

**Universidade do Vale do Paraíba  
Colégio Técnico Antônio Teixeira Fernandes**

**Introdução as Linguagens de Programação  
(POO-II)**

***Prof. Responsável  
Wagner Santos C. de Jesus***

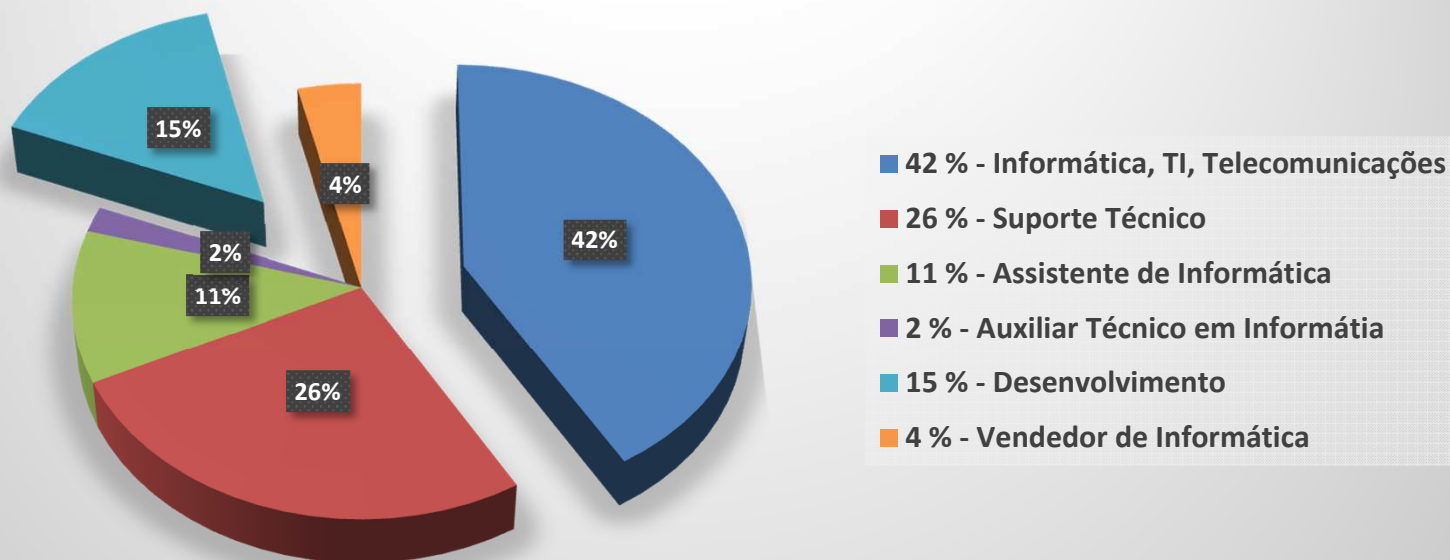
***[www1.univap.br/wagner](http://www1.univap.br/wagner)  
[wagner@univap.br](mailto:wagner@univap.br)***

1. Conceito de Linguagem de Programação;
2. Definição de Linguagem de Programação;
3. Motivação para uso de uma Linguagem de Programação;
4. Compiladores ou Interpretadores;
5. Origem de Algumas Linguagens;
6. Programação para Microcontroladores;
7. Linguagem para Desenvolvimento de Jogos;
8. Linguagens API (Interface de Aplicação de Programas);

- Banco de Dados
- Modelagem de dados
- Mineração de dados
- Geoprocessamento
- Arquitetura de dados
- Administração de dados

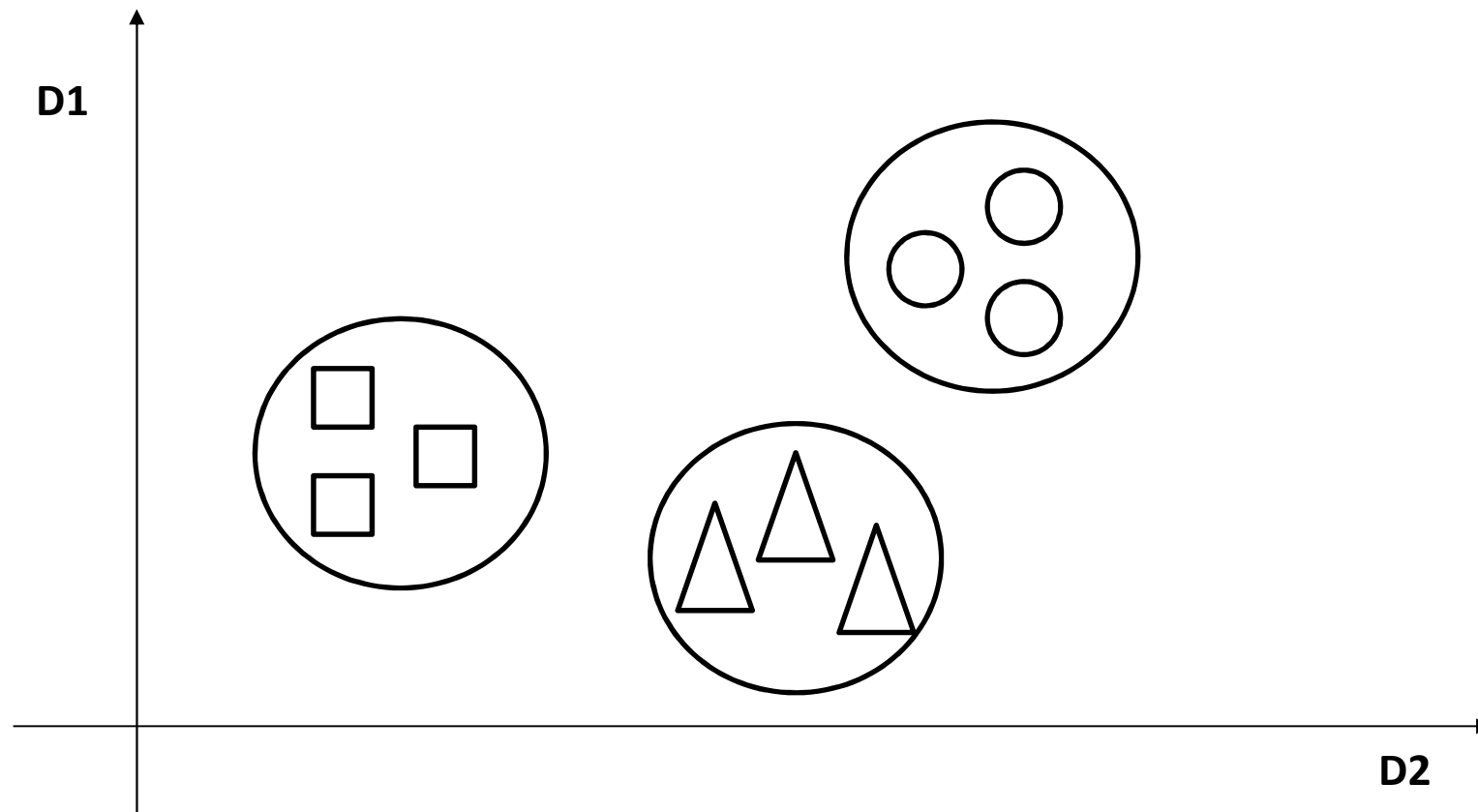


## Ofertas de empregos para Informática



Disponível em: <<https://www.google.com/search?ibp=htl;jobs&hl=pt-BR&kgs=b0ef31d4331047e6&q=empregos+em+informatica&shndl=-1&source=sh/x/im/textlists/detail&entrypoint=sh/x/im/textlists/detail#fpstate=tl&detail&htidocid=3dfzAnhkFPYhCiRAAAAAA%3D%3D&htiq=empregos%20em%20informatica&htivrt=Jobs>>. Acesso em 22 de Jan. 2019.

## Técnica de Mineração de Dados (CLUSTER)



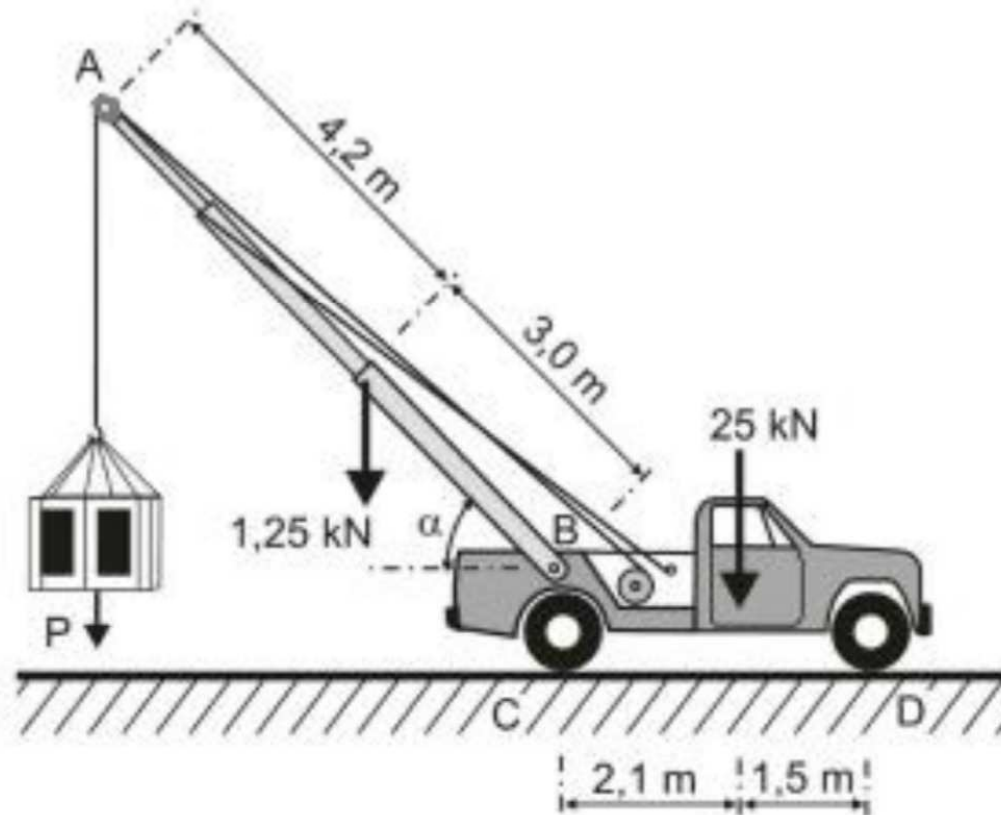
# Conceito de Linguagem de Programação (LP)

Conhecimentos necessário para o aprendizado de programação:

- Matemática (Estatística)
- Linguagens naturais (Inglês, Português ..).
- Comunicação verbal e interpretativa.



04. No guindaste montado no caminhão, determine o menor ângulo  $\alpha$  para que o caminhão não tombe quando a carga  $P$  de 15 kN for levantada.



## Alunos

## Pergunta dos alunos



Mas professor,  
onde vou usar  
isso na minha  
vida ??

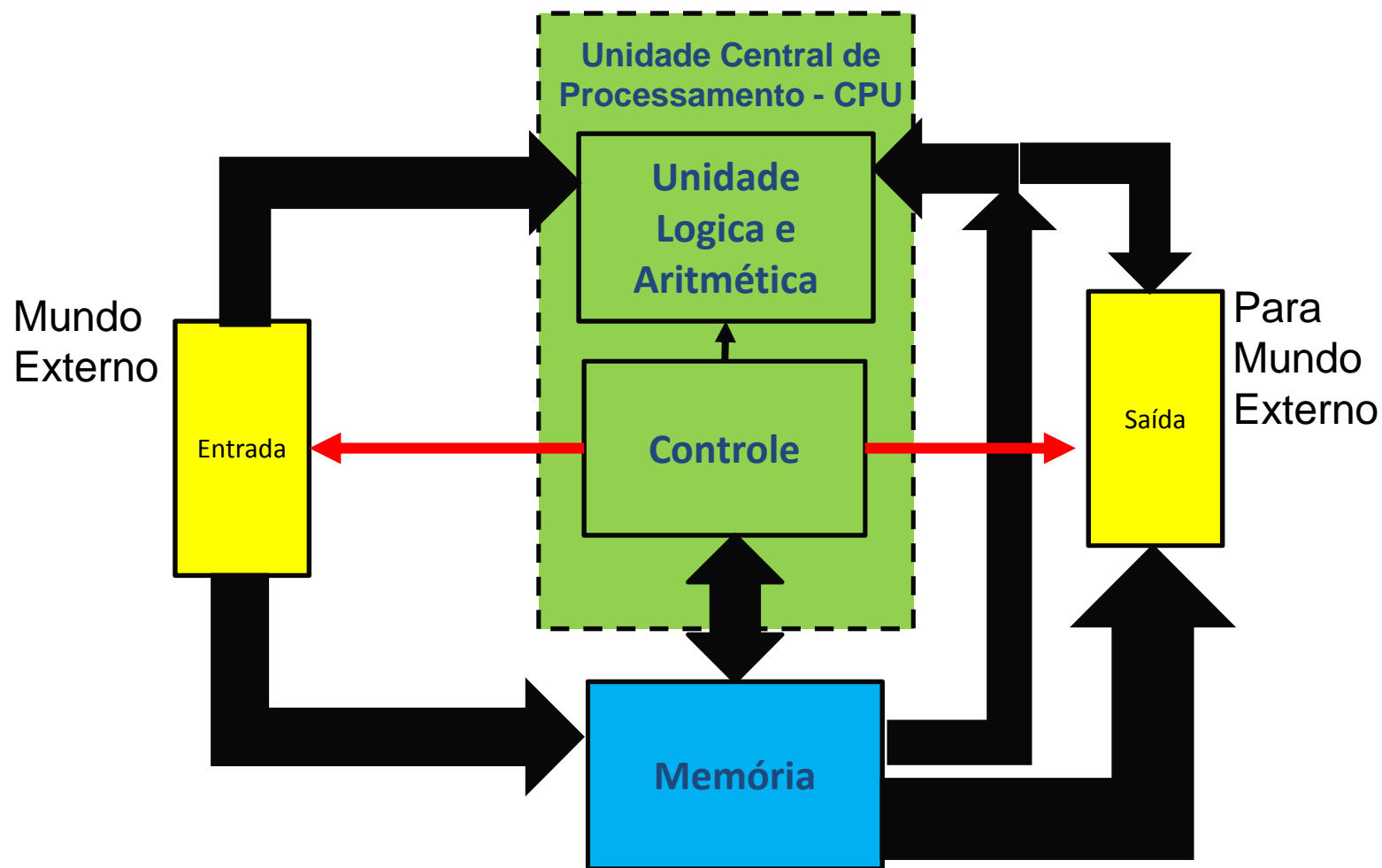


# Máquina de Von Neumann

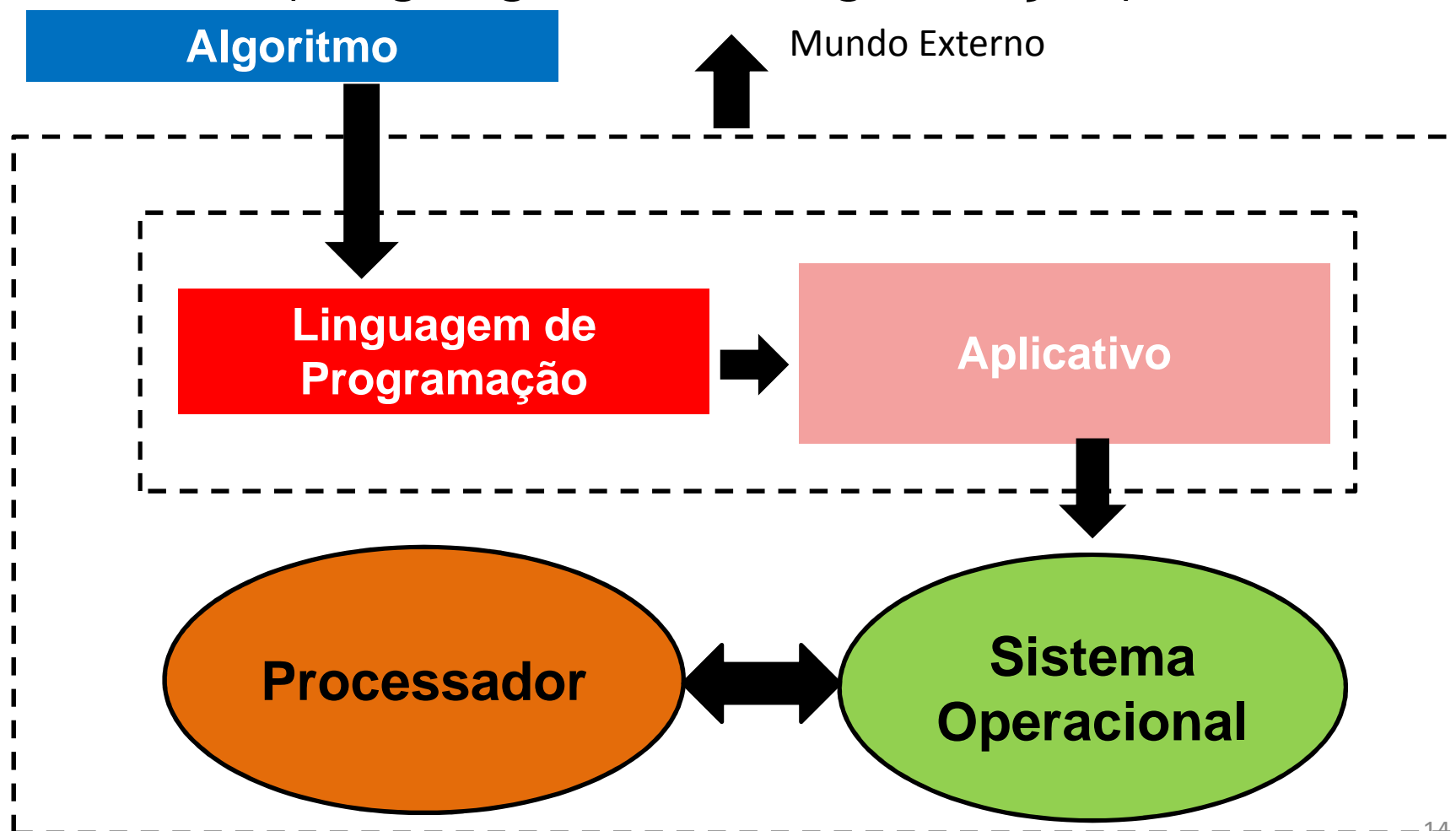


**John Von Neumann**,  
nascido **Margittai Neumann**  
**János Lajos** (Budapeste, 28  
de dezembro de 1903 —  
Washington, D.C., 8 de  
fevereiro de 1957) foi  
um matemático húngaro de  
origem judaica,  
naturalizado estadunidense.

## Proposto por Von Neumann



## Função de uma LP (Linguagem de Programação)



# Definição de Linguagem de Programação

# Definição

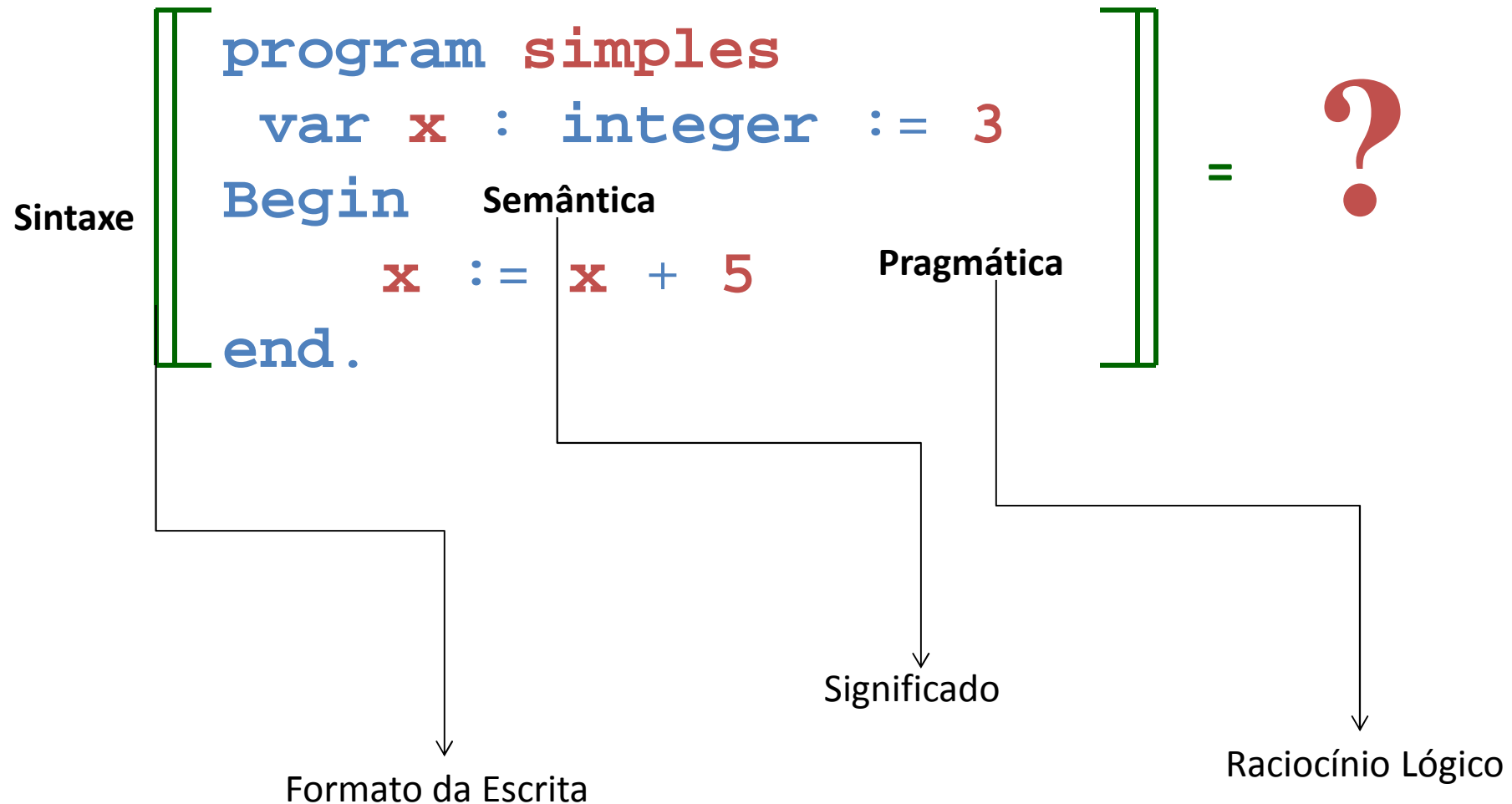
Uma **linguagem de programação** é um método padronizado para comunicar instruções para um computador. É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador. Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.



# Característica principais de uma LP

- Sintaxe
- Semântica
- Pragmática

# Exemplo de Características de uma LP



# Motivação para uso de uma Linguagem de Programação

# Porque usar uma LP

Uma das principais metas das linguagens de programação é que programadores tenham uma maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado com a linguagem que um computador entende nativamente (código de máquina).

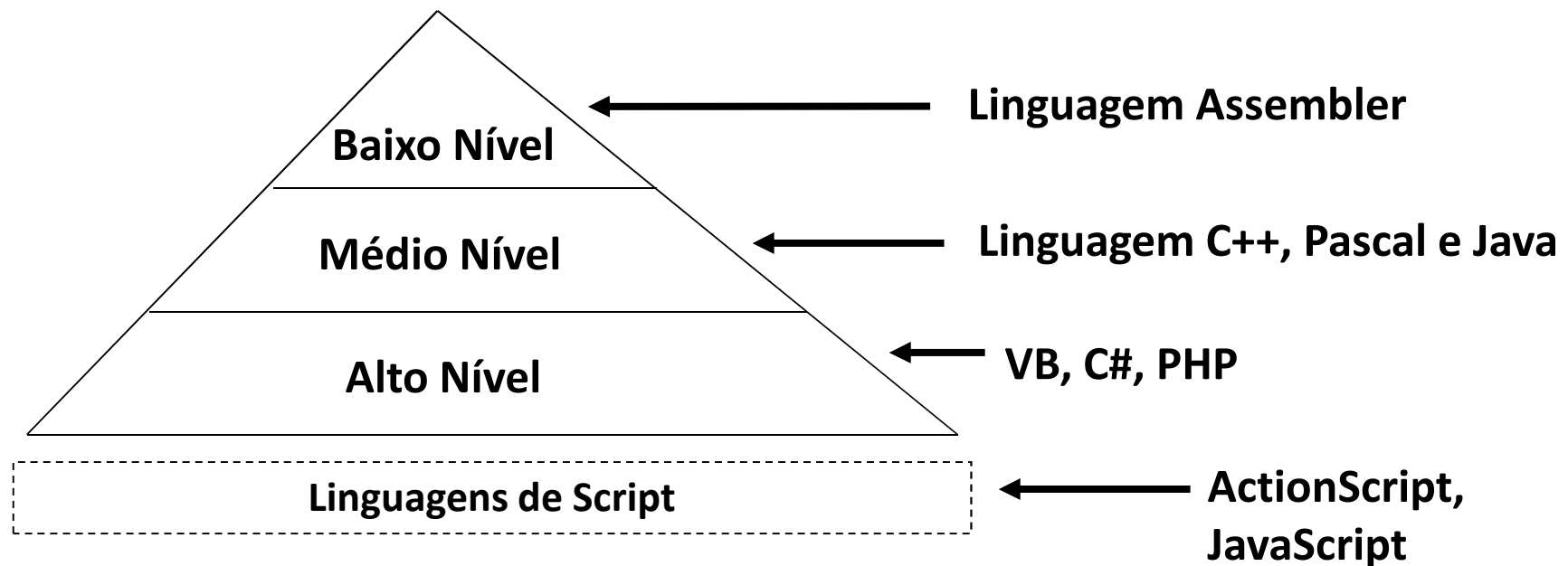
# Mapeamento Hexadecimal

```

CODE:00404DFF 0F 85 C4 00 00 00 jnz loc_404EC9
CODE:00404E05 68 D4 4E 40 00 push offset LibFileName ; "DbdDevAPI.dll"
CODE:00404E0A E8 C9 EB FF FF call LoadLibraryA
CODE:00404E0F A3 20 B1 40 00 nov ds:hModule, eax
CODE:00404E14 83 3D 20 B1 40 00+ cmp ds:hModule, 0
CODE:00404E1B 0F 84 A8 00 00 00 jz loc_404EC9
CODE:00404E21 68 E4 4E 40 00 push offset aDbddevopen_0 ; "DbdDevOpen"
CODE:00404E26 A1 20 B1 40 00 nov eax, ds:hModule
CODE:00404E2B 50 push eax ; hModule
CODE:00404E2C E8 77 EB FF FF call GetProcAddress
CODE:00404E31 A3 04 D3 40 00 nov ds:DbdDevOpen, eax
CODE:00404E36 68 F0 4E 40 00 push offset aDbddevclose_0 ; "DbdDevClose"
CODE:00404E3B A1 20 B1 40 00 nov eax, ds:hModule
CODE:00404E40 50 push eax ; hModule
CODE:00404E41 E8 62 EB FF FF call GetProcAddress
CODE:00404E46 A3 08 D3 40 00 nov ds:DbdDevClose, eax
CODE:00404E4B 68 FC 4E 40 00 push offset aDbddevgetinfo ; "DbdDevGetInfo"
CODE:00404E50 A1 20 B1 40 00 nov eax, ds:hModule
CODE:00404E55 50 push eax ; hModule
CODE:00404E56 E8 4D EB FF FF call GetProcAddress
CODE:00404E5B A3 0C D3 40 00 nov ds:DbdDevGetInfo, eax
CODE:00404E60 68 0C 4F 40 00 push offset aDbddevregistercallback_0 ; "DbdDevRegisterCallback"
CODE:00404E65 A1 20 B1 40 00 nov eax, ds:hModule

```

# Classificação das LP



# Objetivo prático

Linguagens de programação são ferramentas importantes para que programadores e engenheiros de software possam escrever programas mais organizados e com maior rapidez.

# Evolução das Linguagens de Programação



Primeira geração - linguagem de máquina

Segunda geração - linguagens de montagem (*assembly*).

Terceira geração - Linguagens procedurais.

Quarta geração - Linguagens aplicativas

Quinta geração - Linguagens voltadas a Inteligência artificial como as linguagens lógicas (Prolog) e as linguagens funcionais (Lisp).



# Classificação da Linguagem Programação

- Genéricas
- Específicas

# Genérica

Possibilitam o desenvolvimento para qualquer tipo de aplicação. (Científica e ou Comercial).

---

Exemplo (Java, C++, C# e Basic)

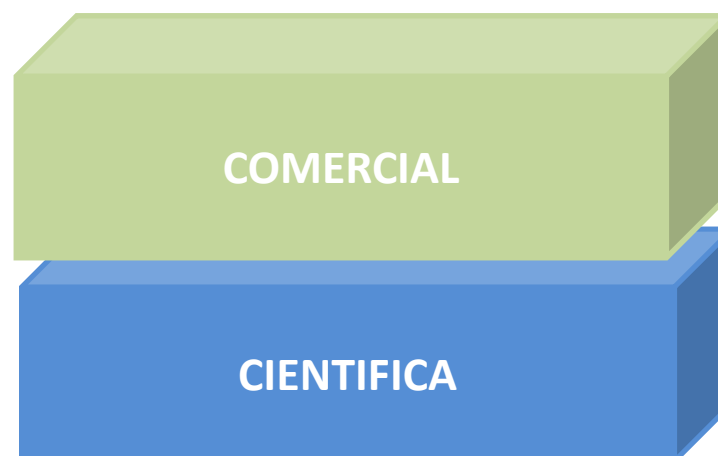
# Específica

Possibilitam o desenvolvimento para qualquer um tipo de aplicação.

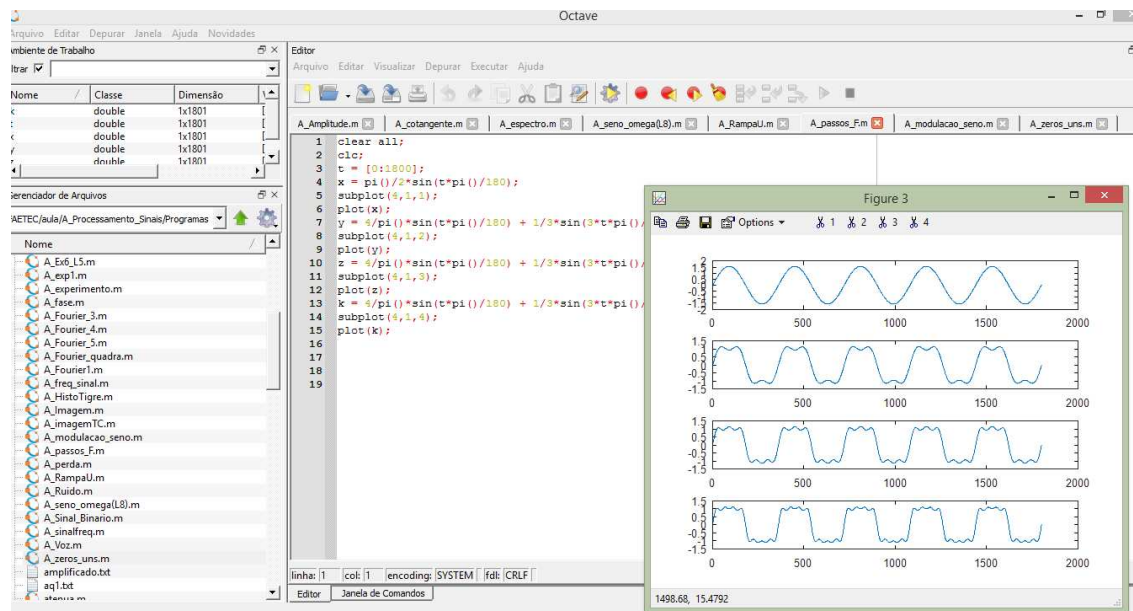
---

Exemplo (Php, ActionScript, CSS e HTML)

# Áreas de atuação de uma Linguagem de Programação



# Exemplo de Linguagem Científicas



O **Octave** originalmente foi criado por volta de 1988, com o propósito de ser um software de apoio a um livro de graduação em projetos de reator químico, escrito por James B. Rawlings da Universidade de Wisconsin-Madison e John G. Ekerdt da Universidade do Texas. Originalmente foi idealizado como ferramenta muito especializada relacionado à criação de reatores químicos. Posteriormente, após constatar as limitações dessa abordagem, optou-se pela construção de uma ferramenta mais flexível.

# Exemplo de Linguagem Comercial

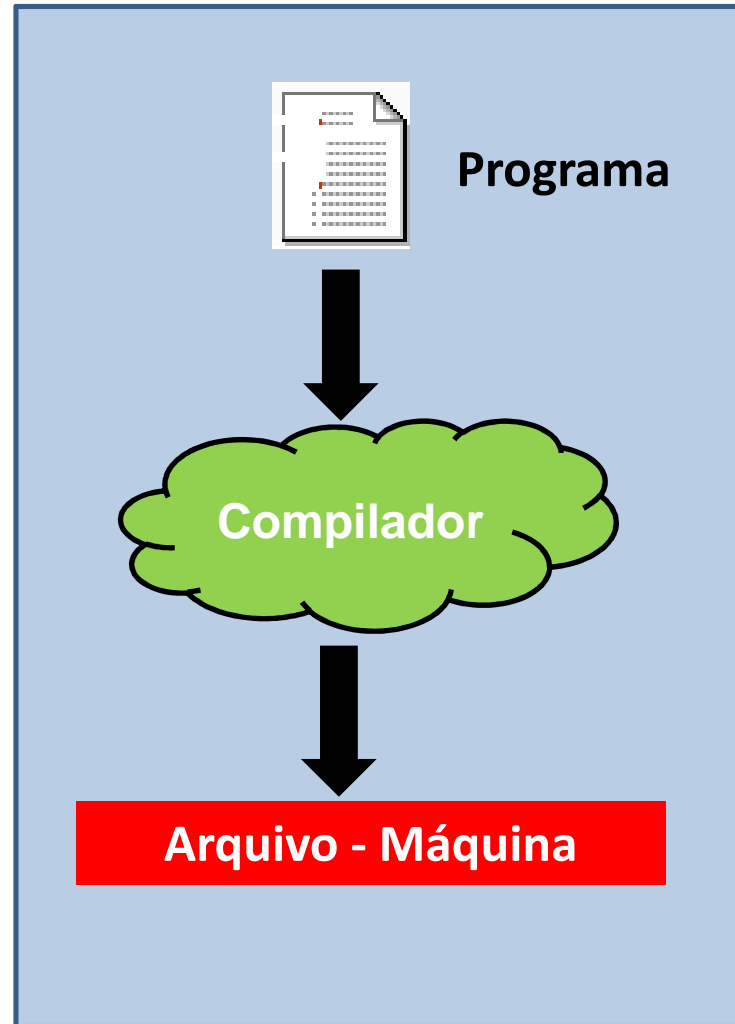
**COBOL** (sigla de ***CO**mmun **B**usiness **O**riented **L**anguage*) - Linguagem Comum Orientada para os Negócios é uma linguagem de programação orientada para o processamento de banco de dados comerciais.

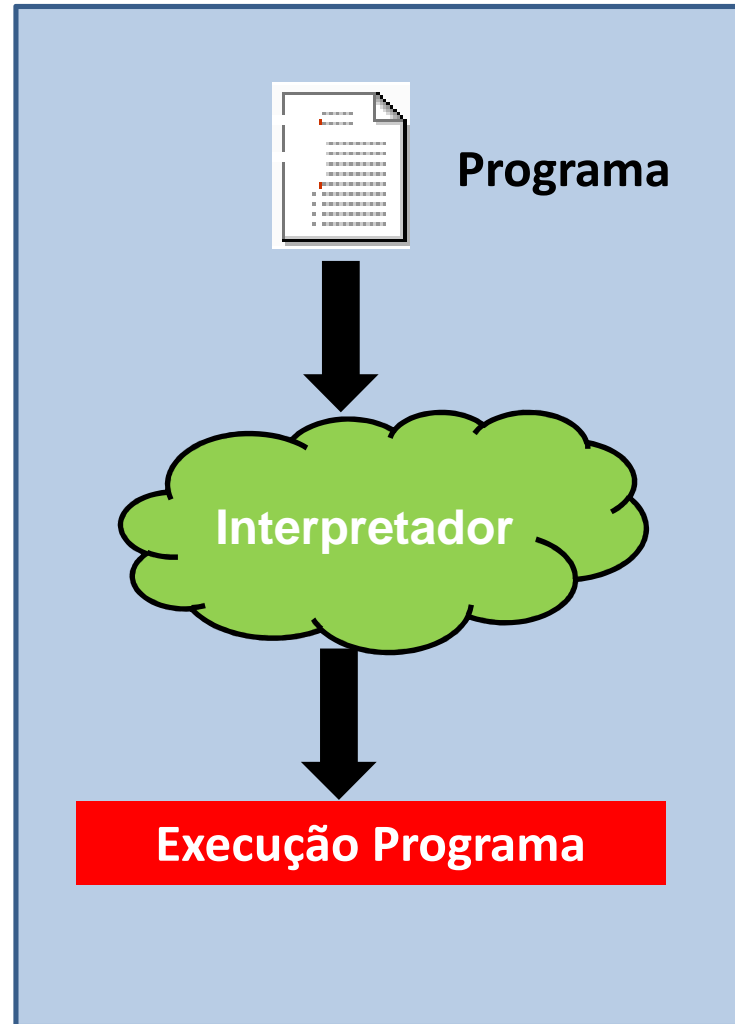
- 1 - IDENTIFICATION DIVISION**
- 2 - ENVIRONMENT DIVISION**
- 3 – DATA DIVISION**
- 4 – PROCEDURE DIVISION**

# Funcionamento das Linguagens de Programação

# LP – Compiladas ou Interpretadas







# Origem de Algumas Linguagens

# Criador da LP (Pascal)



Criada Tendo em mente encorajar o uso de código estruturado.

Niklaus Wirth 1970 (Suíço)

```
program Soma_Pares;  
uses crt;  
var superior, soma, num: integer;  
begin  
    soma:=0;  
    write ('Entre com o limite superior');  
    readln (superior);  
    num:=2;  
    repeat  
        soma:=soma+num;  
        num:=num+2;  
    until (num > superior);  
    writeln('A soma dos números pares de 2 até ', superior, ' é ', soma);  
    readln;  
end.
```



Linguagem criada para desenvolvimento do kernell do sistema operacional Unix

**Ken Thompson e Dennis Ritchie (da esquerda pra direita)**

```
main(){
    int superior, soma, num, i;
    soma = 0;
    printf("Entre com limite Superior:");
    scanf("%d",&superior);
    num = 2;
    do {
        soma += num;
        num += 2;
    } while(num <= superior);

    printf("Soma dos numeros pares de 2 ate %d = %d \n",superior,soma);
    printf("----- Fim -----");
    int d;
    scanf("%s",&d);
}
```

# A Linguagem Java

- Desenvolvida na década de 90 por uma equipe de programadores chefiada por **James Gosling**, na empresa Sun Microsystems.

Os mentores do projeto eram Patrick Naughton, Mike Sheridan, e James Gosling.



**Programador Canadense**  
**Foto 2005**



```
import java.util.Scanner;
public class Somapar {
    public static void main(String args[]){
        int superior, soma, num, i;
        soma = 0;
        Scanner entrada = new Scanner(System.in);
        System.out.println("Entre com limite Superior:");
        superior = entrada.nextInt();
        num = 2;
        do {
            soma += num;
            num += 2;
        } while(num <= superior);

        System.out.printf("Soma dos numeros pares de 2 ate %d = %d \n",superior,soma);
    }
}
```

# Linguagens Residentes (Embarcadas)



CP 200

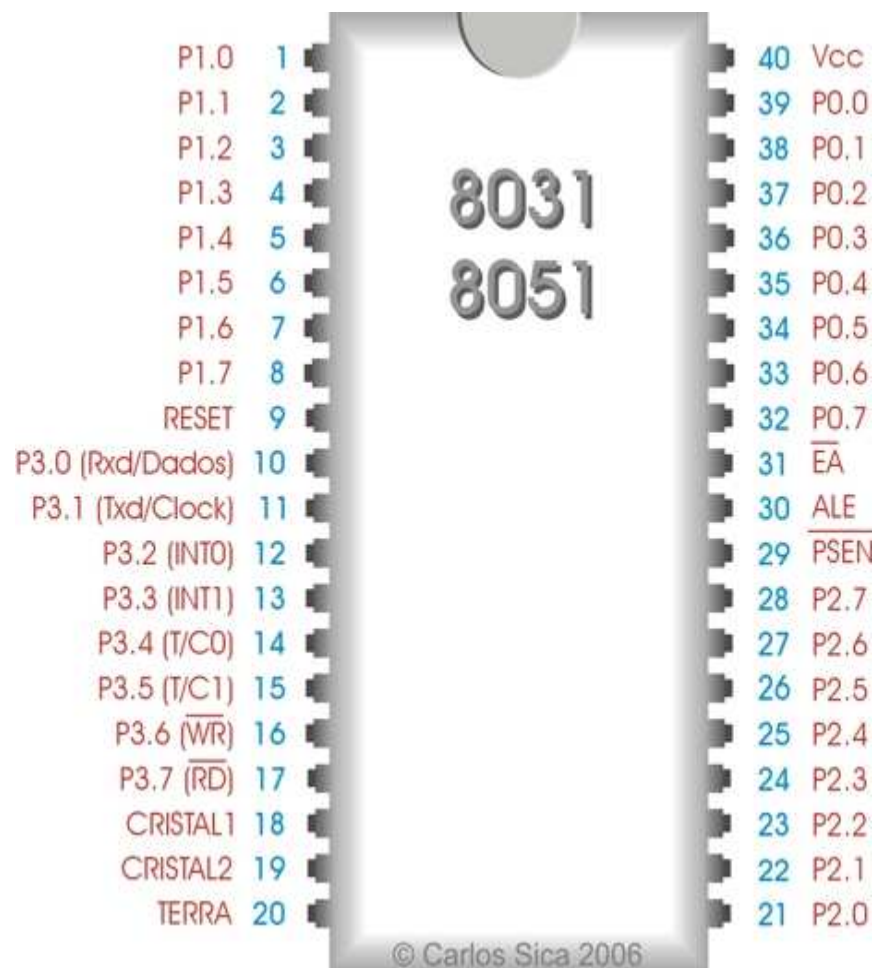


CP 500



TK 90X

# Programação para Microcontrolador



```
#include <8051.h>
void main(void) {
    while(true) {
        if(P1_0 == 1)
        {
            P2_0 = ~P2_0;
        }
    }
}
```

# Exemplo de Linguagem para Desenvolvimento de Jogos



```
onClipEvent (enterFrame) {
    if(Key.isDown (Key.RIGHT )){
        _root["Atirador"]._x+=10
    }

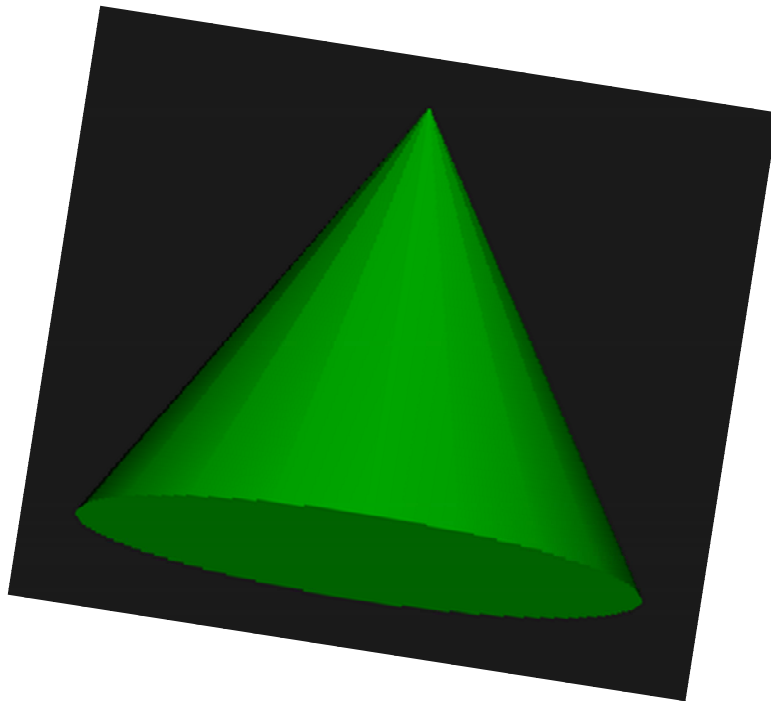
    if(Key.isDown (Key.LEFT )){
        _root["Atirador"]._x-=10
    }

    if(Key.isDown (Key.SPACE)) {
        _root["Tiro"]._x = _root["Atirador"]._x
        _root["Tiro"]._y = _root["Atirador"]._y

        _root["Tiro"]._visible = true;
        _root.pontos = 140;
        //_root.controle_tiro = 1;
    }
}
```

# Linguagens API (Interface de Aplicação de Programas)





```
class Cone {

    public static void main(String[] args) {

        Navigation.NavigationInfo("Cone");

        Materialpr material = new Materialpr();
        material.setdiffuseColor("0 1 0");
        BaseCirclepr basecircle = new BaseCirclepr();

        basecircle.setheight("2");
        basecircle.setbottomRadius("1");
        basecircle.setside("TRUE");
        basecircle.setbottom("TRUE");

        Conepr cone = new Conepr (material,basecircle,"Cone");

        Viewpr.display("Cone");

    }
}
```