



INSTITUTO FEDERAL
SANTA CATARINA

Programação de Computadores I - PRG1

Engenharia Elétrica

Edilson Hipolito da Silva

edilson.hipolito@ifsc.edu.br - <http://www.hipolito.info>

Aula 06 - Introdução a algoritmos

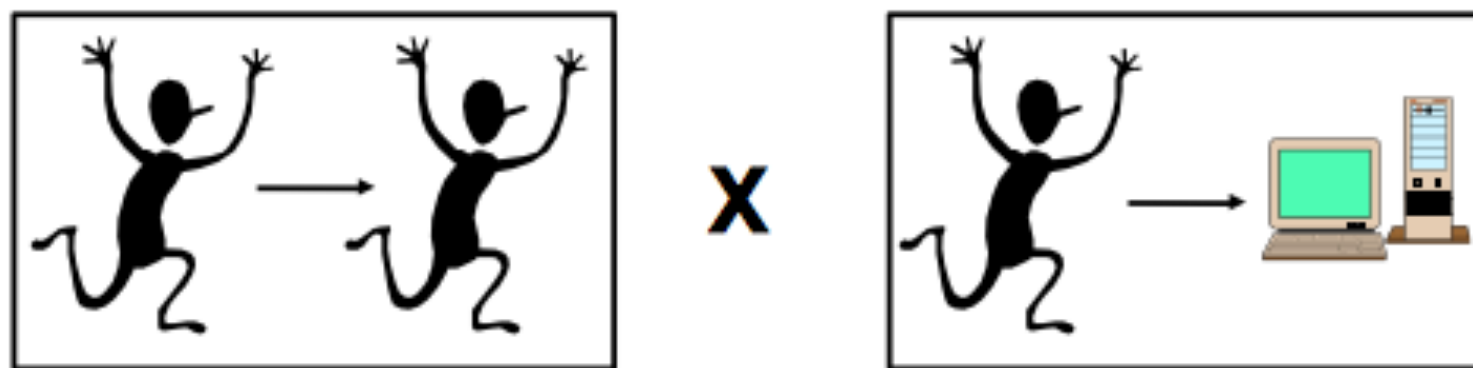
Roteiro

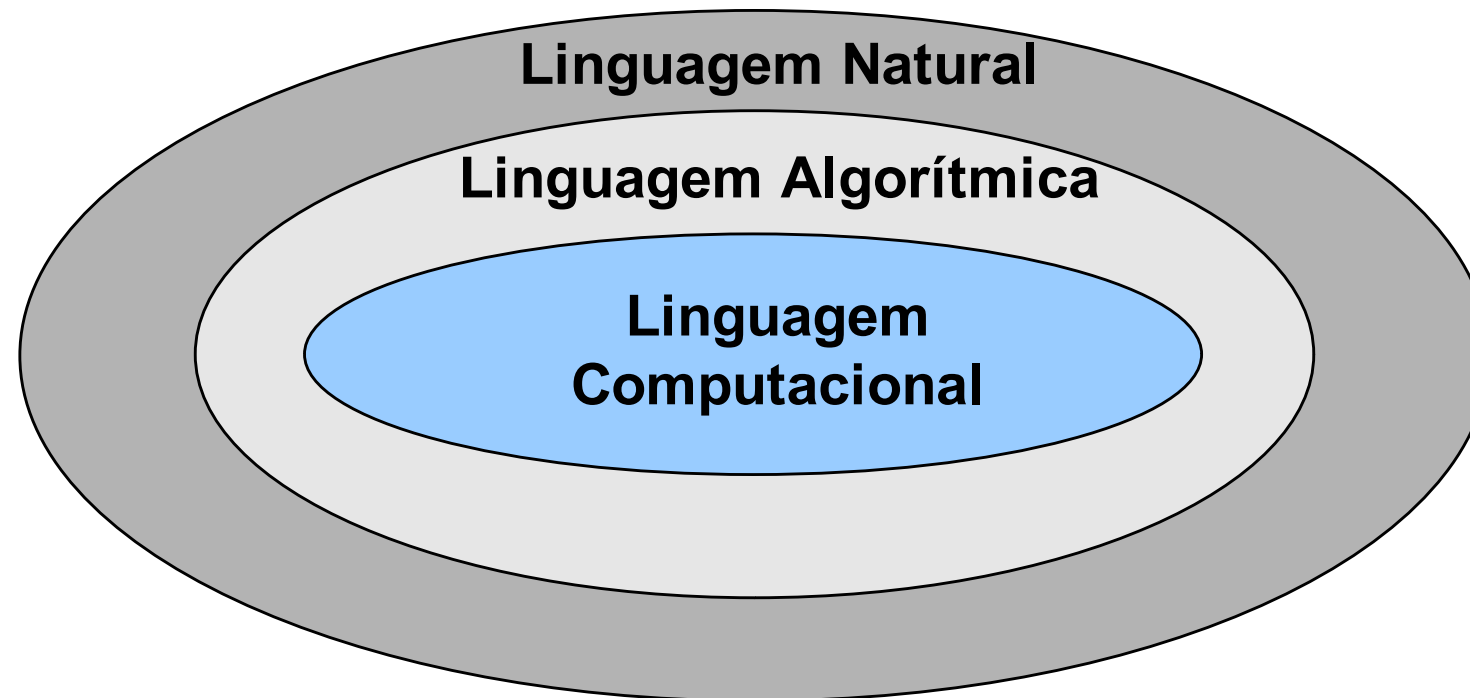


INSTITUTO FEDERAL
SANTA CATARINA

- Algoritmos
- Formas de representação de algoritmos
- Exercícios

- **Ambiguidade:** relacionada à sintaxe e semântica de linguagens naturais.
- “O artista produz formas perfeitas.”
- **Linguagens naturais são muito extensas, possuindo ambiguidade;**
- Linguagem Natural X Linguagem Computacional





- Nos algoritmos, devemos utilizar construções que executarão uma ação específica bem definida;
- Uma linguagem algorítmica deve ter sintaxe rígida e semântica bem definida;
- Toda linguagem possui sintaxe e semântica.



- **Sintaxe:** dita as regras de como as sentenças e cada um de seus elementos devem ser construídos corretamente;
 - Regras Sintáticas
- **Semântica:** se preocupa com o significado de uma sentença construída.



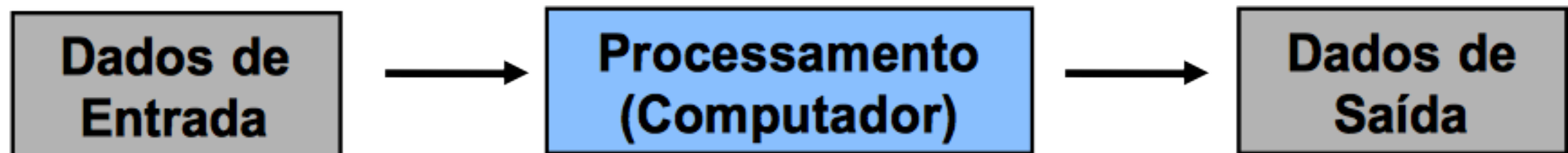
- A **sintaxe** por si só **não define completamente uma linguagem**;
- **Exemplo:**
 - “Descoloridas idéias verdes sonham furiosamente” (Chomsky)
- Por outro lado:
 - “Nóis foi, mais já vortemo”

Algoritmo



INSTITUTO FEDERAL
SANTA CATARINA

- Algoritmos geralmente descrevem algum processamento de dados:



- Estes dados precisam ser representados (descritos) nos algoritmos;



- Um algoritmo para fins computacionais é a **descrição de ações que manipulam objetos em função do tempo.**
 - **Objetos descrevem o estado computacional;**
 - Objetos **têm função específica;**
 - **Ações** são **direcionadas aos objetos;**
 - **Objetos** são definidos como **parte do programa;**
 - **Estados diferentes** requerem **objetos diferentes.**
- * **Objetos** frequentemente serão **chamados de variáveis.**

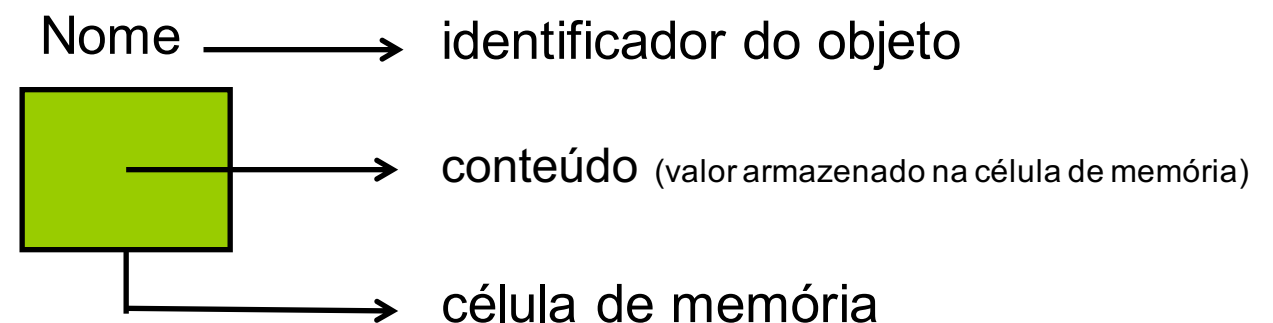


- A cada objeto é associado um nome que identifica este objeto ao longo da programação.
- Este nome é chamado **identificador**.
- De forma geral, podemos dizer que um **identificador** está **associado à uma célula de memória**.
- O termo **variável** é frequentemente utilizado como **sinônimo de identificador**.



- Em geral, as linguagens de programação exigem nomes de identificadores com as seguintes características:
 - Use somente letras e números;
 - O primeiro caracter do nome deve ser uma letra;
 - Não é permitido o uso de caracteres especiais.
- **Exemplos válidos:** ImpostoDeRenda, F1, soma, produto, desconto, SalarioHora.
- **Exemplos inválidos:** 1f, Imposto-renda, salario hora.

- Atributos (características) de um objeto:
 - Nome;
 - Célula de memória;
 - Valor associado;
- Tipo: **define a categoria dos dados que podem ser armazenados no objeto.**





- **Tipo de dados:**
 - especifica uma classe de valores que podem ser armazenados na célula de memória associada ao identificador;
 - define as operações válidas sobre o identificador;



- **Tipo de dados:**

- Um identificador **armazena um único dado** num instante, mas este dado **não pode ser qualquer um**;
- Os objetos serão **classificados** segundo um **tipo**;
- Os dados podem ter **estruturas físicas diferentes**;
- As células de **memória** do computador precisam ser **compatíveis** com o **tipo de dado** que vão armazenar;

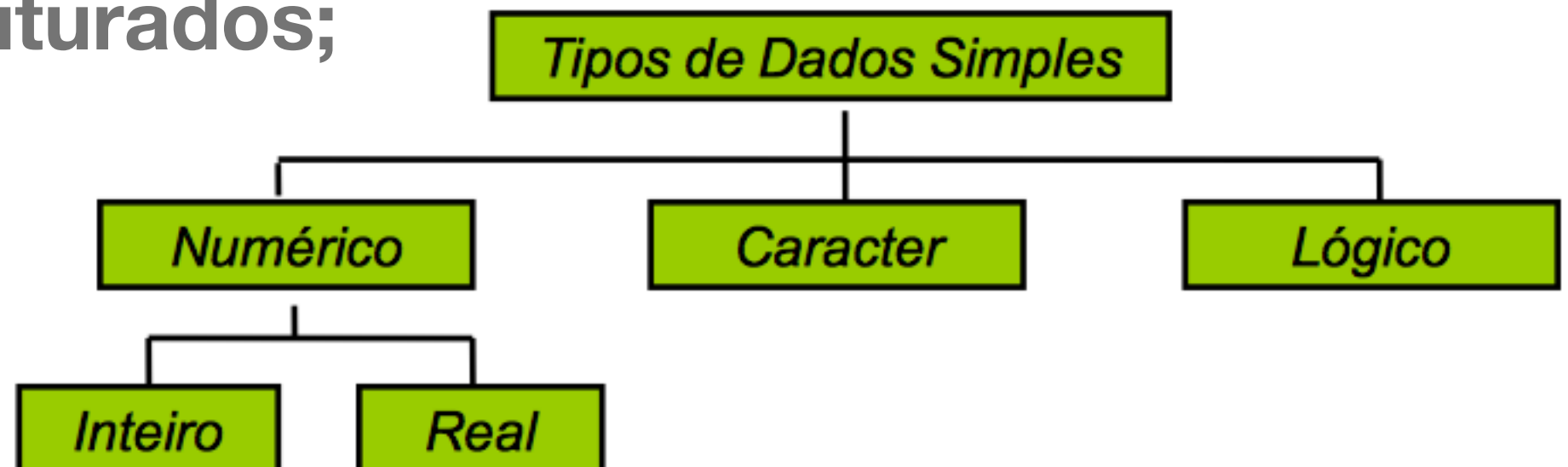


- Devemos informar ao computador qual o tipo de dado que vamos armazenar;
- Esta informação determina **quantas células de memória serão necessárias** para armazenar o dado;
- Esta informação define quais serão as **operações possíveis** de se aplicar sobre os dados;
- **Exemplo:** em um dado numérico pode-se aplicar uma operação aritmética.



Algoritmo

- Os tipos de dados podem variar de uma linguagem para outra.
- Existem dois tipos básicos:
 - **Tipos simples;**
 - **Tipos estruturados;**





- Os Tipos de Dados Numéricos são divididos basicamente em dois grandes conjuntos:
 - **Inteiros;**
 - **Reais;**
- Os inteiros podem ser positivos, negativos ou nulos, mas **não possuem um componente decimal;**



- Os inteiros são compatíveis com os reais, mas os reais não são compatíveis com os inteiros;
- Assim, um objeto real pode receber um valor inteiro, mas um objeto inteiro não pode receber um valor real;



Algoritmo

- Os reais podem ser positivos, negativos ou nulos, e possuem sempre um componente decimal;
- **Exemplos:** 2,34; 0,0; -214,123;
- Computacionalmente, existem muitas diferenças entre o armazenamento de números inteiros e números reais;



- Os tipos de dados **não numéricos**:
 - Caracter: formado pelas letras (de A a Z, e de a a z), dígitos (de 0 a 9) e caracteres especiais (~, ., ?, >, <, ...);
 - A diferença entre caracteres e dados numéricos estão na forma de armazenamento, e nas operações legais permitidas sobre os mesmos;
 - Exemplo: operações aritméticas sobre os dados numéricos.



- Objetos declarados como do tipo **caractere** poderão armazenar um **único caractere**;
- Para armazenar uma **sequência** de caracteres será utilizada a **estrutura cadeia de caracteres**, que consiste num **tipo** de dado **estruturado**;
- **Exemplo:**
 - ‘Z’ é um caractere;
 - ‘ANA’ é uma cadeia de caracteres;



- **Tipo Lógico:** os valores lógicos podem assumir uma entre duas possibilidades (**verdadeiro** ou **falso**);
- Só poderá armazenar um destes dois valores;
- Muito utilizados em controle do fluxo lógico do algoritmo;



- Escolha do tipo de um identificador: deve-se conhecer bem a solução do problema.
 - **Exemplos de inteiros:**
 - número de pessoas;
 - um número par ou ímpar;
 - idade de uma pessoa;
 - quantidade de objetos;
 - **Exemplos de reais:**
 - o peso de uma pessoa;
 - um percentual;
 - nota de uma avaliação;
 - média;
 - preço de um produto;

Algoritmo



INSTITUTO FEDERAL
SANTA CATARINA

- Situações de dúvida
- Exemplos:
 - raíz quadrada de um número;
 - divisão entre dois números inteiros;
- Quando houver alguma possibilidade de um identificador numérico receber um valor não inteiro, deve ser declarado como real.





Algoritmo: Entrada e Saída

- **Entrada:** objetos cujos valores são conhecidos em tempo de execução;
- **Saída:** objetos que armazenam a solução objetivo de um problema;



Formas de Representação

- **Descrição em Linguagem Natural**
- **Fluxograma**
- Diagrama de Nassi-Schneiderman ou de Chapin
- Linguagem Algorítmica ou Pseudocódigo ou Portugol
- **Linguagem de Programação**



Linguagem Natural

- **Linguagem natural:** algoritmos expressos diretamente em linguagem natural, como nas receitas.
- **Vantagens:** Facilidade de compreensão, linguagem já conhecida.
- **Desvantagens:** Pode levar a ambiguidade (várias interpretações); Não há rigor, portanto um algoritmo pode ser resumido ou detalhado demais;
- Por exemplo, em um algoritmo de troca de pneu
 - A instrução “afrouxar ligeiramente os parafusos”, está sujeita interpretações diferentes.
- **Uma instrução mais precisa seria:**
 - “afrouxar a porca, girando-a de 30° no sentido anti-horário”.



Exemplo

- Faça um Algoritmo para calcular a média aritmética entre duas notas de um aluno e mostrar sua situação, o que pode ser APROVADO ou REPROVADO.
- **Passo 1:** Receber os valores das duas notas
- **Passo 2:** Calcular a média aritmética
- **Passo 3:** Mostrar a média aritmética
- **Passo 4:** Se a média aritmética for maior ou igual a 7, então situação do aluno é APROVADO; caso contrário, a situação é REPROVADO.



Fluxogramas

- **Fluxogramas:** representação gráfica, por meio de símbolos geométricos.
- Cada tipo de operação é representado por um símbolo diferente.
- Permite o acompanhamento visual do fluxo do algoritmo.
- **Vantagem:** O entendimento de elementos gráficos é mais simples que o entendimento de textos.
- **Desvantagem:** Necessidade de aprender a simbologia; algoritmo resultante não apresenta muitos detalhes, dificultando sua transição para um programa.

Fluxogramas



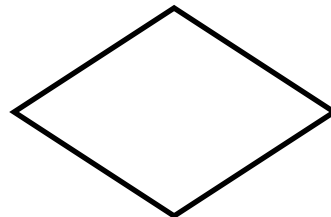
INSTITUTO FEDERAL
SANTA CATARINA



Início e fim de algoritmo



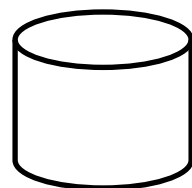
Atribuições e cálculos de valores



Decisões



Entrada de dados

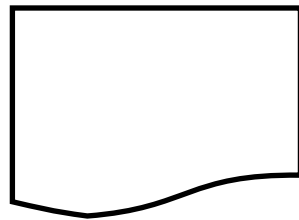


Discos

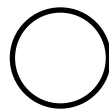
Fluxogramas



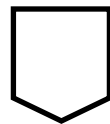
INSTITUTO FEDERAL
SANTA CATARINA



Impressão de resultados



Conector na página



Conector fora da página

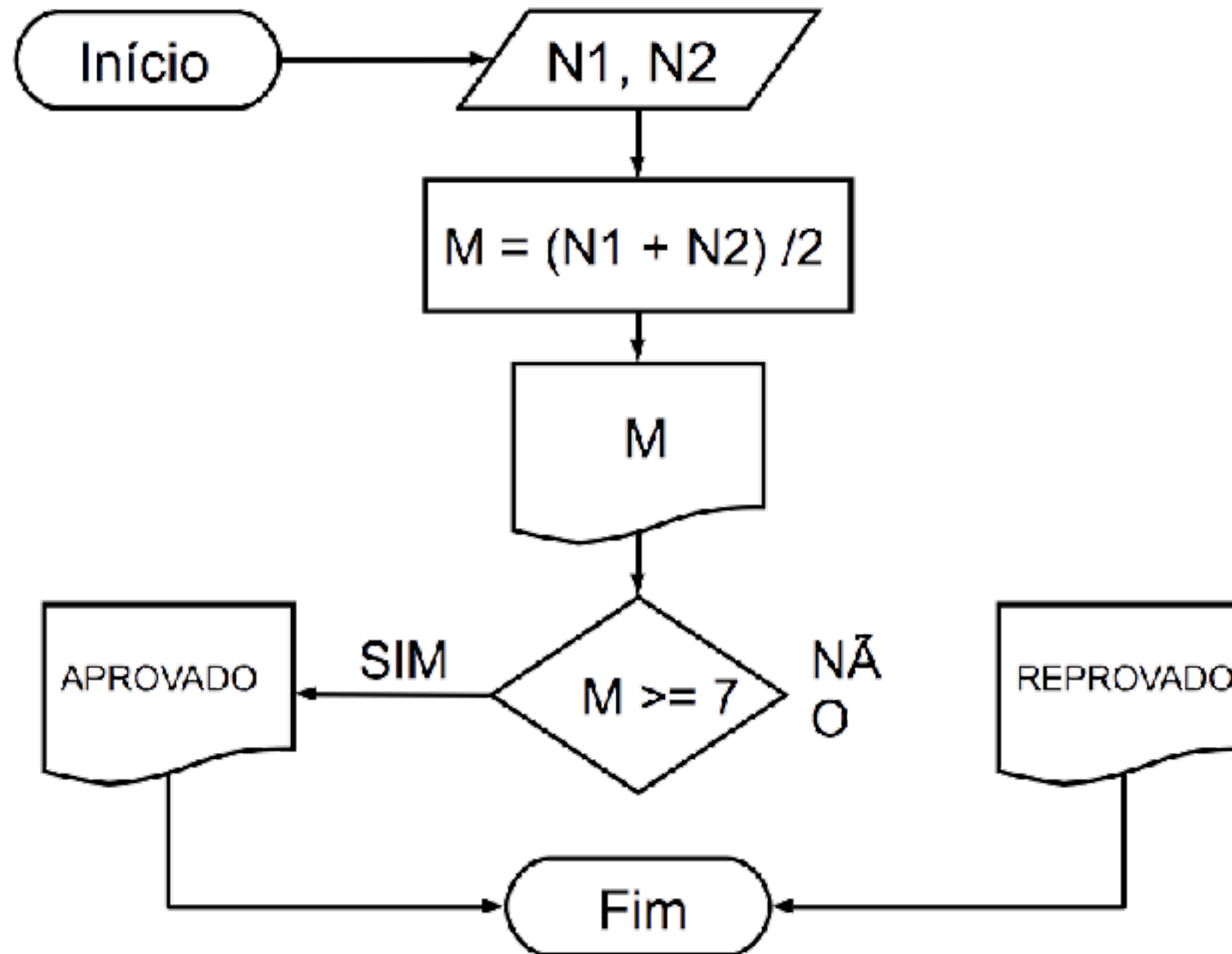


Entrada manual



Indica o sentido do fluxo e serve de ligação entre blocos existentes.

Exemplo





Pseudocódigo

- **Pseudo-linguagem:** emprega linguagem intermediária entre linguagem natural e linguagem de programação
- Este método procura misturar as facilidades da linguagem natural com a precisão das linguagens de programação
- A representação da solução em pseudo código possibilita ao programador a abstração dos detalhes da sintaxe da linguagem escolhida.



ALGORITMO

DECLARE N1, N2, M **NUMÉRICO**

ESCREVA "Digite as duas notas"

LEIA N1, N2

$M \leftarrow (N1 + N2) / 2$

ESCREVA "Média = ", M

SE M \geq 7

ENTÃO ESCREVA "Aprovado"

SENÃO ESCREVA "Reprovado"

FIM_ALGORITMO.



Linguagem de programação

- A própria linguagem de programação é uma forma de representação de algoritmos.
- **Neste caso os algoritmos estão codificados na linguagem de programação específica.**

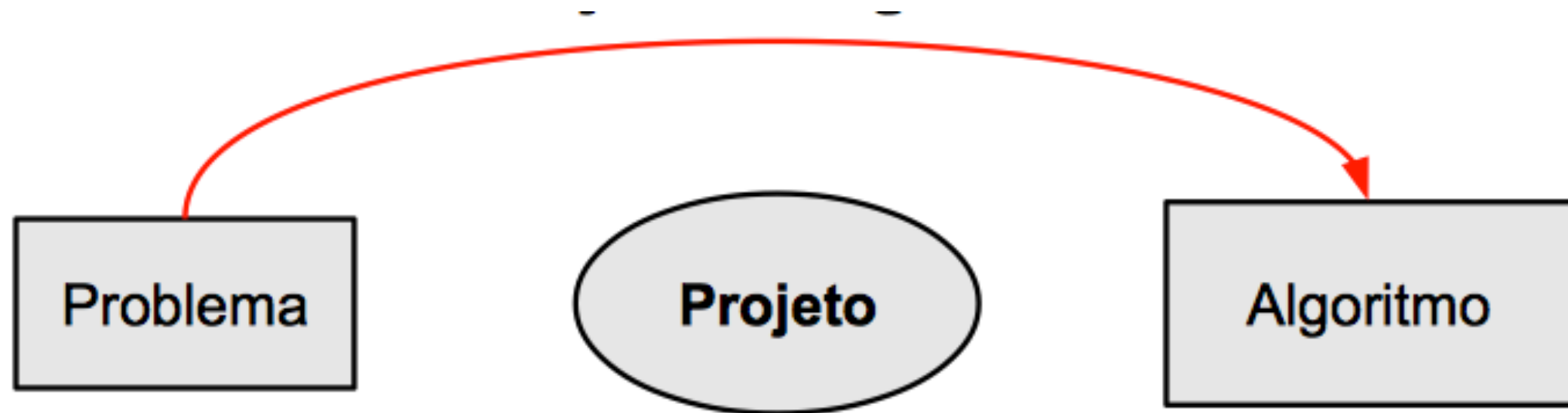
Linguagem de programação



INSTITUTO FEDERAL
SANTA CATARINA

```
media.c x
1 #include <stdio.h>
2
3 int main(){
4     float n1, n2, m;
5
6     printf("Digite as duas notas:\n");
7     scanf("%f", &n1);
8     scanf("%f", &n2);
9
10    m = (n1 + n2) / 2;
11
12    printf("Média: %.2f\n", m);
13    if(m >= 7.0){
14        printf("Aprovado\n");
15    }
16    else{
17        printf("Reprovado\n");
18    }
19 }
```





**Muitas vezes nos esquecemos do projeto.
Passamos direto para a codificação, o que é errado.**



Técnicas de construção

- **Divisão para a conquista:** Técnica de divisão do problema em partes menores.
- As soluções das partes menores combinam-se para a composição da solução global.
- **Exemplo:** receita de um bolo (modo de preparo do bolo, preparo do recheio, preparo da cobertura)
- Permite que seja preparado por mais de uma pessoa e simplifica a execução
- **Tarefas independentes → paralelismo.**

Exercícios



INSTITUTO FEDERAL
SANTA CATARINA

- Ler dois números inteiros e imprimir a soma.
- Entrar com dois números inteiros e imprimir a média aritmética.
- Escreva um algoritmo que calcule o diâmetro, a área e a circunferência de um círculo, sabendo que o único dado disponível é o seu raio.
 - $\text{Diâmetro} = 2 * \text{Raio}$
 - $\text{Área} = \text{Pi} * \text{Raio} * \text{Raio}$
 - $\text{Circunferência} = 2 * \text{Pi} * \text{Raio}$
- Criar um algoritmo que solicite e imprima os valores da diagonal maior, diagonal menor e calcule a área de um losango.
- Fórmula: $(\text{diagonal maior} * \text{diagonal menor})$ dividido por 2

Exercícios



INSTITUTO FEDERAL
SANTA CATARINA

- Elaborar um algoritmo para calcular e apresentar o volume de uma lata de óleo, utilizando a fórmula:
 - $VOLUME = 3.14159 * (R * R) * ALTURA.$
- Escreva um algoritmo que pergunte o nome a altura (em metros) e a massa (em Kg) do usuário. Em seguida o programa deverá exibir uma mensagem dizendo o nome do usuário e a sua densidade corporal.
 - $Densidade = peso / altura^2$

Dúvidas?



INSTITUTO FEDERAL
SANTA CATARINA

