

# Compilação e interpretação

## Código fonte e objeto

## Máquina virtual

<http://educandoweb.com.br>

Prof. Dr. Nelio Alves

**Código fonte:** é aquele escrito pelo programador em linguagem de programação

```
#include <stdio.h>

int main() {
    double x, y, media;

    printf("Digite o primeiro numero: ");
    scanf("%lf", &x);
    printf("Digite o segundo numero: ");
    scanf("%lf", &y);
    media = (x + y) / 2.0;
    printf("Media = %.1f\n", media);
    return 0;
}
```

```
#include <iostream>

using namespace std;

int main() {
    double x, y, media;

    cout << "Digite o primeiro numero: ";
    cin >> x;
    cout << "Digite o segundo numero: ";
    cin >> y;
    media = (x + y) / 2.0;
    cout << "Media = " << media << endl;
    return 0;
}
```

```
using System;

namespace programa {
    class Program {
        static void Main(string[] args) {
            double x, y, media;

            Console.Write("Digite o primeiro numero: ");
            x = double.Parse(Console.ReadLine());
            Console.Write("Digite o segundo numero: ");
            y = double.Parse(Console.ReadLine());
            media = (x + y) / 2.0;
            Console.WriteLine("Media = " + media);
        }
    }
}
```

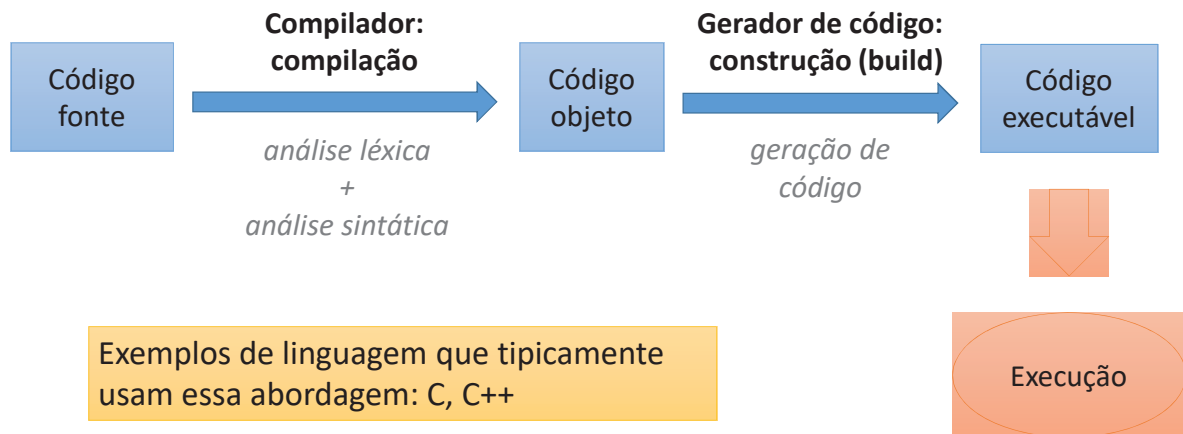
```
import java.util.Scanner;

public class Main {

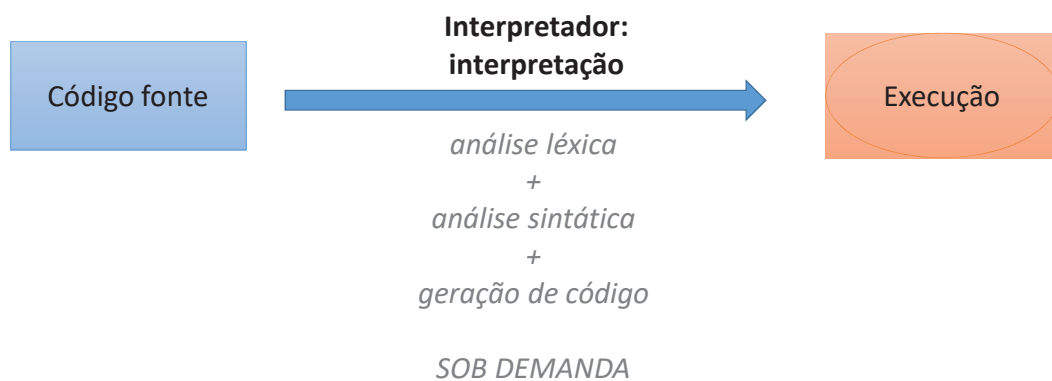
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        double x, y, media;

        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();
    }
}
```

## Compilação

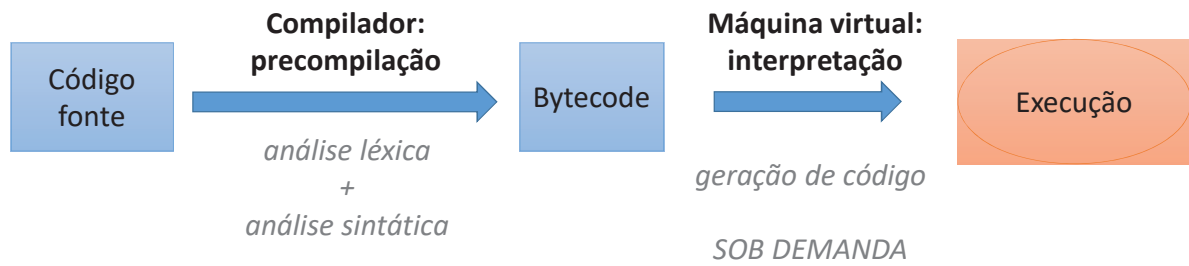


## Interpretação



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

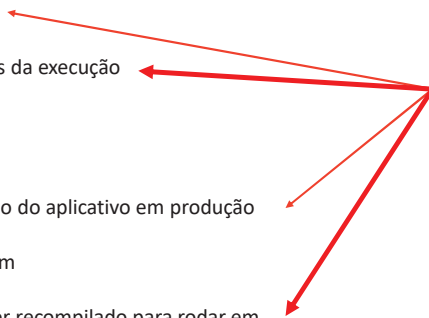
### Compilação:

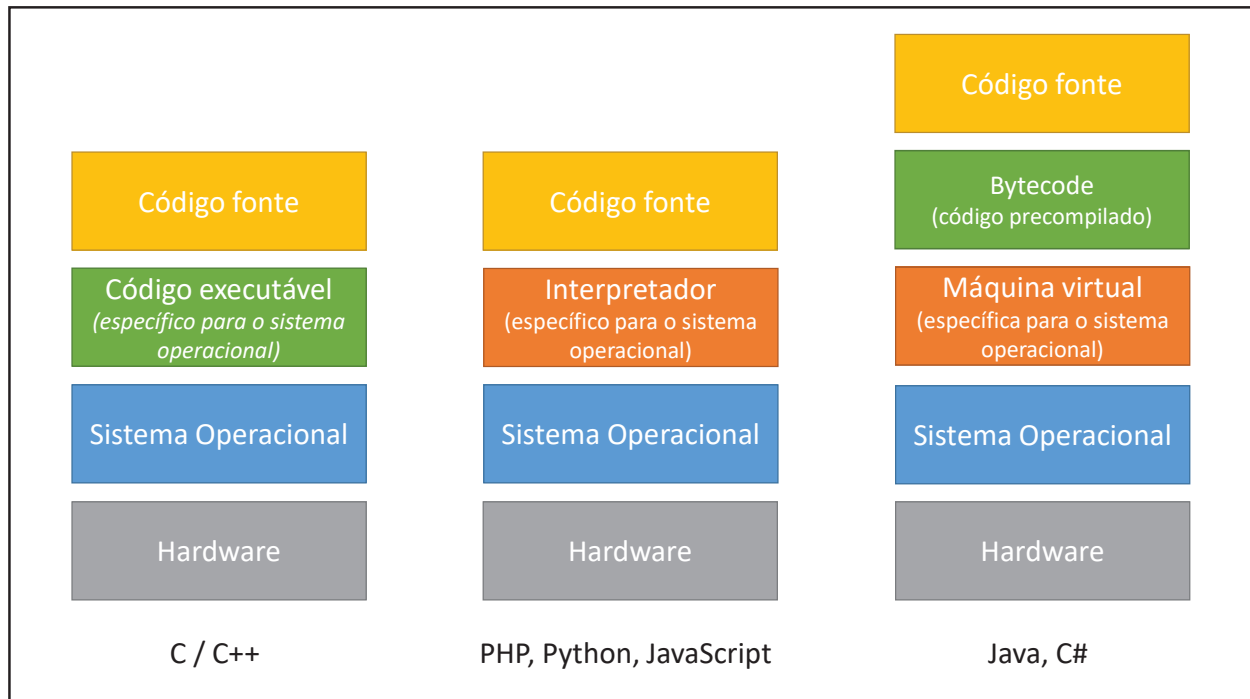
- velocidade do programa
- auxílio do compilador antes da execução

### Interpretação:

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

### Abordagem híbrida





## Resumo da aula

- Tipos de código
  - Código fonte
  - Código objeto / bytecode
- Modelos de execução:
  - Compilação
    - Gerador de código
  - Interpretação
  - Abordagem híbrida
    - Máquina virtual