



Introdução à Programação

Conteúdo

- Algoritmos
 - Representação
 - Exercícios
- Linguagens de Programação
- Compilador
- Interpretador

Lógica

- O que é lógica?
 - Ciência que estuda as leis do raciocínio.
 - Correção/validação do pensamento.
 - Encadeamento/ordem de idéias.
 - Arte de pensar bem.

Raciocínio Lógico

- Precisamos mais do que fórmulas, precisamos aprender a pensar!
- É preciso aprender a pensar sobre o problema, extraindo o máximo de informações sobre ele.
- Lógica ensina a colocar ordem no pensamento.

Lógica de Programação

- Permite o **aperfeiçoamento** de nossa forma de pensar e raciocinar sobre um problema computacional, a fim de obter uma solução eficaz e/ou eficiente
- A lógica de programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a **sequência lógica** para o desenvolvimento.

Sequência Lógica

- São passos executados até atingir um objetivo ou solução de um problema.
- Exemplo:
 - “Chupar uma bala”
 - Pegar a bala;
 - Retirar o papel;
 - Chupar a bala;
 - Jogar o papel no lixo;

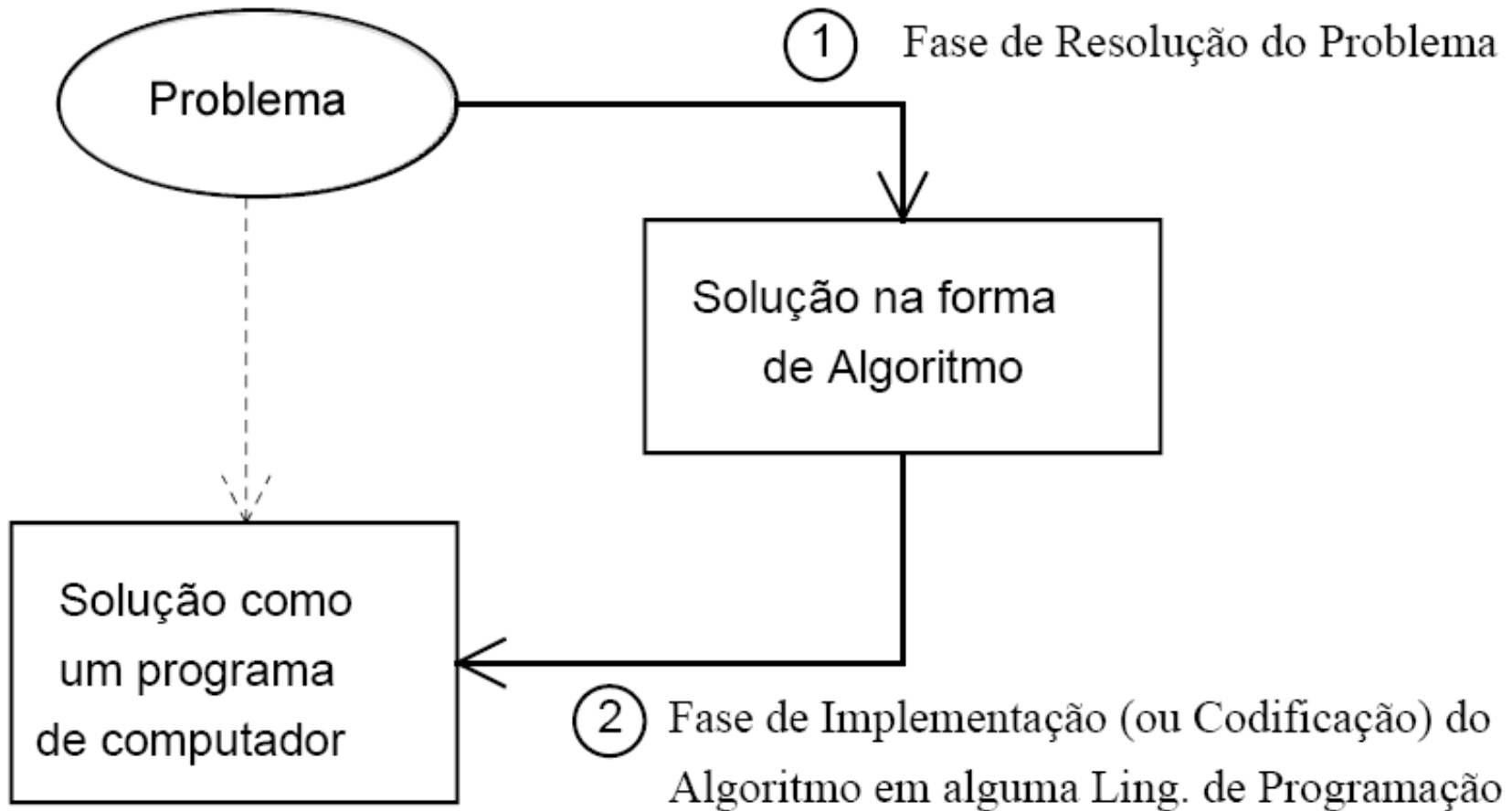
Algoritmo

- É uma sequência finita de passos, descritos em ordem lógica, que visam a atingir um objetivo bem definido
- Ao definir uma sequência de passos é necessário pensar ordenadamente, utilizar lógica.

Por que usar algoritmos?

- **Abstração** – Todo esforço é utilizado na resolução do problema, e não em detalhes computacionais.
- **Portabilidade** – Uma solução algorítmica pode ser traduzida para qualquer linguagem de programação.

Fases da Programação



Regras para criação do algoritmo

- *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.*

Regras para criação do algoritmo

■ Exemplo

- 1. Encher a chaleira com água
- 2. Colocar a chaleira para ferver
- 3. Colocar duas colheres de sopa de pó de café no coador
- 4. Aguarde a água ferver
- 5. Acrescente a água ao pó aos poucos
- 6. Aguarde coar
- 7. Adoce à gosto

Algoritmo

- Formas de representação
 - **Narrativa**: uso de português
 - **Fluxograma**: símbolos gráficos para representar fases e componentes dos algoritmos
 - **Pseudocódigo**: Definição de uma pseudo-linguagem de programação, cujos comandos são em português

Algoritmo

- Formas de representação

- **Narrativa**

- Fluxograma

- Pseudocódigo

Algoritmo

Descrição Narrativa

- Receita de bolo:
 1. Misture os ingredientes
 2. Bata os ingredientes em uma vasilha
 3. Unte a forma com manteiga
 4. Despeje a mistura na forma
 5. Se houver coco ralado
 - então despeje-o sobre a mistura
 6. Leve a forma ao forno
 7. Enquanto não corar
 - deixe a forma no forno
 8. Retire do forno
 9. Deixe esfriar

Algoritmo

Descrição Narrativa

- Troca de pneus:
 1. Afrouxar ligeiramente as porcas
 2. Suspender o carro
 3. Retirar as porcas e o pneu
 4. Colocar o pneu reserva
 5. Apertar as porcas
 6. Abaixar o carro
 7. Dar o aperto final nas porcas

Algoritmo

Descrição Narrativa

- Obter o status de um aluno:
 1. Obter as suas 2 notas de provas
 2. Calcular a média aritmética
 3. Se a média for maior que 7
 - o aluno foi aprovado
 - senão ele foi reprovado

Trocar uma Lâmpada

- Sequenciação
 1. pegar uma escada
 2. posicionar a escada embaixo da lâmpada
 3. buscar uma lâmpada nova
 4. subir na escada
 5. retirar lâmpada velha
 6. colocar lâmpada nova

Trocar uma Lâmpada

- SE estiver queimada = Decisão
 1. pegar uma escada
 2. posicionar a escada embaixo da lâmpada
 3. buscar uma lâmpada nova
 4. acionar o interruptor
 5. **se** a lâmpada não acender, **então**
 - 5.1 subir na escada
 - 5.2 retirar lâmpada queimada
 - 5.3 colocar lâmpada nova

Trocar uma Lâmpada

- SE estiver queimada $v_2 =$ (decisão)
- evitar pegar escada e lâmpada
 1. acionar o interruptor;
 2. se a lâmpada não acender, então
 - 2.1 pegar uma escada
 - 2.2 posicionar a escada embaixo da lâmpada
 - 2.3 buscar uma lâmpada nova
 - 2.4 subir na escada
 - 2.5 retirar lâmpada queimada
 - 2.6 colocar lâmpada nova

Trocar uma Lâmpada...

- SE estiver queimada v3 = (decisão)
 - Re-teste após a troca
 1. acionar o interruptor
 2. se a lâmpada não acender, então
 - 2.1 pegar uma escada
 - 2.2 posicionar a escada embaixo da lâmpada
 - 2.3 buscar uma lâmpada nova
 - 2.4 subir na escada
 - 2.5 retirar lâmpada queimada
 - 2.6 colocar lâmpada nova
 - 2.7 acionar o interruptor
 - 2.8 se a lâmpada não acender, então
 - retirar lâmpada queimada
 - colocar lâmpada nova
-



Repetir
várias
vezes!!!

Trocar uma Lâmpada...

- SE estiver queimada v4 = **Repetição!!**
 1. acionar o interruptor
 2. se a lâmpada não acender, então
 - 2.1 pegar uma escada
 - 2.2 posicionar a escada embaixo da lâmpada
 - 2.3 buscar uma lâmpada nova
 - 2.4 subir na escada
 - 2.5 retirar lâmpada queimada
 - 2.6 colocar lâmpada nova
 - 2.7 acionar o interruptor
 - 2.8 enquanto a lâmpada não acender, faça:**
 - retirar lâmpada queimada
 - colocar lâmpada nova
 - acionar o interruptor

Trocar 10 Lâmpadas...

- SE estiverem queimadas (v1)
 1. acionar o interruptor **do primeiro soquete**
 2. se a lâmpada não acender, então
 - 2.1 pegar uma escada
 - 2.2 posicionar a escada embaixo da lâmpada
 - 2.3 buscar uma lâmpada nova
 - 2.4 subir na escada
 - 2.5 retirar lâmpada queimada
 - 2.6 colocar lâmpada nova
 - 2.7 acionar o interruptor
 - 2.8 enquanto a lâmpada não acender, faça:
 - retirar lâmpada queimada
 - colocar lâmpada nova
 - acionar o interruptor
 3. acionar o interruptor **do segundo soquete**
 4.

Trocar 10 Lâmpadas...

- SE estiverem queimadas (v2)
 1. acionar o interruptor **do primeiro soquete**
 2. Enquanto número de soquetes for menor ou igual a 10 faça
 - 2.1 se a lâmpada não acender, então
 - 2.1.1 pegar uma escada
 - 2.1.2 posicionar a escada embaixo da lâmpada
 - 2.1.3 buscar uma lâmpada nova
 - 2.1.4 subir na escada
 - 2.1.5 retirar lâmpada queimada
 - 2.1.6 colocar lâmpada nova
 - 2.1.7 acionar o interruptor
 - 2.1.8 enquanto a lâmpada não acender, faça:
 - » retirar lâmpada queimada
 - » colocar lâmpada nova
 - » acionar o interruptor
 - 2.2 acionar o interruptor **do segundo soquete**

Exercícios

Descrição Narrativa

- Descreva os seguintes algoritmos
 1. Passos para chegar a UFRPE e assistir aula de introdução a programação.
 2. Passos para comprar um DVD pela internet

Exercícios

Descrição Narrativa

3. Tendo como exemplo os algoritmos desenvolvidos para solucionar o problema da troca de lâmpadas, elabore algoritmos com os passos necessários para trocar um pneu furado em cada as seguintes situações:
 1. Trocar o pneu traseiro esquerdo
 2. Trocar o pneu traseiro esquerdo e, antes, verificar se o pneu reserva está em condições de uso
 3. Verificar se existe algum pneu furado; se houver, verificar o pneu reserva e, então, trocar o pneu correto.

Para cada algoritmo faça um refinamento do anterior, introduzindo novas ações e alterando o fluxo de execução de forma compatível com as situações apresentadas

Descrição Narrativa

- Vantagens
 - Linguagem Natural
- Desvantagens
 - Ambiguidade
 - Imprecisão
 - Pouca Confiabilidade
 - Extensão

Algoritmo

- Formas de representação

- Narrativa

- **Fluxograma**

- Pseudocódigo

Algoritmo

Fluxograma

- Fluxograma (Diagrama de Blocos) :
 - **Representação gráfica** de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos
 - Fácil visualização

Principais Símbolos Utilizados



Início e final do diagrama



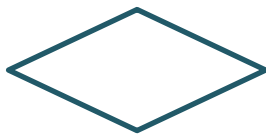
Operação de entrada de dados



Operação de saída de dados



Cálculo, operações de atribuição.

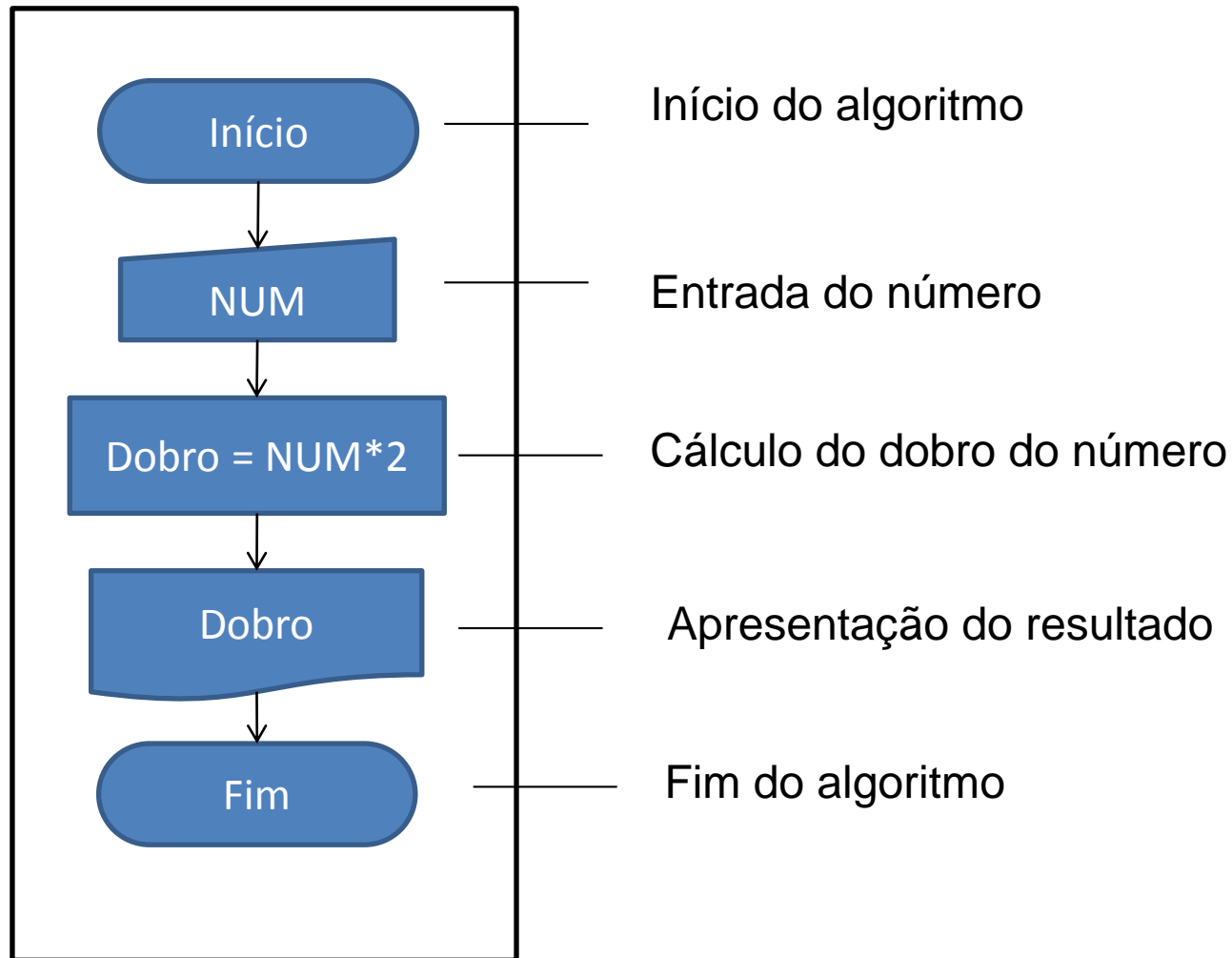


Decisão



Seta do fluxo de dados

Diagrama de Blocos- Exemplo



Fluxograma

Divisão de Dois Números

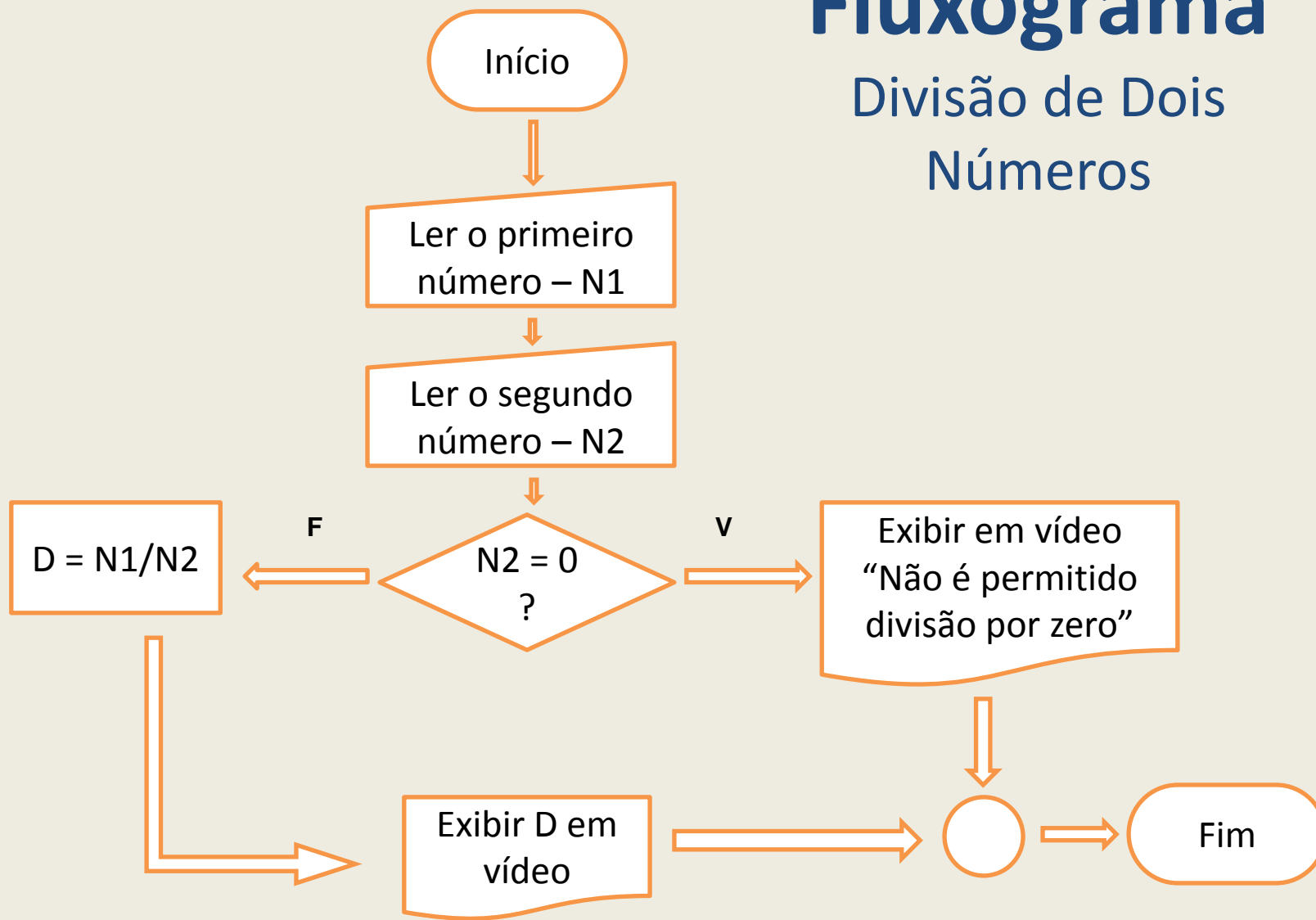


Diagrama de Blocos - Atividade

- Criar diagrama de bloco de um algoritmo que:
 - Recebe duas notas de um aluno.
 - Calcula Média
 - Diz se o aluno foi aprovado (média ≥ 7) ou reprovado(média < 7).

Exercícios

Fluxograma

- Construa fluxogramas para os seguintes algoritmos
 - Calcular a área de um retângulo (**AREA = BASE x ALTURA**)
 - Verificar se o individuo é maior de idade ou não (**maior de 18 anos**)

Fluxograma

Troca de 10 lâmpadas

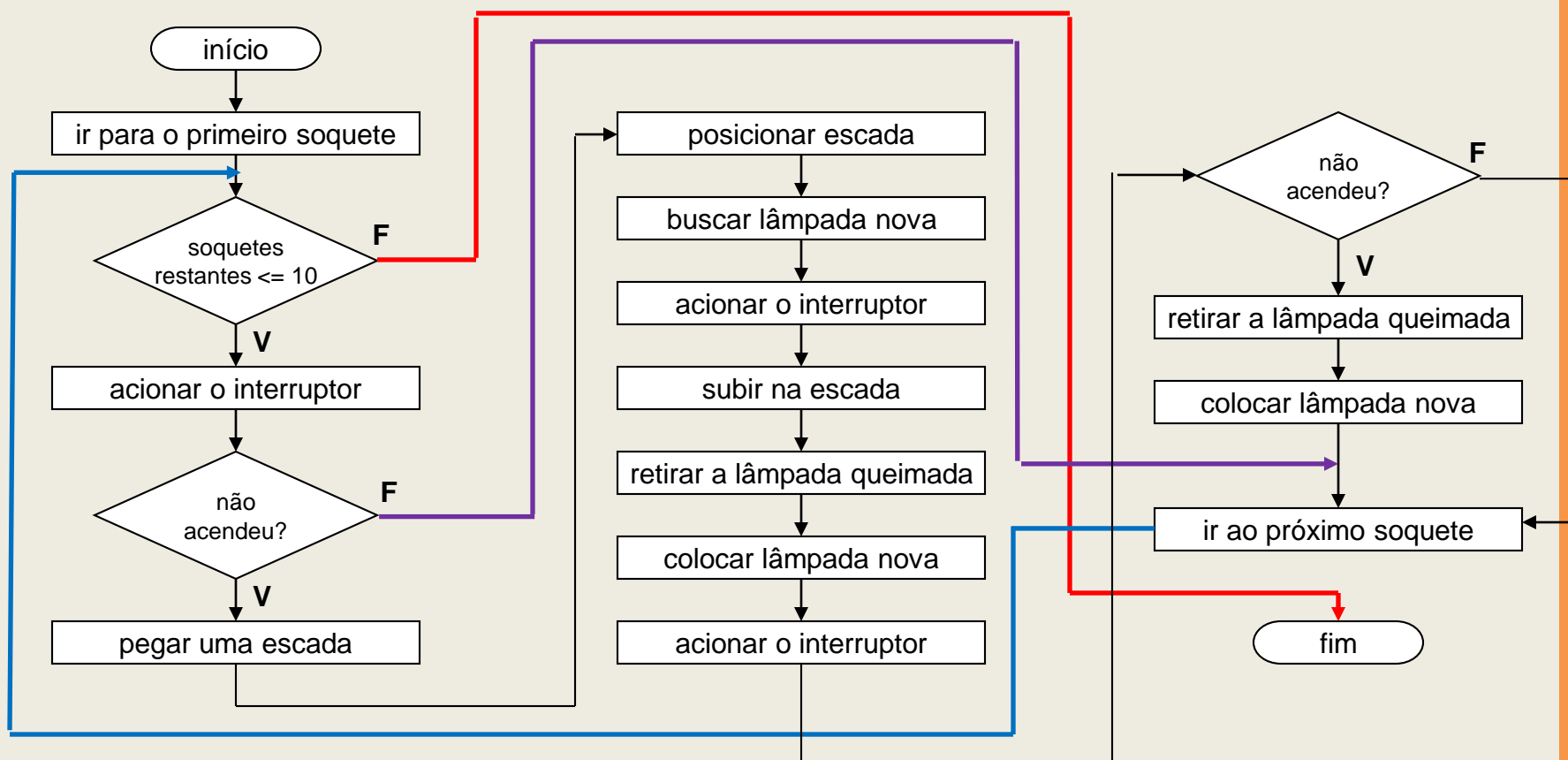


Diagrama de Blocos - Atividade

- Criar diagrama de bloco de um algoritmo que:
 - Recebe duas notas de um aluno.
 - Calcula Média
 - Diz se o aluno foi aprovado (média ≥ 7) ou reprovado (média < 7).

Algoritmo

- Formas de representação

- Narrativa

- Fluxograma

- **Pseudocódigo**

Pseudocódigo

- pseudolinguagem de programação
 - comandos em português
- Representação suficientemente geral para permitir uma tradução simples de um algoritmo nela representado para uma linguagem de programação específica.

Pseudocódigo - Estrutura

- Algoritmo <Nome do Algoritmo>
 <declaração_de_variáveis>

Início

 <corpo_do_algoritmo>

Fim

Pseudocódigo - Exemplo

- Algoritmo **Calcula_Dobro**
Var Num, Dobro: Inteiro;
Inicio
 Leia Num;
 Dobro <- Num*2;
 Escreva Dobro;
Fim

Algoritmo

Pseudo-código

■ Principais Comandos

- Atribuição (= ou := ou ←) : Utilizado para guardar um valor em uma variável
 - `total = 0`
 - `SOMA ← SOMA + 1`
 - `mensagem := "Erro de Digitação !"`
- Entrada de Dados : Utilizado para ler dados do usuário, de dispositivos externos
 - `leia(Idade)`
 - `leia('d:\arquivo.txt')`
- Saída de dados : Utilizado para exibir dados
 - `escreva(Idade)`
 - `escreval('d:\arquivo.txt')`

Exercício

- Criar um algoritmo em pseudo-código para:
 - Calcular a área de um retângulo (**AREA = BASE x ALTURA**)

Pseudocódigo

■ Vantagens

- Usa o português como base.
- Pode-se definir quais e como os dados vão estar estruturados.
- Passagem quase imediata de um algoritmo para uma linguagem qualquer.

■ Desvantagens

- Exige a definição de uma linguagem não natural para trabalho.
- Não é padronizado.

Aplicações - Exemplo

Java

Pseudocódigo

```
Var: num1, num2, soma: int;  
Inicio  
  num1 = 2;  
  num2 = 3;  
  soma = num1 + num2;  
  Imprime Soma;  
Fim
```



```
public class Algoritmo{  
    public static void main(){  
        int num1, num2, soma;  
  
        num1=2;  
        num2=3;  
        soma = num1 + num2;  
        system.out.println (soma);  
    }  
}
```

Aplicações - Exemplo

Pseudocódigo

```
Var: num1, num2, soma: int;  
Inicio  
  num1 = 2;  
  num2 = 3;  
  soma = num1 + num2;  
  Imprime Soma;  
Fim
```

C

```
int main(){  
  int num 1, num2, soma;  
  
  num1 =2;  
  num2 = 3;  
  soma = num1+num2;  
  printf("Soma = %d", Soma);  
  return 0;  
}
```

Aplicações - Exemplo

Pseudocódigo

```
Var: num1, num2, soma: int;  
Inicio  
  num1 = 2;  
  num2 = 3;  
  soma = num1 + num2;  
  Imprime Soma;  
Fim
```



Python

```
num1 = 2  
num2 = 3  
soma = num1+num2  
print soma
```

Características de um Algoritmo

- **Finitude** – Um algoritmo tem que terminar com um número finito de passos.
- **Definitude** – Cada passo do algoritmo deve ser definido com precisão.
- **Entrada** – Um algoritmo pode ter zero ou mais entradas.
- **Saída** – Um algoritmo tem uma ou mais saídas.

Características de um Algoritmo

- **Eficácia** – Todas as operações feitas por um algoritmos devem ser básicas.
- **Não ser ambíguo**
- **Ser efetivo** – Todas as etapas devem ser alcançáveis em um tempo finito.

Linguagens de Programação

- Uma linguagem de programação é um **vocabulário** e um **conjunto de regras** usadas para escrever programas de computador
- Divididas em três tipos, com relação à similaridade com a linguagem humana:
 - **Linguagem de máquina**
 - **Linguagem Simbólica**
 - **Linguagem de Alto Nível**

Linguagem de Máquina

- É linguagem de mais baixo nível de entendimento pelo ser humano e a **única** entendida pelo processador (UCP)
- Constituída inteiramente de números **(0's e 1's)**
- Uma instrução típica em linguagem de máquina seria algo como:
 - **0100 1111 1010**

Linguagens de Programação

Linguagem Simbólica

- Assembly:
 - linguagem de nível **imediatamente acima** da linguagem de máquina.
 - Possui a **mesma estrutura** e conjunto de instruções que a linguagem de máquina, porém permite que o programador utilize **nomes** (mnemônicos) e símbolos em lugar de **números**
- A conversão da linguagem simbólica para a linguagem de máquina se chama **montagem**, e é feita por um programa chamado **montador** (assembler).

Linguagens de Programação

Linguagem Simbólica

- Exemplo de instrução:

ADD A, B

MOV AX, 6

- Classificada como linguagem de **segunda** geração
- Assim como a linguagem de máquina, é considerada uma linguagem de **baixo nível**

Linguagens de Programação

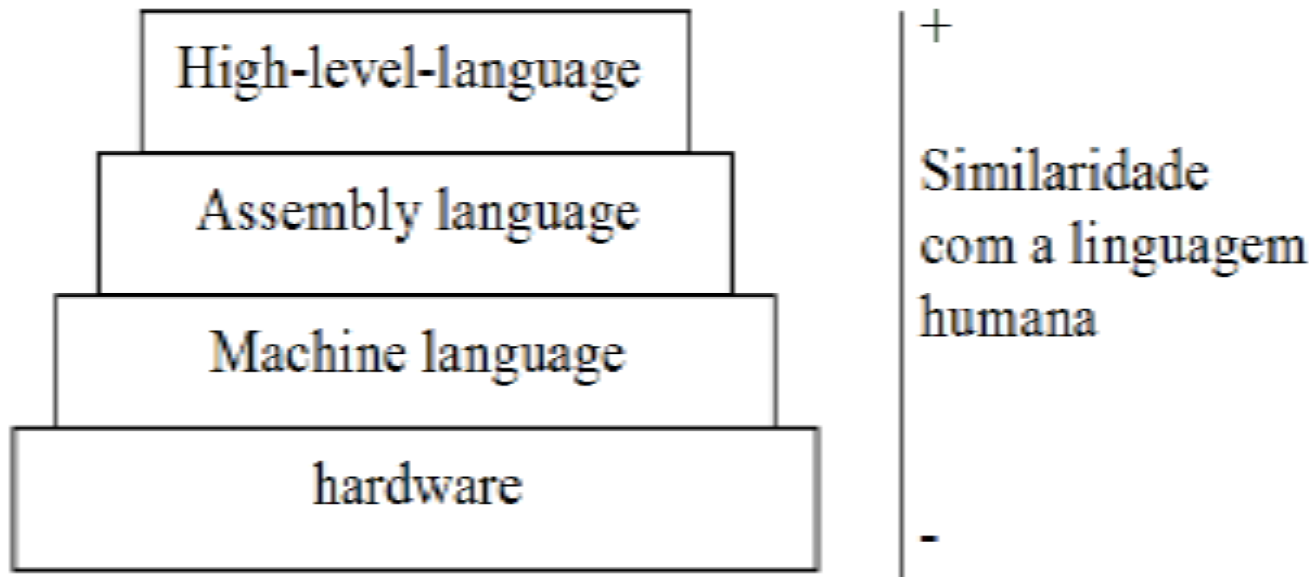
Linguagem de Alto Nível

- linguagens de programação que possuem uma estrutura e palavras-chave que são **mais próximas** da linguagem humana
 - **C, C++, Java, Python, etc ...**
- Programas escritos nessas linguagens são convertidos para a linguagem de baixo nível através de um programa denominado **compilador** ou de um **interpretador**

Linguagens de Programação

Linguagem de Alto Nível

- Exemplo de instrução de uma linguagem de alto nível:
 - **if (A>10) then A=A-7;**



Compilador

- Compilação : transformação de um programa em **código fonte** para **linguagem de máquina**
 - programa em código fonte = programa escrito pelo programador (***source language***)
 - programa em linguagem de máquina = programa executável (***target language***)

Interpretador

- Executa instruções em uma determinada linguagem:
 - Traduz o código fonte em uma representação intermediária e o executa **imediatamente**
- **Python**, **Matlab**, **Java**, etc.
- Principal desvantagem: Eficiência

Perguntas???