Iniciando

SUMÁRIO

- Software, Programa e Algoritmo
- Tipos de Software
- Algoritmos
- Linguagem de Programação
- Ciclo de Vida do Software

Algoritmo, Programa e Software

- **Um algoritmo descreve a sequência de passos de um programa em uma linguagem próxima de um idioma, como o português**

```
Algoritmo para dividir dois valores numéricos
1-Receba o dividendo
2-Receba o divisor
3-Se divisor vale zero
   3.1 Exibir mensagem
         "Divisão por zero não é possível".
   3.2 Terminar a execução.
4. Calcular a divisão:
         divisão= dividendo/divisor
5. Exibir mensagem "Resultado=divisão"
6. Terminar a execução
```

Algoritmo, Programa e Software

- Um **programa** é uma sequência de instruções descritas por um algoritmo que permite com que o computador realize uma tarefa.

```c
#include <stdio.h>
#include<stdio.h>
int main()
{
   float dividendo, divisor,divisao;

   printf("Dividendo="); scanf("%f",&dividendo);
   printf("Divisor="); scanf("%f",&divisor);

   if(divisor==0){
       printf("Divisão por zero não é possível!!");
       return 0;
   }
   divisao=dividendo/divisor;
   printf("Resultado=%.2f",divisao);
   return 0;
}
```

```
Dividendo=10
Divisor=0
Divisão por zero não é possível!!
```

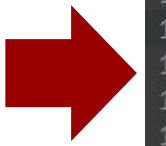
```
Dividendo=10
Divisor=3
Resultado=3.33
```

Algoritmo, Programa e Software

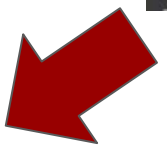
- Um **software** é formado por um ou mais programas e estruturas de dados escritos para realizar alguma tarefa que resolva um problema ou torne mais fácil e segura a vida de pessoas ou de organizações.

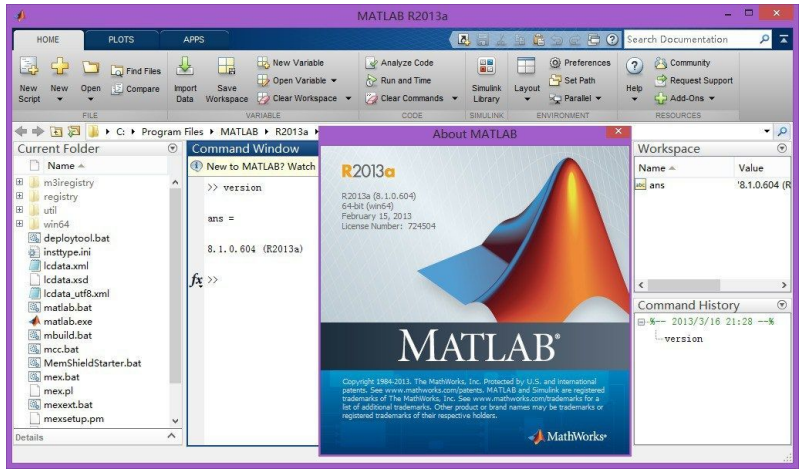
Algoritmo, Programa e Software

```
Algoritmo
1-Receba o dividendo
2-Receba o divisor
3-Se divisor vale zero
      3.1 Exibir mensagem
      "Divisão por zero não é possível".
      3.2 Terminar a execução.
4. Calcular a divisão:
      Divisão= dividendo/divisor
5. Exibir mensagem "Resultado=divisão"
6. Terminar a execução
```



```c
#include <stdio.h>
#include<stdio.h>
int main()
{
   float dividendo, divisor,divisao;

   printf("Dividendo="); scanf("%f",&dividendo);
   printf("Divisor="); scanf("%f",&divisor);

   if(divisor==0){
       printf("Divisão por zero não é possível!!");
       return 0;
   }
   divisao=dividendo/divisor;
   printf("Resultado=%.2f",divisao);
   return 0;
}
```

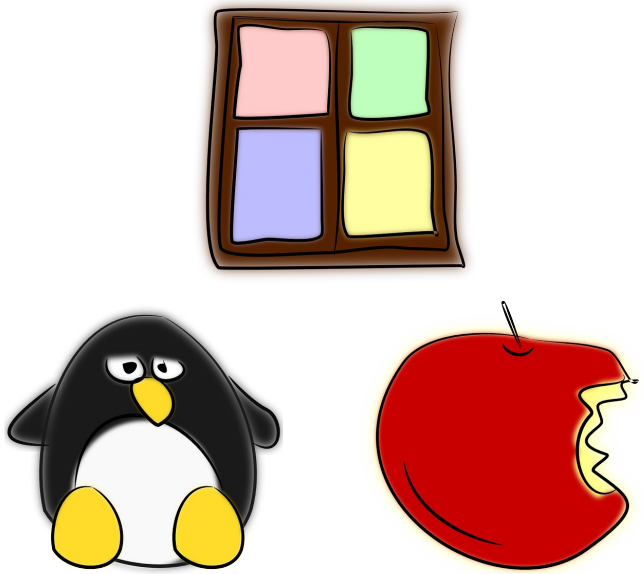




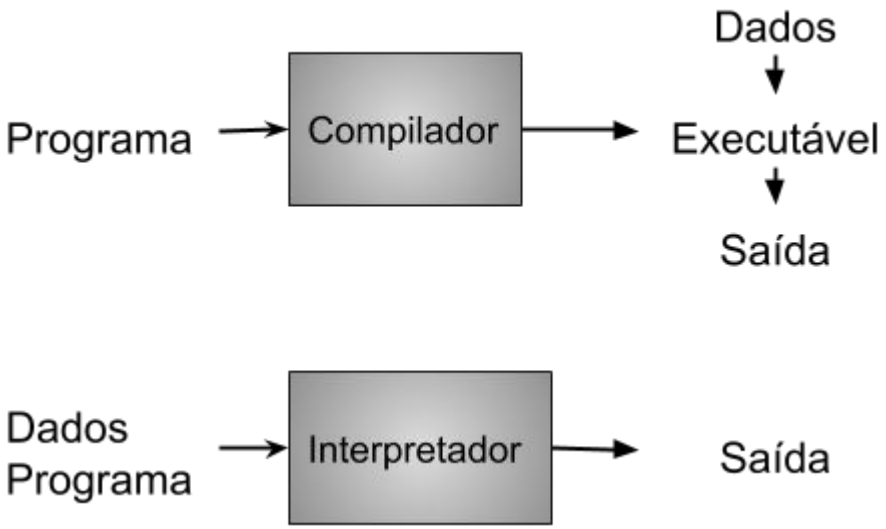
Tipos de Software

- **Software básico ou de sistema:** coleção de programas escritos para apoiar outros programas. Possui contato mais direto com o hardware, gerenciando a execução de outro software.

Sistemas Operacionais



Compiladores e Interpretadores



Tipos de Software

- **Software utilitário:** programas úteis ao bom funcionamento do sistema computacional.
- **Software de aplicação:** programas que ajudam o usuário a realizar uma dada tarefa.

Utilitários

Aplicação

Tipos de Software

- **Plug-in:** código que adiciona características específicas a outro software.
- **Software embarcado:** frequentemente é um firmware dentro de sistemas embarcados ou de dispositivos de uso específico.

Plug-in

Firmware

Algoritmo

- **Conjunto finito de regras, bem definido (sem ambiguidades), voltado à solução de um problema em tempo finito.**
- **Logo, deve possuir 3 qualidades:**
    1. **Cada passo do algoritmo deve ser uma instrução que possa ser realizada**
    2. **A ordem dos passos deve ser precisamente determinada**
    3. **O algoritmo deve ter fim**

Algoritmo

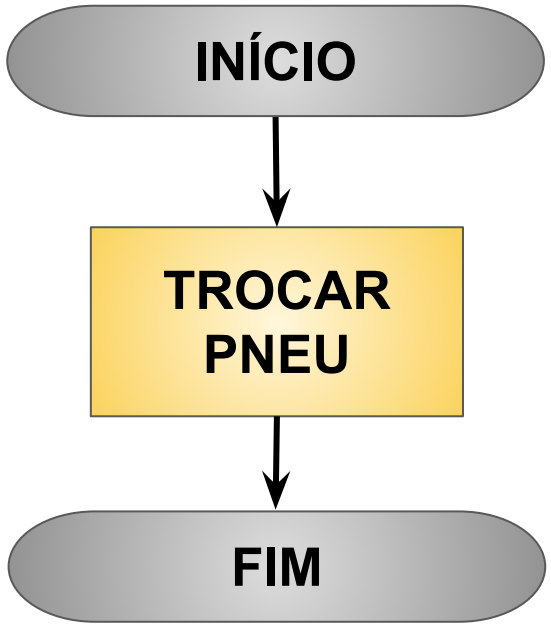
Dois aspectos devem ser levados em consideração no projeto de algoritmos:

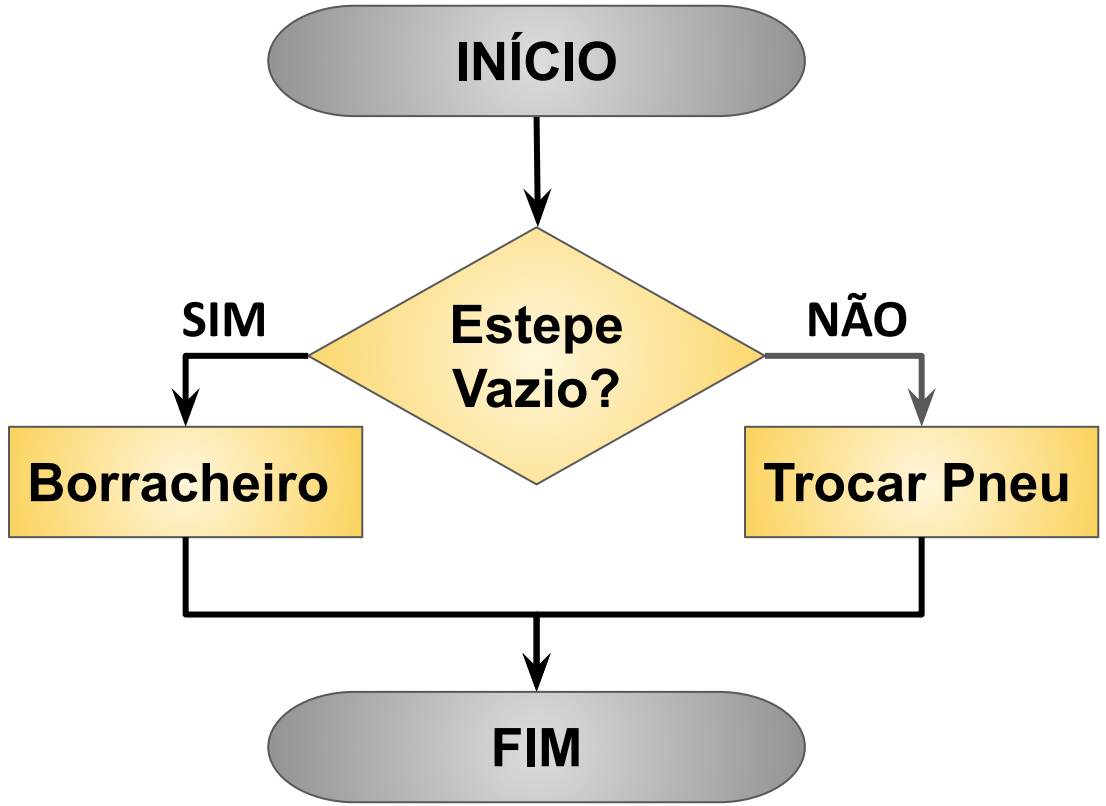
- **Corretude:** **para cada entrada possível, o algoritmo terá fim e produzirá a resposta desejada;**

- **Eficiência:** **a quantidade de recursos (tempo e armazenamento) consumidos na execução do algoritmo.**

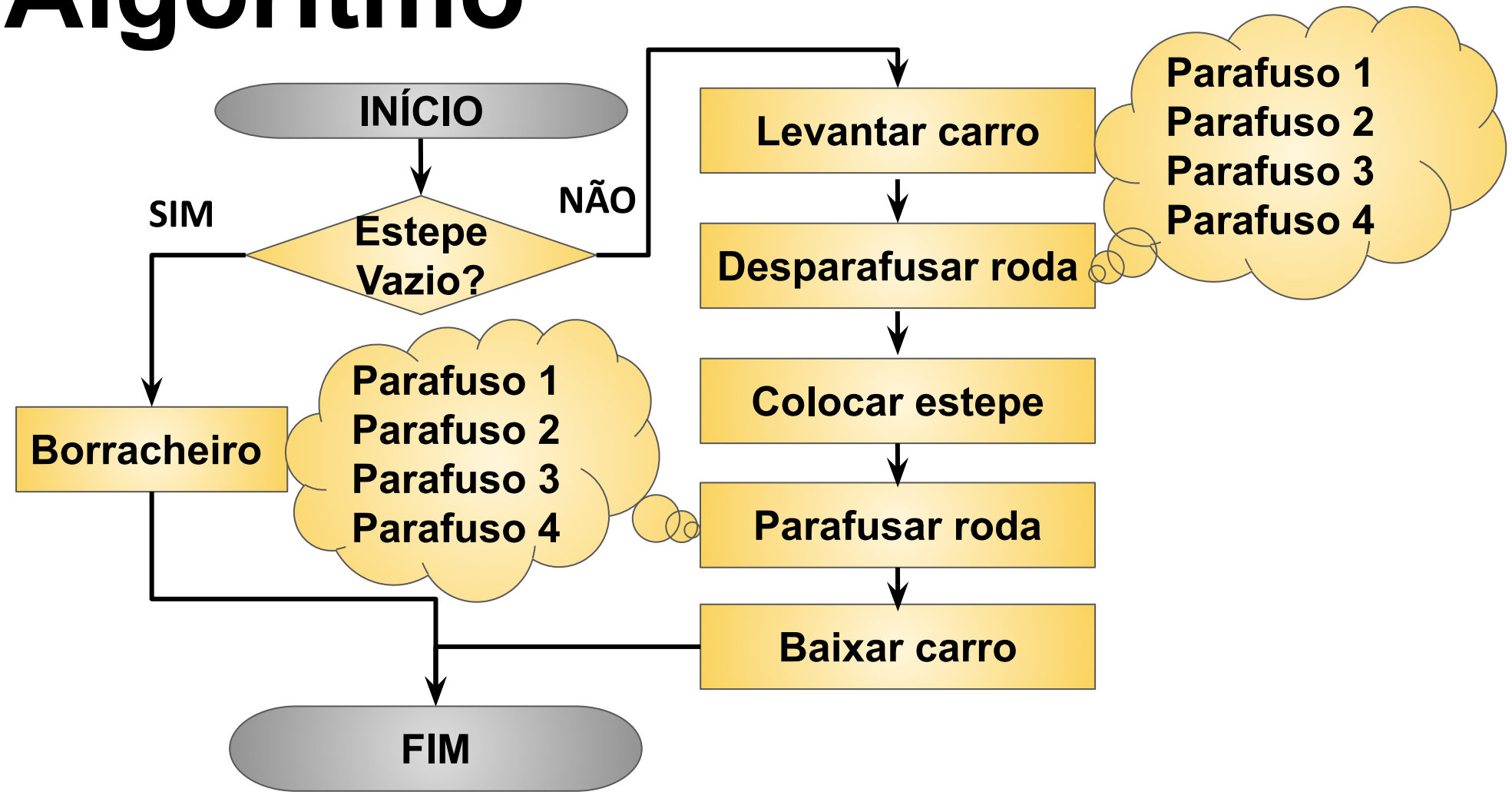
Algoritmo

TROCAR PNEU DE UM CARRO

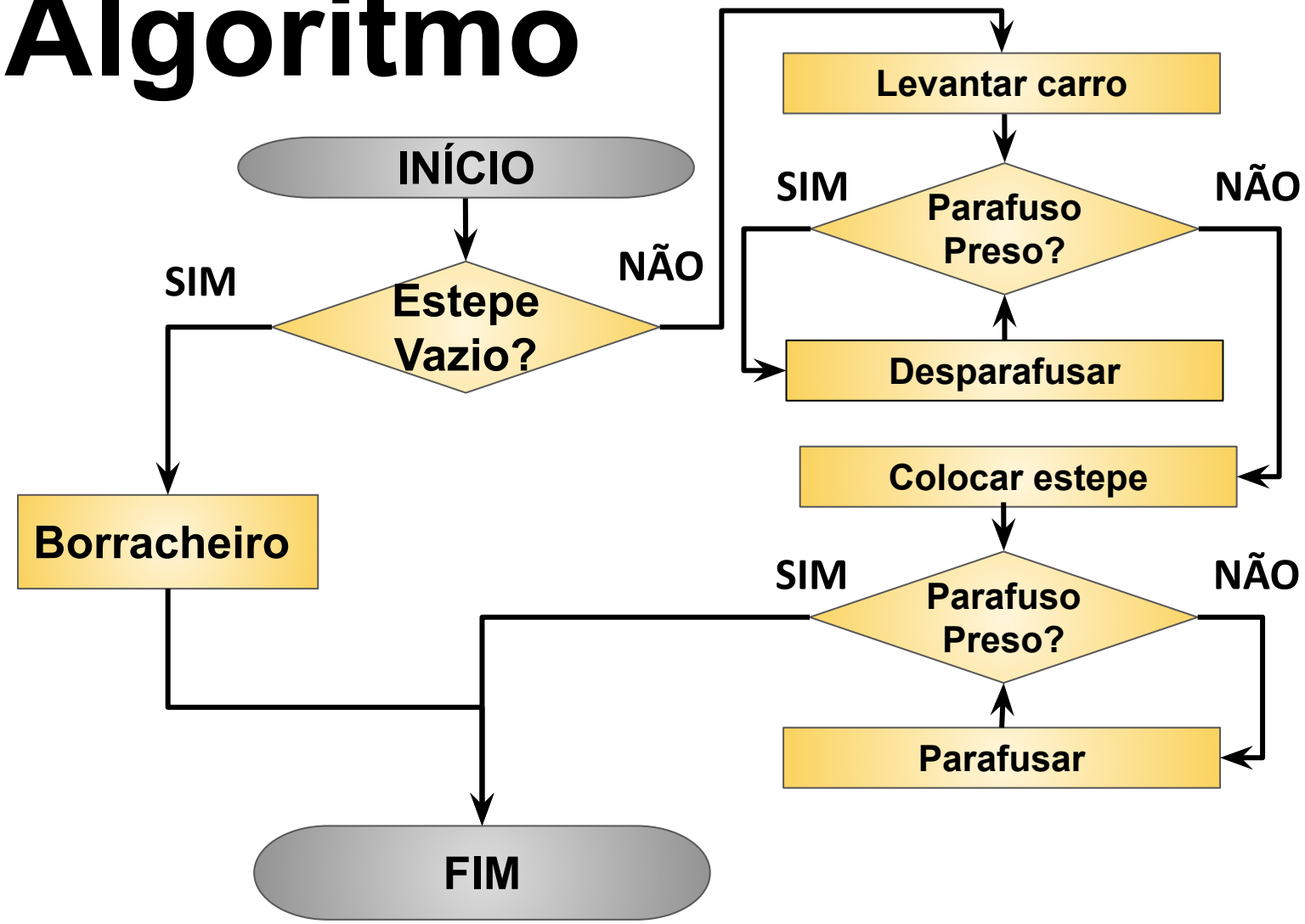




Algoritmo


INÍCIO
Estepe Vazio?
SIM
NÃO
Borracheiro
Parafuso 1
Parafuso 2
Parafuso 3
Parafuso 4
Levantar carro
Desparafusar roda
Parafuso 1
Parafuso 2
Parafuso 3
Parafuso 4
Colocar estepe
Parafusar roda
Baixar carro
FIM

Algoritmo


INÍCIO
Estepe Vazio?
SIM
NÃO
Borracheiro
Levantar carro
Parafuso Preso?
SIM
NÃO
Desparafusar
Colocar estepe
Parafuso Preso?
SIM
NÃO
Parafusar
FIM

Algoritmo

- **Pseudocódigo é uma forma simples de escrever um algoritmo através de uma linguagem simples.**
- **Pode seguir algumas regras predefinidas na sua escrita**
- **Facilita o mapeamento para uma linguagem de programação**

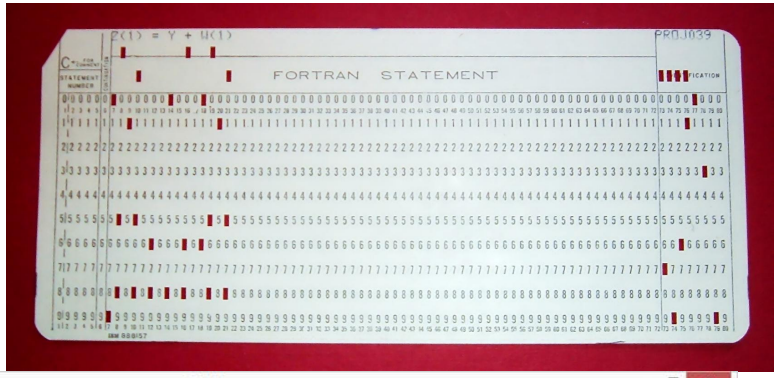
```
Inicio.
   declare n1, n2, soma
    escreva(“Digite dois números”)
    leia(n1, n2)
    soma = n1 + n2
    escreva(“Soma é igual a”, soma)
Fim.
```

Linguagem de Programação

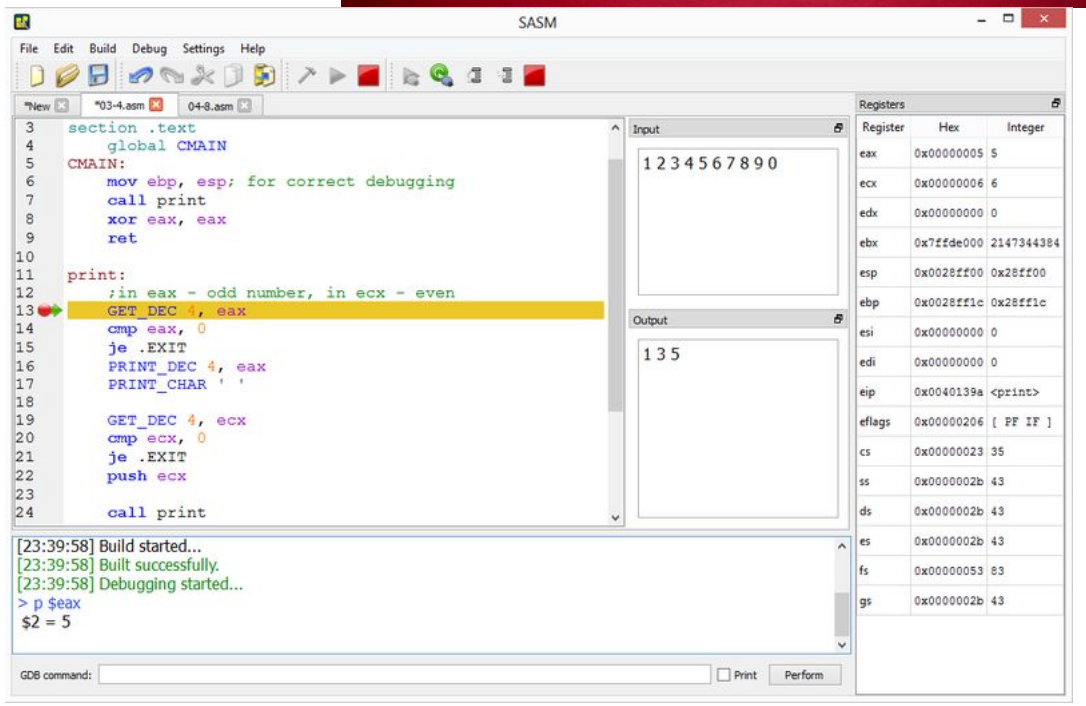
- **Permite descrever, de forma compreensível para o computador, os passos a serem por ele realizados na execução de uma tarefa.**
- **Toda linguagem de programação possui um conjunto limitado de símbolos, estruturas de dados e comandos utilizados para criar os programas.**

Linguagem de Programação

Linguagens de máquina
Primeira Geração



Linguagens de montagem
Segunda Geração



Linguagem de Programação

Linguagens de alto nível
Terceira Geração

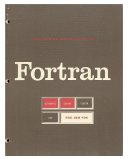
```c
#include <stdio.h>
#include<stdio.h>
int main()
{
   float dividendo, divisor,divisao;

   printf("Dividendo="); scanf("%f",&dividendo);
   printf("Divisor="); scanf("%f",&divisor);

   if(divisor==0){
       printf("Divisão por zero não é possível!!");
       return 0;
   }
   divisao=dividendo/divisor;
   printf("Resultado=%.2f",divisao);
   return 0;
}
```

Linguagens de muito alto nível
Quarta geração

Linguagem de Programação

Paradigmas:

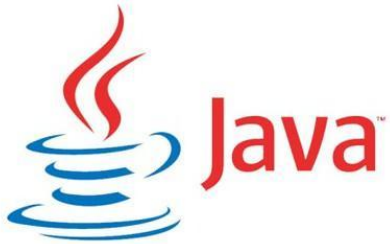
- **Imperativo**



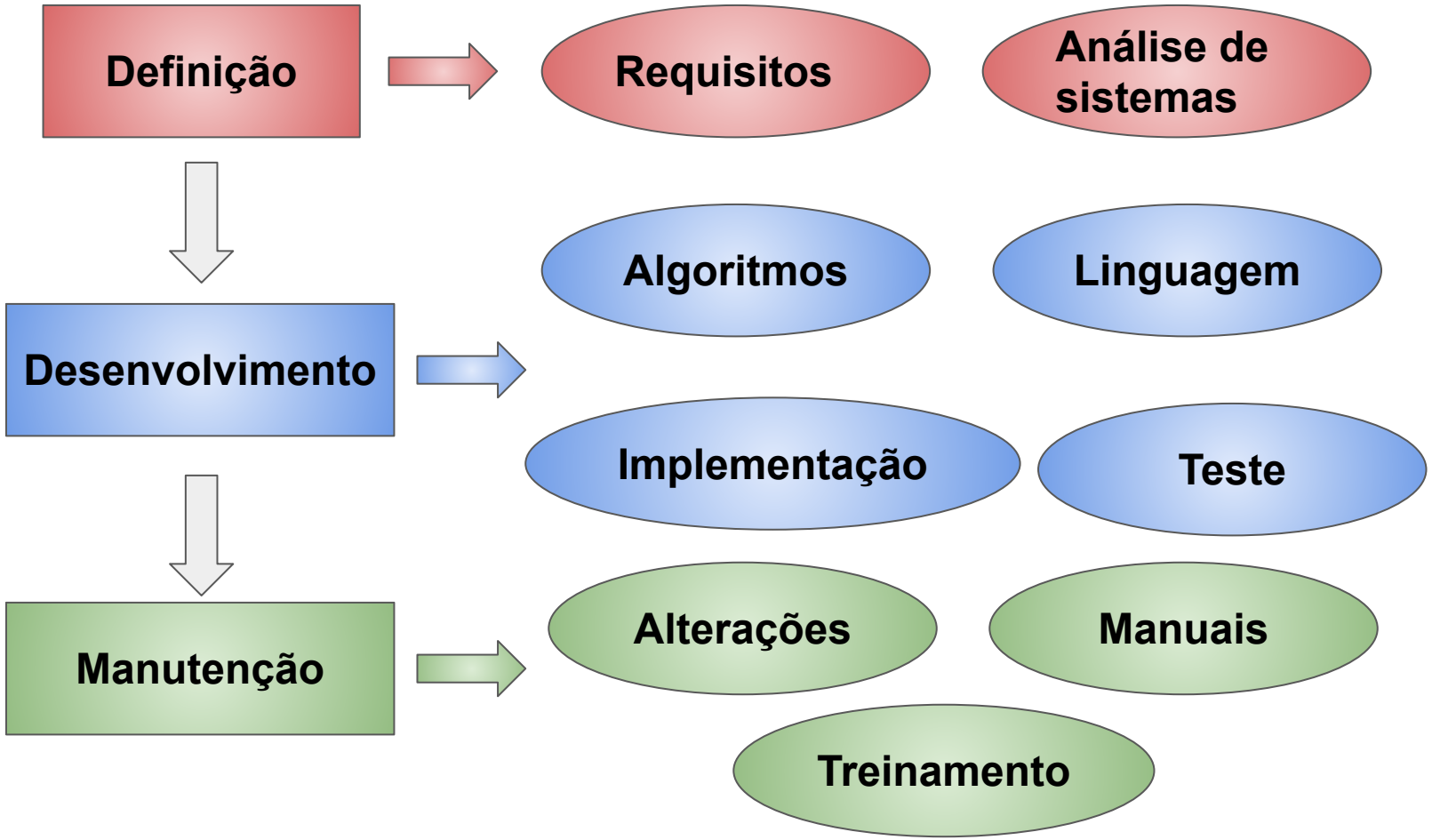
- **Funcional**

- **Lógico**

- **Orientado a objetos**



Ciclo de Vida do Software



Python

- Linguagem de programação de propósito geral usada para codificar qualquer tipo de programa que não demande acesso direto ao hardware do computador
- Utiliza interpretador ao invés de compilador.
  - Interpretado versus compilado considera se a sequência de instruções escritas pelo programador (código-fonte), é executada diretamente por um interpretador, ou se é primeiro convertida pelo compilado em uma sequência de operações primitivas ao nível da máquina.
  - Logo, o interpretador traduz o código em tempo de execução, enquanto o compilador faz isso antes do tempo de execução.
  - Isso causa uma sobrecarga adicional que resulta na execução mais lenta do código em linguagens interpretadas do que em linguagens compiladas.

Python

- Vantagens
  - Uma das linguagens de programação mais populares do mundo com uma vasta gama de bibliotecas usadas em vários aplicativos: Numpy, Keras, Seaborn, TensorFlow, SciKit-Learn, Scrapy, Pandas, Matplotlib e Plotly.
  - Sintaxe concisa quando comparada a outras linguagens de programação de alto nível como Java.
  - Portável, ou seja, capaz de gerar aplicativos executados em vários sistemas operacionais (SO), exceto se o programa inclua chamadas específicas do sistema. Logo, você executa seu programa Python no Windows, Mac OS e Linux sem modificações no código.
  - Vários casos de uso em áreas em crescimento como Ciência de dados
  - Aprendizado de máquina. Estatísticas ,Segurança cibernética, Desenvolvimento de jogos, etc.

Python

- Desvantagens
  - Lenta quando comparada a outras linguagens de programação de alto nível. Causas:
    - Uso de um interpretador,
    - Uso de tipagem dinâmica,
    - Uso de coletor de lixo,
    - Tipo de modelo de objeto do Python
    - Python não foi projetado como uma linguagem com eficiência de memória.
  - A nomenclatura do Python se desvia das normas com as quais outras linguagens de programação podem concordar, por exemplo, o tipo de dados ao qual nos referimos como um dicionário em Python também é conhecido como hash em Java e C++.
  - O código pode se tornar complexo com o aumento do número de linhas, se boas práticas de programação não forem aplicadas.