

## Linguagem de programação

A função das linguagens de programação é servir de um meio de comunicação entre computadores e humanos, assim como os humanos fazem com os humanos: usam da linguagem para poder conversar.

Quando programamos em uma linguagem de programação primeiramente criamos um arquivo de texto comum contendo a lógica do que deve ser feito, ou seja, é onde falamos ao computador, o que queremos.

Este arquivo de texto é chamado de programa, cada palavra de ordem dentro do programa é chamada de instrução. Após criarmos o programa, este deve ser traduzido para linguagem entendida pelo computador a partir da linguagem que estamos utilizando.

Um segundo arquivo que chamamos de executável ou programa, será gerado e interpretado diretamente pelo computador.

## Algoritmo

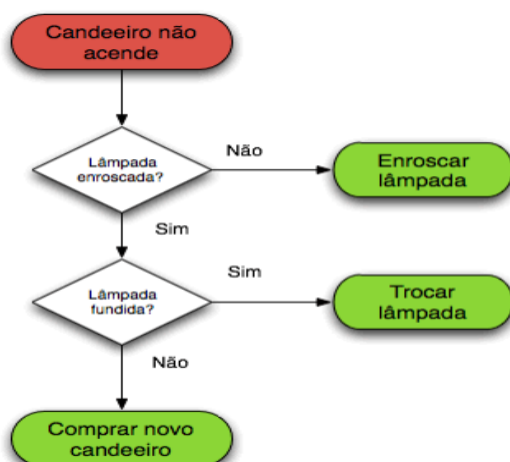
Algoritmo é uma sequência lógica de passos que levam a um determinado objetivo. Apesar de a palavra algoritmo parecer estranha e desconhecida, é claro que todos nós sabemos construir algoritmos.

Se isto não fosse verdade, não conseguiríamos sair de casa pela manhã, ir à escola, decidir qual o melhor caminho para chegar a um lugar, voltar para casa etc.

Para que tudo isto seja feito é necessário uma série de entradas do tipo: a que hora acordar, a que horas sair de casa, qual o melhor meio de transporte etc.

Um fator importante é que pode haver mais de um algoritmo para resolver um problema. Por exemplo, para ir de casa até o colégio, podemos escolher diversos meios de transportes em função do preço, conforto, rapidez etc.

Como por exemplo



Algoritmos podem ser divididos em partes

**Início;**

**Entrada:** dados que devem ser lidos, ou fornecidos pelo usuário;

**Processamento:** onde acontecem os cálculos em si, os dados de entrada são manipulados a fim de se gerar uma resposta (um resultado);

**Saída:** são os resultados de seu processamento.;

**Fim.**

### O que é preciso para se fazer um programa de computador?

Vamos precisar de:

- Uma linguagem de programação: regras léxicas e sintáticas para se escrever o programa
- Uma IDE: software para editar e testar o programa
- Um compilador: software para transformar o código fonte em código objeto
- Um gerador de código ou máquina virtual: software que permite que o programa seja executado

## léxica e sintática


### Léxica

Diz respeito à correção das **palavras** "isoladas" (ortografia).

**Exemplo (Português):**

cachorro

caxorro



**Linguagem de programação:**

main

maim




### Sintática

Diz respeito à correção das **sentenças** (gramática).

**Exemplo (Português):**

O cachorro está com fome.


A cachorro está com fome.



**Linguagem de programação:**

x = 2 + y;

x = + 2 y;



## Exemplo de um programa:

Suponha um programa que solicita do usuário dois números e depois mostra a média aritmética deles:

## Solução em linguagem Java

```
import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        System.out.print("Digite o primeiro numero: ");
        x = sc.nextDouble();
        System.out.print("Digite o segundo numero: ");
        y = sc.nextDouble();
        media = (x + y) / 2.0;
        System.out.println("Media = " + media);
        sc.close();
    }
}
```

### IDE – Ambiente Integrado de Desenvolvimento

É um conjunto de softwares utilizado para a construção de programas. Exemplos:

C/C++ : Code Blocks

Java : Eclipse, NetBeans

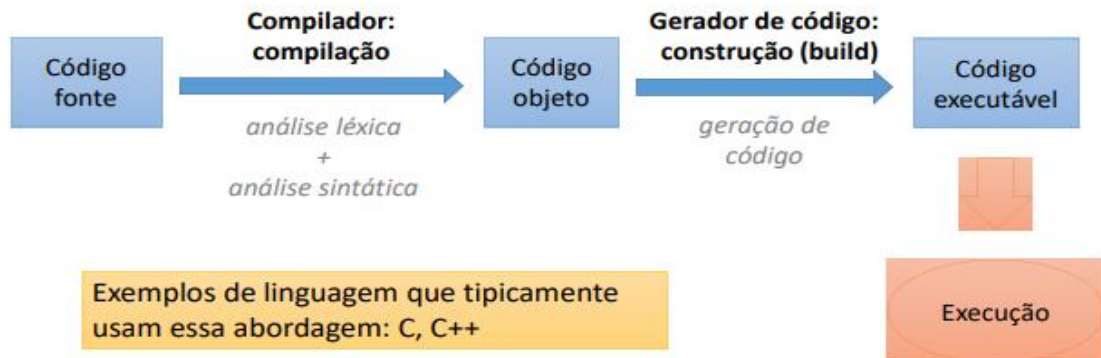
C# : Microsoft Visual Studio

### Funcionalidades de uma IDE

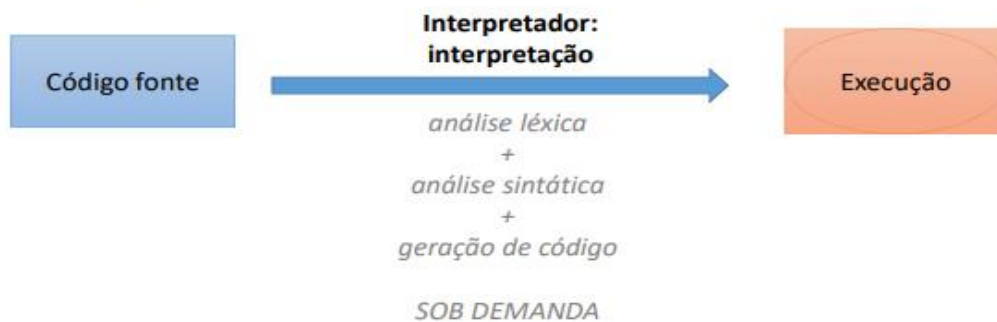
- Edição de código fonte (indentação, autocompletar, destaque de palavras etc.)
- Depuração e testes
- Construção do produto (build)
- Sugestão de modelos (templates)
- Auxiliar em várias tarefas do seu projeto

## COMPILAÇÃO, INTERPRETAÇÃO E ABORDAGEM HÍBRIDA

### Compilação

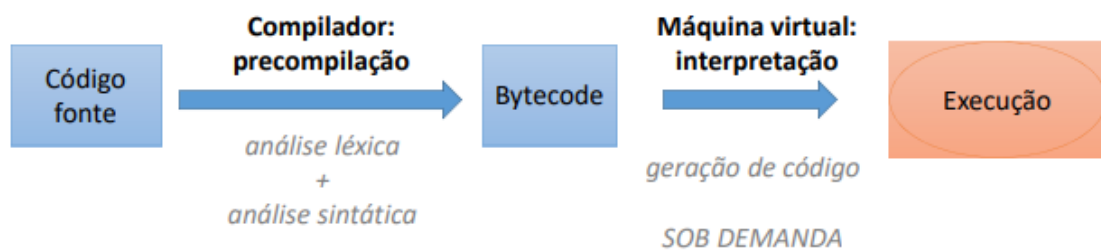


### Interpretação



Exemplos de linguagem que tipicamente usam essa abordagem: PHP, JavaScript, Python, Ruby

### Abordagem híbrida



Exemplos de linguagem que tipicamente usam essa abordagem: Java (JVM), C# (Microsoft .NET Framework)

## Vantagens

### Compilação:

- velocidade do programa
- auxílio do compilador antes da execução

### Interpretação:

- flexibilidade de manutenção do aplicativo em produção
- expressividade da linguagem
- código fonte não precisa ser recompilado para rodar em plataformas diferentes

### Abordagem híbrida

The diagram illustrates the 'Abordagem híbrida' (Hybrid Approach) by showing red arrows pointing from the 'Abordagem híbrida' label to specific advantages in both the 'Compilação' and 'Interpretação' sections. The arrows point to 'velocidade do programa', 'auxílio do compilador antes da execução', 'flexibilidade de manutenção do aplicativo em produção', and 'código fonte não precisa ser recompilado para rodar em plataformas diferentes'.