

Prof. Dr. Daniel Leal Souza

# Autômatos e Compiladores (EC8MA)

# Compiladores (CC6MA)

---

[daniel.leal.souza@gmail.com](mailto:daniel.leal.souza@gmail.com)  
[daniel.souza@prof.cesupa.br](mailto:daniel.souza@prof.cesupa.br)

2024



## Compiladores

### Pré-requisitos:

Linguagens Formais e Autômatos



Programação



Esforço próprio



### Vai ser importante para:

Programação de Sistemas em geral: desktop, web, games, mobile.

# Apresentação

O que eu espero de vocês?

1. Presença;
2. Pontualidade;
3. Envolvimento nas aulas e nas atividades (**pontuação extra**);
4. Matéria dada é matéria estudada (**HOJE!**).

# Apresentação

O que eu NÃO espero de vocês?

1. Plágio (crime)
2. Passividade extrema
3. Falta de comprometimento
4. Atraso na entrega de atividades

## **Autômatos e Compiladores (EC8MA)**

Dias da semana e horários:

- Quarta de 11h40 às 13h20 (EC8MA)
- Quinta de 08h00 às 09h40 (EC8MA)

## Conteúdo resumido:

1. Introdução à compilação;
  - a. O que é um compilador?
  - b. O que é um interpretador?
  - c. Comparação compilador x interpretador.
2. Análise léxica (com implementação);
3. Análise sintática (com implementação);
4. Análise semântica (com implementação);

# Apresentação

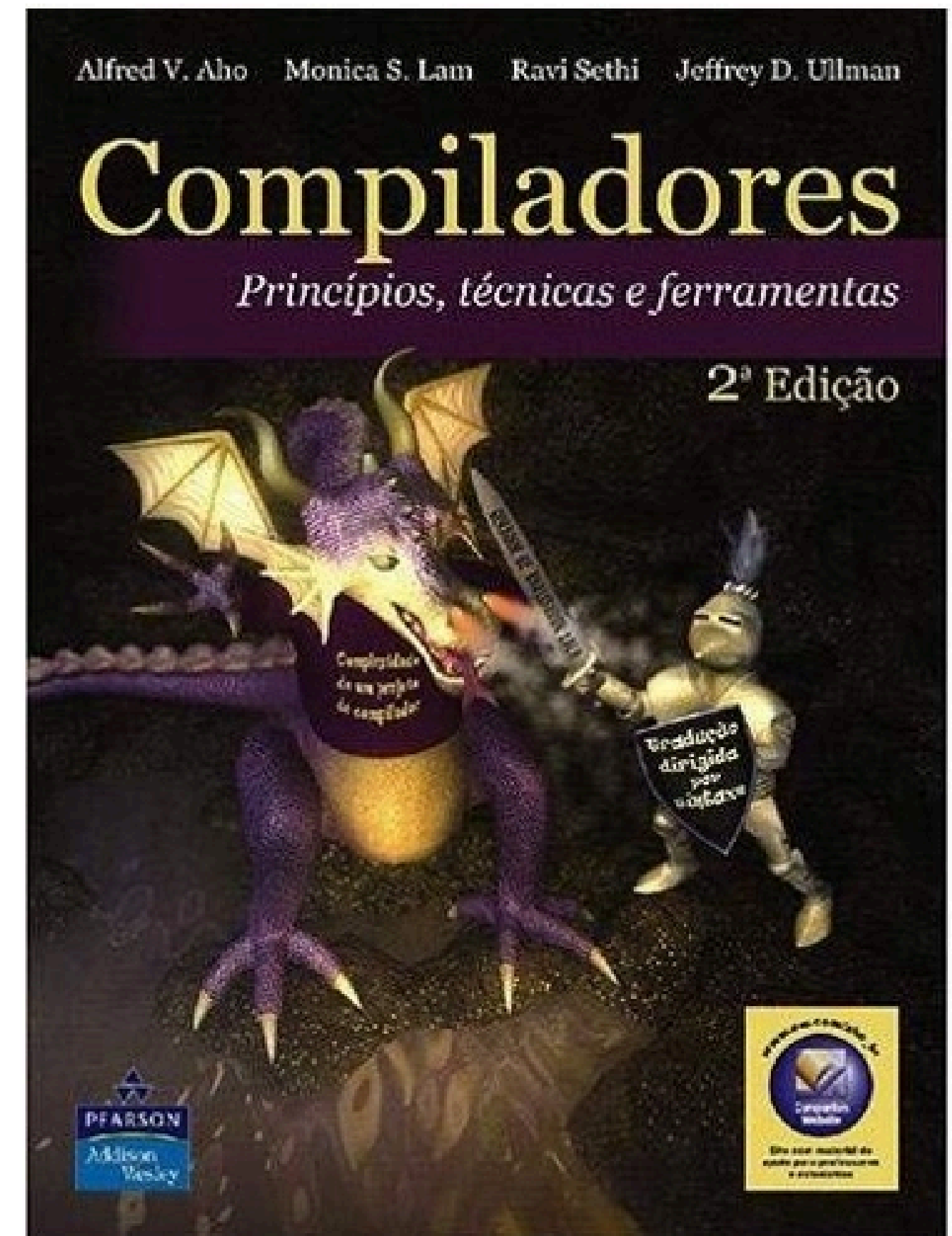
## Referência 1 (principal)

Título: Compiladores – princípios, técnicas e ferramentas

Autores: Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman

Ano: 2007

Editora: Pearson



# Apresentação

## Referência 2

Título: Introdução à compilação

Autor: João José Neto

Ano: 2016

Editora: Elsevier





# Apresentação

## Referência 3

Título: Construindo compiladores

Autores: Keith Cooper & Linda Torczon

Ano: 2014

Editora: Elsevier

2ª edição



# Introdução

Que está envolvido em uma solução computacional?



# Introdução



**PROGRAMADOR**

Domina:

C/C++

Java

Python

HTML

CSS

Javascript

- 
- 
- 



**USUÁRIO**

Domina:

Windows

Word

Excel

Power point

E-mail

Sistemas corp.

- 
- 
-

## **Definição de compilador:**

**“um programa que recebe como entrada um programa em uma linguagem de programação – a linguagem fonte – e o traduz para um programa equivalente em outra linguagem – a linguagem objeto.”**

Aho, Lam, Sethi e Ullman (2008). Compiladores: princípios, técnicas e ferramentas

**Já não existem compiladores clássicos e bem estabelecidos no mercado?**

R: Sim!

**Vale a pena criar um novo compilador?**

R1: <http://www.youtube.com/watch?v=kazE4zPRsxc&t=3m36s>

R2: Linguagem Égua - <https://github.com/eguatech/egua>

## O que são compiladores?

- Compilador é um processador de linguagem
  - Objetivo é TRADUZIR
    - Entrada: Linguagem de programação (C, C++, Java)
    - Saída: Linguagem de máquina
    - De forma que o computador consiga “entender”
- Por que existem?
  - A linguagem de máquina é muito “ruim” de programar
  - Dizemos que é “de baixo nível”
  - Fica difícil raciocinar sobre ela



# Introdução



# Introdução



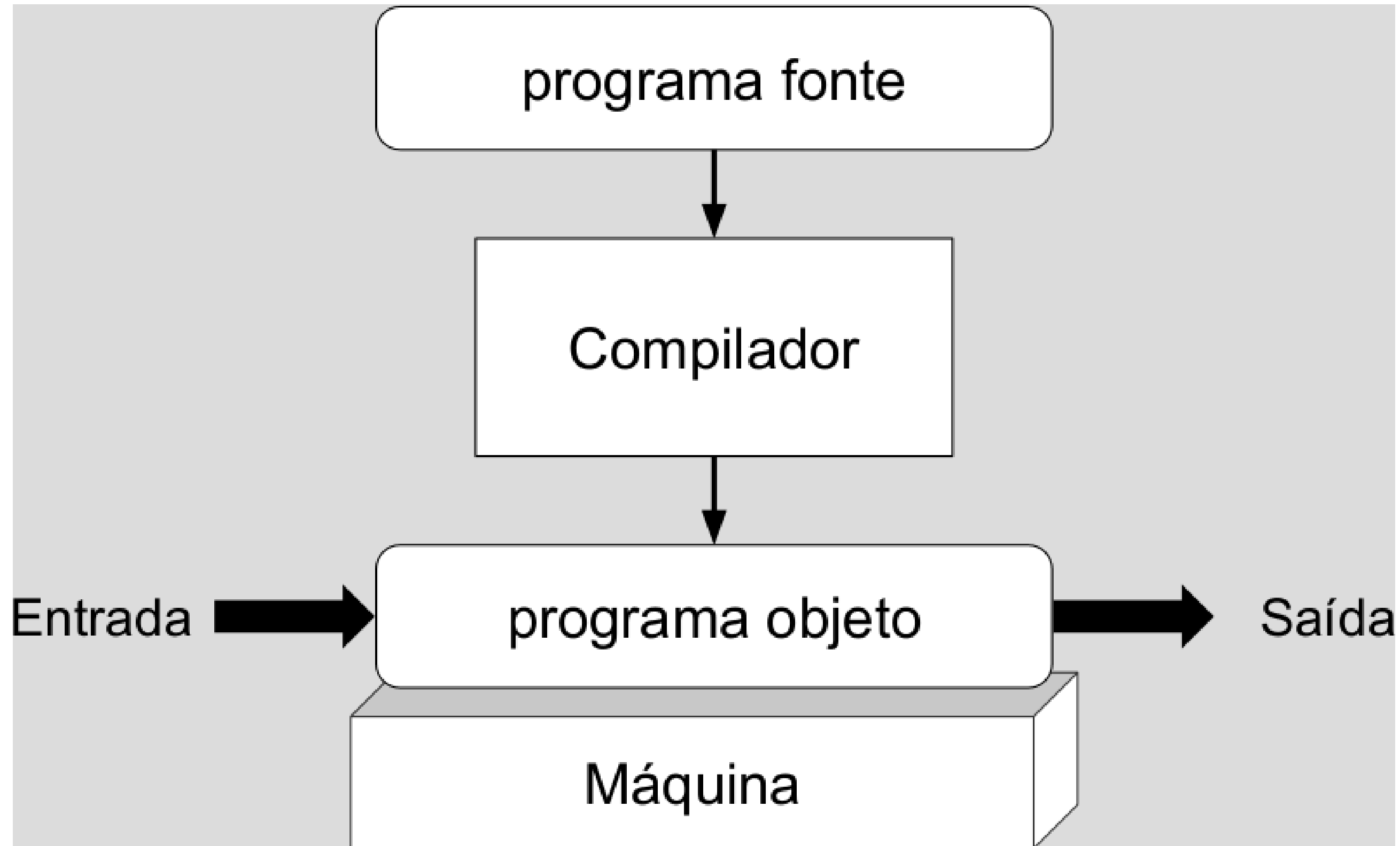


## O que são compiladores?

- O compilador é basicamente isso:
  - Algo (software) que consegue traduzir nossos desejos em linguagem que o computador entende
  - O compilador entende a linguagem de alto nível
  - E a traduz para uma linguagem de baixo nível
  - Seu trabalho então termina
- De forma que podemos “ensinar” ao computador alguma tarefa
  - E depois ele faz a tarefa sozinho

# Introdução

Compilador:



## O que são compiladores?

- Também possuem uma função extra
  - Ajudar o programador
    - Apontando erros no programa fonte
- Ex:
  - `if ( a > 10 a += 3;`
    - Falta um fecha parêntese depois do 10
      - Como você sabe disso?
      - Como uma máquina pode saber disso?
      - Não é uma tarefa simples!!!

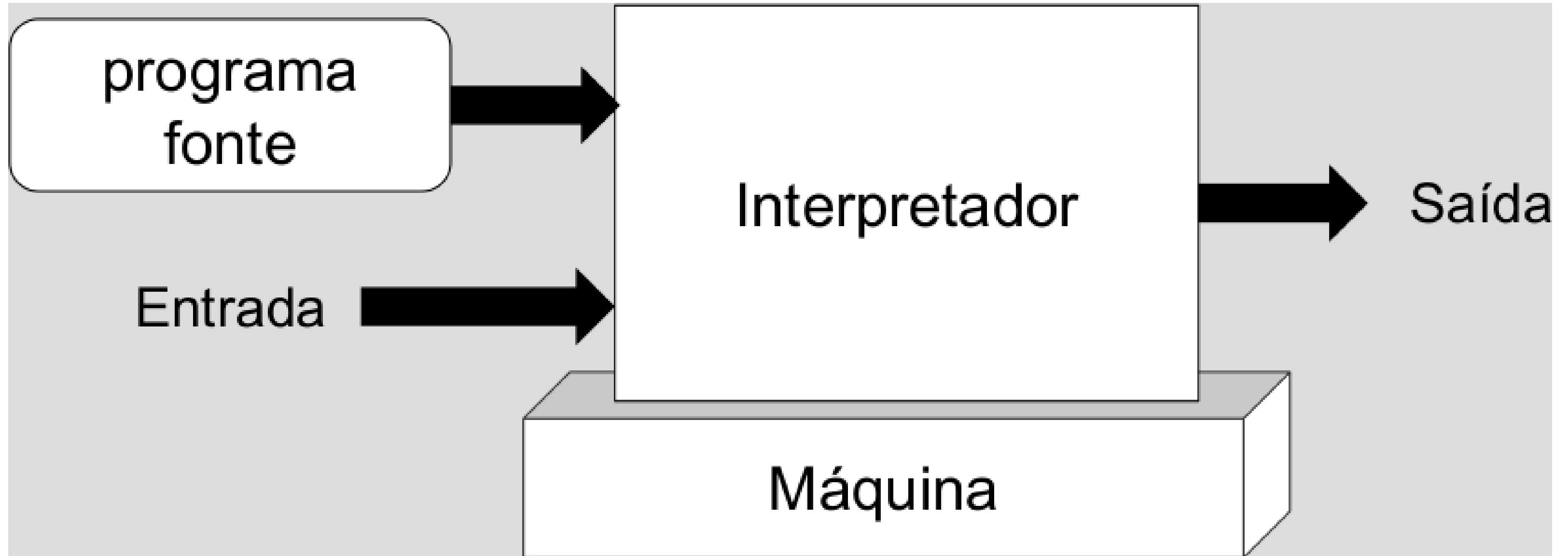
# DÚVIDAS?

## Interpretador

- Existe outro tipo de processador de linguagem
  - Chamado de interpretador
  - Como o compilador, o interpretador entende a linguagem de alto nível
  - Mas ele mesmo executa as tarefas
    - É como se o treinador de cachorros fosse buscar o jornal
- Outra forma de ver:
  - O interpretador traduz o programa fonte diretamente em ações

# Introdução

Interpretador:



## Interpretador x Compilador:

	LP interpretada	LP compilada
Erros são detectados antes do programa rodar?	Não, apenas em tempo de execução.	Sim, o programa só roda se todos os erros forem corrigidos.
Depende do Sistema Operacional?	Não, depende do interpretador.	Sim, se foi compilado no Windows não vai funcionar nos outros SOs.
Como fica a performance?	Mapeamento entrada/saída mais lento (Precisa traduzir toda vez que é executado).	Mapeamento entrada/saída mais rápido (Traduz 1 vez para executar n vezes).
Diagnóstico do erro.	Na tradução, perde-se informação (Dificulta o diagnóstico de erros).	O diagnóstico de erros é normalmente melhor (Mais flexível).

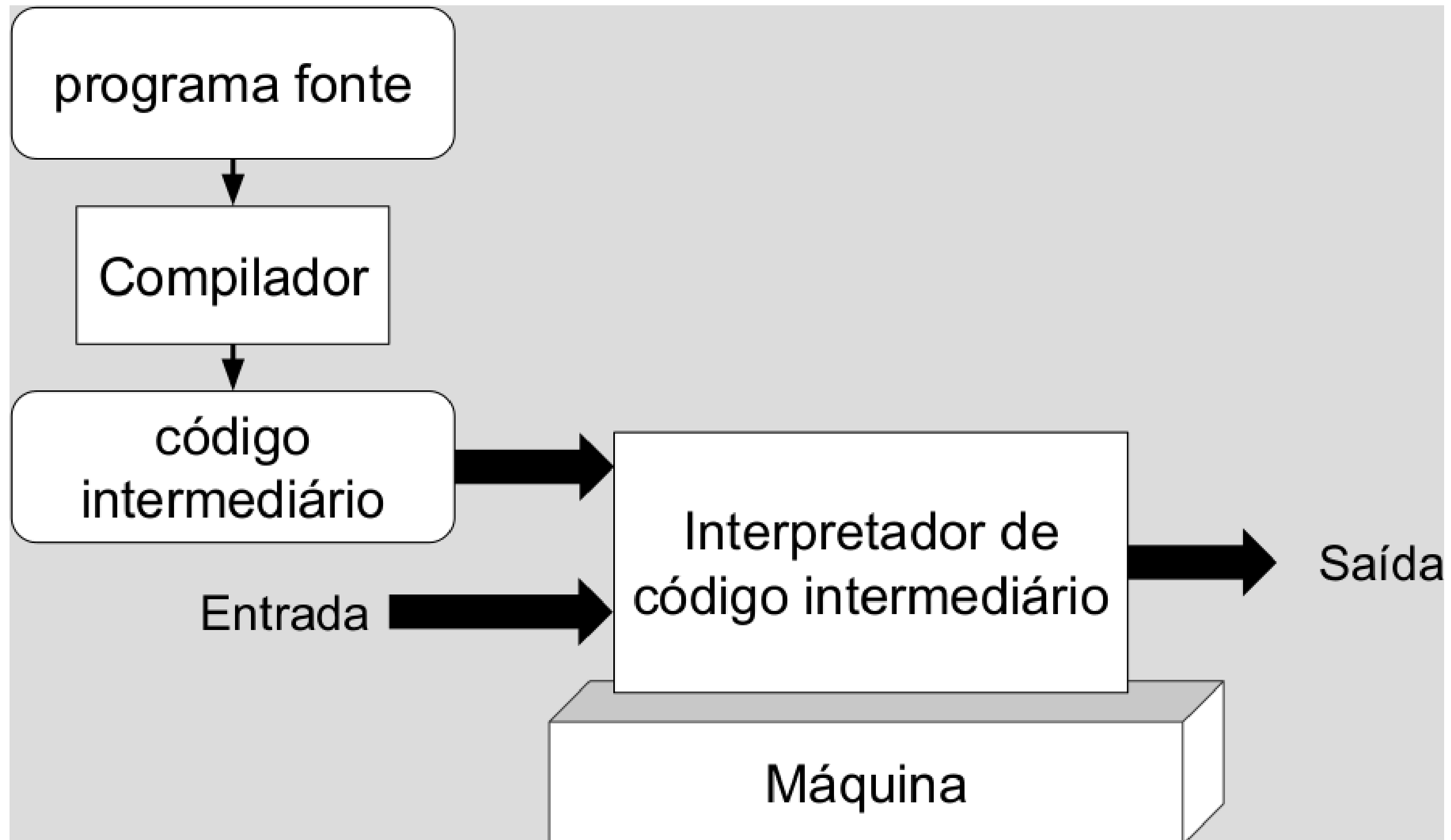
## Interpretadores e compiladores

- É possível combinar as duas abordagens
  - E buscar os benefícios de ambas
- Ex: Java, .NET
  - Usam compilação + interpretação
  - Abordagem híbrida



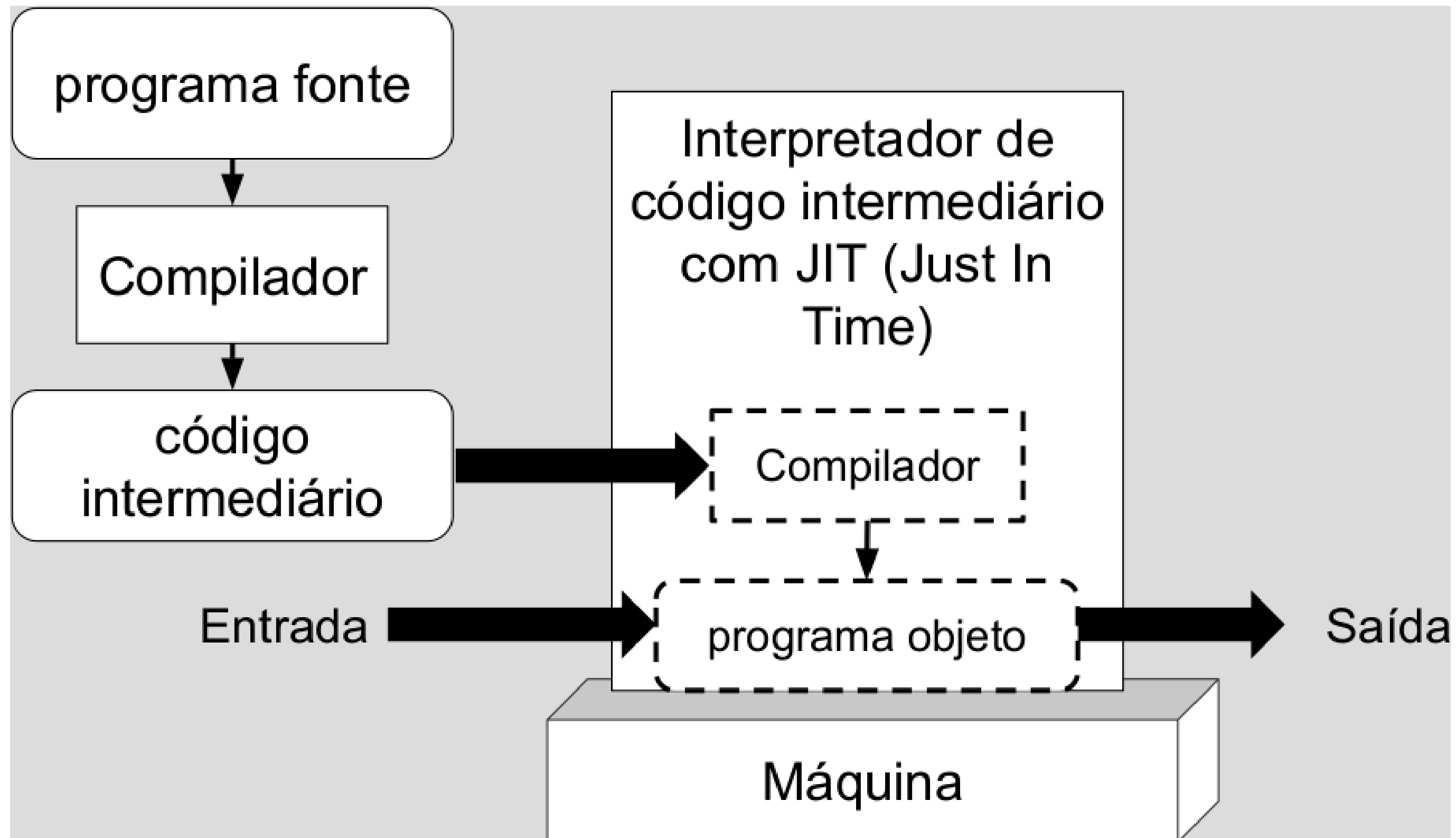
# Introdução

## Interpretadores e compiladores



# Introdução

## Interpretadores e compiladores



## Interpretadores e compiladores

- Na disciplina, muitos conceitos são comuns
- Foco é no processo de tradução
  - Compreensão da linguagem fonte
  - Detecção de erros

# DÚVIDAS?

# Introdução

## Exercício 1:

Traduza o algoritmo abaixo para Java e para Python.

algoritmo Tabuada

Inteiro n;

imprima("Número para calcular a tabuada:");

leia(n);

para (Inteiro i<-1; i<10; i++)

imprima(n," x ", i, " = ", n\*i);

fim\_para

fim\_algoritmo



Pausa para você tentar.  
Agora é a sua vez!

# Exercícios

1. Pesquise sobre TRANSPILADOR, especificando:
  - a. O que é.
  - b. Quais suas diferenças quando comparado com compilador e interpretador.
  - c. Qual sua utilidade.
  - d. Busque um trabalho científico que descreve um transpilador (bases: Google Scholar, Scielo).

# Exercícios

2. Construa um mapa mental tendo como tópico central **COMPILADOR**, resumizando o conteúdo que foi visto apenas nesta aula. Recomenda-se o uso do programa XMIND.

# Referências

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman.  
Compiladores - princípios, técnicas e ferramentas. Pearson, 2007.  
José Neto, João. Apresentação à compilação. 2ª edição. Rio de Janeiro: Elsevier, 2016.  
Notas de aula do professor Daniel Lucrédio - UFSCar.