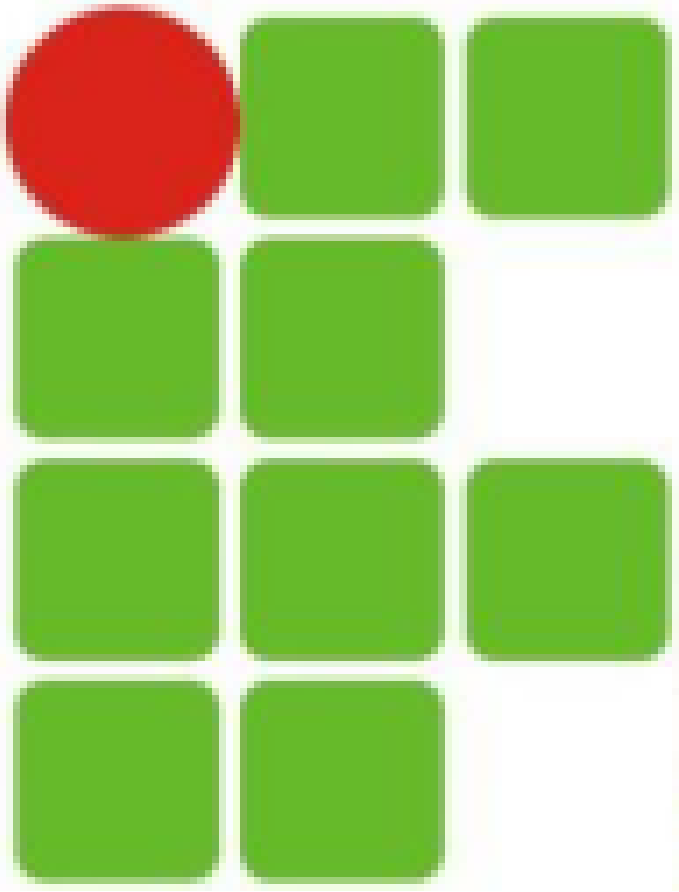


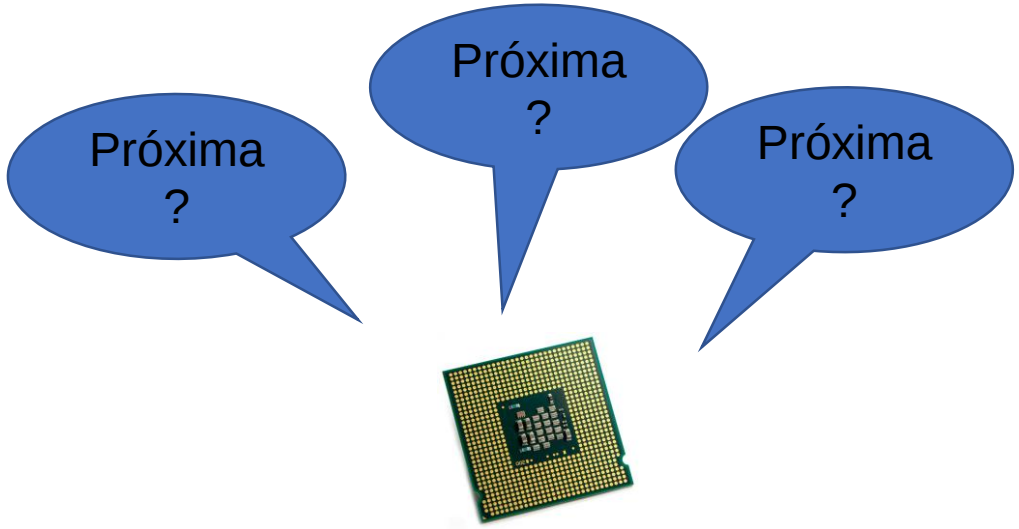
Introdução à Programação de Computadores

Galileu Batista de Sousa

Galileu.batista -at +ifrn -edu +br

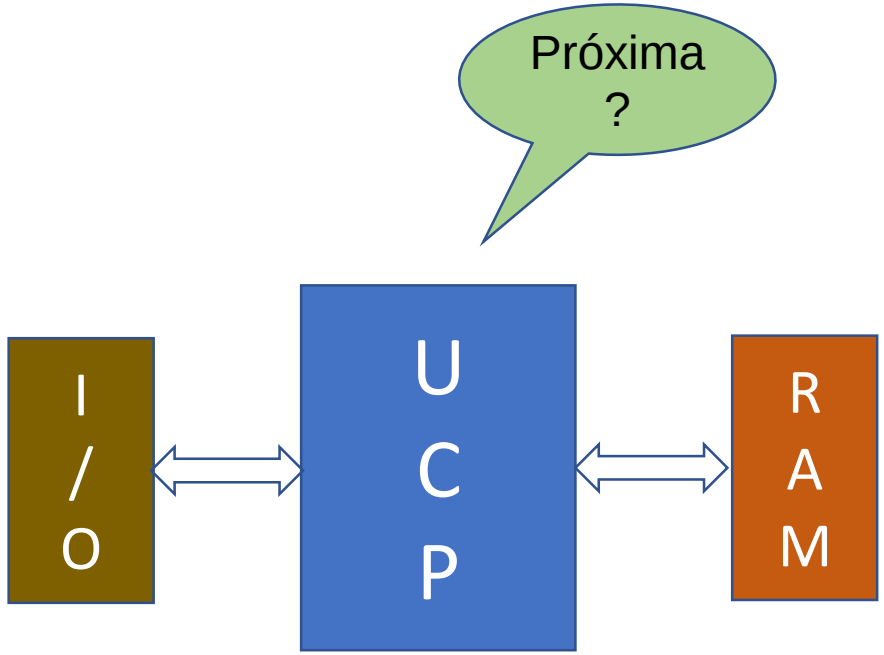
Computadores e programas

- Computadores são construídos para automatizar tarefas
- A questão é? Como dizer ao computador o que fazer?
- Os usuários não se preocupam: alguém já colocou as instruções lá (os programas). É só usar.



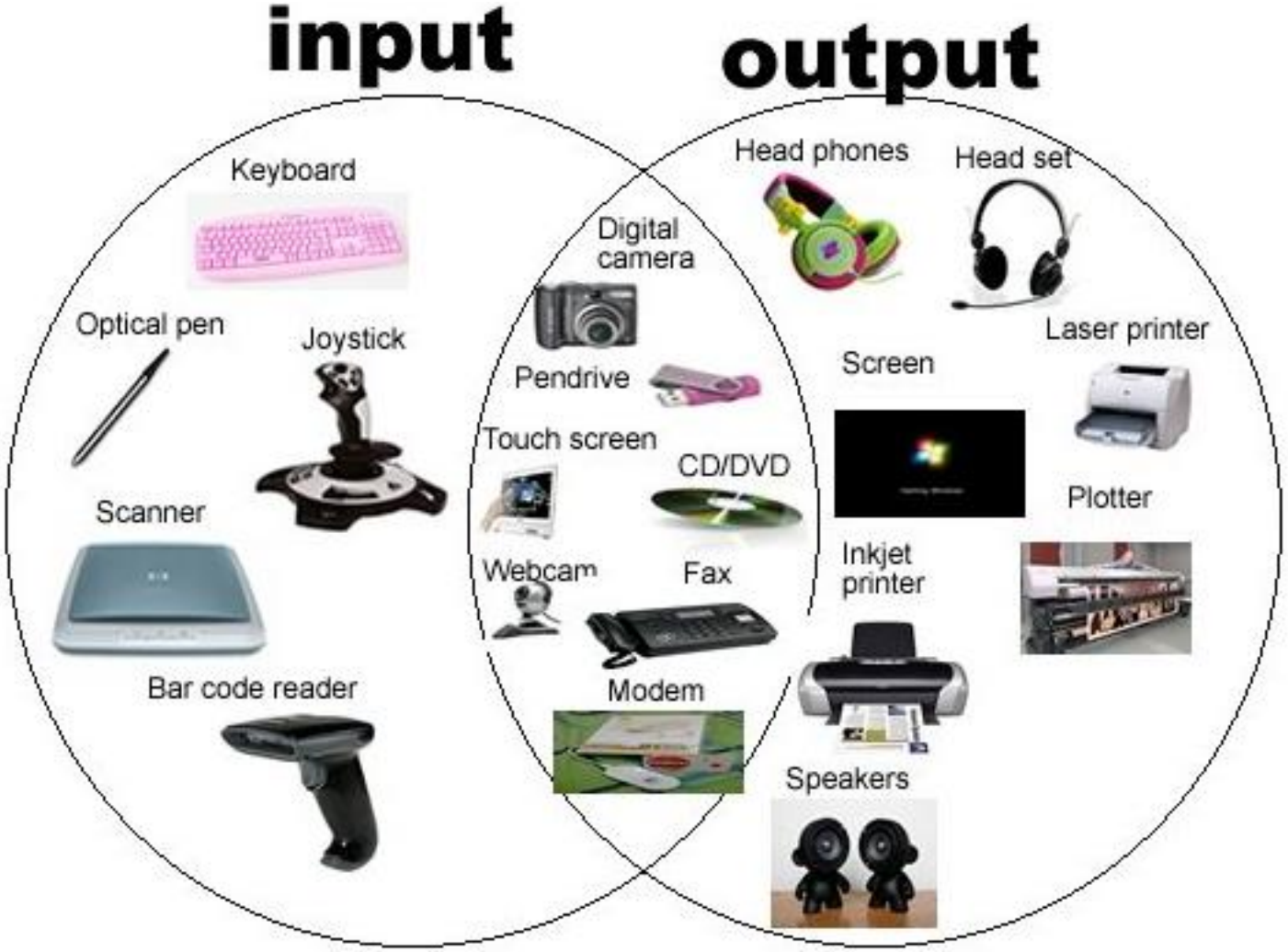
Computadores

- **UCP**: Unidade Central de Processsamento
  - próxima? próxima? ....
- **RAM**: Memória de acesso randômico
  - rascunho: de onde vêm:
    - O que fazer?
    - Sobre quem fazer?
    - Onde guardar os resultados
- **I/O**: discos, mouse, teclado, vídeo...
  - Interface com o mundo exterior



Entrada e saída (I/O)

- Muitas vezes a UCP é só intermediária entre:
  - I / O e RAM
- **Próxima**?
  - Ler input ➡ memória
  - Ler memória ➡ output



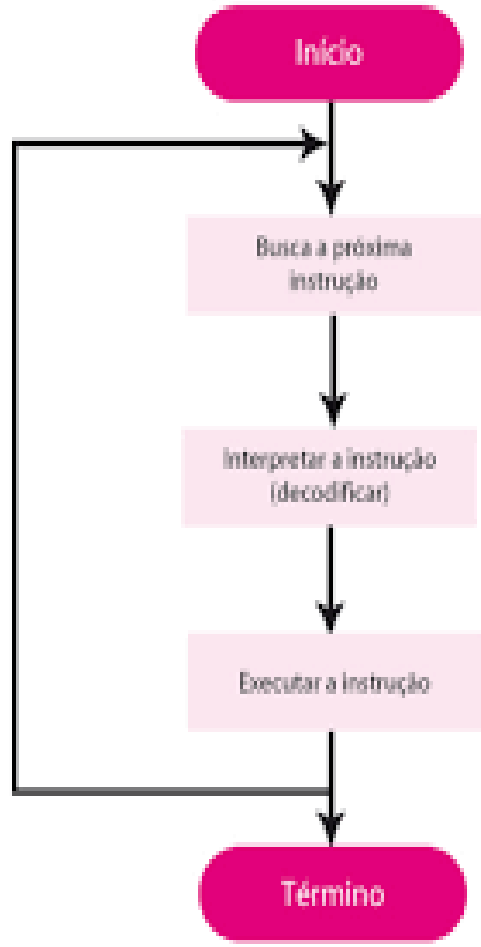
Memória (RAM)

- O local onde guardar:
  - Instruções
  - Operandos
  - Resultados
- Cada posição tem um endereço
- Não resiste à falta de energia

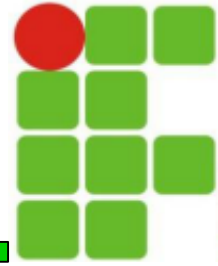
00	dado/inst
01	dado/inst
02	dado/inst
03	dado/inst
04	dado/inst
05	dado/inst
06	dado/inst
07	dado/inst
....	....
....	dado/inst
....	dado/inst
....	dado/inst
....	dado/inst

UCP – como 'instruí-la'

- A UCP quer fazer (Próxima?), mas:
  - Sabe fazer poucas coisas
    - Operações aritméticas: add, sub, ..., mult, div
    - Operações lógicas: and, or, not
    - Operações relacionais: eq, gt, gte, low, ...
    - Movimentação: mov, in, out
  - **Por padrão executa em sequência**
  - **Permite execução condicional**
    - A posição da próxima instrução depende de uma condição (em geral, uma comparação)



Instruções da UCP



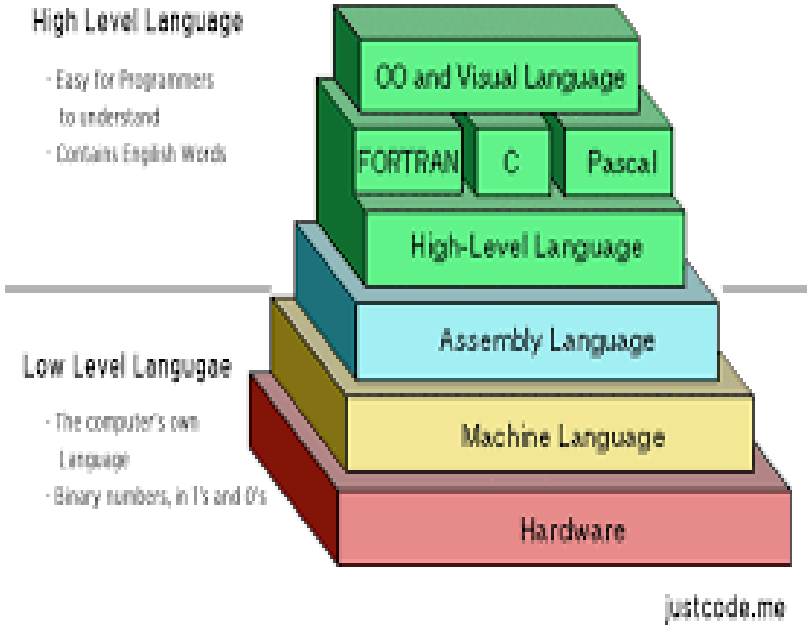
- São de muito baixo nível
  - Instruções primitivas
  - Tem que informar a posição de memória:
    - Das instruções,
    - Dos operandos,
    - Do resultado
- Melhor usar algo de mais sofisticado
  - Uma linguagem de programação de alto nível
- Problema: a UCP não entende ...

Nível das linguagens

- Linguagem de máquina: 0's e 1's (valores binários)
  - Difícil de escrever, difícil de entender
  - Muitas instruções para fazer algo simples

- Linguagem de montagem (*assembly*):
  - Dá um nome para instrução de 0's e 1's

- Linguagens de alto-nível (+humana)
  - Cada comando, muitas instruções
  - Python, C/C++, Java, PHP, C#, Ruby, ...



Programa - definição

- Um conjunto **finito** e **não-ambíguo** de **passos** (instruções) para resolver um problema.

- Usamos "programas" para atividades do dia-a-dia, mas:
  - Nossa língua, tende a ser:
    - Ambígua
    - Prolixa
    - Redundante
    - Contexto envolvido

Compiladores x Interpretadores

- **Compiladores**:
  - Convertem um programa em linguagem de alto nível nas instruções da máquina
    - Realizam a atividade uma vez só.
    - É como um tradutor de livros.

- **Interpretadores:**
  - Convertem cada comando de alto nível em instruções da máquina
    - É como um intérprete.

Erros em programas

- **Erros sintáticos**:
  - O compilador/interpretador não entende o programa
    - **(8 + 5) * 3 -**
- **Erros de execução:**
  - Só são verificáveis executando o programa
    - **(5 + 3*4) / (12 – 2 * 6)**
- **Erros semânticos:**
  - Programa executa, mas não faz o que deveria
    - **Vá a escola todos os dias, até chegar o dia 30/02**

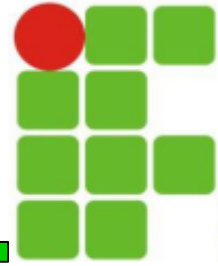
Python

- É uma linguagem de programação interpretada:

  Python deixa você trabalhar mais rapidamente e integrar seus sistemas mais efetivamente.

- Projetada por *Guido van Rossum* nos anos 1990.
  - Disponível em várias plataformas
  - Código aberto
  - Versões 2.x e 3.X

Como usar python (python3)

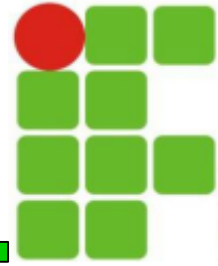


```
c:\Users\geral\Desktop>python
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

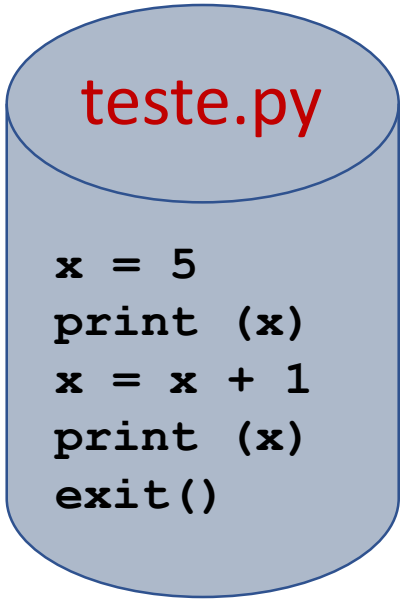
Próxima ?

```
c:\Users\geral\Desktop>python
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> x = 5
>>> print (x)
5
>>> x = x + 1
>>> print (x)
6
>>> exit()
c:\Users\geral\Desktop>
```

Python - Interativo *vs* scripts



- Além do uso interativo, é possível colocar os comandos em python num arquivo (***.py***)
  - Executar o interpretador passando o nome do arquivo




```
c:\Users\geral\Desktop>python teste.py
5
6
c:\Users\geral\Desktop>
```