



Processamento da Informação BC-05045

Prof Dr. Francisco Isidro



Aula 1 - Introdução

Aula Teórica

Roteiro da Aula



- Introdução
- Algoritmos
- Programas
- Pseudocódigos

Introdução



- Lógica
 - Necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas
 - Permite definir a seqüência lógica para o desenvolvimento de sistemas
 - **Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo**

Introdução



- **Seqüência Lógica são passos executados até atingir um objetivo ou solução de um problema.**
- Convém ressaltar que uma ordem isolada não permite realizar o processo completo, para isso é necessário um conjunto de instruções colocadas em ordem seqüencial lógica.
- Por exemplo, se quisermos fazer uma omelete de batatas, precisaremos colocar em prática uma série de instruções: descascar as batatas, bater os ovos, fritar as batatas, etc...

Introdução



- **Instruções no contexto da computação:**
 - Instrução é a informação que indica a um computador um ação elementar
 - Em ordem isolada não permite realizar o processo completo
 - Para isso, é necessário um conjunto de instruções colocadas em ordem seqüencial lógica.

Algoritmo



- Um algoritmo é formalmente uma seqüência **finita** de passos que levam a execução de uma **tarefa**
- Uma tarefa **não pode ser redundante nem subjetiva na sua definição**, deve ser **clara** e **precisa**.
- Um algoritmo pode não ser a **ÚNICA** solução de um determinado problema pois podem existir **diferentes algoritmos** que resolvem o mesmo problema

Algoritmo - Exemplos



- Algoritmo para Chupar uma bala
 - Pegar a bala
 - Retirar o papel
 - Chupar a bala
 - Jogar o papel no lixo
- E se não jogarmos o papel no lixo? O algoritmo continua correto?

Algoritmo - Exemplos



- **Efetuar uma ligação de um telefone público:**
 1. Retirar o telefone do gancho
 2. Esperar sinal
 3. Colocar o cartão
 4. Discar o número
 5. Falar no telefone
 6. Terminada a ligação, retirar o cartão
 7. Colocar o telefone no gancho

Algoritmo - Exemplos



- **Fritar um ovo:**
 1. Pegar a frigideira, ovo, óleo e sal
 2. Colocar o óleo na frigideira
 3. Acender o fogo
 4. Colocar a frigideira no fogo
 5. Esperar o óleo esquentar
 6. Colocar o ovo
 7. Colocar o sal
 8. Retirar quando estiver pronto
 9. Desligar o fogo

Algoritmo - Exemplos



- Nem todos os algoritmos executam todas as ações ou executam somente uma vez um única ação
- Muitas vezes é necessário:
 - Escolher qual ação deverá se realizada
 - Repetir uma determinada instrução até que um evento ocorra ou uma condição seja atingida

Algoritmo - Exemplos



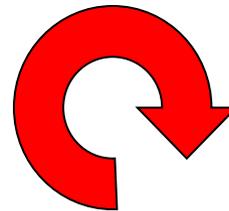
- **Descascar uma batatas**
 1. Pegar faca, bacia e batatas
 2. Colocar água na bacia
 3. Descascar batatas

E para descascar várias batatas?

Algoritmo - Exemplos



- **Descascar várias batatas (utiliza laços de repetição)**
 1. Pegar faca, bacia e batatas
 2. Colocar água na bacia
 3. Enquanto (houver batatas)
 Descascar batatas



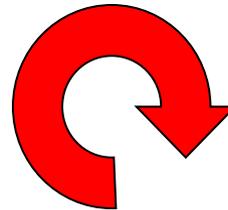
E se a batata estiver podre?

Algoritmo - Exemplos



- **Descascar várias batatas (utiliza condição)**

1. Pegar faca, bacia e batatas
2. Colocar água na bacia
3. Enquanto (houver batatas)
 1. Se batata **Não** estiver podre
Descascar batatas



Algoritmo - Exemplos



- **Trocar Lâmpada (utiliza condição)**
 1. Se (lâmpada estiver fora do alcance)
 - Pegar escada
 2. Pegar lâmpada
 3. Se (lâmpada estiver quente)
 - Pegar um pano
 4. Tirar lâmpada queimada
 5. Colocar lâmpada boa

Algoritmo



- Exemplos:
 - Algoritmos das operações básicas (adição, multiplicação, divisão e subtração) de números reais decimais;
 - Manuais de aparelhos eletrônicos, como um vídeo-cassete, que explicam passo-a-passo como por exemplo, gravar um evento.
- Desenvolvimento de Algoritmos
 - Tarefa difícil
 - **Exige muito exercício**
 - Contribui para o desenvolvimento de uma lógica própria de programação

Regras para construção de um Algoritmo



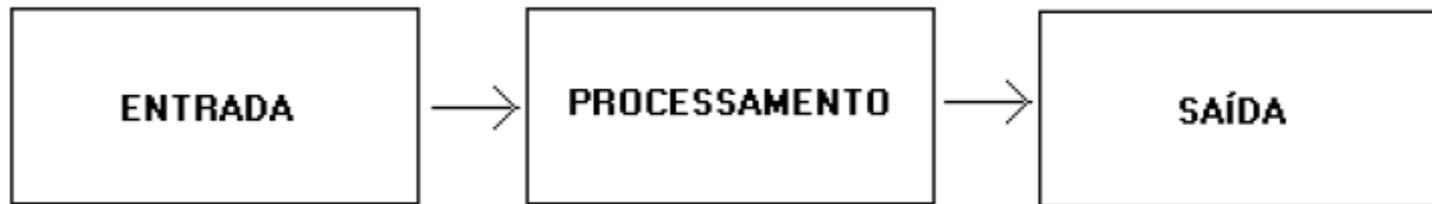
- Para escrever um algoritmo precisamos descrever a seqüência de instruções, de maneira simples e objetiva.
- Para isso utilizaremos algumas técnicas:
 - **Usar somente um verbo por frase**
 - **Imaginar que você está desenvolvendo um algoritmo para pessoas que não trabalham com informática**
 - **Usar frases curtas e simples**
 - **Ser objetivo**
 - **Procurar usar palavras que não tenham sentido dúbio**

Fases de um Algoritmo



- Basicamente um algoritmo é dividido em **três fases**:
 - ***Fase de entrada***: corresponde aos possíveis dados de entrada de um algoritmo
 - ***Fase de processamento***: fase em que condições ou relações devem ser satisfeitas para transformar dados de entrada em uma saída aceitável
 - ***Fase de Saída***: exhibe os resultados provenientes das fases anteriores

Fases de um Algoritmo



Fases de um Algoritmo



- **Exemplo:** Algoritmo para calcular a média de duas notas
 - **Entrada:** informar as duas notas: nota1 e nota2
 - **Processamento:** realizar o cálculo da média, sendo que a média é igual a soma de nota1 e nota2 dividido por 2
 - **Saída:** apresentar o resultado do cálculo realizado na fase anterior, ou seja, a média obtida das duas notas

Fases de um Algoritmo



Imagine o seguinte problema: Calcular a média final dos alunos da 3ª Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4.

Onde:

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

Para montar o algoritmo proposto, faremos três perguntas:

a) Quais são os dados de entrada?

R: Os dados de entrada são P1, P2, P3 e P4

b) Qual será o processamento a ser utilizado?

R: O procedimento será somar todos os dados de entrada e dividi-los por 4 (quatro)

$$\frac{\mathbf{P1 + P2 + P3 + P4}}{4}$$

c) Quais serão os dados de saída?

Fases de um Algoritmo



Imagine o seguinte problema: Calcular a média final dos alunos da 3^a Série. Os alunos realizarão quatro provas: P1, P2, P3 e P4.

Fases de um Algoritmo



R: O dado de saída será a média final

Algoritmo

Receba a nota da prova1
Receba a nota de prova2
Receba a nota de prova3
Receba a nota da prova4
Some todas as notas e divida o resultado por 4
Mostre o resultado da divisão

Pseudocódigo



- Os algoritmos podem ser descritos em uma linguagem chamada **pseudocódigo**.
- Este nome é uma alusão à posterior implementação em uma linguagem de programação
- Ao contrário de uma linguagem de programação não existe um formalismo rígido de como deve ser escrito o algoritmo.
- O algoritmo deve ser fácil de se interpretar e fácil de codificar. Ou seja, **ele deve ser o intermediário entre a linguagem falada e a linguagem de programação.**

Exemplo de Pseudocódigo



programa Teste1

inicio

leia(p1)

→ Entrada

leia(p2)

leia(p3)

media $\leftarrow (p1+p2+p3)/3$ →

Processamento

escreva(media)

↘
saída

fim

Teste de Mesa



- Após desenvolvermos um algoritmo precisamos testá-lo para verificar se está correto
- Funcionamento
 - Seguir as instruções descrita no algoritmo de maneira precisa
 - Verificar se resultado obtido casa o com o esperado

Exemplo



- Considere o seguinte boletim do aluno
 - Utilize os dados deste boletim e faça o teste de mesa do pseudocódigo de calculo das médias

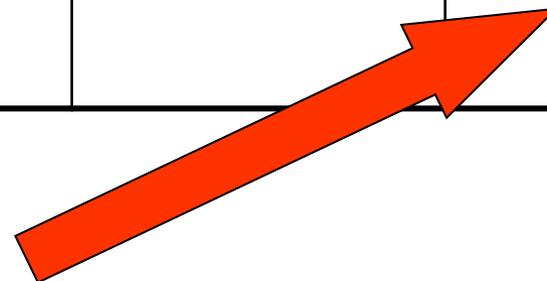
Disciplinas:

Matemática	9,0
Geografia	9,5
História	9,5
Educação Artística	10,0
<i>Média Final</i>	<i>9,5</i>

Teste de Mesa



Prova1	Prova2	Prova3	Prova4	Média
9.0	9.5	9.5	10.0	9.5



```
programa Teste1
inicio
leia(p1)
leia(p2)
leia(p3)
leia(p4)
media <-
(p1+p2+p3+p4)/ 4
escreva(media)
fim
```

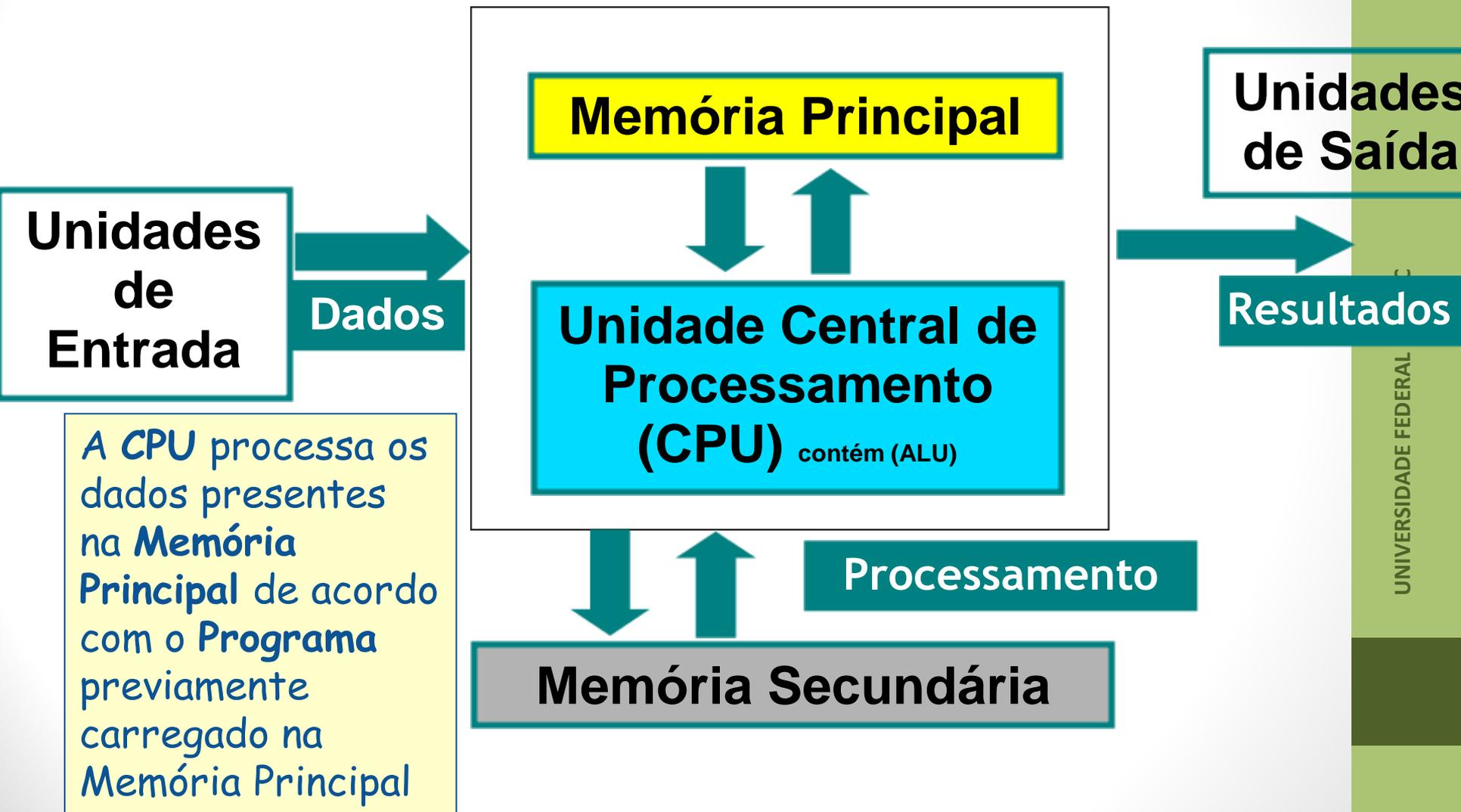
$$\text{Média} = \frac{\text{prova1} + \text{prova2} + \text{prova3} + \text{prova4}}{4}$$

Programas

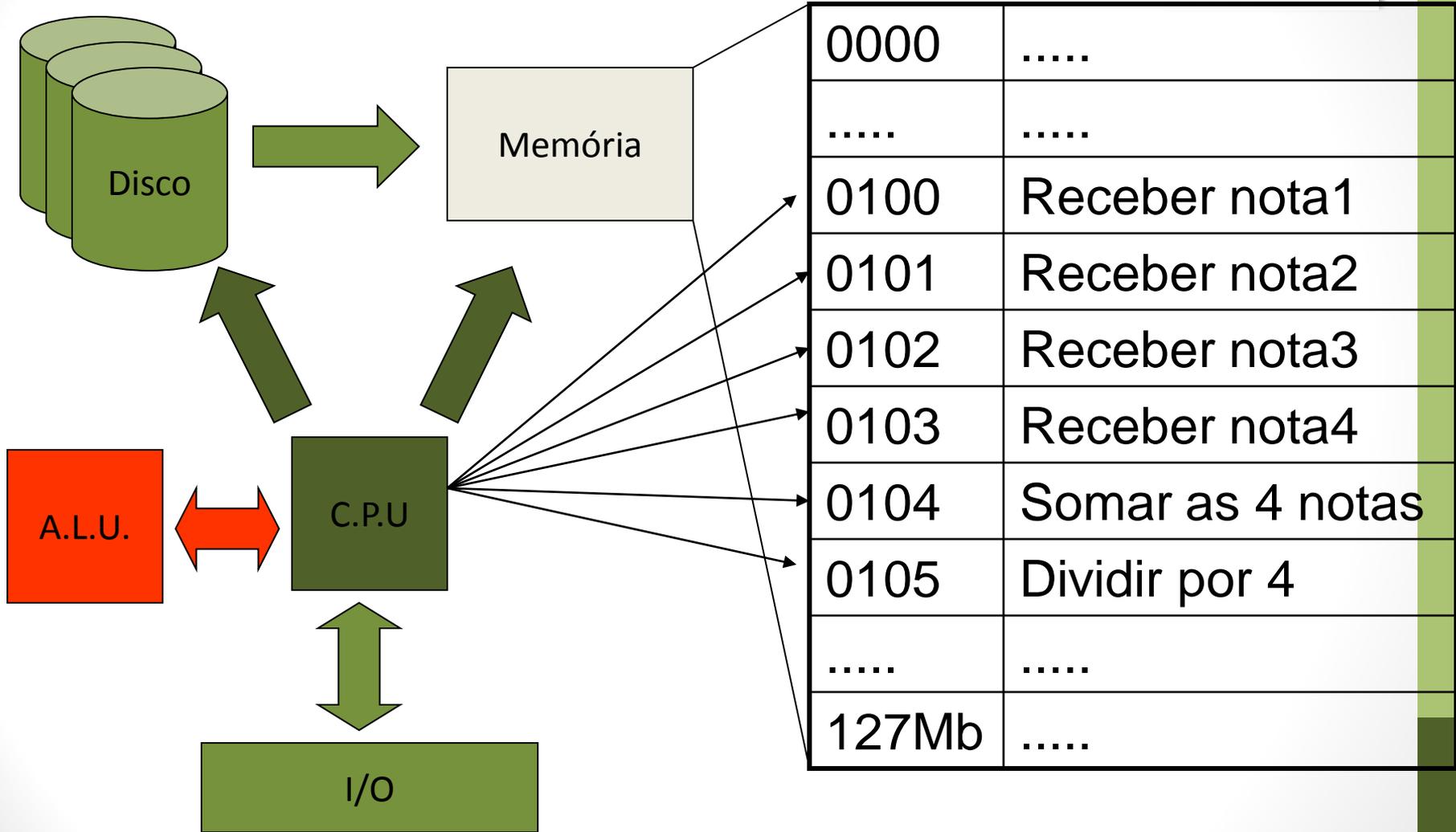


- Os programas de computadores nada mais são do que algoritmos escritos numa linguagem de computador (Pascal, C, Cobol, Fortran, Visual Basic entre outras) e que são interpretados e executados por uma máquina, no caso um computador.
- Notem que dada esta interpretação rigorosa, um programa é por natureza muito específico e rígido em relação aos algoritmos da vida real.

John Von Neumann (1946)



Execução de um Programa



Lógica



- Juntar opções para formar uma idéia maior
- Para Fritar um Ovo:
 - tem ovo? **E**
 - tem óleo? **E**
 - tem frigideira? **E**
 - tem fogão
- Se algum desses elementos faltar, não consegue-se obter o ovo frito
- Característica do conectivo **E (AND)**
 - Todos TÊM que acontecer

Lógica



- Para Trocar a Lâmpada
 - Tem lâmpada igual **OU**
 - Tem lâmpada semelhante
- Qualquer uma das afirmações que acontecerem, existe uma forma de substituir a lâmpada queimada
- Característica do conectivo **OU (OR)**
 - Se apenas um acontecer, já está bom!
- Para mapear, utiliza-se o conceito dos dois estados, zeros e uns, na forma textual, Falso e Verdadeiro

Lógica



- Para o conectivo **E**
 - Todos devem ser Verdadeiros para a afirmação toda ser verdadeira!
- Para o conectivo **OU**
 - Qualquer uma sendo Verdadeiro, torna a afirmação verdadeira!
 - Caso contrário, a afirmação é Falsa

		E	OU
Verdadeiro	Verdadeiro	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Falso	Verdadeiro
Falso	Verdadeiro	Falso	Verdadeiro
Falso	Falso	Falso	Falso

Lógica



- Conectivo **NÃO (NOT)**
 - Negação – inverte o estado, de verdadeiro para falso e vice-versa
 - Exemplo:

NÃO

‘Pedro mora na Vila Rica’

‘Pedro não mora na Vila Rica’

Não é verdade que não é verdade que não é verdade
que João é casado?

Verdade ou Mentira?

Lógica - Exemplo



- Fritar Ovo

- Tem ovo? → Verdadeiro
- Tem óleo? → Verdadeiro
- Tem frigideira? → Verdadeiro
- Tem fogo? → Verdadeiro

Verdadeiro

- Consigo preparar meu ovo, pois todas as respostas foram Verdadeiro
- Se faltar algum, ou seja, possuir uma resposta Falso, não consigo preparar meu ovo frito

Lógica - Exemplo



- Trocar a Lâmpada
 - Tem lâmpada igual **Verdadeiro**
 - Tem lâmpada semelhante **Verdadeiro**
 - Existe troca: possui os dois tipos de lâmpadas
- Outro caso
 - Tem lâmpada igual **Falso**
 - Tem lâmpada semelhante **Verdadeiro**
 - Existe troca: coloca a lâmpada semelhante

Fluxograma



- Fluxograma: utiliza-se de figuras geométricas para ilustrar os passos a serem seguidos para a resolução dos problemas. Diagrama de Blocos. É bastante utilizado;

Fluxograma



- Cada instrução ou ação a ser executada é representada por um símbolo gráfico.



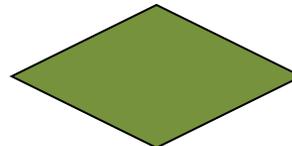
Terminal: representa o início e o final do fluxograma.



VÍdeo: representa a saída de informações por meio do monitor de vídeo.



Processamento: representa a execução de operações ou ações.



Decisão: representa uma ação lógica que resultará na escolha de uma das seqüências de instruções.



Teclado: representa a entrada de dados para as variáveis por teclado.

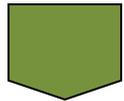


Preparação: representa uma ação de preparação para o processamento.

Fluxograma



Conector: utilizado para interligar partes do fluxograma ou para desviar o fluxo corrente para um determinado trecho do fluxograma.



Conector de Páginas: utilizado para interligar partes do fluxograma em páginas distintas.

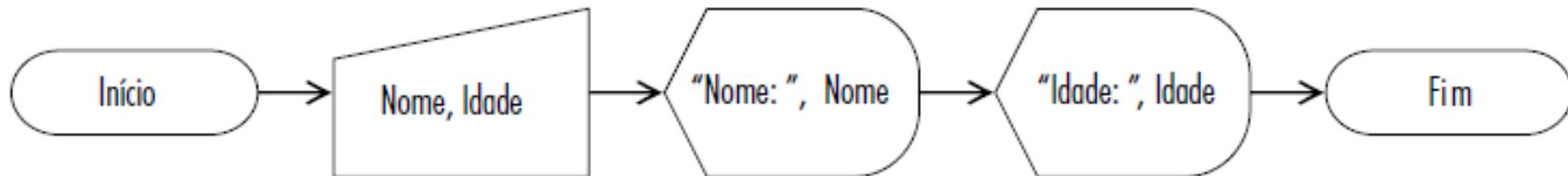


Seta de orientação do fluxo.

Fluxograma

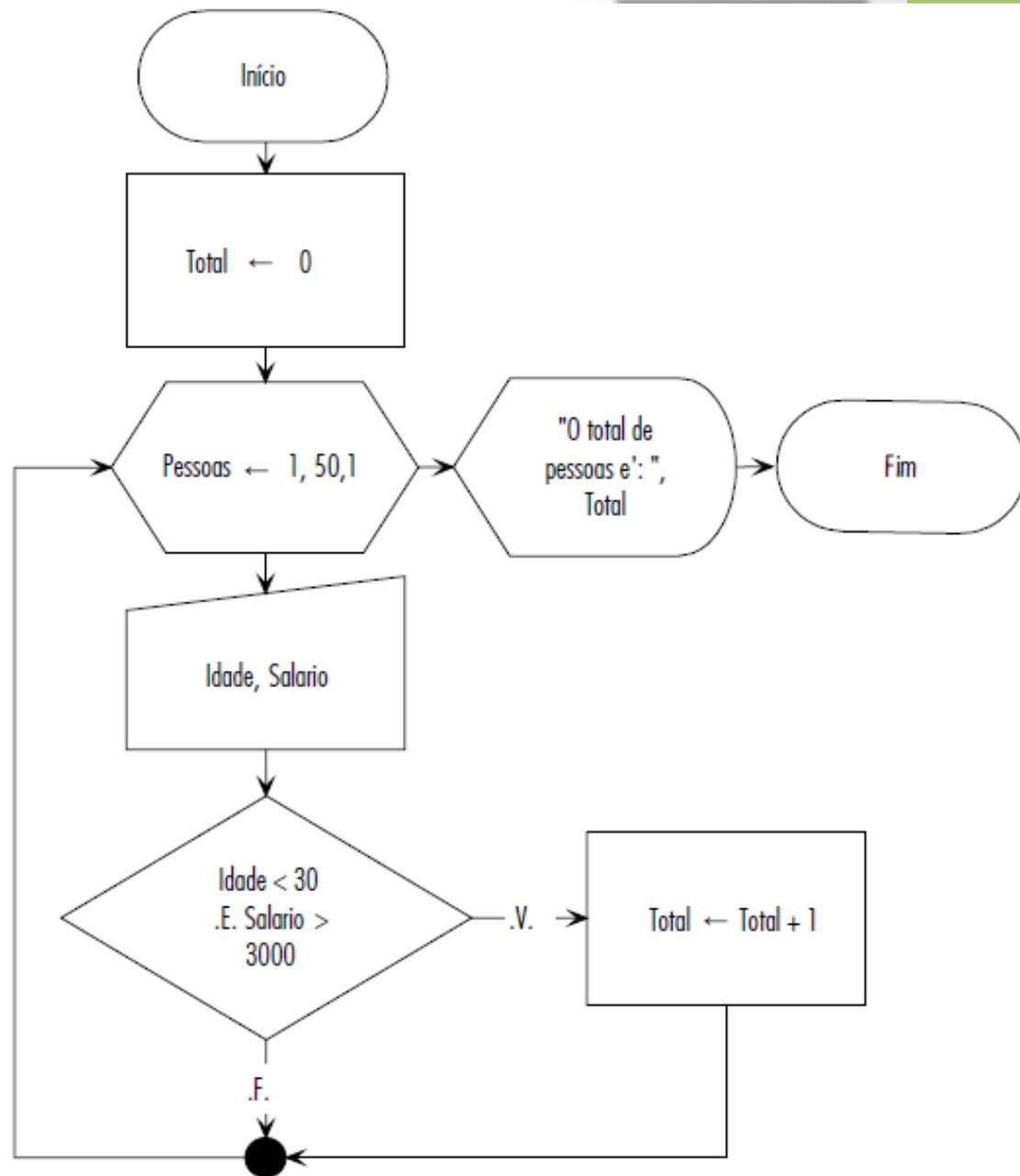


- Ler o nome e a idade de uma pessoa e mostrar na tela.



Fluxograma

- Fluxograma para ler a idade e o salário de 50 pessoas e verificar quantas têm idade inferior a 30 anos e salário superior a R\$ 3.000,00.



Algoritmos – Exercícios



1. Fazer a verificação se uma pessoa está apta a votar ou não, baseado em sua idade
2. Verificar se uma determinada pessoa já pode aposentar
3. Informar qual o dia do rodízio de carros baseado no final da placa do veículo
4. Verificar qual é o maior de 3 números distintos

Exercício 1 - resolução



- Fazer a verificação se uma pessoa está apta a votar ou não, baseado em sua idade (1/2)

INICIO

1. Receber a idade
2. Analisar a idade
3. Mostrar resultado da análise

FIM

Exercícios



- Fazer a verificação se uma pessoa está apta a votar ou não, baseado em sua idade (1/2)

INICIO

1. Receber a idade
2. Analisar a idade
3. Mostrar resultado da análise

FIM

2.1. Se (idade for maior ou igual a 16) então
resultado: está apto a votar
senão
resultado: não está apto a votar

Exercícios



- Fazer a verificação se uma pessoa está apta a votar ou não, baseado em sua idade (2/2)

INICIO

1. Receber idade
2. Se idade ≥ 16 então
 resultado = está apto a votar
senão
 resultado = não está apto a votar
3. Mostrar resultado

FIM

Referência



- Lógica de Programação (Notas de Aula)
Professor Paulo Sérgio de Moraes
Unicamp – Centro de Computação
- Lógica de Programação
André L. V. Forbellone e Henri F. Eberspächer
- PUGA, Sandra; RISSETTI, Gerson. Logica De Programação e Estruturas De Dados Com Aplicações Em Java, Prentice Hall Brasil, 2a. Ed. 2008

Instruções para a atividade Prática



```
1 programa soma
2
3 declaracoes
4 inteiro num1, num2, result
5
6 inicio
7
8 escreva ("Digite o primeiro valor: ")
9 leia (num1)
10 escreva ("\nDigite o segundo valor: ")
11 leia (num2)
12
13 result <- num1+num2
14
15 escreva("\n0 Resultado da soma é= ", result)
16
17 fim
```

Observe que a
Entrada de dados precisa
Ser declarada

Ao declarar, é necessário
dizer qual o tipo da
entrada

Esse é o conceito de
variável vista na disciplina
de Bases Computacionais

Na próxima aula vamos
recordar bem os conceitos
de variável.