

inicio
   escreva("Oi Mundo!")
fimalgoritmo
```



Comentário

Definição

É uma estrutura da linguagem que permite ao desenvolvedor fazer uma breve explicação do código escrito.

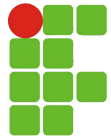
Exemplo

```
// Esse programa exemplifica o uso de comentários.
// Autor: Diego
algoritmo "ExemploComentario"

inicio
   escreva("Vamos programar?") // Exibe a mensagem na tela.
   //escreva("Fim.")
fimalgoritmo
```

Importante

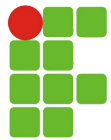
O que for escrito no bloco de comentário será ignorado pelo interpretador.

Ementa do Curso



Variável

- Uma variável representa uma posição de memória;
- Possui um nome e tipo;
- Seu conteúdo pode variar ao longo do tempo, durante a execução do programa;
- Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante; e
- Não existe limite para o número de variáveis em um programa, porém cada variável criada ocupa um espaço de memória de acordo com seu tipo e seu tamanho. Em outras palavras, quanto maior o número de variáveis utilizadas, maior o gasto de memória pelo programa desenvolvido.



Variável (cont.)

Ilustração

Endereço	Conteúdo	Variável
⋮	⋮	⋮
1001	???	
1002	2450	i
1003	???	
1004	225.345	f
1005	11331	j

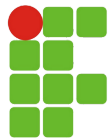
Tipos de Dados

Definição

O tipo de dados indica qual valor pode ser armazenado em uma posição de memória.

Os tipos de dados mais utilizados são:

- numérico;
- lógico; e
- literal ou caracteres.



Tipos de Dados (cont.)

Tipo Numérico

Os dados numéricos dividem-se em dois grupos:

- inteiros; e
- reais.

Os números **inteiros** podem ser positivos ou negativos e não possuem parte fracionária.

Exemplos

2 -1 0 10 8 -255

Os dados **reais** podem ser positivos ou negativos e possuem parte fracionária.

Exemplos

8.5 -82.96 0.0 1.238 -1.5

Tipos de Dados (cont.)

Tipo Lógico

São também chamados de dados *booleanos* (por causa da álgebra de *Boole*).

Exemplos

verdadeiro ou *falso*

Tipo Literal ou Caracteres

São formados por um único caractere ou por uma cadeia de caracteres. Esses caracteres podem ser as letras maiúsculas, as letras minúsculas, os números e os caracteres especiais.

Exemplos

“D” “Aluno” “sala1” “@ifrn.edu.br” “(84) 9999-9999”

Formação de Identificadores

Definição

Os identificadores são os nomes das variáveis, dos programas, dos procedimentos e das funções.

As regras básicas para a formação dos identificadores são:

- Os caracteres que você pode utilizar são: os números, as letras maiúsculas, as letras minúsculas e o caractere sublinhado;
- O primeiro caractere deve ser sempre uma letra ou um caractere sublinhado;
- Não são permitidos espaços em branco e caracteres especiais; e
- Não podemos usar as palavras reservadas nos identificadores, ou seja, palavras que pertençam a uma linguagem de programação.

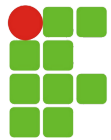
Exemplos

A a nota NOTA dia1 data_nascimento

Formação de Identificadores no Visualg

Exemplo

```
algoritmo "ExemploIdentificadores"

var nome :caracter
    matricula :inteiro
    nota1, nota2, nota3, media :real

inicio

fimalgoritmo
```

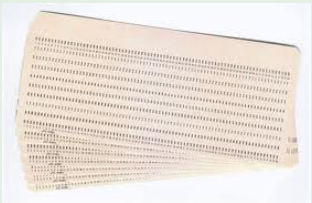


Instrução de Entrada

Definição

É o meio pelo qual as informações (mais especificamente os dados) são transferidas pelo usuário ou pelos níveis secundários de memória ao computador.

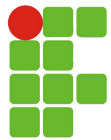
Os dispositivos de entradas mais comuns

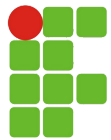

Instrução de Entrada no Visualg

Exemplo

```
algoritmo "ExemploInstrucaoEntrada"

var nome :caracter

inicio
    escreva("Digite seu nome:")
    leia(nome)
    escreva("Seu nome é ", nome, ".")
fimalgoritmo
```



Ementa do Curso



Operador de Atribuição

Definição

O comando de atribuição é utilizado para conceder valores ou operações a variáveis, sendo representado pelo símbolo $\leftarrow$.

Do lado esquerdo ao operador de atribuição fica a variável à qual está sendo atribuído o valor, e a direita operador pode-se escrever qualquer expressão (constantes, variáveis ou expressões numéricas), desde que seu resultado tenha tipo igual ao da variável.

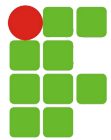
Exemplos

nome $\leftarrow$ "Diego"
aprovado $\leftarrow$ *verdadeiro*
media $\leftarrow$ 9.5
a $\leftarrow$ 10
b $\leftarrow$ a
soma $\leftarrow$ 5 + 3

Operador de Atribuição no Visualg

Exemplo

```
algoritmo "ExemploOperadorAtribuicao"

var nome :caracter
    a, b, soma :inteiro
    media :real

inicio
   nome <- "João"
   a <- 10
   b <- a
   soma <- 5 + 3
   media <- 9.5
   escreval(nome)
   escreval(a)
   escreval(b)
   escreval(media)
fimalgoritmo
```



Operadores Aritméticos

Definição

A aritmética é o ramo da matemática que lida com números e com as operações possíveis entre eles.

As operações aritméticas tradicionais são:

- Adição (+);
- Subtração (-);
- Multiplicação (*); e
- Divisão (/).

Podemos utilizar os operadores: resto (mod), divisão inteira (div) e potência ($\wedge$).

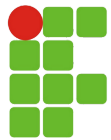
Exemplos

$10 + 2 = 12$	$5 \;/\; 2 = 2.5$	$2 \wedge 3 = 8$
5 - $2 = 3$	5 div $2 = 2$	
2 * $3 = 6$	5 mod $2 = 1$	

Operadores Aritméticos no Visualg

Exemplo

```
algoritmo "ExemploOperadores"

var a, b, c, d, e :inteiro

inicio
   a <- 2 + 3
   b <- a - 1
   c <- b * 2
   d <- 5 mod 2
   e <- 5 div 2
   escreval(a)
   escreval(b)
   escreval(c)
   escreval(d)
   escreval(e)
fimalgoritmo
```



Expressão Aritmética

Definição

Uma expressão constitui-se em um conjunto de variáveis e/ou valores, separados por caracteres especiais, que indicam as operações que devem ser executadas.

Exemplo

resultado $\leftarrow 2 + 8/2$

Importante

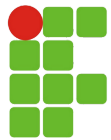
Os operadores devem obedecer uma ordem de precedência:

1. Parênteses;
2. Potenciação;
3. Multiplicação, Divisão e Resto; e
4. Adição e subtração.

Expressão Aritmética no Visualg

Exemplo

```
algoritmo "OrdemOperadores"

var a, b, c, d :inteiro

inicio
   a <- 2 + 8 / 2
   b <- (2 + 8) / 2
   c <- 4 / 2 ^ 2 - 1
   d <- (4 / 2) ^ (2 - 1)
   escreval(a)
   escreval(b)
   escreval(c)
   escreval(d)
fimalgoritmo
```



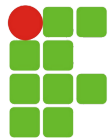
Teste de Mesa

Definição

O teste de mesa simula a execução de um algoritmo sem utilizar o computador, empregando apenas papel e caneta.

Os passos necessários para realizar um teste de mesa são:

- Identifique as variáveis envolvidas em seu algoritmo;
- Crie uma tabela com linhas e colunas, no qual corresponde, respectivamente, ao número de instruções observadas pelo teste de mesa e é o número de variáveis envolvidas; e
- De cima para baixo, preencha cada uma das linhas da tabela com o número da linha que identifica cada instrução, seguido dos valores assumidos pelas variáveis do programa após a execução daquela instrução.



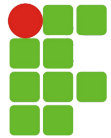
Teste de Mesa na Prática

Exemplo

```
1  algoritmo "CalculoMedia"
2
3  var nota1, nota2, soma, media: real
4
5  inicio
6        escreva ("Digite a primeira nota:")
7        leia (nota1)
8        escreva ("Digite a segunda nota:")
9        leia (nota2)
10       soma <- nota1 + nota2
11       media <- soma / 2
12       escreva ("Média =", media)
13 fimalgoritmo
```

Execução	nota1	nota2	soma	media	Saída
6	?	?	?	?	Digite a primeira nota:
7	5	?	?	?	
8	5	?	?	?	Digite a segunda nota:
9	5	8	?	?	
10	5	8	13	?	
11	5	8	13	6,5	
12	5	8	13	6,5	Média = 6,5

Ementa do Curso



O que é Estrutura de Seleção?

Definição

Também citado na literatura por **Estrutura Condicional**, é a representação de um ou mais comandos de decisão que são responsáveis por mudar o fluxo das instruções de um algoritmo em tempo de execução, permitindo que diferentes instruções de entrada sejam executadas de acordo com a entrada do programa.

Exemplo de uma compra de um produto

Se tem dinheiro para comprá-lo **então**
    coloque-o no carrinho
senão
    Devolva-o para a prateleira
    Escolha uma marca mais barata

Estrutura de Seleção Simples

Definição

É uma estrutura para desvio de fluxo do programa formada apenas pelo comando de decisão *se-então/senão*.

Exemplo

```
algoritmo "ExemploPositivoNegativo"
var numero :real

inicio
   escreva("Digite um número inteiro:")
   leia(numero)
   se numero >= 0 entao
      escreva("O número é positivo!")
   senao
      escreva("O número é negativo!")
   fimse
fimalgoritmo
```

Importante

O senão não é obrigatório.

Operadores Relacionais

Definição

Os operadores relacionais estabelecem uma relação entre seus operandos. E o valor resultante de uma relação pode ser: *verdadeiro* ou *falso*.

As relações podem ser:

- Igualdade: $=$;
- Diferença: $<>$;
- Maior que: $>$;
- Menor que: $<$;
- Maior ou igual a: $>=$; e
- Menor ou igual a: $<=$.

Exemplos

$100 = 100$ *(verdadeiro)*
$0 <> 0$ *(falso)*
$5 > 2$ *(verdadeiro)*
$0 < -50$ *(falso)*
$7 >= 10$ *(falso)*
$600 <= 600$ *(verdadeiro)*

Operadores Lógicos

Definição

Os operadores lógicos definem as maneiras como as relações podem ser conectadas. E o resultado de uma conexão pode ser: *verdadeiro* ou *falso*.

Os operadores lógicos podem ser:

- Negação: não;
- Conjunção: e ;
- Disjunção: ou; e
- Disjunção exclusiva: xou.

Exemplos

$(10 = 10)$ e $(0 < 1)$ *(verdadeiro)*

não$(0 <> 0)$ *(verdadeiro)*

$(0 < -50)$ ou $(0 < 1)$ *(verdadeiro)*

$(10 < 20)$ xou $(0 > -1)$ *(falso)*

Tabela-verdade

Construção

A	B	A e B	A ou B	A xou B	Não A
V	V	V	V	V	F
V	F	F	V	F	F
F	V	F	V	F	V
F	F	F	F	V	V

Estrutura de Seleção Aninhada

Definição

É uma estrutura para desvio de fluxo do programa formada pelo comando de decisão *se-então/senão* mais sub-estruturas de decisão.

Exemplo:

```
algoritmo "ExemploFormaPagamento"

var tipo :caracter

inicio
   escreva("Digite o tipo de pagamento:")
   leia(tipo)
   se numero = "d" entao
      escreva("Pagamento em débito.")
   senao
      se tipo = "c" entao
         escreva("Pagemento em crédito.")
      senao
         escreva("Tipo de pagamento inválido!")
      fimse
   fimse
fimalgoritmo
```

Estrutura de Seleção *Escolha*

Definição

Também citada na literatura por Seleção de Múltipla Escolha ou Caso. O comando escolha pode ser visto como uma especialização do comando *se*, e compara um dado valor a uma constante, desviando o fluxo de código para o ponto indicado pela primeira constante onde há casamento.

Exemplo da escolha de uma disciplina pela siglas

Escolha siglas

AL: Algoritmo

IA: Inteligência Artificial

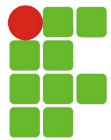
BD: Banco de Dados

LC: Lógica Computacional

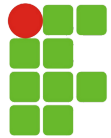
Comando *Escolha* no Visualg

Exemplo

```
algoritmo "ExemploDeposito"

var tipo:caracter

inicio
   escreva("Informe o tipo de depósito:")
   leia(tipo)
   escolha tipo
      caso "cc"
          escreva("Conta corrente.")
      caso "cp"
          escreva("Conta poupança.")
      outrocaso
          escreva("Tipo de conta inválido!")
   fimescolha
fimalgoritmo
```



Ementa do Curso



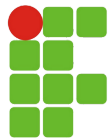
O que é Estrutura de Repetição?

Definição

Uma estrutura de repetição é uma estrutura de desvio do fluxo de controle presente em linguagens de programação que realiza e repete diferentes computações ou ações dependendo se uma condição é *verdadeira* ou *falsa*, em que a expressão é processada e transformada em um valor *booleano*.

O processo de controle de parada das estruturas de repetição podem ser:

- Repetição controlada por variável de controle (Estrutura: *para*);
- Repetição pré-testada (Estrutura:*enquanto*); ou
- Repetição pós-testada (Estrutura: *repita*).



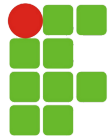
Estrutura de Repetição: *para*

Definição

A construção *para*, ou repetição com variável de controle, é uma estrutura de repetição que designa uma variável de controle para cada iteração do bloco, e uma operação de passo a cada iteração.

Exemplo de uma contagem automática de 1 até 10

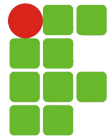
Para contador **de** 1 **até** 10 **faça**
    escreva contador



Estrutura de Repetição *para* no Visualg

Exemplo

```
algoritmo "ExemploContador"

var i :inteiro

inicio
   para i de 1 ate 10 faca
       escreval(i)
   fimpara
fimalgoritmo
```



Comando *Interrompa* no Visualg

Definição

O comando *interrompa* permite parar uma execução de uma instrução de repetição toda vez que o mesmo for invocado, ignorando, caso ainda existam, outras instruções a serem executadas.

Exemplo

```
algoritmo "ExemploContador"

var i :inteiro

inicio
   para i de 1 ate 10 faca
        escreval(i)
        se i = 5 entao
           interrompa
        fimse
   fimpara
fimalgoritmo
```

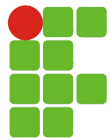
Estrutura de Repetição: *enquanto*

Definição

A construção *enquanto* (também chamada repetição pré-testada) é a mais difundida estrutura de repetição. O processo de repetição continua enquanto o valor da expressão de controle for *verdadeiro*.

Exemplo de uma contagem automática de 1 até 10

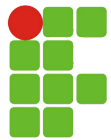
Enquanto contador $\leq$ 10 **faça**
    escreva contador
    incrementa contador



Estrutura de Repetição *enquanto* no Visualg

Exemplo

```
algoritmo "ExemploContador"

var i :inteiro

inicio
   i <- 1
   enquanto i <= 10 faca
       escreval(i)
       i <- i + 1
   fimenquanto
fimalgoritmo
```



Estrutura de Repetição: *repita*

Definição

A construção *repita* (também chamada repetição pós-testada) é uma variação da construção apresentada anterior *enquanto*, e difere pois a verificação da condição é feita após uma execução do bloco. O processo de repetição continua enquanto o valor da expressão de controle for *falso*.

Exemplo de uma contagem automática de 1 até 10

Repita

escreva contador

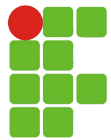
incrementa contador

Até contador $>$ 10

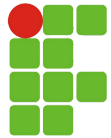
Estrutura de Repetição *Repita* no Visualg

Exemplo

```
algoritmo "ExemploContador"

var i :inteiro

inicio
   i <- 1
   repita
        escreval(i)
        i <- i + 1
   ate i > 10
fimalgoritmo
```



Ementa do Curso



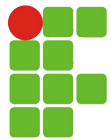
O que são Procedimentos?

Definição

São subrotinas (módulos ou métodos) de programas, capazes de executar uma tarefa definida pelo programador, mas que não retorna nenhum valor. Os programas desenvolvidos com procedimentos são ditos modulares.

As principais vantegens em criar programas usando subrotinas são:

- Melhor organização do programa;
- Reutilização da subrotina em outras partes do programa; e
- Facilidade de manutenção do código.

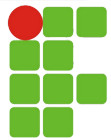


Estrutura de um Procedimento

- Todo procedimento deve ter um identificador;
- Pode possuir um conjunto de parâmetros (não é obrigatório);
- Permite declaração de variáveis locais; e
- Possui um bloco de instruções.

Exemplo

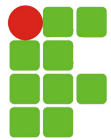
```
procedimento nome (parâmetros)
variáveis locais
início
        instruções ...
fim.
```



Procedimento no Visualg I

Exemplo

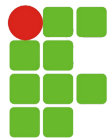
```
algoritmo "ExemploProcedimento"

procedimento exibir_cumprimento
inicio
   escreva("Oi Mundo!")
fimprocedimento

inicio
   exibir_cumprimento
fimalgoritmo
```



Procedimento no Visualg II

Exemplo

```
algoritmo "ExemploProcedimento"

procedimento exibir_cumprimento(n:caracter)
inicio
   escreva("Oi ", n,"!")
fimprocedimento

var nome: caracter

inicio
   escreva("Digite seu nome:")
   leia(nome)
   exibir_cumprimento(nome)
fimalgoritmo
```



Tipos de Passagem de Parâmetros

A passagem de parâmetros pode ser de dois tipos:

- por valor; e
- por referência.

Valor

Informamos o valor a ser trabalhado e indiferente de quais modificações serão feitas com essas informações, seus valores originais permenecem o mesmo.

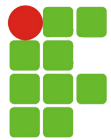
Referência

Informamos o valor a ser trabalhado e de acordo com as mudanças que vão sofrendo, os valores originais vão sendo atualizados. Passagem por referência somente é feita passando-se variáveis como parâmetro.

Procedimento no Visualg III

Exemplo

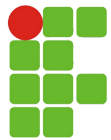
```
algoritmo "ExemploPassagemPorValor"

procedimento incrementar(n:inteiro)
inicio
   n <- n + 1
fimprocedimento

var numero: inteiro

inicio
   escreva("Digite um número inteiro:")
   leia(numero)
   incrementar(numero)
   escreva("O incremento unário é ", numero, ".")
fimalgoritmo
```



Procedimento no Visualg IV

Exemplo

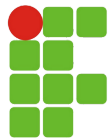
```
algoritmo "ExemploPassagemPorReferencia"

procedimento incrementar(var n:inteiro)
inicio
   n <- n + 1
fimprocedimento

var numero: inteiro

inicio
   escreva("Digite um número inteiro:")
   leia(numero)
   incrementar(numero)
   escreva("O incremento unário é ", numero, ".")
fimalgoritmo
```



O que são Funções?

Definição

São rotinas similares aos procedimentos, só que retornam um valor após cada chamada. Uma função não deverá simplesmente ser chamada, como no caso dos procedimentos, **mas deverá ser atribuída à alguma variável**.



Estrutura de uma Função

- Toda função deve ter um identificador;
- Pode possuir um conjunto de parâmetros;
- Obrigatoriamente deve retornar um valor;
- Permite declaração de variáveis locais; e
- Possui um bloco de instruções.

Exemplo

```
função nome (parâmetros) : tipo de retorno
variáveis locais
início
        instruções ...
fim.
```

Função no Visualg

Exemplo

```
algoritmo "ExemploFuncao"

funcao somar(a,b:inteiro):inteiro
        var r:inteiro
inicio
   r <- a + b
   retorne r
fimfuncao

var valor1, valor2, resultado :inteiro

inicio
   escreva("Digite o primeiro valor:")
   leia(valor1)
   escreva("Digite o segundo valor:")
   leia(valor2)
   resultado <- somar(valor1, valor2)
   escreva("A soma é ", resultado)
fimalgoritmo
```