



# Introdução à Programação

*Eduardo Silva Lira*

*XLVIII Programa de Verão do IME-USP*

*São Paulo - SP, Jan 2019*



INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
UNIVERSIDADE DE SÃO PAULO



# Sobre o professor

- Bacharel em Computação (2009 - 2013)
- Mestrado em Computação (2015 - 2016)
- Doutorando em Computação

# Requisitos para aprovação - Avisos importantes

- Não pratique **plágio!**
  - O que é plágio?!
- Caso enfrente alguma **dificuldade** ou **imprevisto**, entre em contato com o professor ou monitora.
  - Estamos aqui para atendê-los!

# Requisitos para aprovação - Avisos importantes

- Ao entregar as **listas escritas/impresas**:
  - Caso utilize alguma **fonte** para suas respostas, cite (site, livros, artigos, etc.).
  - A solução deve **conter os passos**, e não apenas o resultado final.

# Demais avisos

- Assuntos administrativos devem ser tratados na Secretaria do Programa de Verão - Sala 14 do bloco B do IME
  - email: [verao@ime.usp.br](mailto:verao@ime.usp.br)
- Para receber o certificado, é necessário satisfazer os critérios de aprovação no curso.
- Não serão aceitos alunos ouvintes.

# Demais avisos

- Alunos do Programa de Verão do IME podem utilizar o bandeirão da USP
  - Compre na tesouraria (bloco A), das 9h às 11h e das 13h às 16h, no valor de R\$10,00 (dez reais) cada.
- Caso seu nome não esteja na lista de presença:
  - Adicione seu nome no final da lista
  - Informe a Secretaria do Programa de Verão
- Alunos bolsistas: comparecer à Secretaria do Curso de Verão, sala 14-B, para retirar a ficha de acompanhamento. A assinatura da mesma será feita sempre na segunda-feira.

# Agora tem Wi-fi!

Utilize a rede wireless do programa de verão!



Rede: **verao**

Senha: **programaverao2019**

# Bibliografia

DEITEL, H.M. e DEITEL, P.J., "Como Programar em C", 2a ed., Livros Técnicos e Científicos, 1999.

ASCENCIO, A. F. G. e CAMPOS, E. A. V Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++. Prentice Hall. 2007.

FORBELLONE, A. L. V. e EBERSPACHER, H. F. Lógica de Programação – A Construção de Algoritmos e Estrutura de Dados. 3a Edição. Prentice Hall, 2005.

SOUZA, M. A. F. et al. Algoritmos e Lógica de Programação. C. Learning. 2008.



# **Revisão dos tópicos anteriores**

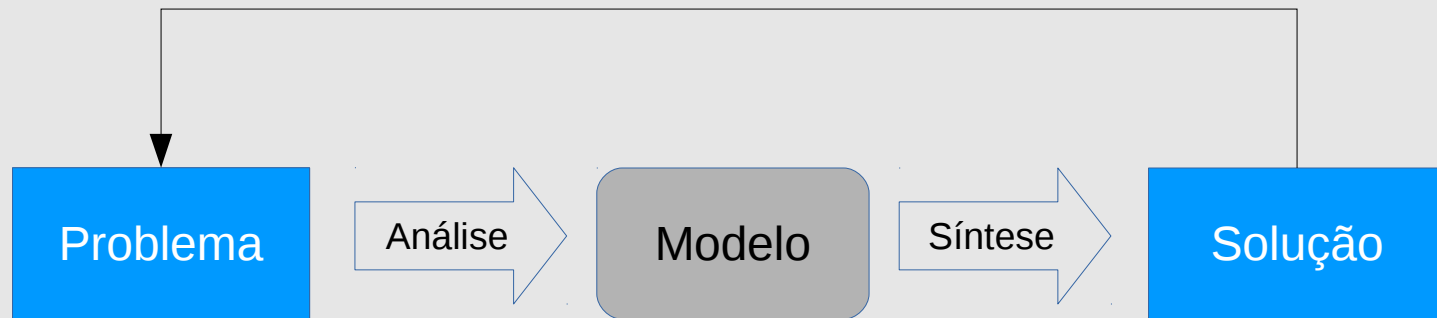
# Algoritmo

# Algoritmo

- Uma sequência finita de instruções bem definidas e não ambíguas
- Passos para realizar uma tarefa
- Não representa, necessariamente, um programa de computador

# Algoritmo

- Pense sobre o problema
  - Engenharia da solução



# Algoritmo

- Características:
  - Finitude
  - Exatidão
  - Efetividade

# Algoritmo

- Características:
  - Finitude
    - Número finito de passos
    - Quantidade finita de tempo
    - Quantidade finita de recursos
  - Exatidão
    - Definição dos passos com clareza e precisão
  - Efetividade
    - Cumprir o seu propósito

# Algoritmo

Quais algoritmos vocês fizeram na aula passada?

# Algoritmo

Outras sugestões:

1. Aprovação de crédito por um banco
2. Converter polegadas para centímetros
3. Dada uma lista de idades de pessoas, informar a idade da pessoa mais velha



# Algoritmo

Outras sugestões:

1. Aprovação de crédito por um banco
2. Converter polegadas para centímetros

**Entrada:** valor em polegadas

1. Devolva o valor em polegadas multiplicado por 2,54

3. Dada uma lista de idades de pessoas, informar a idade da pessoa mais velha

# Algoritmo

3. Dada uma lista de idades de pessoas, informar a idade da pessoa mais velha.
  - Encontrar o maior valor em uma lista de números naturais

**Entrada:** lista com  $n$  números naturais

1. Defina o máximo como zero
2. Para cada natural  $x$  da lista, compare com o máximo.  
Se for maior, atualize o máximo com o valor de  $x$ .
3. Devolva o máximo.

# Algoritmo

4. Responder se existe, no estoque da empresa, um parafuso com certa medida

# Algoritmo

4. Responder se existe, no estoque da empresa, um parafuso com certa medida

- Responder SIM ou NÃO para a pergunta: “existe um número igual a um certo valor em uma lista de números”?

**Entrada:** lista com **n** números reais, um valor **p**

1. Para cada número **x** da lista, compare com **p**.  
Se for igual, devolva SIM.
3. Devolva Não.

# Linguagem de programação

- O que é uma linguagem?

# Linguagem de programação

- O que é uma linguagem?
  - Linguagem de sinais
  - Linguagem escrita

*“**Linguagem** pode se referir tanto à capacidade especificamente humana para aquisição e utilização de sistemas complexos de comunicação, quanto à uma instância específica de um **sistema de comunicação complexo**”*

# Linguagem de programação

"...é um método **padronizado** para comunicar instruções para um computador."

- Permite especificar:
  - Os dados envolvidos
  - Ações a serem tomadas
  - Tratamento de diversas situações

# Linguagem de programação

"...é um método **padronizado** para comunicar instruções para um computador."

- Exemplos de linguagens de programação:
  - C
  - C++
  - Java
  - Python
  - Pascal
  - PHP
  - Fortran
  - Objective C



# Linguagem de programação

## Exemplo de código em Python

```
1 # -*- coding: utf-8 -*-
2 import pygame, sys, time
3 from pygame.locals import *
4 from menu import *
5 from game import *
6 from getInformacoes import *
7
8 pygame.init()
9 pygame.display.set_caption('Pegue o queijo!')
10
11 while True:
12     while True:
13         optionMenuPrincipal = menuPrincipal(DISPLAYSURF)
14         if (optionMenuPrincipal == 2): #SAIR
15             pygame.quit()
16             sys.exit()
```

# Linguagem Interpretada x Compilada

- Tradução do código fonte para código de máquina
- Interpretada
  - O programa é executado conforme vai sendo traduzido
- Compilada
  - Todo o código fonte é traduzido antes de ser executado.

# Paradigma

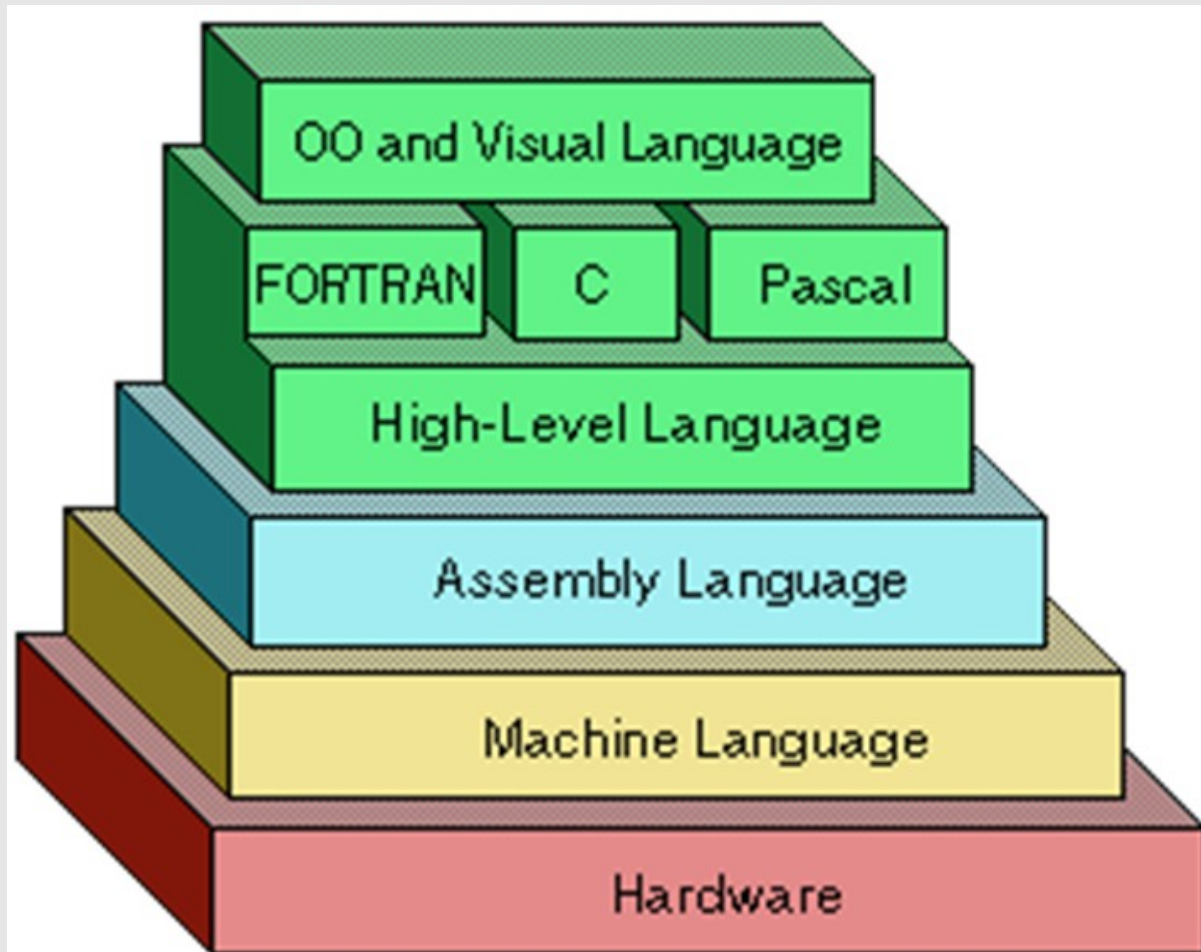
- o Estrutural

- Sequência, decisão e iteração
- Estruturas, como sub-rotinas e funções
- Resolver problemas mais simples e diretos
- C, Cobol, Pascal e Perl

- o Orientada a Objetos

- Modelar o mundo real no domínio do problema

# Alto x Baixo nível



# Alto x Baixo nível

- Baixo nível
  - Linguagem de máquina

**0s e 1s**

- Assembly

**LOAD   NOTA**

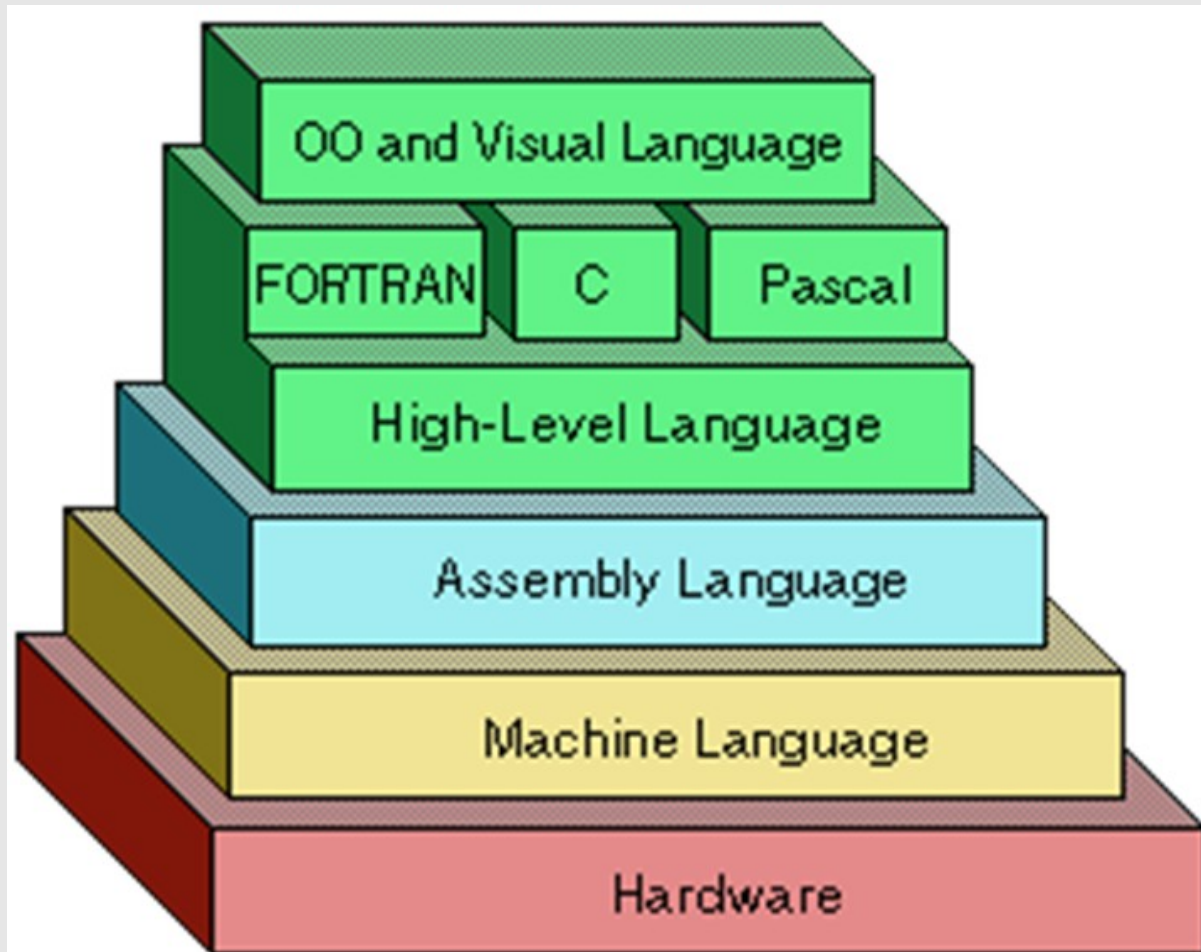
**ADD     NOTAEXTRA**

**STORE NOTAFINAL**

- Alto nível

**NOTAFINAL = NOTA + NOTAEXTRA**

# Alto x Baixo nível



# Prepare-se!

- Programas necessários:
  - Editor de código
  - Compilador

# Prepare-se!

## Ambiente **Windows**

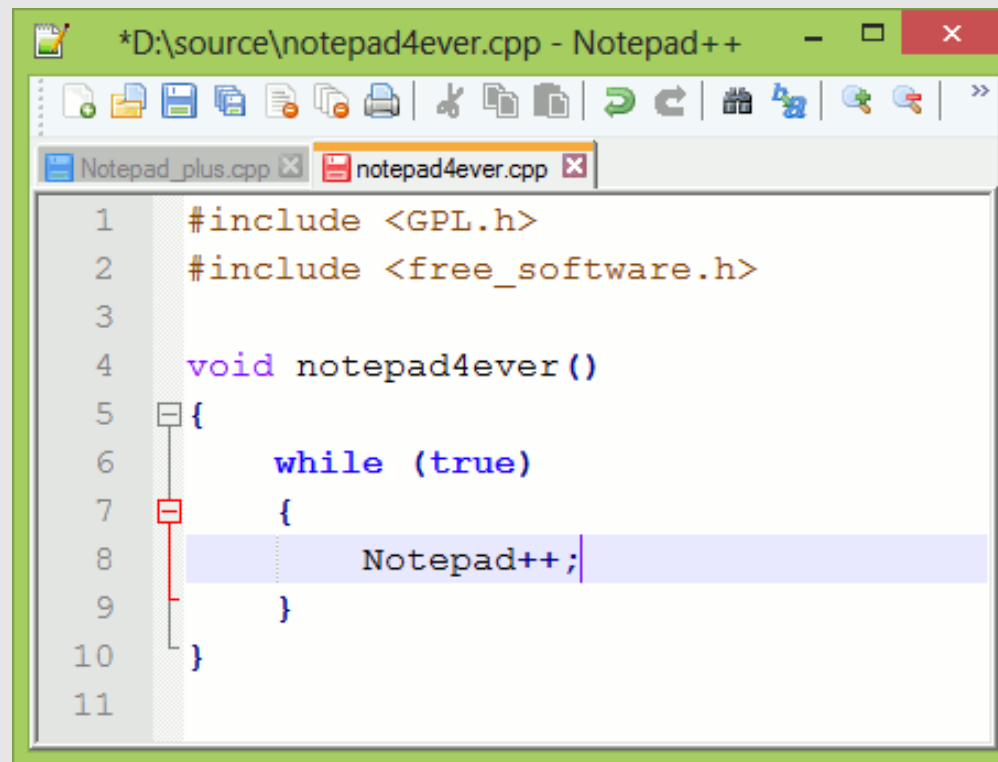
- Programas necessários:
  - Editor de código
    - **Notepad++** (<http://notepad-plus-plus.org/>)
    - Komodo Edit
    - Bloco de notas
  - IDE (*Integrated Development Environment*)
    - Dev-C++ (<http://www.bloodshed.net/devcpp.html>)
    - Code::Blocks (<http://www.codeblocks.org/>)
  - Compilador
    - **MingW** (*Minimalist GNU for Windows*)
    - Acesse <http://www.mingw.org/>.



# Prepare-se!

Ambiente **Windows**

**Notepad++** (<http://notepad-plus-plus.org/>)

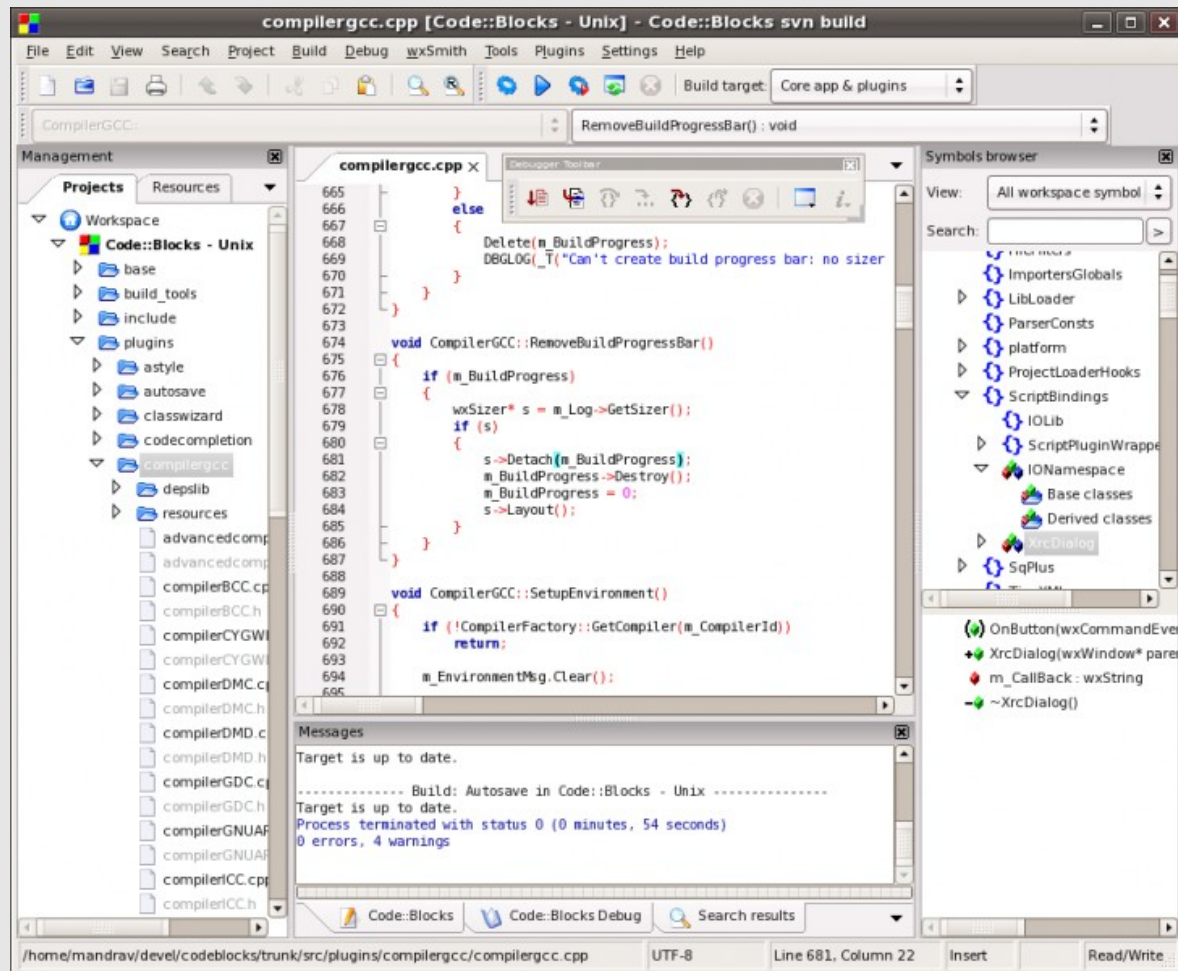


The screenshot shows the Notepad++ application window. The title bar reads '\*D:\source\notepad4ever.cpp - Notepad++'. The menu bar includes File, Edit, Format, View, and Plugins. The toolbar contains icons for opening, saving, printing, and other standard text editor functions. Two tabs are open: 'Notepad\_plus.cpp' and 'notepad4ever.cpp'. The 'notepad4ever.cpp' tab is active, displaying the following C++ code:

```
1  #include <GPL.h>
2  #include <free_software.h>
3
4  void notepad4ever ()
5  {
6      while (true)
7      {
8          Notepad++;
9      }
10 }
11
```

# Prepare-se!

Code::Blocks (<http://www.codeblocks.org/>) - disponível para Linux e Windows



# Prepare-se!

## Ambiente **Linux**

- Programas necessários:
  - Editor de código
    - **Gedit**
    - Kate
    - Vim
    - Emacs
  - IDE
    - Code::Blocks (<http://www.codeblocks.org/>)
  - Compilador
    - **Gcc**

# Prepare-se!

## **Vídeo Tutorial para Download do CodeBlocks**

Windows:

<https://www.youtube.com/watch?v=CiwPDUOvIMU>

Linux:

<https://www.youtube.com/watch?v=-rxhENQKCHw>

## **Ambiente online para programação C**

[https://www.tutorialspoint.com/compile\\_c\\_online.php](https://www.tutorialspoint.com/compile_c_online.php)

# Linguagem C

- Compilada, estruturada e imperativa
- Criada em 1972 (Dennis Ritchie), no AT&T Bell Labs
- Padrão ANSI C (C90)

[http://en.wikipedia.org/wiki/ANSI\\_C](http://en.wikipedia.org/wiki/ANSI_C)

# Nosso primeiro programa

- Abra o editor escolhido e vamos trabalhar!

**Dúvidas?**