

Linguagens de programação

Marco A L Barbosa
malbarbo.pro.br

Departamento de Informática
Universidade Estadual de Maringá



Este trabalho está licenciado com uma Licença Creative Commons - Atribuição-Compartilhável 4.0 Internacional.

<http://github.com/malbarbo/na-programacao>

Como descrever o que um computador deve fazer?

Usando uma linguagem.

Mas o que é uma linguagem e qual a sua função?

Um sistema estruturado para comunicação.

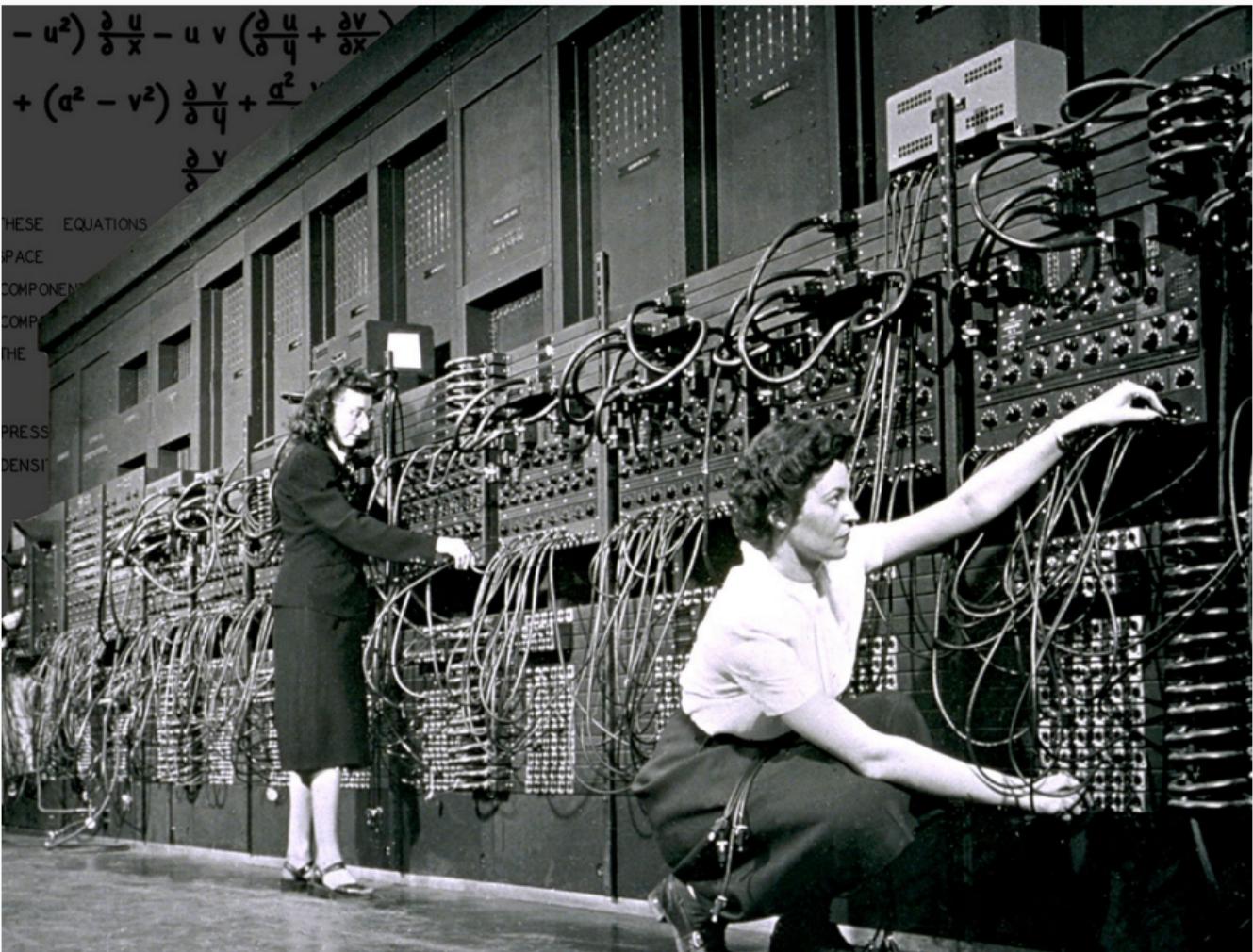
Podemos utilizar o português para programar um computador? Não, porque o computador não entende o português! Ainda que o computador entendesse o português, ele não seria uma linguagem adequada para programação porque não é preciso o bastante.

E que linguagem o computador entende? Cada tipo de computador entende um linguagem específica, a sua linguagem de máquina.

Então, precisamos programar o computador usando a linguagem de máquina? Hoje em dia não, mas os primeiros computadores eram programados usando linguagem de máquina!

$$-u^2) \frac{\partial u}{\partial x} - u v \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ + (a^2 - v^2) \frac{\partial v}{\partial y} + \frac{a^2}{\partial y} \\ \frac{\partial v}{\partial y}$$

THESE EQUATIONS
SPACE
COMPONENT
COMP
THE
PRESS
DENSI



Se não queremos utilizar linguagem de máquina, como podemos programar os computadores? Como duas entidades que falam linguagens diferentes podem se comunicar? Usando um tradutor.

Se a linguagem fonte e a linguagem alvo forem bem definidas, podemos escrever um programa que faça a tradução de forma automática. A linguagem de máquina (alvo) é bem definida, precisamos de uma linguagem fonte que também seja bem definida.

Uma **linguagem de programação** é uma linguagem formal (bem definida) adequada para programar computadores.

Uma linguagem de programação é caracterizada por diversos aspectos, entre eles

- A sintaxe e a semântica
- Nível de abstração
- Modo de execução

A **sintaxe** de uma linguagem é o conjunto de regras que define como os símbolos podem ser agrupados para criar as sentenças na linguagem.

Operação de soma com dois número:

- Na maioria das linguagens: $20 + 17$
- No Lisp: $(+ 20 17)$

A **semântica** de uma linguagem define o significado das construções da linguagem.

A expressão `12 + "3"` é válida sintaticamente em muitas linguagens, mas o significado é diferente entre elas

- Lua: significa que o texto "3" deve ser convertido para um número e depois somado com 12.
- Javascript: significa que o número 12 deve ser transformado para um texto e depois juntado com "3".
- Na maioria linguagens: é um erro semântico.

Uma linguagem de programação pode requerer (ou permitir) que o programador controle com mais detalhes ou menos detalhes o computador. Quanto mais detalhes são necessários, mais baixo nível é a linguagem, quanto menos detalhes, mais alto nível.

A programming language is low level when its programs require attention to the irrelevant.
(Alan Perlis)

Ser relevante ou não depende do contexto, por exemplo, para sistemas embarcados, controlar com detalhes o uso da memória pode ser bastante relevante, já para um *script* que será executado em um computador de trabalho, nem tanto.

Uma linguagem de programação pode ser compilada ou interpretada.

Em uma linguagem compilada o código do programa é primeiro traduzido (compilado) para código de máquina e posteriormente executado.

Em uma linguagem interpretada o código do programa é lido e executado diretamente pelo interpretador.

Características de programas compilados

- Execução eficiente
- Ciclo de desenvolvimento lento (tem que esperar o compilador)

Características de programas interpretados

- Execução menos eficiente
- Ciclo de desenvolvimento mais rápido (não precisa esperar o compilador)

Exemplo em C++ - Raiz quadrada

Parte do arquivo `raiz.cpp`

```
// Entrada
double n = stof(argv[1]);
double c = stof(argv[2]);

// Processamento
while (abs(c * c - n) >= 0.1) {
    double c_ = (c + n / c) / 2;
    c = c_;
}

// Saída
cout << c << endl;
```

Windows

Compilação

```
g++ -o raiz.exe raiz.cpp
```

Execução

```
> raiz.exe 4 1
2.00061
```

Linux

Compilação

```
g++ -o raiz raiz.cpp
```

Execução

```
> ./raiz 4 1
2.00061
```

Exemplo em Python - MDC

Parte do arquivo `mdc.py`

```
# Entrada
a = int(sys.argv[1])
b = int(sys.argv[2])
# Processamento
if a < b:
    a, b = b, a
while b != 0:
    c = a - b
    a = b
    b = c
    if a < b:
        a, b = b, a
# Saída
print(a)
```

Interpretação

```
> python3 mdc.py 52 36
4
```

Existem centenas de linguagens de programação, cada uma com suas próprias características e usos, então, qual delas aprender?

Nenhuma em particular. O mais importante não são as linguagens em si, mas os princípios que são utilizados para criar os programas nas linguagens.

Dominando os fundamentos de algoritmos e paradigmas de programação, aprender uma nova linguagem não é uma tarefa difícil!

Mas precisamos de uma linguagem para começar. Nessa disciplina vamos utilizar a linguagem C++!

C++ é uma das linguagens mais poderosas que existem, mas também uma das mais complicadas! Mas não se preocupe, vamos utilizar apenas construções básicas!

Vamos começar?

1. Tente relacionar as construções do fluxograma do algoritmo que calcula a raiz quadrada com construções do código em C++.
2. Tente relacionar as construções do pseudo código que calcula o MDC com construções do código em Python.