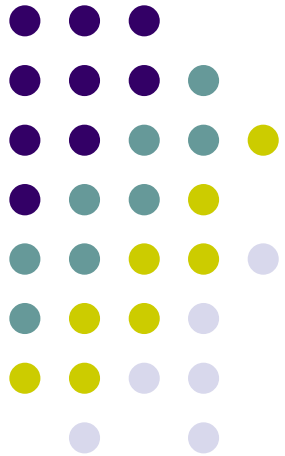
Fundamentos de Programação

Instituto Federal de Mato Grosso

Tecnologia em Desenvolvimento de Sistemas para Internet

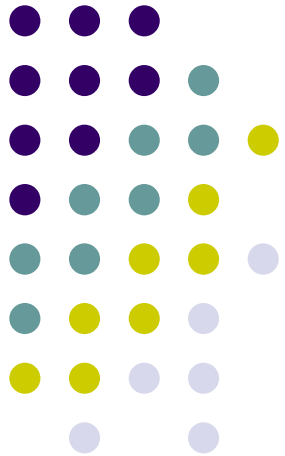


Prof. Alberto Sales

Objetivos

- O que é um Processo
- Conhecer o conceito de “Algoritmos”;
- Detalhar os conceitos de programa computacional;

Conceitos de Processos



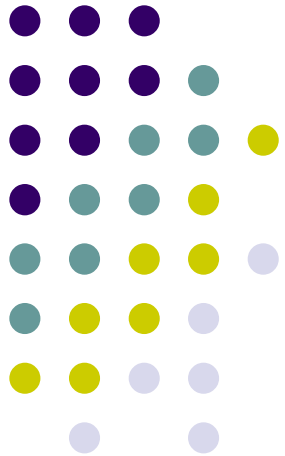
Prof. Alberto Sales

Processos

- Quando falamos de **processo** estamos nos remetendo à um conceito que representa "**módicamente**" a **transformação** de algo (**dados** ou **informações**) em algum "**produto**" com objetivo de atendimento de alguma necessidade seja de uma empresa, pessoa ou mesmo um equipamento.
- Processo requer alguma **entrada** (por ex.: dados), transformação (**processamento** propriamente dito) e uma **saída** (resultado esperado)

entrada → processamento → saída

Conceitos de Algoritmos

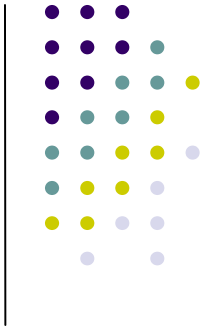


Prof. Alberto Sales

Algoritmos

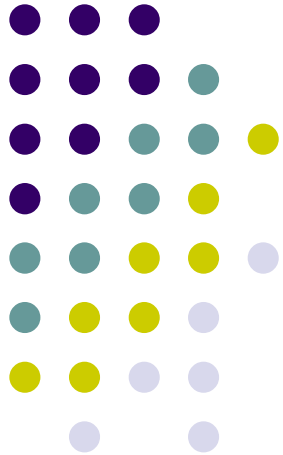
- Um algoritmo é uma **sequência** extremamente precisa de instruções que, quando lida e executada por uma outra pessoa, produz o resultado esperado, isto é, a solução de um problema.
  - Esta sequência de instruções é nada mais nada menos que um registro escrito da sequência de passos necessários que devem ser executados para manipular informações, ou dados, para se chegar na resposta do problema.

O que é um programa?

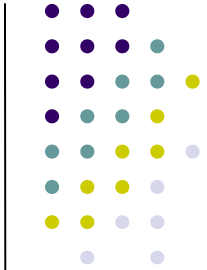


- Um **programa** é a **codificação** em alguma **linguagem formal** que garanta que os passos do **algoritmo** sejam executados da maneira como se espera por quem executa as instruções.
  - Um **programa**, neste sentido, é um **algoritmo** escrito de forma tão detalhada quanto for necessário para quem executa as instruções.
    - O algoritmo pode ser mais "**genérico**", o programa não.

Conceitos elementares

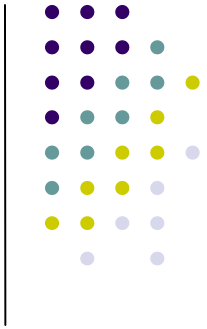


Algoritmos e linguagens de programação



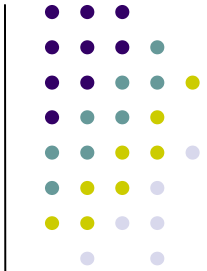
- Os algoritmos devem ser escritos em um nível de detalhamento suficiente para que o compilador consiga fazer a tradução do código para linguagem de máquina.

Algoritmos e linguagens de programação



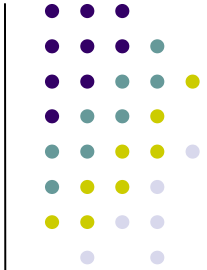
- **Variáveis**
  - São utilizadas para armazenar dados.
  - Neste sentido, uma variável pode ser utilizada para receber dados do usuário para depois ser de alguma forma processada.
  - É utilizada também para armazenar resultado de algum processamento
  - ex.:
    - inteiro idade = 16
    - texto nome = “Maria José da Silva”

Algoritmos e linguagens de programação



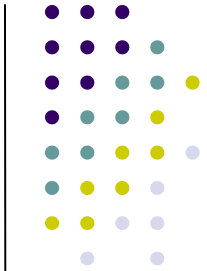
- **Comandos/Funções**
  - São utilizadas para determinar ou auxiliar algum processamento.
  - ex.:
    - **escreva**(“a idade do joão é 16 anos”)

Algoritmos e linguagens de programação



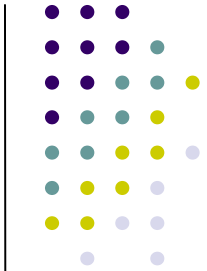
- Os algoritmos devem ser escritos em um nível de detalhamento suficiente para que o compilador consiga fazer a tradução do código para linguagem de máquina.

Algoritmos e linguagens de programação



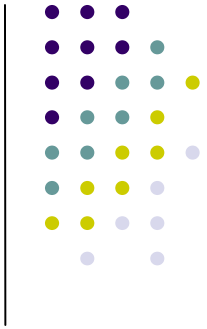
- A base das linguagens de programação, em geral, é constituída por:
  - a noção de fluxo de execução de um programa;
  - os comandos da linguagem que manipulam os dados em memória e a interação com o usuário (atribuição, entrada e saída de dados);

Algoritmos e linguagens de programação



- A base das linguagens de programação, em geral, é constituída por:
  - as expressões da linguagem que permitem a realização de cálculos aritméticos e lógicos;
  - os comandos da linguagem que modificam o fluxo de execução de um programa.
  - **Atenção: cada linguagem de programação utiliza uma sintaxe própria para a escrita dos comandos.**

Fluxo de execução de um programa



- Utilizaremos um contexto de programação que nos permitirá “traduzir” para o português, o **PORTUGOL**.

Exemplo:

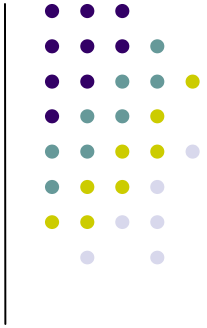
```
1 programa OlaMundo
2 {
3    funcao inicio ()
4    {
5          escreva("Olá Mundo!\n")
6    }
7 }
```

Comandos de entrada e saída

- Toda linguagem de programação possui comandos que permitem a um usuário fornecer dados ao computador pelo teclado ou outro meio físico (interação) e também receber "**retornos**" do computador através do monitor de video, impressora é outro exemplo.

```
1  programa NumeroDigitado
2  {
3      funcao inicio ()
4      {
5              inteiro num
6              escreva("Digite um número inteiro:")
7              leia(num)
8              escreva("O número digitado foi: ", num, "\n")
9      }
10 }
```

Funcionamento de Um Programa



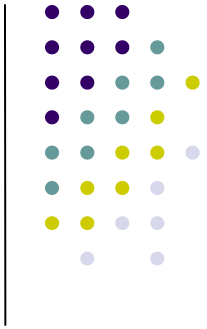
1) Antes de iniciar o programa principal, o compilador solicita ao sistema ope racional que reserve um espaço de memória para a variável a, que deve ser interpretada como um número real;

2) Inicia o programa pela primeira instrução (read(a);), que faz com que o com putador espere o usuário digitar algo no teclado e apertar ENTER. Se o usuário digitar, por exemplo, o número 2, então o endereço de memória associado à a conterá o valor 2.

3) Executa a segunda instrução (write(a);), que faz com que o computador busque na memória o valor (o conteúdo) de a, que é 2, e imprima esta informação, isto é, 2, na tela.

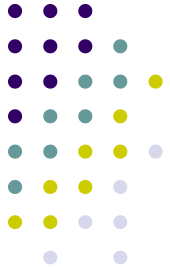
4) O programa termina.

Atribuições



- O comando de atribuição vai depender muito da linguagem de programação utilizada. **Java**, **Dart** e a linguagem **C** utilizam o mesmo caracter "**=**", já Pascal utiliza "**:=**" .
- Este comando serve para armazenar uma informação em uma variável, isto é, em um endereço de memória.
  - Por exemplo: **a = 2**, faz com que a variável a tenha como seu conteúdo o número 2.

Expressões Aritméticas



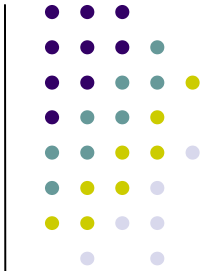
- As expressões aritmétias realizam procedimento matemáticos e calculam equações, funções financeiras avançadas e etc.

```
1  programa
2  {
3      funcao inicio()
4      {
5              real a, b, soma, sub, mult, div
6              escreva("Digite o primeiro número: ")
7              leia(a)
8              escreva("Digite o segundo número: ")
9              leia(b)
10             soma = a + b // Soma os dois valores
11             sub  = a - b // Subtrai os dois valores
12             mult = a * b // Multiplica os dois valores
13             div  = a / b // Divide os dois valores
14             escreva("\nA soma dos números é igual a: ", soma)          // Exibe o resultado da soma
15             escreva("\nA subtração dos números é igual a: ", sub)      // Exibe o resultado da subtração
16             escreva("\nA multiplicação dos números é igual a: ", mult) // Exibe o resultado da multiplicação
17             escreva("\nA divisão dos números é igual a: ", div, "\n")  // Exibe o resultado da divisão
18     }
19 }
```

Fluxograma do algoritmo

```
1  programa
2  {
3     funcao inicio()
4     {
5        inteiro idade
6        real salario, nota1, nota2, nota3
7        cadeia nome, sobrenome
8        escreva("Informe a sua idade: ")
9        leia (idade)              //lê o valor digitado para "idade"
10        escreva("Informe seu salario: ")
11        leia (salario)            //lê o valor digitado para "salario"
12        escreva("Informe o seu nome e sobrenome: ")
13        leia (nome, sobrenome)     //lê o valor digitado para "nome" e "sobrenome"
14        escreva("Informe as suas três notas: ")
15        leia (nota1, nota2, nota3)  //lê o valor digitado para "nota1", "nota2" e "nota3"
16        escreva("Seu nome é:"+nome+" "+sobrenome+"\n")
17        escreva("Você tem "+idade+" anos e ganha de salario "+salario+"\n")
18        escreva("Suas três notas foram:\n")
19        escreva("Nota 1: "+nota1+"\n")
20        escreva("Nota 2: "+nota2+"\n")
21        escreva("Nota 3: "+nota3+"\n")
22     }
23  }
```

Exemplificação de atribuição em variável



- Uma das operações mais comuns em programas é uma variável ser incrementada de uma unidade
  - Por exemplo, a variável **i** recebe inicialmente o valor **zero** e imprime este **zero** na tela.
  - O comando seguinte (**i = i + 1**) faz o incremento de uma unidade no valor de i e por isso o segundo comando de impressão imprime o valor 1

```
1  programa
2  {
3          funcao inicio()
4          {
5          real i
6          i = 0  // atribui zero à variável i
7          escreva("O valor de i é: ", i) // Chama a função no escreva
8          I =  i + 1
9          escreva("\nO novo valor de i é: ", i)
10     }
11 }
```

Exemplos em Portugol:

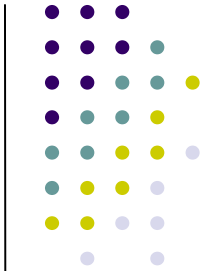
- Utilizando **Portugol Web Studio**
  - Algoritmo em portugol

```
programa
{
    funcao inicio ()
    {
        cadeia nome
        escreva("Digite seu nome: ")
        leia(nome)
    }
}
```

- Portugol em Pascal
  - Escrita semelhante ao **Portugol Web Studio**

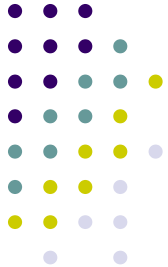
```
Algoritmo Soma_dois_numeros;

    Var
        N1, N2, Soma : real;

    Início
        Leia N1, N2;
        Soma := N1 + N2;
        Escreva Soma;
    Fim.
```

Exemplos com Entrada de Dados e Processo Aritmético



- O exemplo ANTERIOR pede ao usuário que informe dois números. Logo após:
  - calcula e exibe:
    - A soma entre os números
    - A subtração entre os números
    - A multiplicação entre os números
    - A divisão entre os números

Exemplo: Entrada de Dados e Cálculo – Portugol Web Studio

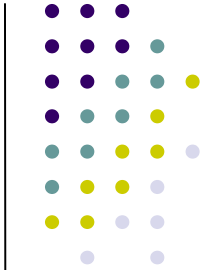


```
programa
{
  funcao inicio()
  {
    real a, b, soma, sub, mult, div
    escreva("Digite o primeiro número: ")
    leia(a)

    escreva("Digite o segundo número: ")
    leia(b)

    soma = a + b // Soma os dois valores
    sub  = a - b // Subtrai os dois valores
    mult = a * b // Multiplica os dois valores
    div  = a / b // Divide os dois valores

    escreva("\nA soma dos números é igual a: ", soma)     // Exibe o resultado da soma
    escreva("\nA subtração dos números é igual a: ", sub)     // Exibe o resultado da subtração
    escreva("\nA multiplicação dos números é igual a: ", mult)  // Exibe o resultado da multiplicação
    escreva("\nA divisão dos números é igual a: ", div, "\n")   // Exibe o resultado da divisão
  }
}
```

Interação com o usuário e saída esperada

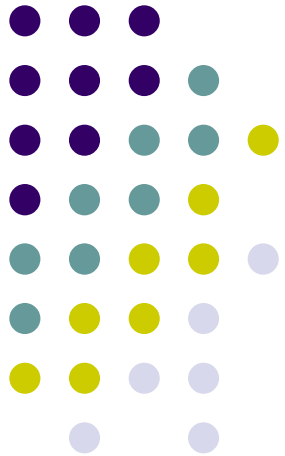
```
Digite o primeiro número: 3
Digite o segundo número: 4

A soma dos números é igual a: 7
A subtração dos números é igual a: -1
A multiplicação dos números é igual a: 12
A divisão dos números é igual a: 0.75
```

Exemplo: Entrada de Dados e Cálculo – Portugol (Pascal)



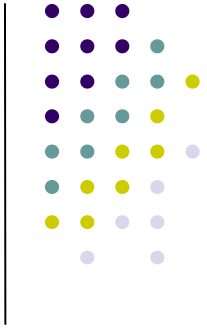
```
Algoritmo Exemplos_declaracao_de_variaveis;

Var
     NOME   : literal[30];
     IDADE  : inteiro;
     ALTURA: real;
     CASADO: lógico;

Início
     NOME   := 'Ivo Mario Mathias';
     IDADE  := 55;
     ALTURA := 1.75;
     CASADO:= .V.;

     Escreva ('Nome completo: ', NOME);
     Escreva ('Idade: ', IDADE);
     Escreva ('Altura: ',ALTURA);
     Escreva ('Casado (S/N): ',CASADO);
Fim.
```

Variáveis e expressões booleanas



Prof. Alberto Sales

Variáveis e expressões booleanas

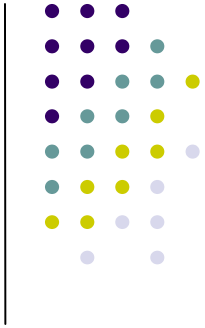


As **expressões booleanas**, também conhecidas como expressões lógicas, são utilizadas para resultar em um valor do tipo **booleano**, que passa a ser o terceiro tipo básico de qualquer linguagem de **Programação.**

- O tipo **booleano** só pode conter dois valores: verdadeiro (**true**) e falso (**false**).
- O caso mais simples de uma expressão booleana é uma variável do tipo booleano. Por exemplo:

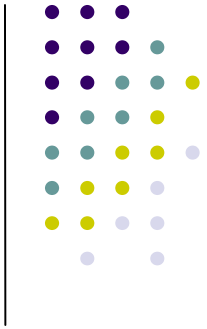
```
logico ok
```

Variáveis e expressões booleanas



- Exemplo
  - **A > 0**
  - **A + B == X + Y**
  - **A == B ou B == C**
  - **A == B e B == C**

Variáveis e expressões booleanas



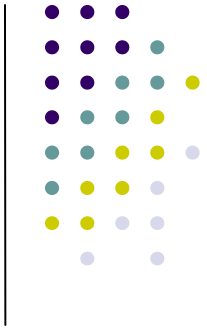
- Assim podemos fazer uma atribuição de um valor verdadeiro ou falso para esta variável:

OK = **verdadeiro**; ou **OK** := **falso (Pascal)**.

- As expressões booleanas mais complexas podem envolver o uso de expressões aritméticas juntamente com os **operadores relacionais**, que são:

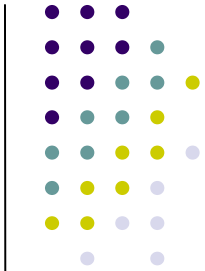
>	(**maior**),
>=	(**maior** ou **igual**),
<	(**menor**),
<=	(**menor** ou **igual**),
==	(**igual**)
!=	(**diferente**)

Variáveis e expressões booleanas



- Também é possível usar conectivos lógicos, que são: o **ou** lógico (OR), o **e** lógico (AND) e a **negação** lógica (NOT).

Tabela Verdade e Expressões Booleanas

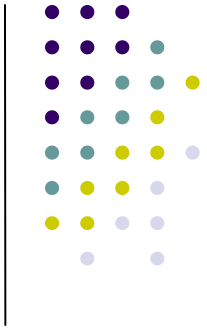


- Valor lógico "**e**" **(&, ∧)**

Variáveis

	A	B	A e B
valores	V	V	V
	V	F	F
	F	V	F
	F	F	F

Tabela Verdade e Expressões Booleanas

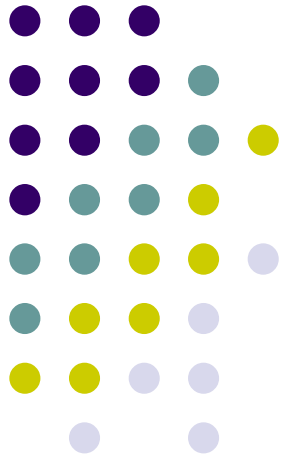


Valor lógico “**ou**” **(|, v)**

Variáveis

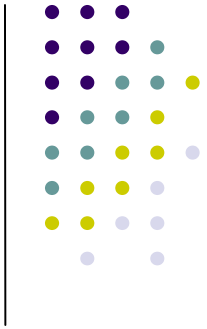
	A	B	A ou B
valores	V	V	V
	V	F	V
	F	V	V
	F	F	F

Estruturas de Decisão



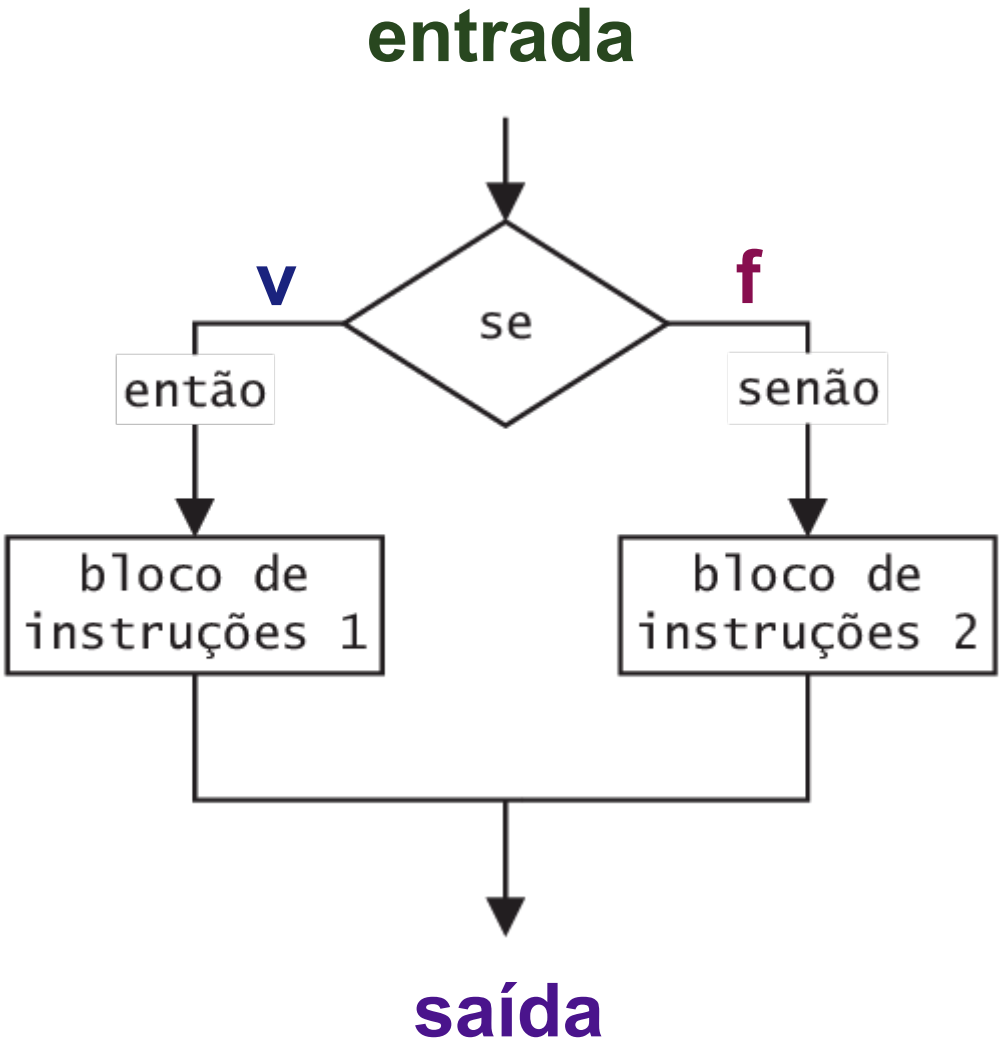
Prof. Alberto Sales

Estruturas de Decisão

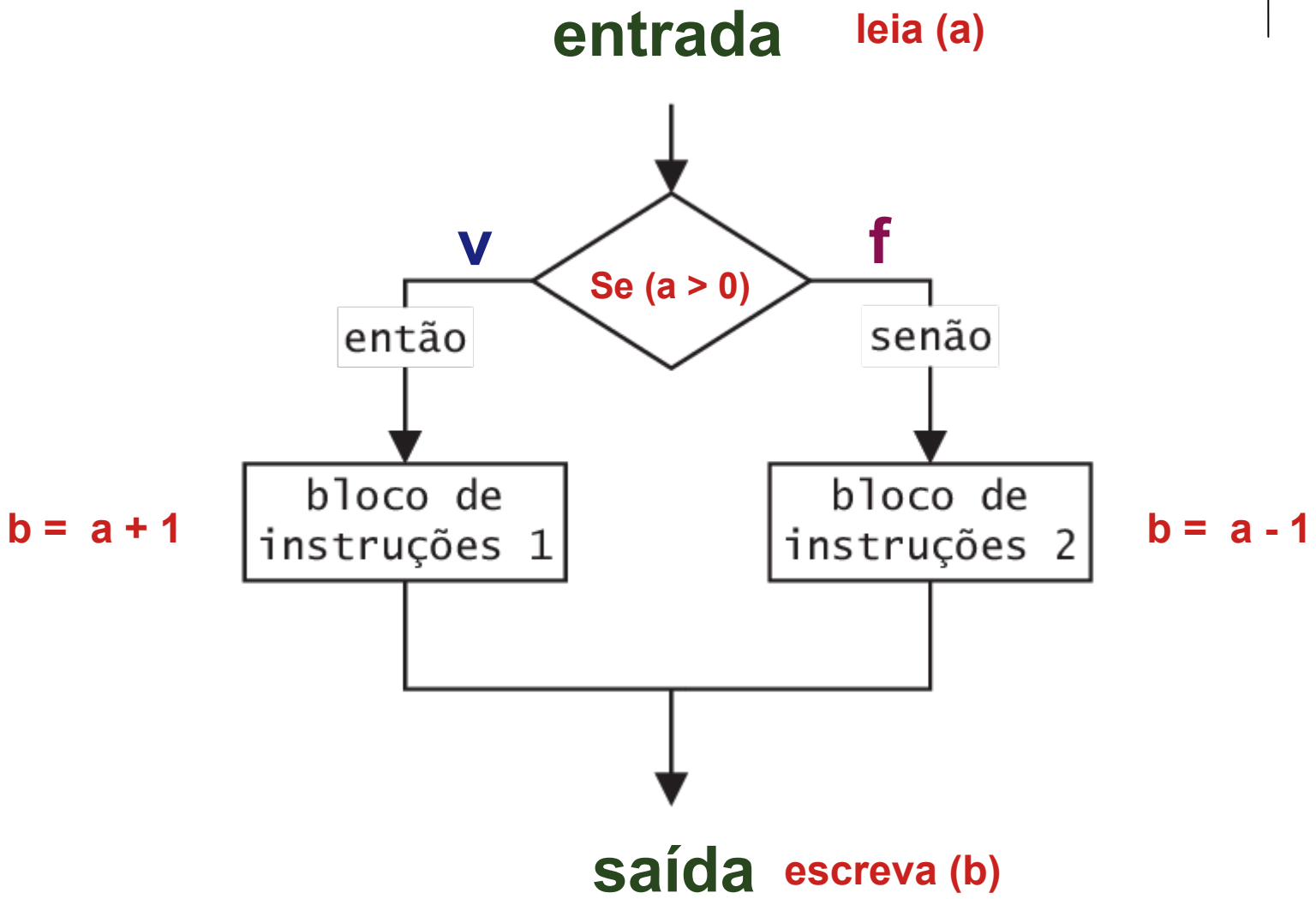


- **Tomadas de decisão** são um dos fundamentos da programação e algoritmos.
- Podemos utilizá-las para **forçar** uma determinada escolha, assim:
  - Uma entrada de dados pode ser em **função de uma decisão**
    - Algo como utilizar uma informação de acordo com uma decisão tomada
    - Realizar uma cálculo em função do tipo de dados
  - OU mesmo realizar uma **processamento específico** em função de uma tomada de decisão

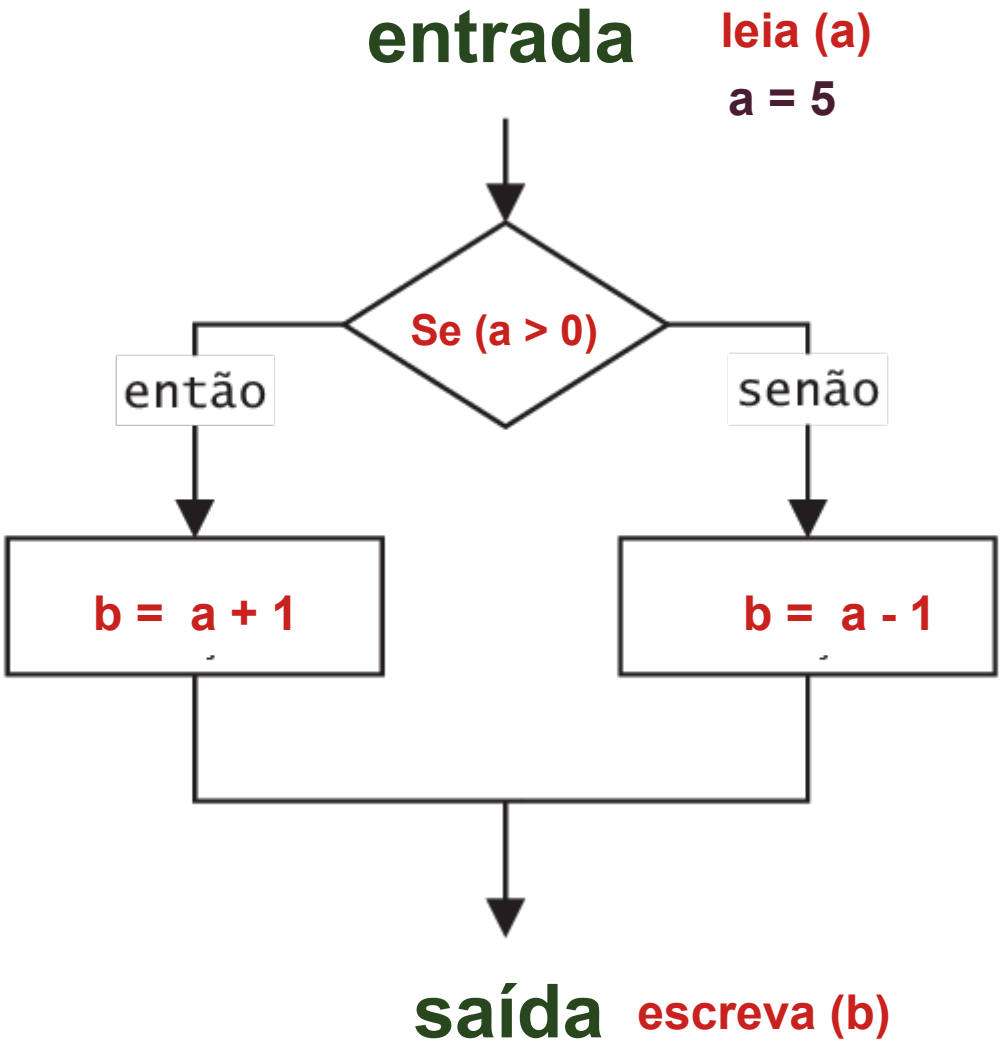
Estruturas de Decisão: fluxograma



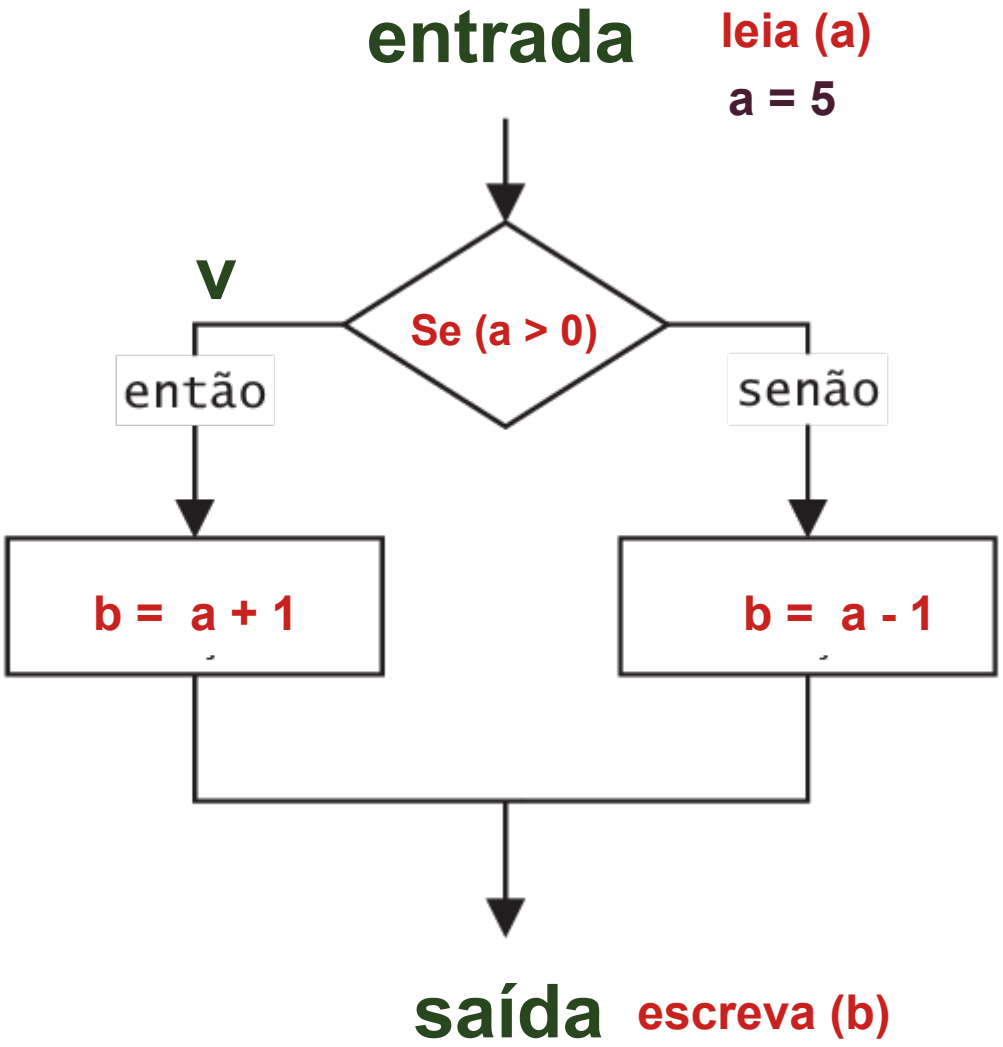
Estruturas de Decisão: fluxograma



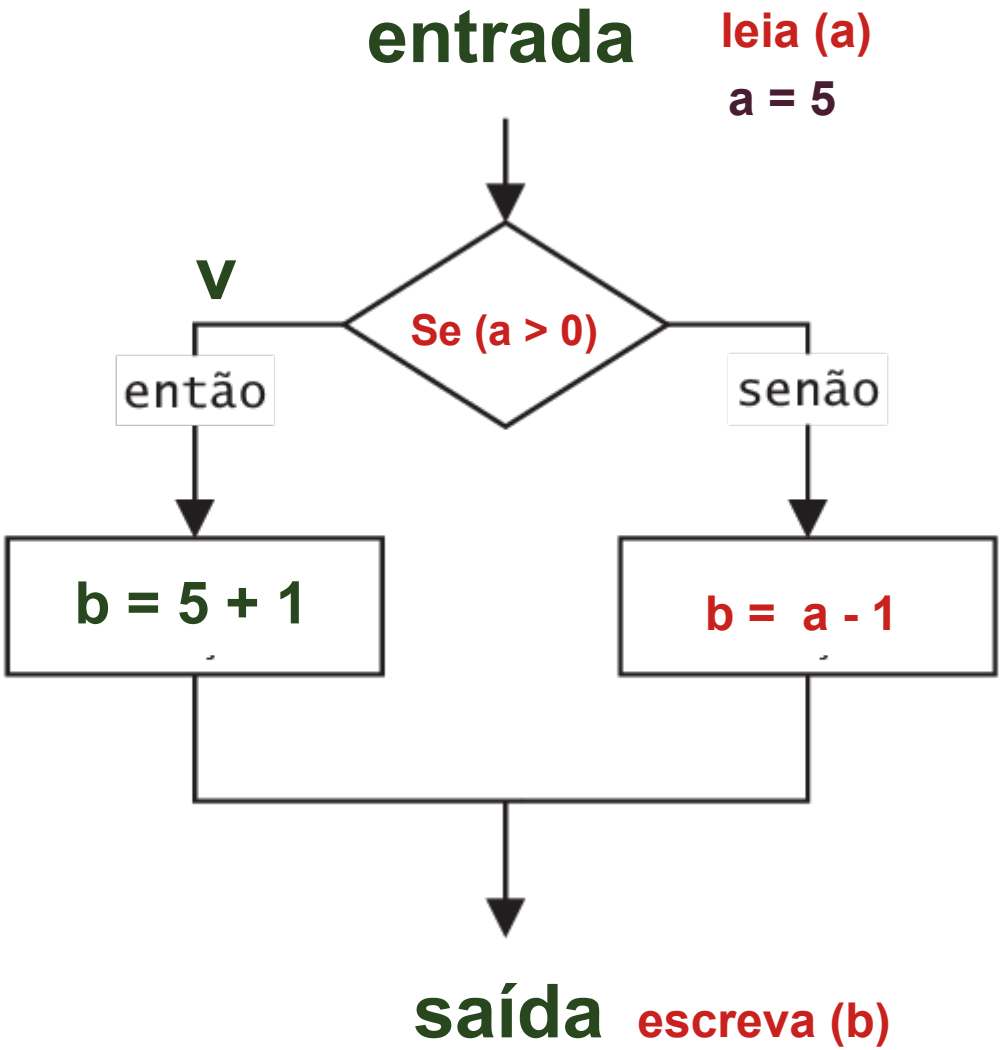
Estruturas de Decisão: fluxograma



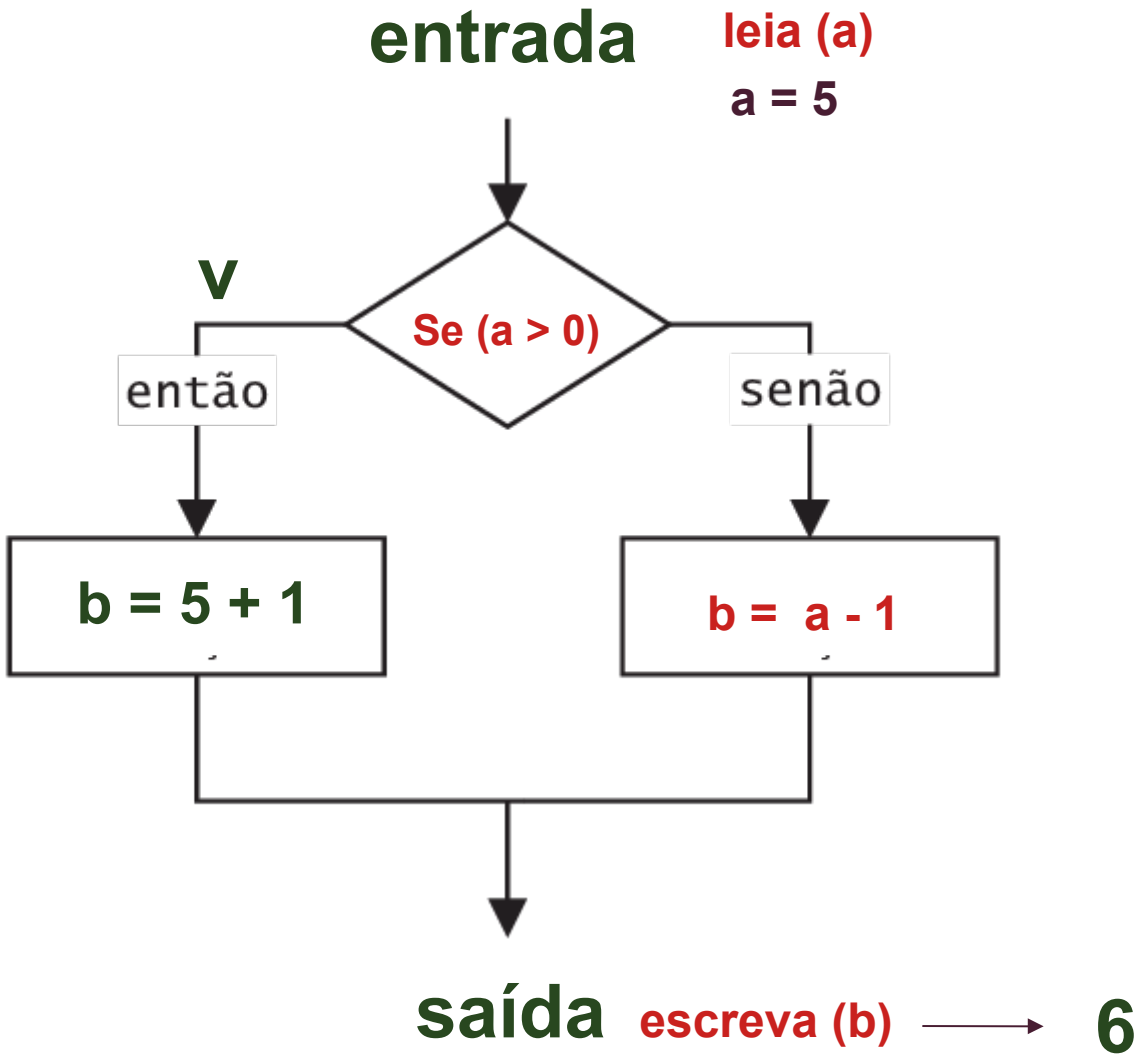
Estruturas de Decisão: fluxograma



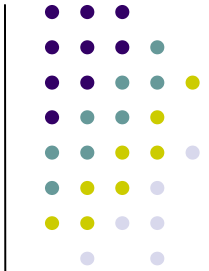
Estruturas de Decisão: fluxograma



Estruturas de Decisão: fluxograma



Estruturas de Decisão: algoritmo em Portugol Web

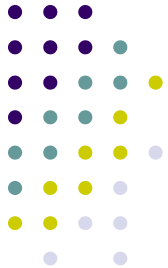


- Exemplo em Portugol Web Studio:

```
se <condição>
{
  <comando a ser executado>
}

Exemplo:
se (m2 < media)
{
  escreva ("A média 2 é menor que a média final\n")
}
```

  - Este algoritmo verifica se a média (**m2**) é menor que a média:
    - **media** = (**m1** + **m2** + **m3**) / **3**

```
1  programa
2  {
3      inclua biblioteca Matematica --> mat
4      funcao inicio ()
5      {
6              real m1, m2, m3, media
7              escreva ("Informe a média 1: " )
8              leia (m1)
9              escreva( "Informe a média 2: ")
10             leia (m2)
11             escreva ("Informe a média 3: ")
12             leia (m3)
13             media = (m1 + m2 + m3) / 3
14             limpa()
15             escreva ("A média final é: ", mat.arredondar(media, 2), "\n\n")
16             se (m1 < media)
17             {
18                     escreva ("A média 1 é menor que a média final\n")
19             }
20              se (m2 < media)
21             {
22                     escreva ("A média 2 é menor que a média final\n")
23             }
24             se (m3 < media)
25             {
26                     escreva ("A média 3 é menor que a média final\n")
27             }
28     }
29 }
```



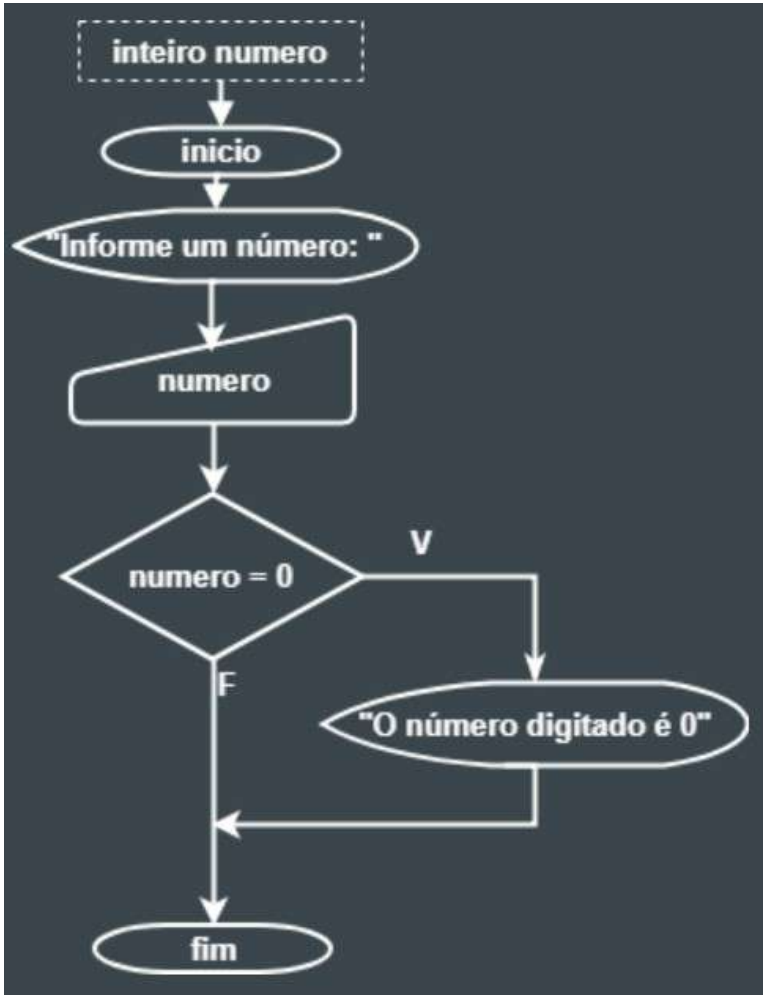
```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro lado_a, lado_b, lado_c
6              escreva ("Informe o primeiro lado do triângulo: ")
7              leia (lado_a)
8              escreva ("Informe o segundo lado do triângulo: ")
9              leia (lado_b)
10             escreva ("Informe o terceiro lado do triângulo: ")
11             leia (lado_c)
12             se (lado_a == lado_b e lado_a == lado_c)
13             {
14                     escreva ("\nEste triângulo é Equilátero\n")
15             }
16             senao
17             {
18                     se (lado_a == lado_b ou lado_b == lado_c ou lado_c == lado_a)
19                     {
20                             escreva ("\nEste triângulo é Isósceles\n")
21                     }
22                     senao
23                     {
24                             escreva ("\nEste triângulo é Escaleno\n")
25                     }
26             }
27     }
28 }
```

// Se os três lados forem
iguais é equilatero

// Se chegou aqui é porque
os três lados não são iguais
// Basta ver se dois deles
são iguais para saber se é isóceles

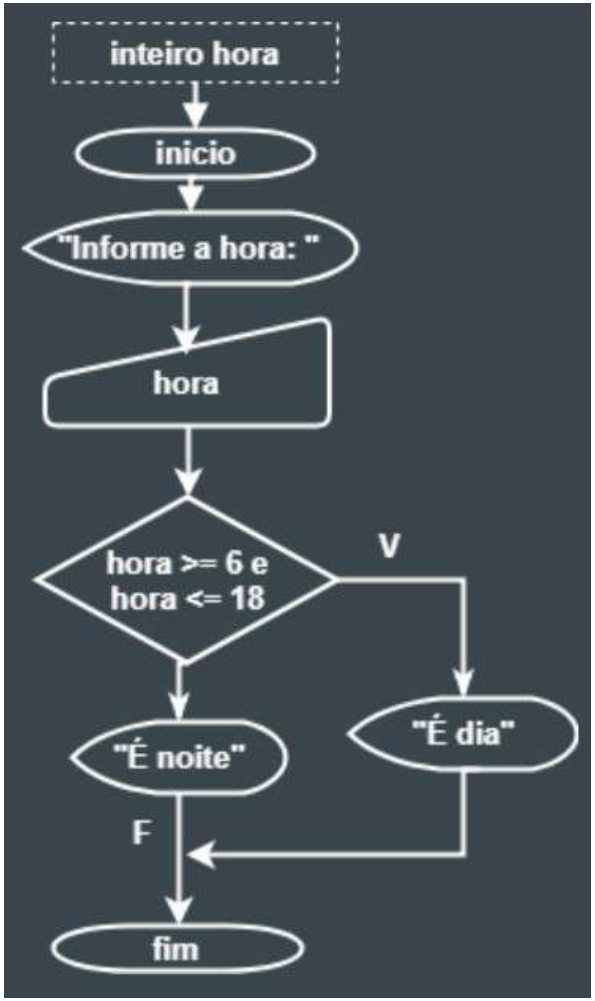
Estruturas de Decisão: Fluxograma

```
1  programa
2  {
3          funcao inicio()
4          {
5
6              inteiro num
7
8              escreva ("Informe um número: ")
9              leia (num)
10
11              se (num == 0)
12              {
13                  escreva ("O número digitado é 0")
14              }
15
16          }
17  }
18
```



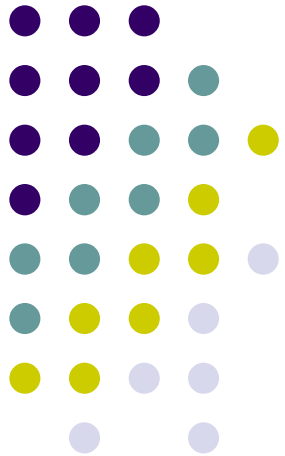
Estruturas de Decisão: Exemplo com Fluxograma

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro hora
6          escreva ("Digite a hora: ")
7          leia (hora)
8
9          se (hora >= 6 e hora <= 18)
10         {
11             escreva ("É dia")
12         }
13         senao
14         {
15             escreva ("É noite")
16         }
17
18     }
19 }
```



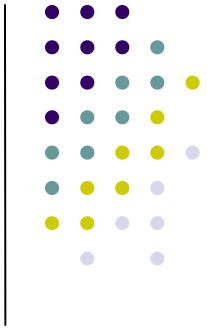
Estruturas de Repetição

Para e Enquanto



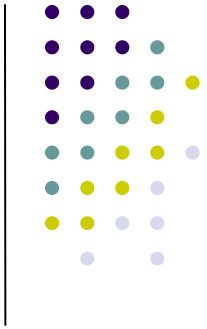
Prof. Alberto Sales

Estrutura de Repetição de Comandos



- Estrutura de repetição em algoritmo
  - Uma estrutura de repetição é utilizada quando um trecho do algoritmo, ou até mesmo o algoritmo inteiro, precisa ser repetido.
  - O número de repetições pode ser fixo ou estar atrelado a uma condição.
  - Assim, existem estruturas para tais situações, descritas a seguir.

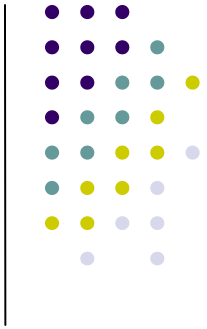
Estrutura de Repetição de Comando PARA



- Estrutura de repetição para número definido de repetições (estrutura **PARA**)
  - Essa estrutura de repetição é utilizada quando se sabe o número de vezes que um trecho do algoritmo deve ser repetido. O formato geral dessa estrutura é:

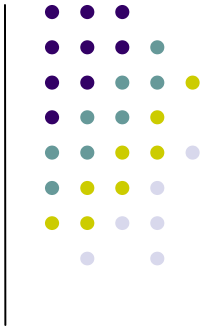
```
para( valor_inicial; valor_final; passo){
    comando 1
    comando 2
}
```

Estrutura de Repetição de Comando PARA



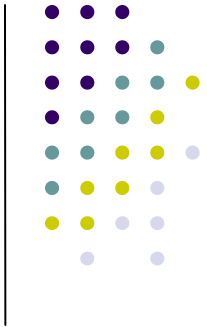
- Estrutura de repetição para número definido de repetições (estrutura **PARA**)
  - **para(valor_inicial; valor_final; [passo n]){**
    comando 1
    comando 2
    }
    - **O comando1, o comando2 e outros serão executados utilizando-se a variável como valor_inicial, e seu conteúdo vai variar do valor_inicial até o valor_final. A informação do PASSO está entre colchetes porque é opcional.**
      - **O PASSO indica como será a variação da variável de controle.**
      - **Por exemplo, quando for indicado PASSO 2, a variável de controle será aumentada em 2 unidades a cada iteração até atingir o valor_final**

Estrutura de Repetição de Comandos



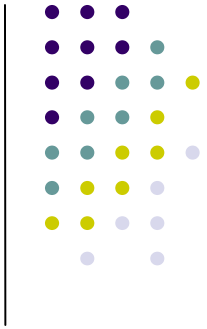
- Repetição de comandos é comumemente utilizado quando se deseja processar um cálculo de forma repetida (várias vezes o mesmo cálculo é executado)
  - ex.: Qual é a idade média dos estudantes?
    - para calcular a média da idade de todos os estudantes é preciso saber quais são as idades de cada um destes estudantes.
      - » Depois de informar a idade de cada estudante, soma-se e divide pelo total já lido no momento.

Estrutura de Repetição de Comandos



- Na repetição de comandos observamos que deve-se resolver um problema aplicados à várias situações diferentes relacionados aos dados propriamente dito, isto é:
  - Temos uma solução de um problema aplicado à vários dados “repetidos”.
    - Por exemplo:
      - Calcular o preço de venda de todos os produtos de uma empresa
      - Calcular o aumento de salários dos funcionários

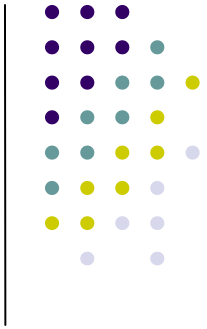
Estrutura de Repetição de Comandos



- Vamos calcular a média das notas dos estudantes de uma turma com 30 estudantes.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

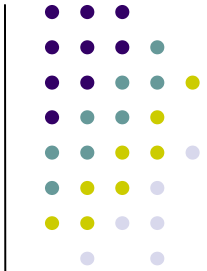
Estrutura de Repetição de Comandos



- Cada estudante é identificado por um código ou matrícula. No exemplo, um código que vai de 1 a 30

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

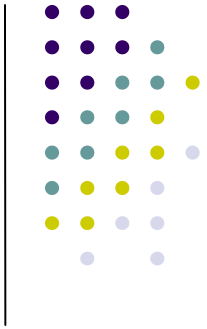
Estrutura de Repetição de Comandos



- Assim, deveremos saber a nota de cada estudante para depois calcular a média geral.

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

Estrutura de Repetição de Comandos



- Passo 1:
  - Ler a nota do primeiro estudante e armazenamos numa variável e já calculamos a média.

Estudante 1

Nota = **8**

média = **8**

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

Estrutura de Repetição de Comandos

- Passo 2:
  - Ler a nota do segundo estudante e armazenamos numa variável e já calculamos a média.

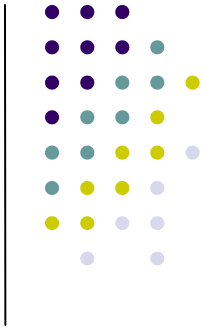
Estudante 2

Nota = **9**

média = **8,5**

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

Estrutura de Repetição de Comandos



- Passo 3:
  - Ler a nota do terceiro estudante e armazenamos numa variável e já calculamos a média.

Estudante 3

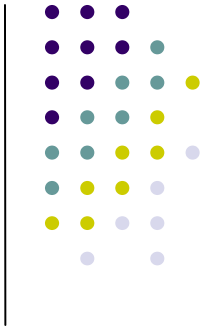
Nota = **5**

Média = **7,33**

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

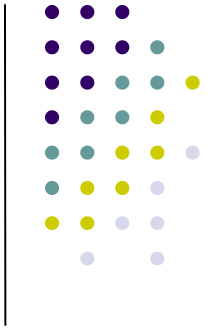
- E assim, faremos para todos….

Tipos de Repetição de Comandos



- **Laço Para (Com Variável de Controle)**
  - **Típico para resolver problemas onde sejam necessárias um número determinado de repetições**
    - **Idade total de 10 estudantes**
    - **Total de 100 produtos vendidos**
    - **Gerar gráfico de vendas referentes a 12 meses do ano**

Estrutura de Repetição de Comandos



- O laço de repetição com variável de controle facilita a construção de algoritmos com número definido de repetições, pois possui um contador (variável de controle) embutido no comando com o incremento automático.
  - O laço com variável de controle possui três partes.
    - A inicialização da variável contadora, (inteiro i = 0)
    - a definição do valor final do contador, (i < 8)
    - e a definição do incremento (contagem). (i = i + 1)

```
1 para ( inteiro i = 0; i < 8 ; i =  i + 1 )
2 {
3     //Codigo a ser executado enquanto a condição for satisfeita.
4 }
```

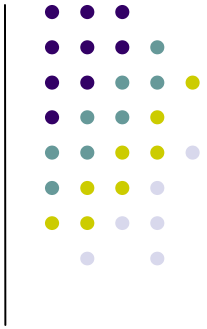
Estrutura de Repetição de Comandos

- O laço com variável de controle possui três partes.
  - **A inicialização da variável contadora, (inteiro i = 0)**
  - a definição do valor final do contador, (i < 8)
  - e a definição do incremento (contagem). (i = i + 1)

```
para ( inteiro i = 0; i < 8 ; i =  i + 1 )
{
   //Código a ser executado enquanto a condição    for satisfeita.
}
```

Primeira Parte

Estrutura de Repetição de Comandos



- O laço com variável de controle possui três partes.
  - A inicialização da variável contadora, (inteiro i = 0)
  - **a definição do valor final do contador, (i < 8)**
  - e a definição do incremento (contagem). (i = i + 1)

```
para ( inteiro i = 0; i < 8 ; i =  i + 1 )
{
   //Código a ser executado enquanto a condição    for satisfeita.
}
```

Segunda Parte

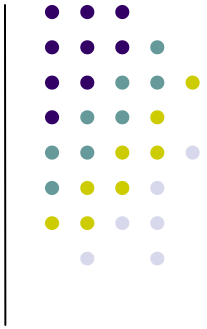
Estrutura de Repetição de Comandos

- O laço com variável de controle possui três partes.
  - A inicialização da variável contadora, (inteiro i = 0)
  - a definição do valor final do contador, (i < 8)
  - **e a definição do incremento (contagem). (i = i + 1)**

```
para ( inteiro i = 0; i < 8 ; i =  i + 1 )
{
   //Código a ser executado enquanto a condição    for satisfeita.
}
```

Terceira Parte

Estrutura de Repetição de Comandos



- Exemplo

```
programa {
    funcao inicio(){
        inteiro i,num, soma = 0; // inicialização da variável SOMA com o valor zero
        para(i = 1; i <= 5; i = i + 1)
        {
            escreva("Digite um número: ");
            leia(num);
            soma = soma + num; // acumulando o valor da variável NUM na variável SOMA
        }
        escreva("Soma = ",soma);
    }
```

Exemplo de algoritmo com repetição PARA

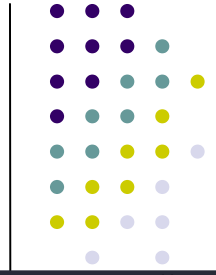
```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                      // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                      // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1		0	

Teste de mesa do algoritmo com repetição PARA



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	0	-

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                      // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

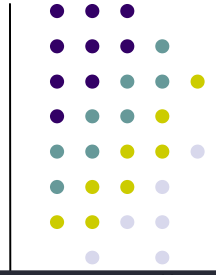
i	numero	soma	media
1	5	5	-

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2			-

Teste de mesa do algoritmo com repetição PARA



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7		-

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3			

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3		

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

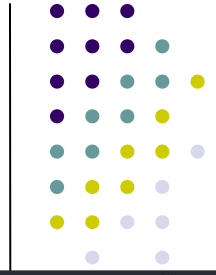
i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4			

Teste de mesa do algoritmo com repetição PARA



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

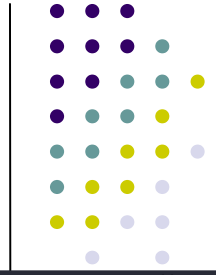
i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10		

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	

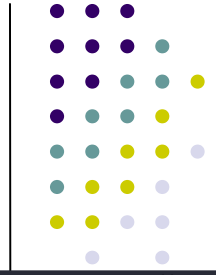
Teste de mesa do algoritmo com repetição PARA



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5			

Teste de mesa do algoritmo com repetição PARA



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5		

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

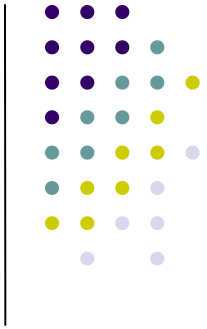
i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5	30	

Teste de mesa do algoritmo com repetição PARA

```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro i = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              para(i = 1; i <= 5; i = i + 1)
10             {
11                     limpa()
12                     escreva("Digite o ", i, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero          // A variavel soma é o acumulador deste exemplo
16                     // Incrementa o contador
17             }
18             media = soma / 5
19
20             limpa()
21             escreva("A média dos números é: ", media, "\n")
```

i	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5	30	
6			6

Estrutura de Repetição de Comandos



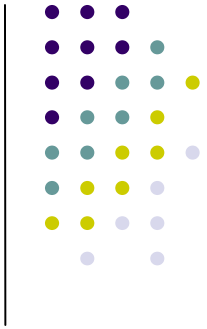
Calculando a tabuada de qualquer Número

```
1  programa
2  {
3    funcao inicio()
4    {
5        inteiro numero, resultado, contador
6
7        escreva("Informe um número para ver sua tabuada: ")
8        leia(numero)
9
10       limpa()
11
12       para (contador = 1; contador <= 10; contador++)
13       {
14           resultado = numero * contador
15           escreva (numero, " X ", contador, " = ", resultado , "\
n")
```

Parâmetro que inicializa a contagem da variável do laço

Laço Para

Estrutura de Repetição de Comandos



Calculando a tabuada de qualquer Número

```
1  programa
2  {
3    funcao inicio()
4    {
5        inteiro numero, resultado, contador
6
7        escreva("Informe um número para ver sua tabuada: ")
8        leia(numero)
9
10       limpa()
11
12       para (contador = 1; contador <= 10; contador++)
13       {
14           resultado = numero * contador
15           escreva (numero, " X ", contador, " = ", resultado , "\
n")
```

Parâmetro que determina o final da contagem do laço

Laço Para

Estrutura de Repetição Para

Calculando a tabuada de qualquer Número

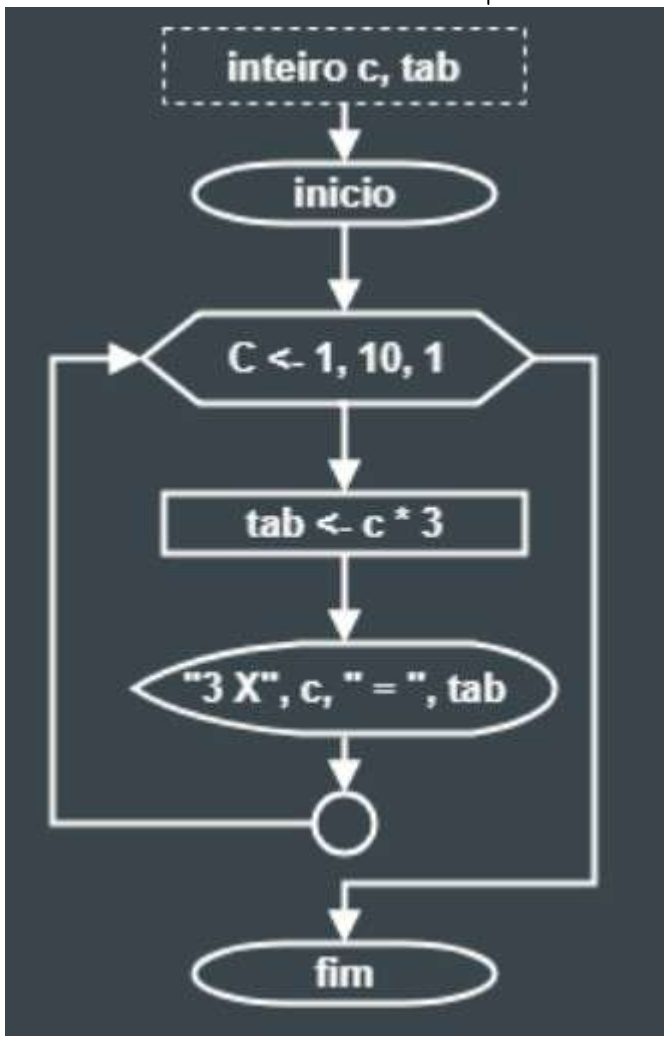
```
1  programa
2  {
3   funcao inicio()
4   {
5       inteiro numero, resultado, contador
6
7       escreva("Informe um número para ver sua tabuada: ")
8       leia(numero)
9
10      limpa()
11
12      para (contador = 1; contador <= 10; contador++)
13      {
14          resultado = numero * contador
15          escreva (numero, " X ", contador, " = ", resultado , "\
n")
```

Parâmetro que atualiza o valor do contador no laço

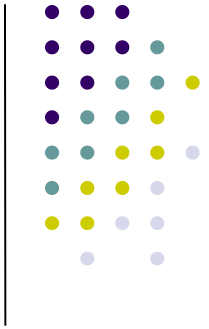
Laço Para

Estrutura de Repetição Para

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro tab
6
7          para (inteiro c=1; c<=10; c++)
8          {
9              tab=c*3
10             escreva ("3 x ", c, " = ", tab, "\n")
11         }
12     }
13 }
```

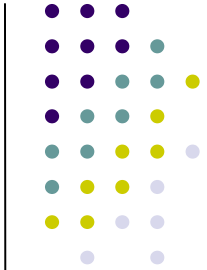


Tipos de Repetição Enquanto



- **Laço Enquanto (Com Variável de Controle)**
  - **Típico para resolver problemas onde sejam necessárias um número determinado de repetições ou quando não sabemos determinar a quantidade de repetições necessárias e dependemos de uma informação dada pelo usuário**
    - **Idade total de 10 estudantes**
    - **Total de 100 produtos vendidos**
    - **Ou, obter dados indefinidamente para se calcular algo. Neste caso, o usuário do sistema terá o controle de parada**

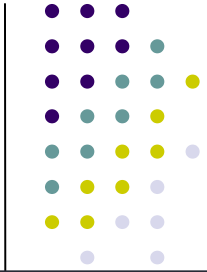
Repetição Enquanto



Condição para entrada

```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

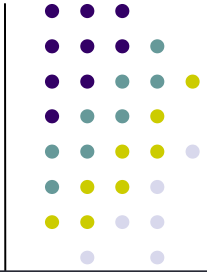
Repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

O comando enquanto testa a condição

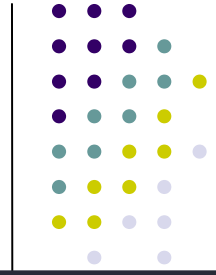
Repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

Delimita o início e fim do bloco do comando enquanto

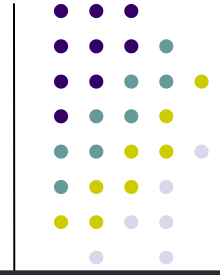
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero       // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19             limpa()
21             escreva("A média dos números é: ", media, "\n")
22     }
```

contador	numero	soma	media

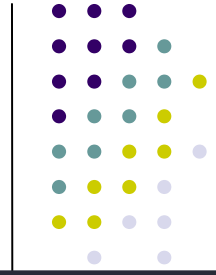
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

contador	numero	soma	media
1		0	

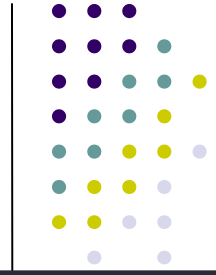
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	0	-

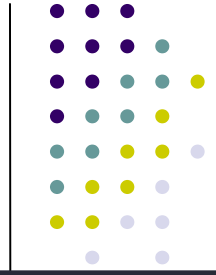
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-

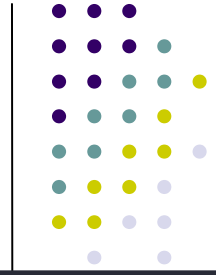
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2			-

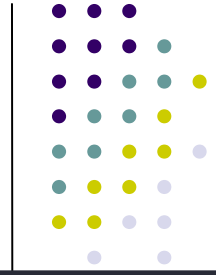
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7		-

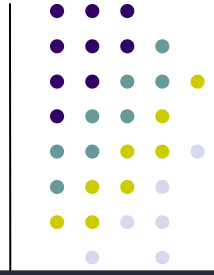
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-

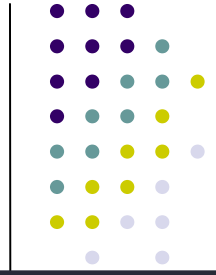
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3			

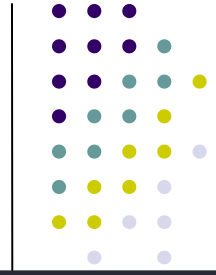
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3		

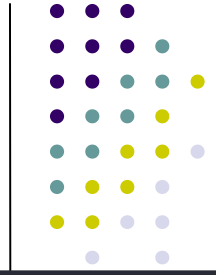
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	

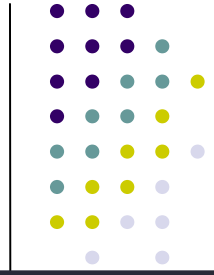
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4			

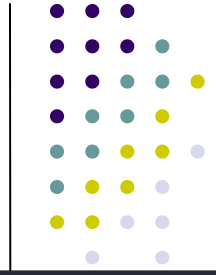
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10		

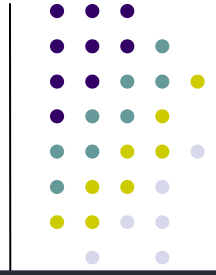
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5			

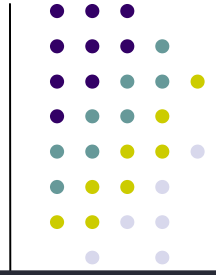
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5              inteiro contador = 1
6              real numero, media, soma = 0.0
7              // Laço que verifica se já foram informados 5 valores
8
9              enquanto(contador <= 5)
10             {
11                     limpa()
12                     escreva("Digite o ", contador, "º número: ")
13                     leia(numero)
14
15                     soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                     contador = contador + 1  // Incrementa o contador
17             }
18             media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5		

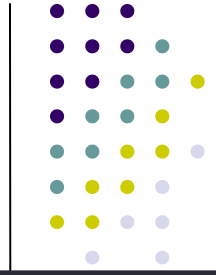
Teste de mesa do algoritmo com repetição Enquanto



```
1  programa
2  {
3      funcao inicio()
4      {
5          inteiro contador = 1
6          real numero, media, soma = 0.0
7          // Laço que verifica se já foram informados 5 valores
8
9          enquanto(contador <= 5)
10         {
11             limpa()
12             escreva("Digite o ", contador, "º número: ")
13             leia(numero)
14
15             soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16             contador = contador + 1  // Incrementa o contador
17         }
18         media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5	30	

Teste de mesa do algoritmo com repetição Enquanto

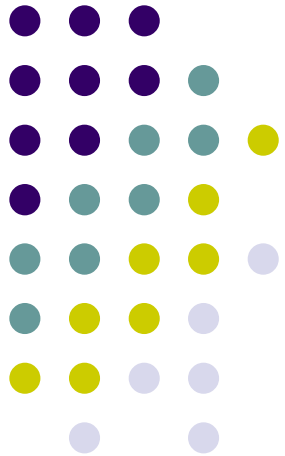


```
1  programa
2  {
3     funcao inicio()
4     {
5           inteiro contador = 1
6           real numero, media, soma = 0.0
7           // Laço que verifica se já foram informados 5 valores
8
9           enquanto(contador <= 5)
10          {
11                limpa()
12                escreva("Digite o ", contador, "º número: ")
13                leia(numero)
14
15                soma = soma + numero     // A variavel soma é o acumulador deste exemplo
16                contador = contador + 1  // Incrementa o contador
17          }
18          media = soma / 5
19
```

contador	numero	soma	media
1	5	5	-
2	7	12	-
3	3	15	
4	10	25	
5	5	30	
6			6

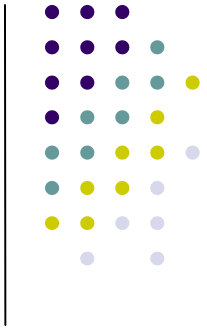
Estruturas de Dados

Vetores e Matrizes



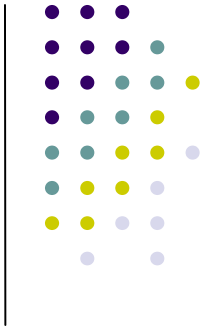
Prof. Alberto Sales

Vetores



- Os vetores são estruturas de dados que permitem o acesso a uma grande quantidade de dados em memória usando-se somente um nome de variável.
  - Esta variável especial é declarada de tal maneira que o programador passa a ter acesso à muitas posições de memória, de maneira controlada

Vetores



- Vetor também é conhecido como variável composta homogênea unidimensional.
  - Isso quer dizer que se trata de um conjunto de variáveis de mesmo tipo, que possuem o mesmo identificador (nome) e são alocadas sequencialmente na memória.
  - Como as variáveis têm o mesmo nome, o que as distingue é um índice que referencia sua localização dentro da estrutura.
  - Exemplo:

```
cadeia nome[] = { "Andre", "Thiago" , "Bruno", "Carlos", "Cassio" }

real altura[] = { 1.71, 1.78, 1.75, 1.87, 1.71 }
```

Vetores

- Exemplo prático
  - vet_nome

0	1	2	3	4
André	Thiago	Bruno	Carlos	Cassio

como declarar no portugol studio

cadeia nome[]

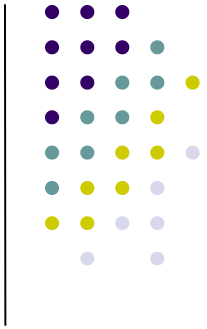
  - vet_altura

0	1	2	3	4
1.71	1.78	1.75	1.87	1.99

como declarar no portugol studio

real altural[]

Vetores



- Exemplo

Faça um programa que receba o nome e a nota de oito alunos e mostre o relatório destes dados.

- Solução:
  - Crie dois vetores: um para os nomes e outro para as notas
  - Para armazenar os respectivos dados crie um laço e solicite as informações do usuário
  - Para apresentar o relatório utilize um laço de repetição e acesse cada posição dos respectivos vetores.

Vetores

```
1   programa{
2     cadeia nome[3]
3     inteiro i
4     real nota[3], soma = 0 , media
5     funcao inicio(){
6       para (i = 0; i < 3; i++) {
7         escreva("Digite o nome do ", i+1,"º aluno: ")
8         leia(nome[i])
9         escreva("Digite a nota de ", nome[i], ": ")
10        leia(nota[i])
11        soma += nota[i]
12        escreva("\n")
13      }
14      escreva("RELATÓRIOS DE NOTAS\n")
15      para (i = 0; i < 3; i++) {
16          escreva(nome[i]," - ",nota[i], "\n")
17      }
18
19      escreva("\nMédia da classe: ", media = soma/3)
20
21    }
22  }
```