

Introdução

Givanaldo Rocha de Souza

givanaldo.rocha@ifrn.edu.br

<http://docente.ifrn.edu.br/givanaldorochoa>

➤ **Lógica de Programação**

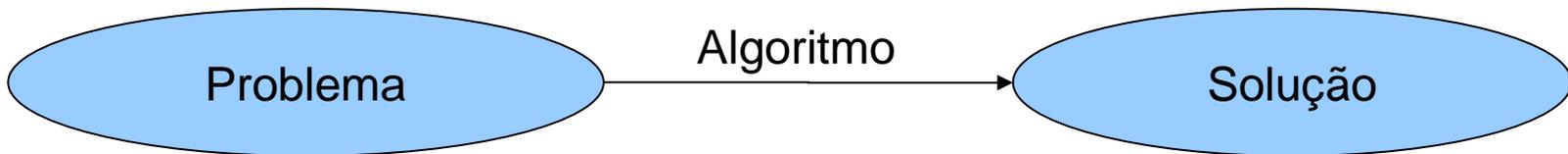
- técnica de encadear pensamentos para atingir determinado objetivo.

➤ **Algoritmo**

- sequência de passos finitos com o objetivo de solucionar um problema.
-

Algoritmos

- Não é a solução para um problema.
- É um conjunto de passos (ações) que levam à solução de um determinado problema.
- Existem várias maneiras de se chegar a solução de um mesmo problema.



Algoritmos

Todos nós nos deparamos com vários problemas no dia-a-dia.

Quantas vezes já vimos um algoritmo e não sabíamos que aquela sequência de passos tinha esse nome?

Ex: *Instruções escritas em um telefone público.*

Algoritmo usar orelhão

- 1) Retirar o telefone do gancho
- 2) Esperar o sinal de discagem
- 3) Colocar o cartão
- 4) Discar o número
- 5) Falar no telefone
- 6) Colocar o telefone no gancho
- 7) Retirar o cartão

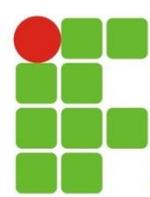


Exercício

Formar grupos e escrever algoritmos para:

- Trocar pneu de um carro;
- Fritar um ovo

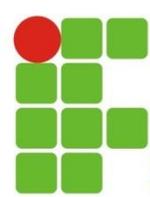




Exercício - resolução

- Separar ingredientes
 - Manteiga
 - ovos
- Separar utensílios
 - Colher
 - Frigideira
- Preparar ambiente
 - Ascender o fogo
 - Colocar a frigideira no fogo
- Modo de preparo
 - Coloque a manteiga na frigideira, depois junte os ovos e mexa. Quando o ovo fritar, desligue o fogo.





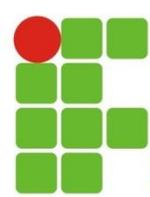
Pseudo-código

A linguagem natural ainda é imprecisa.

- Quanto de manteiga?
- Como mexer?
- O que define que está pronto?

Uso de uma linguagem restrita e estruturada

- Pseudo-código
-



Pseudo-código

fritar_ovos

separar_ingredientes

pegar 2 ovos

pegar 100 g de manteiga

separar_utensílios

pegar frigideira

pegar colher de pau

ascender fogo

colocar a frigideira no fogo

colocar a manteiga na frigideira

deixar a manteiga derreter

quebrar levemente a casca do primeiro ovo deixar a clara e a gema na frigideira

repetir o passo anterior para o segundo ovo

mexer os ovos com a colher

...



Instrução de um algoritmo

Cada linha do algoritmo pode ser chamada de **instrução**.

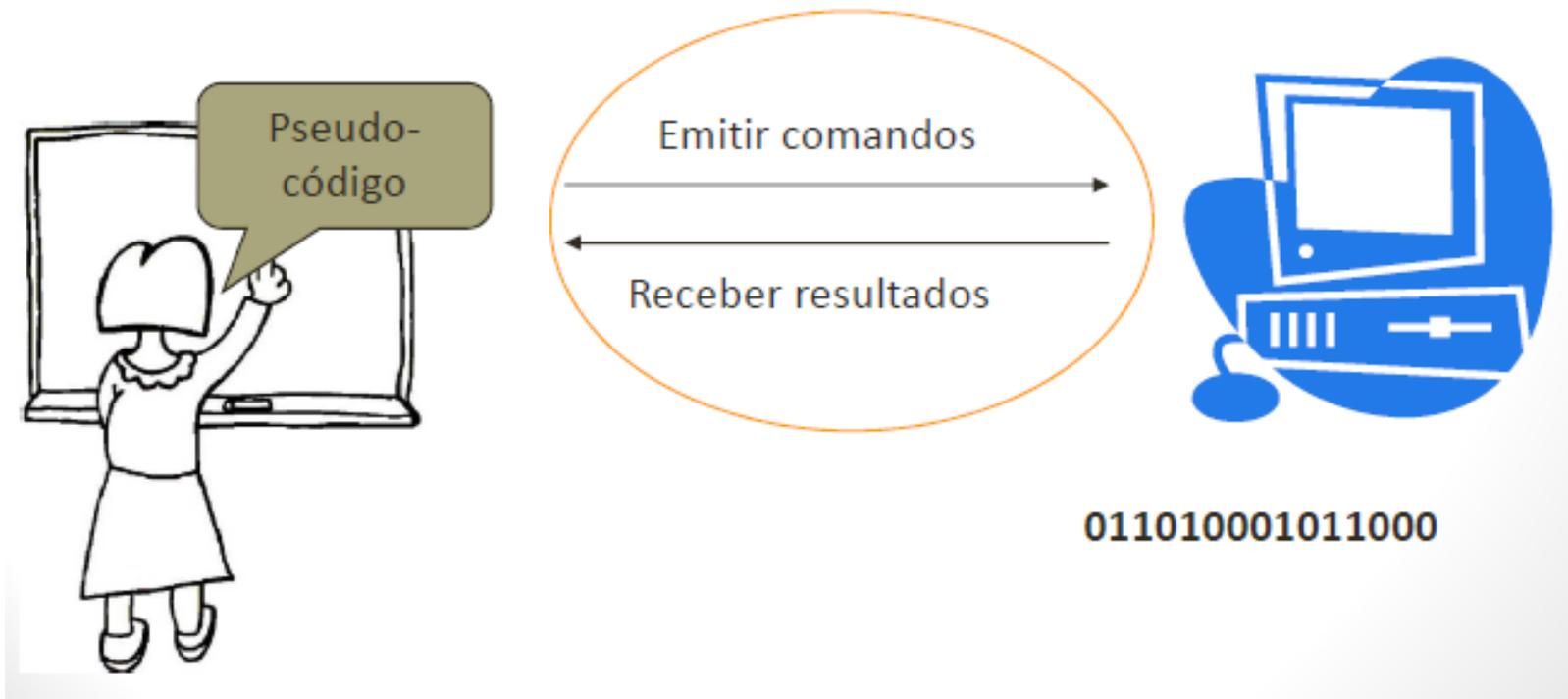
Logo, pode-se dizer que um algoritmo é um conjunto de instruções.

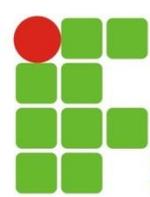
Uma **Instrução** indica a um computador uma ação elementar a ser executada.

Linguagem de Programação

Computadores não entendem pseudo-código.

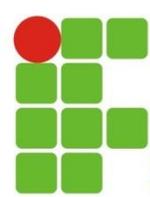
- É preciso utilizar uma **linguagem de programação**.





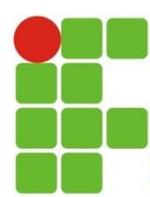
Linguagem de Programação

- Uma **linguagem de programação** é um método padronizado para expressar instruções para um computador.
- É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.
- Uma linguagem de programação pode ser convertida, ou traduzida, em código de máquina por compilação ou interpretação.



Compiladores e Interpretadores

- A única linguagem que o computador entende chama-se linguagem de máquina. Portanto todos os programas que se comunicam com a máquina devem estar em linguagem de máquina.
- A forma como os programas são traduzidos para a linguagem de máquina classifica-os em duas categorias:
 - Interpretados
 - Compilados



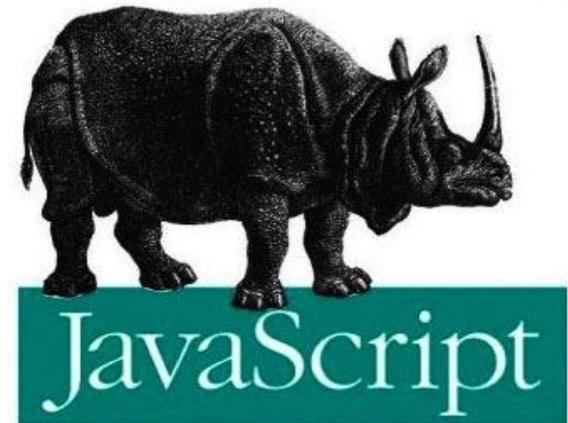
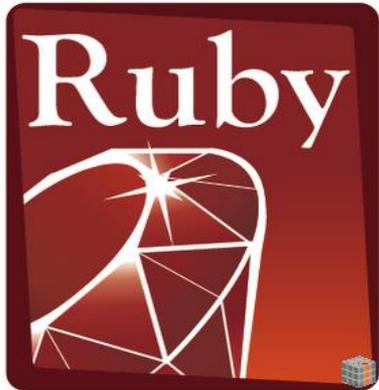
Interpretadores

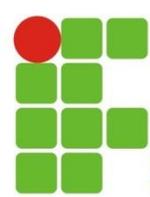
- Interpretadores são programas de computador que leem um código fonte de uma linguagem de programação interpretada e o converte em código executável.
- Em alguns casos, o interpretador lê linha-por-linha e converte em código objeto à medida que vai executando o programa e, em outros casos, converte o código fonte por inteiro e depois o executa.

Interpretadores

- O interpretador precisa estar instalado no computador todas as vezes que vamos executar o nosso programa.

Exemplos de linguagens interpretadas:





Compiladores

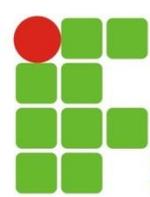
- Um compilador lê a primeira instrução do programa, se não houver erro converte-a para linguagem de máquina e, em vez de executá-la, segue para a próxima instrução repetindo o processo até que a última instrução seja atingida.
- Se não houver erros, o compilador gera um programa em disco.

Compiladores

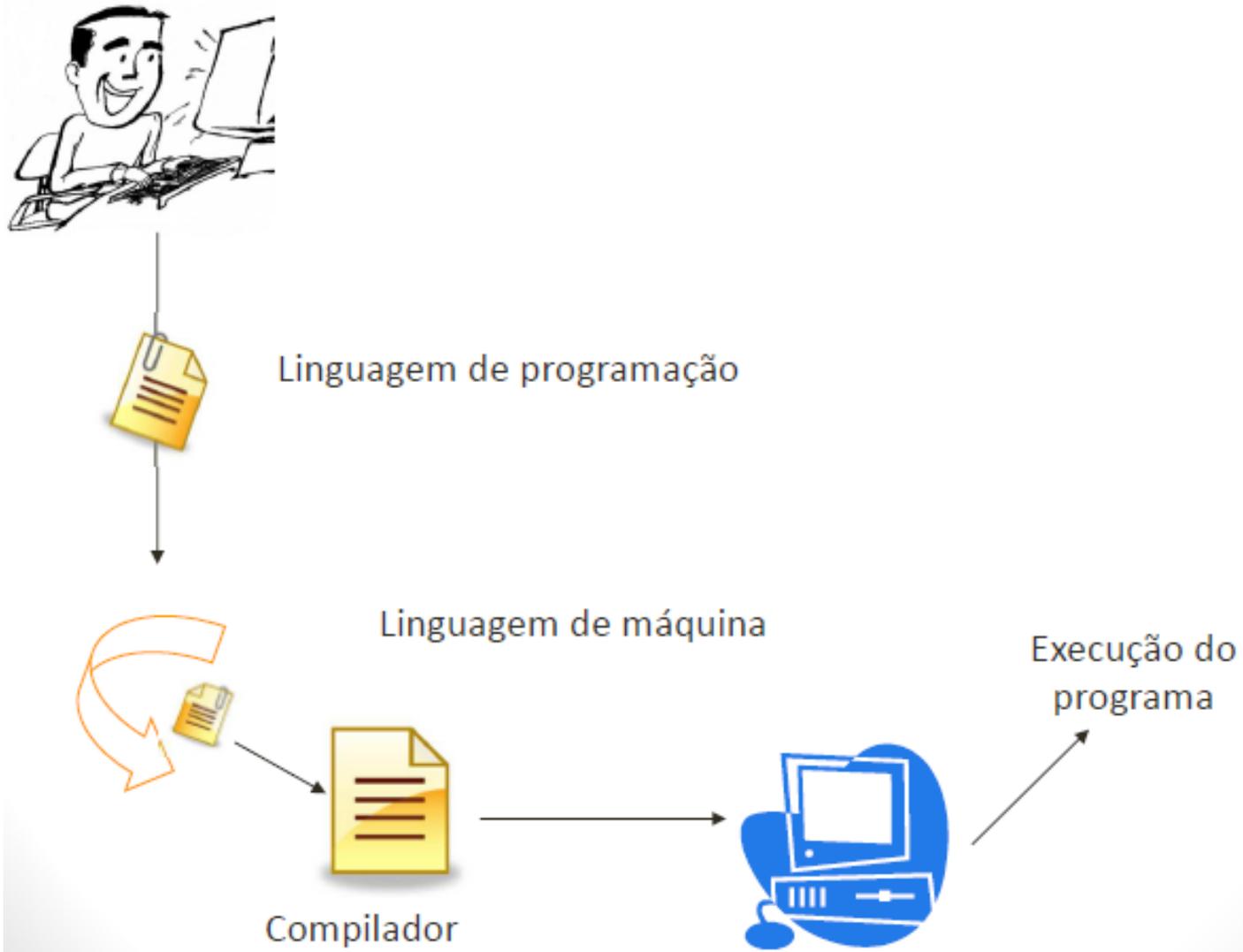
- Após compilado, não é mais necessária a presença do compilador para executar o programa.
- Programas compilados são mais rápidos que interpretados.

Exemplos de linguagens compiladas:

➤ *C, C++, Pascal, Delphi ...*

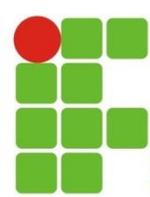


Compiladores



Importância dos algoritmos

- O computador não fará nada que você não mande.
- Ele realiza apenas as instruções fornecidas pelos programadores.
- Com um algoritmo é possível conceber uma solução para um dado problema, **independente de uma linguagem de programação específica**, e até mesmo do computador.

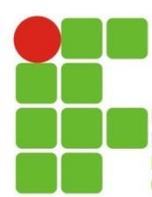


Formas de representação

- **Descrição narrativa**

Receita de Bolo:

- Providencie manteiga, ovos, 2 Kg de massa, etc.
- Misture os ingredientes
- Despeje a mistura na fôrma de bolo
- Leve a fôrma ao forno
- Espere 20 minutos
- Retire a fôrma do forno
- Deixe esfriar
- Prove



Formas de representação

- **Descrição narrativa**
 - Vantagens
 - ✓ O português é bastante conhecido por nós
 - Desvantagens
 - ✓ Imprecisão
 - ✓ Pouca confiabilidade
 - ✓ Extensão

Formas de representação

- **Fluxograma:** símbolos gráficos para representar algoritmos.



Cálculo



Decisão



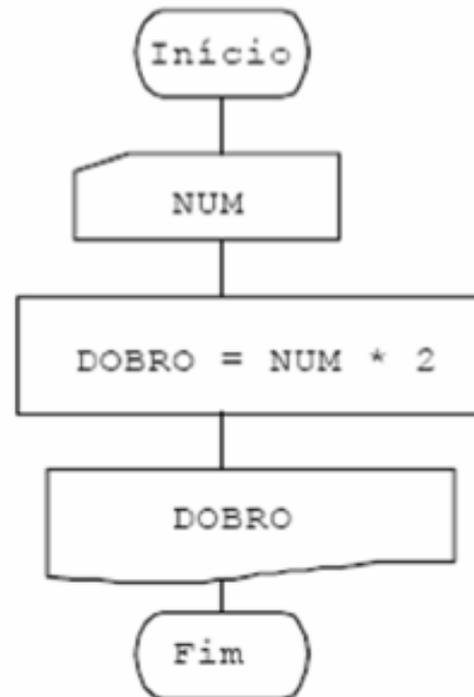
Entrada



Saída



Início/Fim



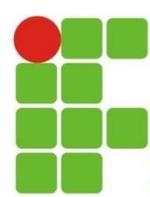
- **Fluxograma**

- Vantagens

- ✓ Imagens dizem mais do que palavras
- ✓ Padrão mundial

- Desvantagens

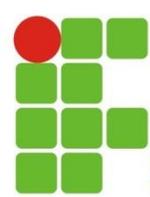
- ✓ Pouca atenção aos dados
 - ✓ Complica-se à medida que o algoritmo cresce
-



Formas de representação

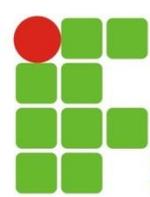
- **Linguagem algorítmica:** pseudo-linguagem de programação.

```
Algoritmo CALCULA_DOBRO  
  NUM, DOBRO : inteiro  
  início  
    Leia NUM  
    DOBRO ← 2 * NUM  
    Escreva DOBRO  
  fim
```



Formas de representação

- **Linguagem algorítmica**
 - Vantagens
 - ✓ Usa o português como base
 - ✓ Pode-se definir quais e como os dados serão estruturados
 - ✓ Passagem quase imediata para uma linguagem de programação
 - Desvantagens
 - ✓ Exige-se a definição de uma linguagem não real para trabalho
 - ✓ Não padronizado



Portugol

- É uma pseudo-linguagem que permite ao programador pensar no problema em si e não no equipamento que irá executar o algoritmo.
- Devem ser considerados a sintaxe (em relação à forma) e a semântica (em relação ao conteúdo ou seu significado).

Portugol:

Inicio

escreva("Olá Mundo!")

fimalgoritmo

C:

```
main( ) {  
    printf("Olá Mundo!")  
}
```

Python:

```
print "Olá Mundo!"
```

Estrutura do algoritmo

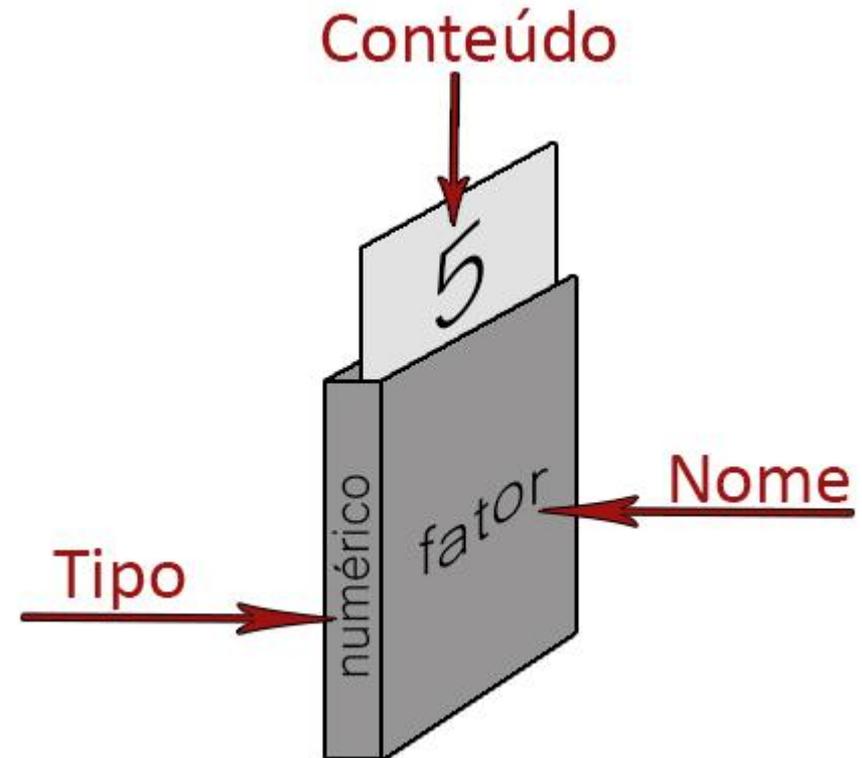
- Formato básico do pseudo-código (portugol):

```
algoritmo "nome_do_algoritmo"  
// Seção de Declarações  
inicio  
// Seção de Comandos  
fimalgoritmo
```

Variáveis

Uma **variável** é um local na memória principal, isto é, um endereço que armazena um conteúdo.

O conteúdo de uma variável pode ser de vários tipos: inteiro, real, caractere, lógico, entre outros.



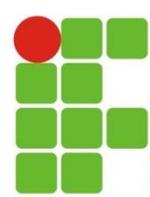
Variáveis

- **Lógico**

- ✓ Também conhecido como booleano.
- ✓ É representado no algoritmo pelos dois únicos valores lógicos possíveis: VERDADEIRO ou FALSO.
- ✓ Existem também outras referências como: *sim/não*, *1/0*, *true/false*

- **Inteiro**

- ✓ Variáveis deste tipo armazenam números inteiros, podendo ser positivos ou negativos.
- ✓ Ex: *0*, *1*, *2*, *-10*, ...



Variáveis

- **Real**

- ✓ Os números reais são aqueles que podem possuir componentes decimais ou fracionários.
- ✓ Ex: 2.5, 33.7, -78.99

- **Caractere**

- ✓ Também conhecido com string, alfanumérico ou literal. Esse tipo de variável armazena dados que contém letras, dígitos e/ou símbolos especiais.
- ✓ Ex: “*Maria*”, “123”, “0”, “A”

Variáveis

Espaço alocado em memória pelas variáveis

Tipo	Tamanho (bits)	Tamanho (bytes)
Lógico (booleano)	1	-
Caractere	8	1
Inteiro	16	2
Real	32	4

Nomes de variáveis

- Os nomes das variáveis devem começar por uma letra e depois podem conter letras, números ou sublinhado.
- Não pode haver duas variáveis com o mesmo nome.

Ex: *x*, *n1*, *media*, *nome_completo*, *nota_2*.



Declaração de variáveis

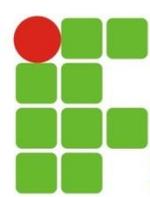
var

a: inteiro

Valor1, Valor2: real

nome_do_aluno: caractere

sinalizador: logico



Importante saber...

- *“A noção de algoritmo é básica para toda a programação de computadores”*
 - KNUTH - Professor da Universidade de Stanford, autor da coleção “The art of computer programming”

- *“O conceito central da programação e da ciência da computação é o conceito de algoritmo”*
 - WIRTH - Professor da Universidade de Zurique, autor de diversos livros na área e responsável pela criação de linguagens de programação como PASCAL

