

# Introdução à Programação

UFRPE

*Prof. Gustavo Callou*  
*gcallou@gmail.com*

# Tópicos

- ▶ Algoritmos
  - Representação
  - Exercícios
- ▶ Linguagens de Programação
- ▶ Compilador
- ▶ Interpretador
- ▶ Ambiente de Desenvolvimento
  - Python
    - Característica
    - Para que serve
    - Onde utiliza

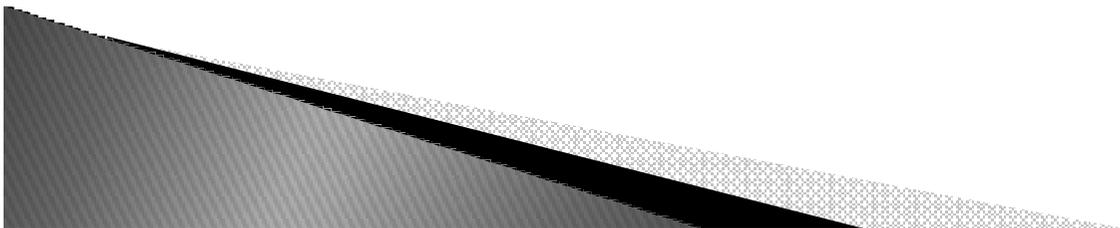
# Algoritmo

- ▶ Conjunto finito de regras que provê uma seqüência de operações para resolver um tipo de problema específico. Knuth, 1972.
- ▶ Seqüencia ordenada, e não ambígua, de passos que levam à solução de um dado problema. Tremblay, 1983.
- ▶ Processo de cálculo, ou de resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições, as regras formais para a obtenção do resultado ou da solução do problema.  
AURÉLIO.

# 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

## Descrição Narrativa – Receita de Bolo

Misture os ingredientes

Junte a forma com manteiga

Despeje a mistura na forma

Se houver coco ralado

então despeje sobre a mistura

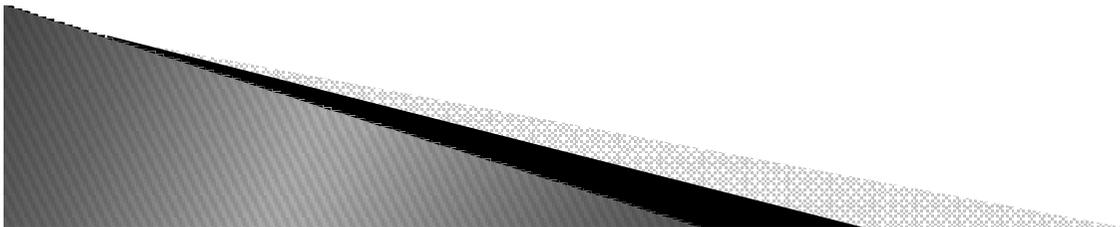
Leve a forma ao forno

Enquanto não corar

deixe a forma no forno

Retire do forno

Deixe esfriar



# Algoritmo

## Descrição Narrativa – Troca de Pneus

Afrouxar ligeiramente as porcas

Suspender o carro

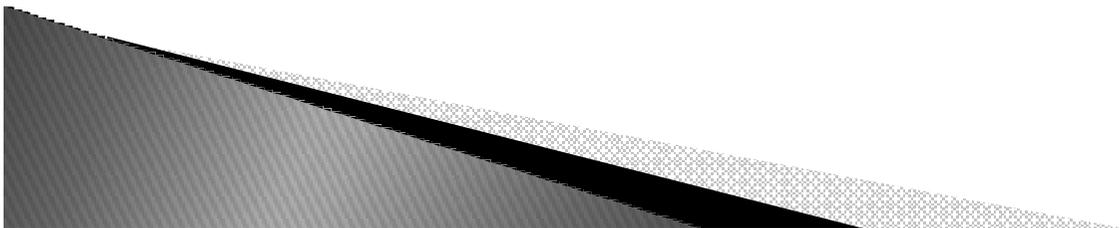
Retirar as porcas e o pneu

Colocar o pneu reserva

Apertar as porcas

Abaixar o carro

Dar o aperto final nas porcas



# Algoritmo

## Descrição Narrativa – Média de um aluno

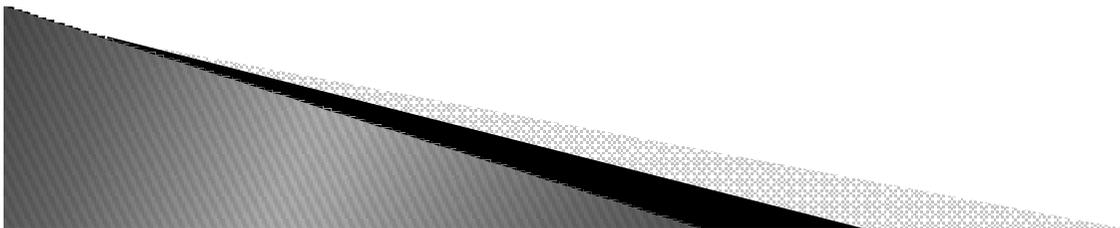
Obter as suas 2 notas de provas

Calcular a média aritmética

Se a média for maior que 7,

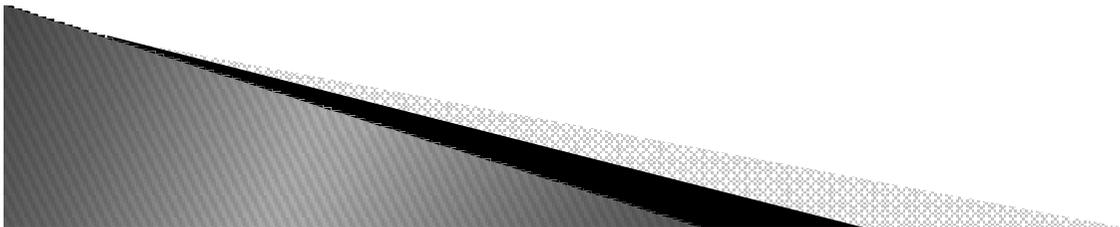
o aluno foi aprovado,

senão ele foi reprovado



# Exercícios de Descrição Narrativa

- ▶ Vamos Praticar !
  1. Passos para chegar a UFRPE e assistir aula de introdução a programação.
  1. Passos para comprar algo via internet.



# Algoritmo

Fluxograma : representação gráfica de algoritmos onde formas geométricas diferentes implicam ações (instruções, comandos) distintos.

## Símbolos



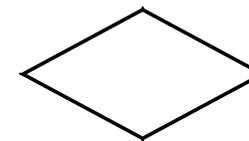
Início e final



Processamento



Entrada de dados



Decisão

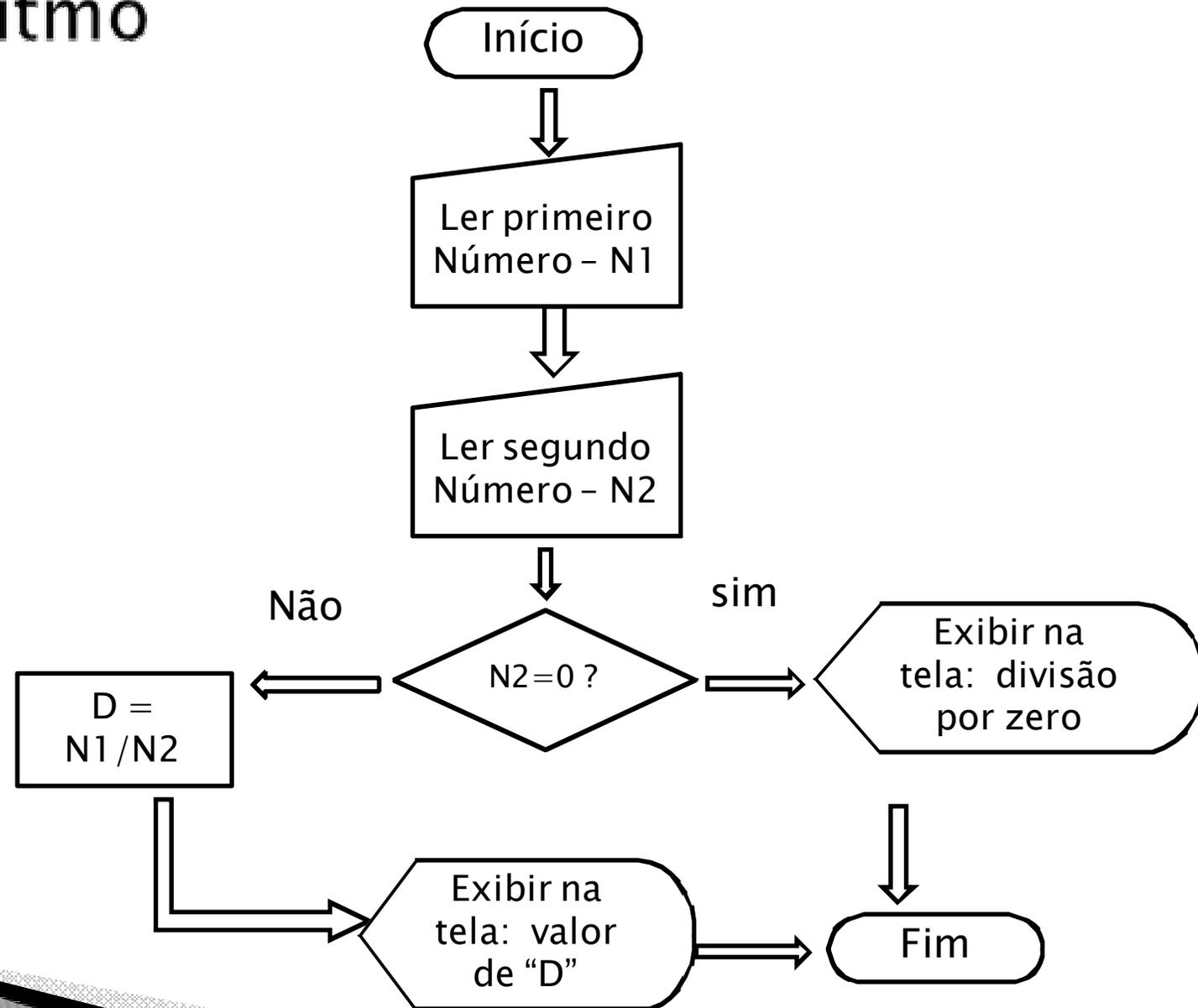


Saída de dados



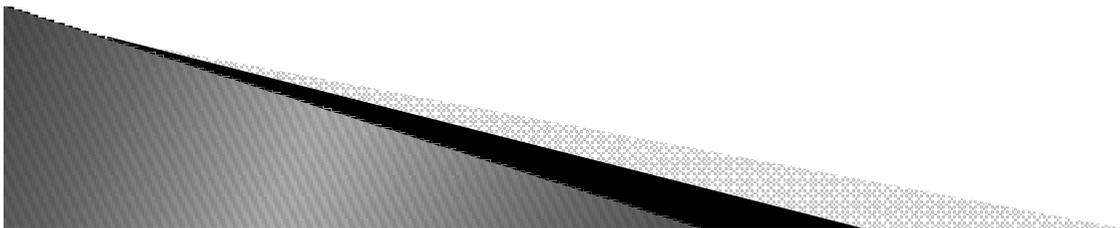
Conector

# Algoritmo



# Exercícios

- ▶ Vamos Praticar !
  1. Calcular a área de um triângulo retângulo.
  2. Verificar se o individuo é maior de idade ou não.



# Algoritmo

## ▶ Pseudo-Código

```
Programa PDOBRO;  
Variáveis NUM, DOBRO : inteiros  
início  
    escreva ('Digite um número: ')  
    Leia (NUM);  
    DOBRO = 2 x NUM;  
    escreva ('O dobro é:', DOBRO);  
fim
```

# Algoritmo

## ▶ Comandos

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

# Linguagens de Programação

- ▶ Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador.
- ▶ Divididas em três tipos, com relação à sua similaridade com a linguagem humana:
  - Linguagem de máquina;
  - Linguagem Simbólica; e
  - Linguagem de Alto Nível (abstrata).

# Linguagens de Programação

- ▶ Linguagem de Máquina : é a linguagem de mais baixo nível de entendimento pelo ser humano e a única, na verdade, entendida pelo processador (CPU).
- ▶ 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.
- ▶ Única para cada tipo de CPU.
- ▶ 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

- ▶ Exemplo de instrução:

ADD A, B

- ▶ Classificada como linguagem de segunda geração, e, assim como a linguagem de máquina, é considerada uma linguagem de baixo nível.

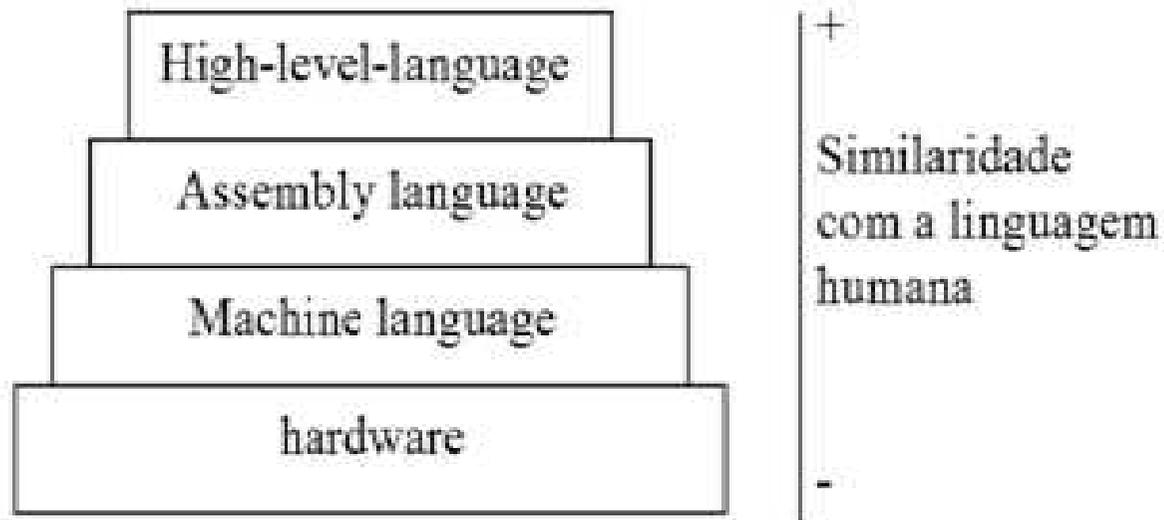
# Linguagens de Programação

- ▶ Linguagens de Alto Nível (abstratas): 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

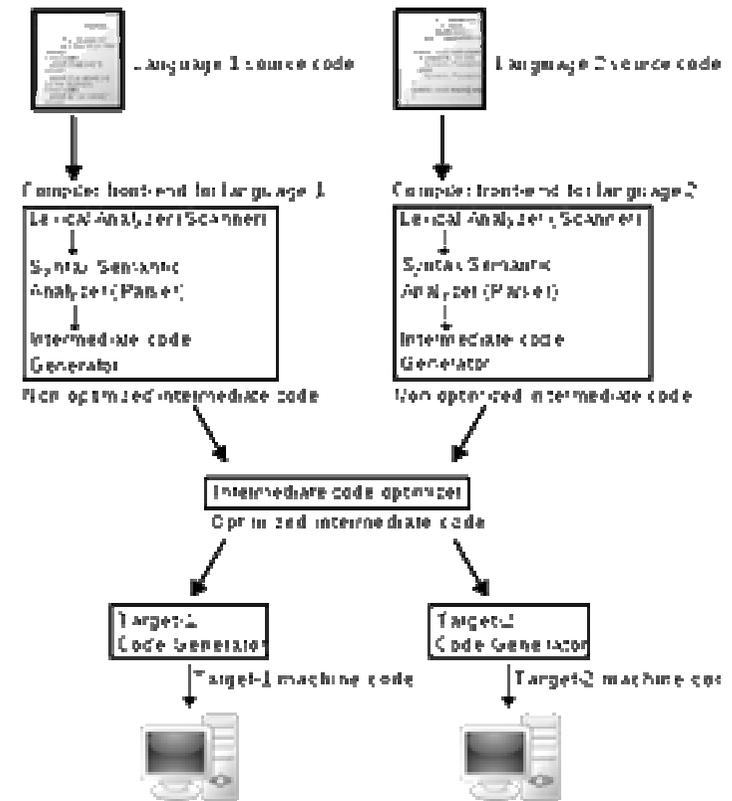
- ▶ Exemplo de instrução de uma linguagem de alto nível:

```
if (A > 10) then A := A - 7;
```



# Compilador

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



# Interpretador

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

# Instalando o Ambiente

- ▶ Entre no link: <http://www.python.org/download/>
- ▶ Escolha a plataforma na qual será instalada a PVM (*Python Virtual Machine*) e Baixe
- ▶ Daqui pra frente depende da plataforma
- ▶ Melhores detalhes no link citado acima
- ▶ Para Windows basta clicar duas vezes no instalador e *Next, Next, Next, Next ... Finish!*
- ▶ *Observação: Toda distribuição Linux já vem com Python instalado*

# Tópicos

- ▶ Caracterizando Python
  - O que é Python?
  - Por que Python?
  - Pra que serve?
  - Quem usa?
  - Como usar?
- ▶ Instalando o ambiente Python
  - Conceitos básicos

# Objetivo

- ▶ Caracterizar a linguagem Python;
- ▶ Conhecer seu ambiente de instalação;
- ▶ Conhecer e executar comando básicos de entrada e saída;

# Características de Python

- ▶ Interpretada
- ▶ Alto nível de abstração
- ▶ Multiplataforma (Windows, Mac OS, Linux. Sun Solaris, AS/400, Amiga, OS/2, BeOS,...)
- ▶ Multiparadigma (OO, imperativa e funcional)
- ▶ Baixa performance (?!?!?)
- ▶ Biblioteca padrão muito rica
- ▶ Estrutura de dados intrínseca poderosa
- ▶ Case Sensitive

# Por que Python ?

- ▶ Alta produtividade (estudos dizem 2–10x mais que C/C++ e Java)
- ▶ Legibilidade de código
- ▶ Fácil aprendizado
- ▶ Facilmente extensível (C/C++, Java)
- ▶ Open Source
- ▶ Madura, desde 1991 (17 anos)

# Para que serve ?

- ▶ Prototipação rápida
- ▶ Desenvolvimento Web
- ▶ Acesso a banco de dados
- ▶ Manipulação de string
- ▶ Computação numérica e científica
- ▶ Jogos
- ▶ 3D
- ▶ Modelagem de Hardware
- ▶ ...



# Quem usa ?

- ▶ Google
- ▶ Yahoo
- ▶ NASA
- ▶ Industrial Light & Magic
- ▶ Nokia
- ▶ Itautec
- ▶ ...



# Como usar ?

- ▶ Python Virtual Machine
- ▶ O código fonte é compilado automaticamente gerando bytecode
- ▶ Arquivos compilados têm extensão “.pyc” ou “.pyo”
- ▶ Pode se embutir a máquina virtual Python dentro de um executável junta com a aplicação
- ▶ Terminal interativo (Shell)
  - Teste de sintaxe
  - Obter ajuda
  - Emitir comandos individualmente

# Bibliografia

- ▶ Python Tutorial –  
<http://www.python.org/doc/current/tut/tut.html>
- ▶ Dive into Python –  
<http://www.diveintopython.org/>
- ▶ Python Brasil –  
<http://www.pythonbrasil.com.br/moin.cgi/DocumentacaoPython#head5a7ba2746c5191e7703830e02d0f5328346bcaac>
- ▶ Slides de Python – Rodrigo José Sarmiento Peixoto