

# **Paradigmas de Linguagem de Programação**

**Prof<sup>a</sup> Maria Adelina Raupp Sganzerla**  
**masganzerla@gmail.com**

**Ulbra – Gravataí**  
**2016/2**

# Linguagens de Programação

- Uma notação formal para descrever a execução de algoritmos de computador.
- É necessário entender a organização das LP, com ênfase na compreensão dos seus conceitos abstratos e dos elementos de sua representação durante o processo de execução.

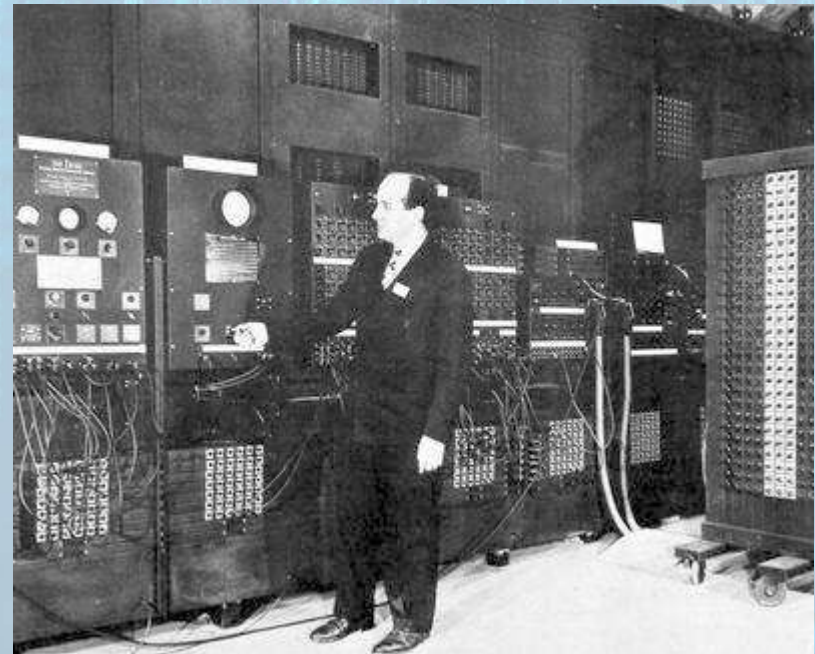


# Histórico das Linguagens de Programação 1/8

- As Linguagens de Programação modernas são o resultado de um processo evolutivo.
- As máquinas eram programadas por meio de Linguagens de baixo nível (Assembly e Linguagem de Máquina);



Década de 1940





# Histórico das Linguagens de Programação 2/8

- Teve início na década de 1950 com o surgimento das primeiras LP de alto nível;
- Programação em Fortran.



Década de 1950



# Histórico das Linguagens de Programação 3/8

- As Linguagens de alto nível são mais ou menos independentes de máquina.
- Linguagens: Lisp, Algol e Cobol



# Histórico das Linguagens de Programação 4/8

- Linguagens: APL, Simula I, Algol 60, Prolog e Basic



Década de 1960



# Histórico das Linguagens de Programação 5/8

- Linguagens: Smaltalk, Pascal e C



Década de 1970



# Histórico das Linguagens de Programação 6/8

- Linguagem: ADA



Década de 1980



# Histórico das Linguagens de Programação 7/8

- Linguagens: Miranda e Orientação a Objetos



Década de 1990



# Histórico das Linguagens de Programação 8/8

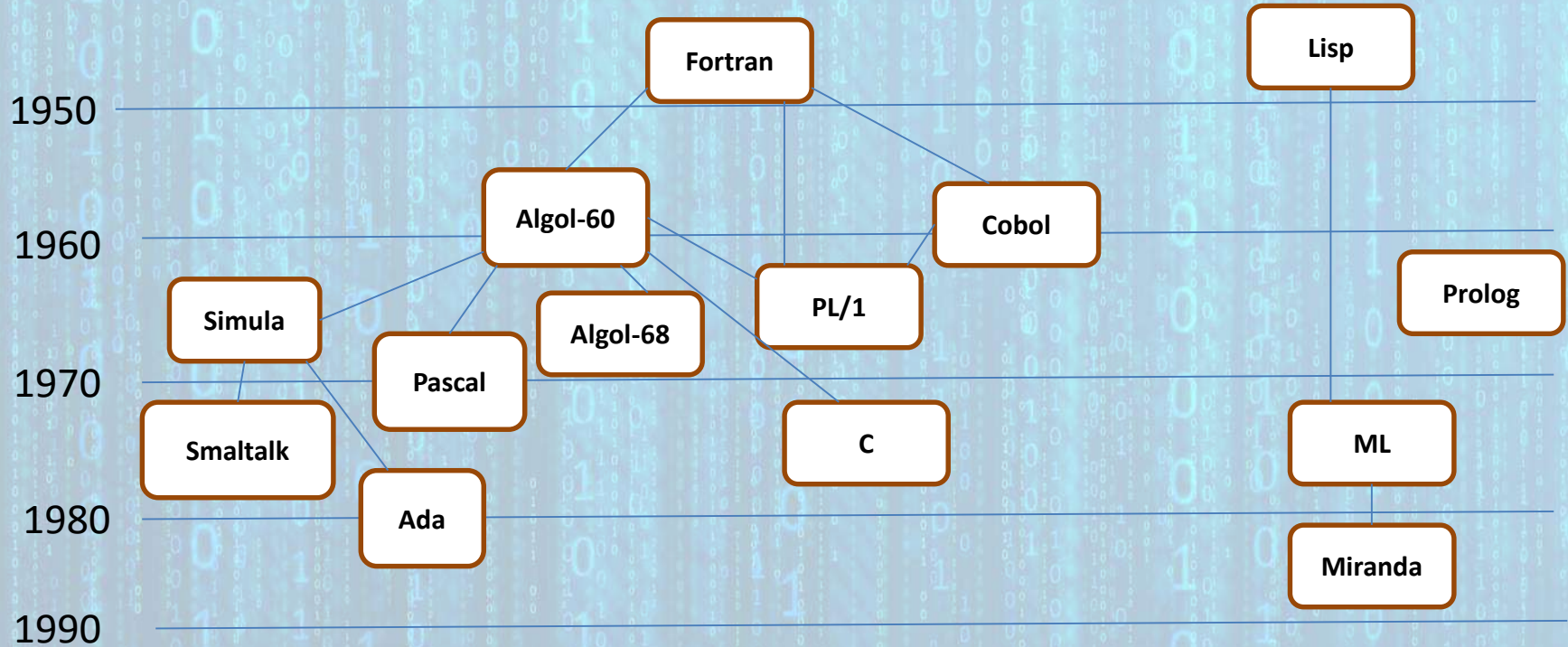


Atualmente





# Evolução das Linguagens de Programação



Orientação a Objetos



# Características de uma Linguagem de Programação 1/5

- Requisitos: Qual o universo de problemas que queremos resolver com a dada Linguagem?
  - Uma boa LP deve combinar poder expressivo – que classe de problemas pode ser abordada – com simplicidade e eficiência.



# Características de uma Linguagem de Programação 2/5

- Expressividade: Qual a forma mais natural de representar os elementos da Linguagem que proveem os requisitos desejados?
  - Linguagens devem permitir naturalmente a implementação de certos problemas, pelo menos em uma área de aplicação científica.



# Características de uma Linguagem de Programação 3/5

- Paradigma: Qual a forma mais adequada para representar os problemas a serem resolvidos (álgebra, lógica, composição de funções, sequência de operações, entre outros)
  - E qual o paradigma de programação mais apropriado para se resolver problemas dos domínios de aplicação desejados?



# Características de uma Linguagem de Programação 4/5

- Implementação: Os requisitos, juntamente com sua forma de representação, são passíveis de implementação?
  - Devem ser implementáveis, ou seja, deve ser possível executar todo programa bem formado (sintaticamente correto).



# Características de uma Linguagem de Programação 5/5

- Eficiência: Os requisitos são implementados em um patamar aceitável de eficiência.



# Conceitos de Linguagens

- Um entendimento básico de conceitos de Linguagens é indispensável para se poder aprender novas Linguagens e decidir que Linguagem é mais adequada para a solução de um problema;
- As formas como os diversos conceitos de Linguagens são agrupados combinado a estilos de programação caracterizam o que chamamos de **Paradigmas de Linguagem de Programação**;
- Atualmente as LP modernas incorporam um ou mais paradigmas.



# Paradigmas de Programação

- São estilos/padrões de programação;
- Paradigmas a serem estudados:
  - Procedural (C)
  - Funcional (Lisp)
  - Lógico (Prolog)
  - Orientada a Objetos (C++)



# Paradigma Procedural

- Estilo de programação convencional (programas são decompostos em “passos” de processamento);
- Rotinas são usadas como unidades de modularização para definir tais “passos” de processamento;
- Exemplos: Pascal e C



# Paradigma Funcional

- Estilo de programação que tem origem na teoria das funções matemáticas;
- Enfatiza o processamento de valores através do uso de expressões e funções;
- As funções são os blocos de construção primários do programa (passagem por parâmetro ou por valor);
- Exemplos: ML e LISP



# Paradigma Orientada a Objetos

- Enfatiza a definição de classes de objetos;
- Instâncias de classes são criadas através do programa conforme a necessidade;
- É baseado na definição de hierarquias de classes e seleção em tempo de execução;
- Exemplos: Smaltalk, C++ e Java



# Paradigma Lógico

- Enfatiza a descrição declarativa de um problema, ao invés da decomposição do problema em uma implementação algorítmica;
- São mais próximos de uma especificação do que uma implementação tradicional;
- São baseadas em lógica ou regras;
- Exemplo: Prolog



# Conceito de Linguagem de Programação

- “Uma Linguagem de Programação é uma linguagem com o objetivo de ser usada por uma pessoa para expressar um processo através do qual um computador pode resolver um problema.”



# Conceito de Linguagem de Programação

- Em outras palavras, a LP faz a ligação entre o pensamento humano e a precisão requerida para o processamento da máquina;
- Quatro componentes chave nesta definição de LP são:
  - Computador
  - Pessoa
  - Processo
  - problema



# Computador

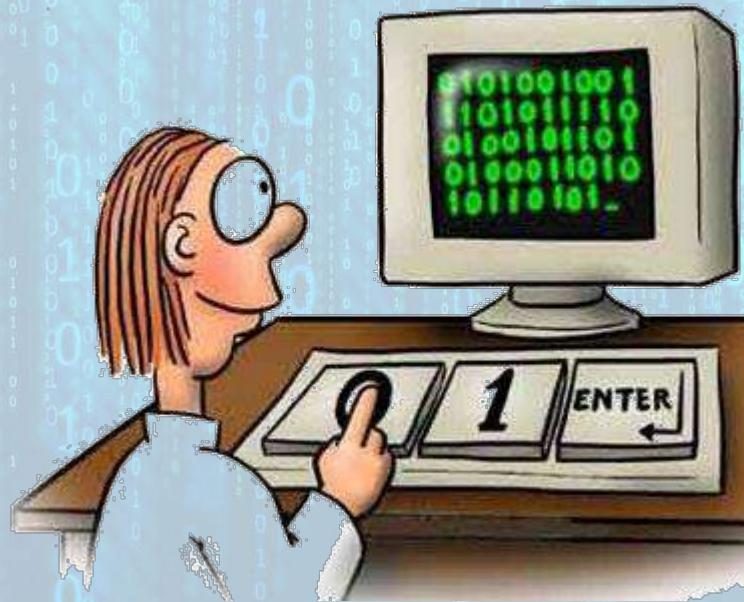
- A máquina que executará o processo descrito através do programa.





# Pessoa

- O programador, que serve como a origem da comunicação.





- A atividade que está sendo descrita através do programa de computador.





# Problema

- O sistema atual ou ambiente onde o problema surgiu.





# Conceitos Básicos

- Paradigma;
- Léxico;
- Sintaxe;
- Semântica.



# Paradigma

- Conjunto de regras estabelecidas no atual momento;
- Fronteiras (domínios) e descrevem como resolver os problemas dentro destas fronteiras;
- Exemplo ou modelo que servem como norma a ser seguida;
- Forma de representar e manipular o conhecimento.



# Léxico

- É o conjunto de palavras que compõem a Linguagem;
- Também chamadas de Palavras Reservadas.



# Sintaxe

- É o conjunto de regras que determina quais construções são corretas para a formação de programas;
- Preocupa-se com a “forma” dos programas (expressões, comandos, declarações, etc.).



# Semântica

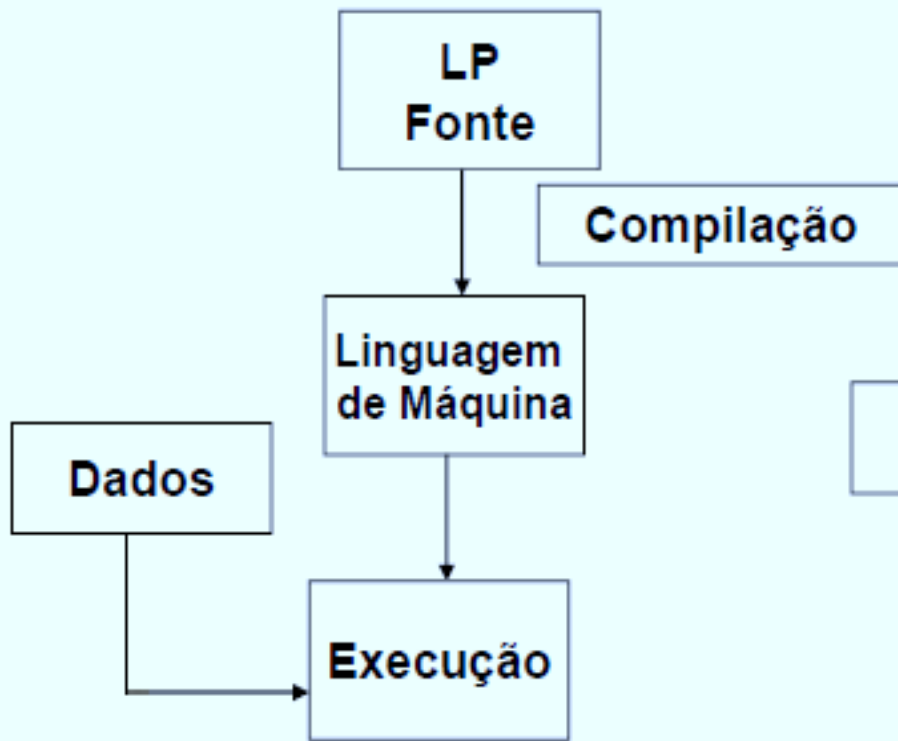
- É a descrição da maneira que um programa sintaticamente correto é interpretado ou executado;
- Preocupa-se com o “significado” dos programas (comportamento ao ser executado).
- Exemplo: a sintaxe do if em Pascal é:  
**IF(<expressão condicional>) THEN  
    <instrução> ELSE <instrução>**

Significado: Se o resultado da expressão condicional for verdadeiro, executa a instrução, do contrário não.

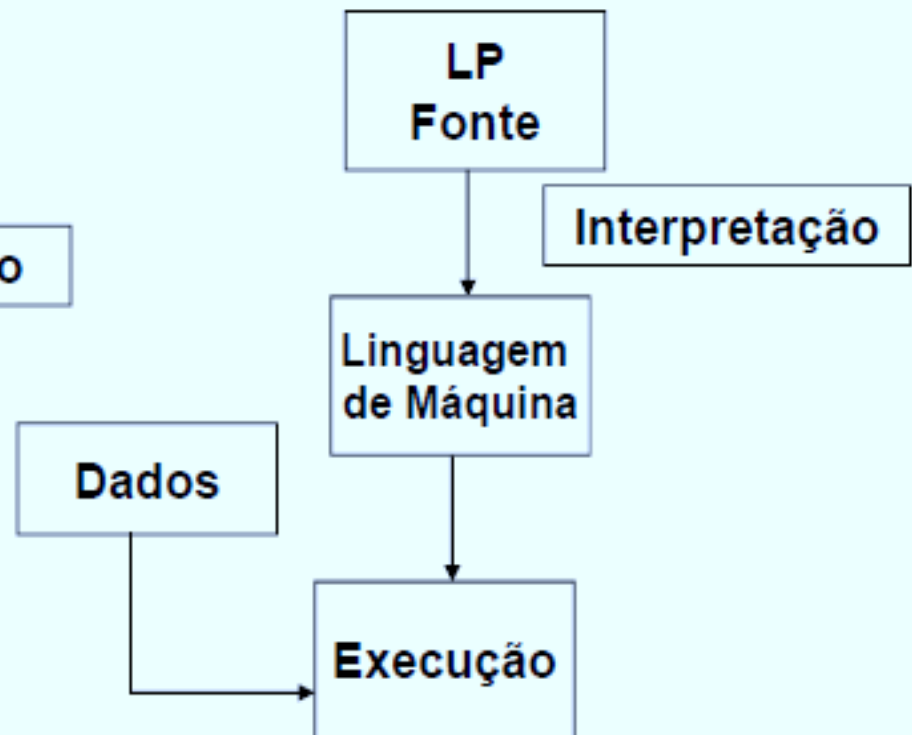


# Implementação de LP

## Compilação

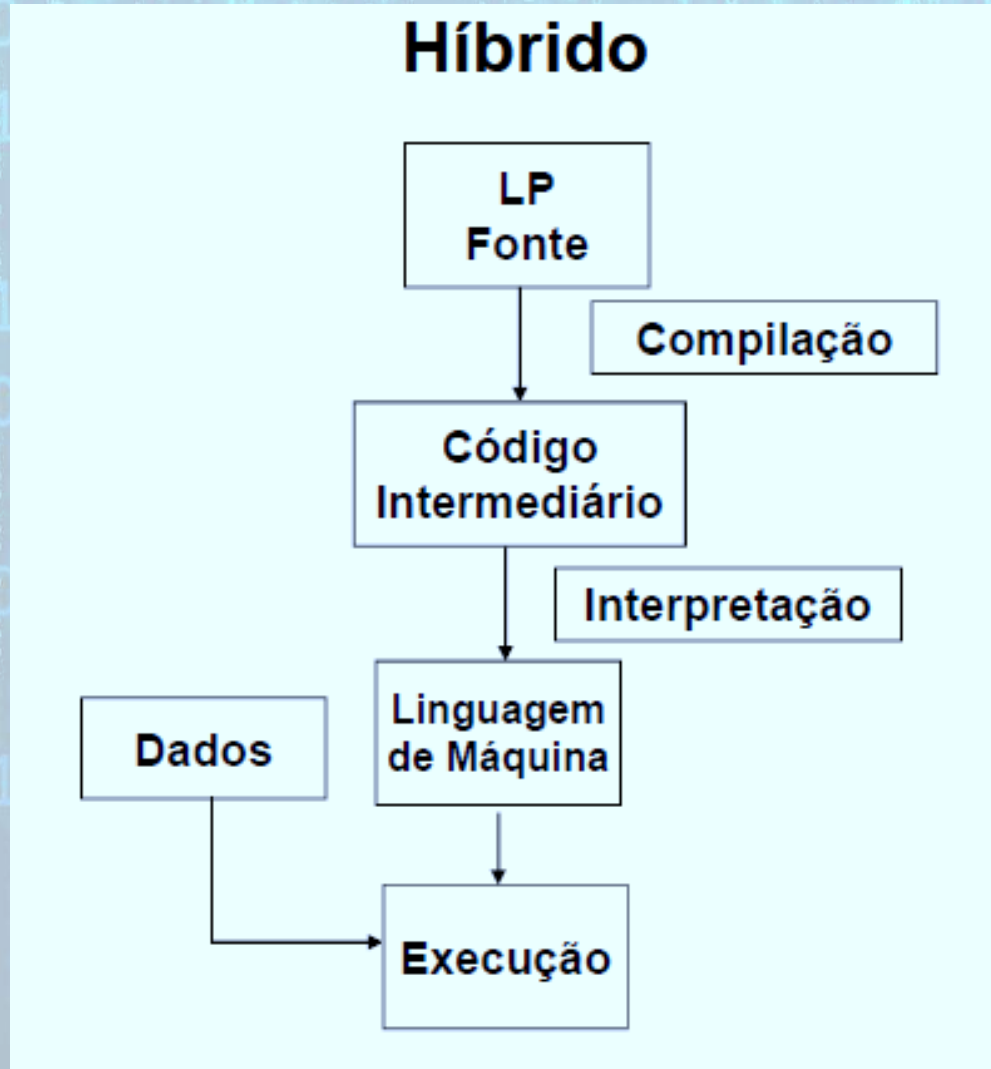


## Implementação Pura





# Implementação de LP





# Implementação de LP

- **Compilação:**
  - Eficiência;
  - Problemas com portabilidade e depuração.
- **Interpretação Pura:**
  - Flexibilidade, portabilidade e facilidade de depuração;
  - Problemas com eficiência.
- **Híbrido:**
  - Une algumas vantagens dos outros métodos;
  - JVM (Máquina Virtual Java)



# Compiladores e Máquinas Virtuais

- Qualquer implementação de uma Linguagem de Programação requer que os programas nessa linguagem sejam analisados e então traduzidos em um formato que possa ser:
  - Executado por um computador (“máquina real”), chamada de **Compilação**;
  - Executado por um interpretador (um software que simule uma “máquina virtual” e seja executado em uma máquina real), chamada de **interpretação**.

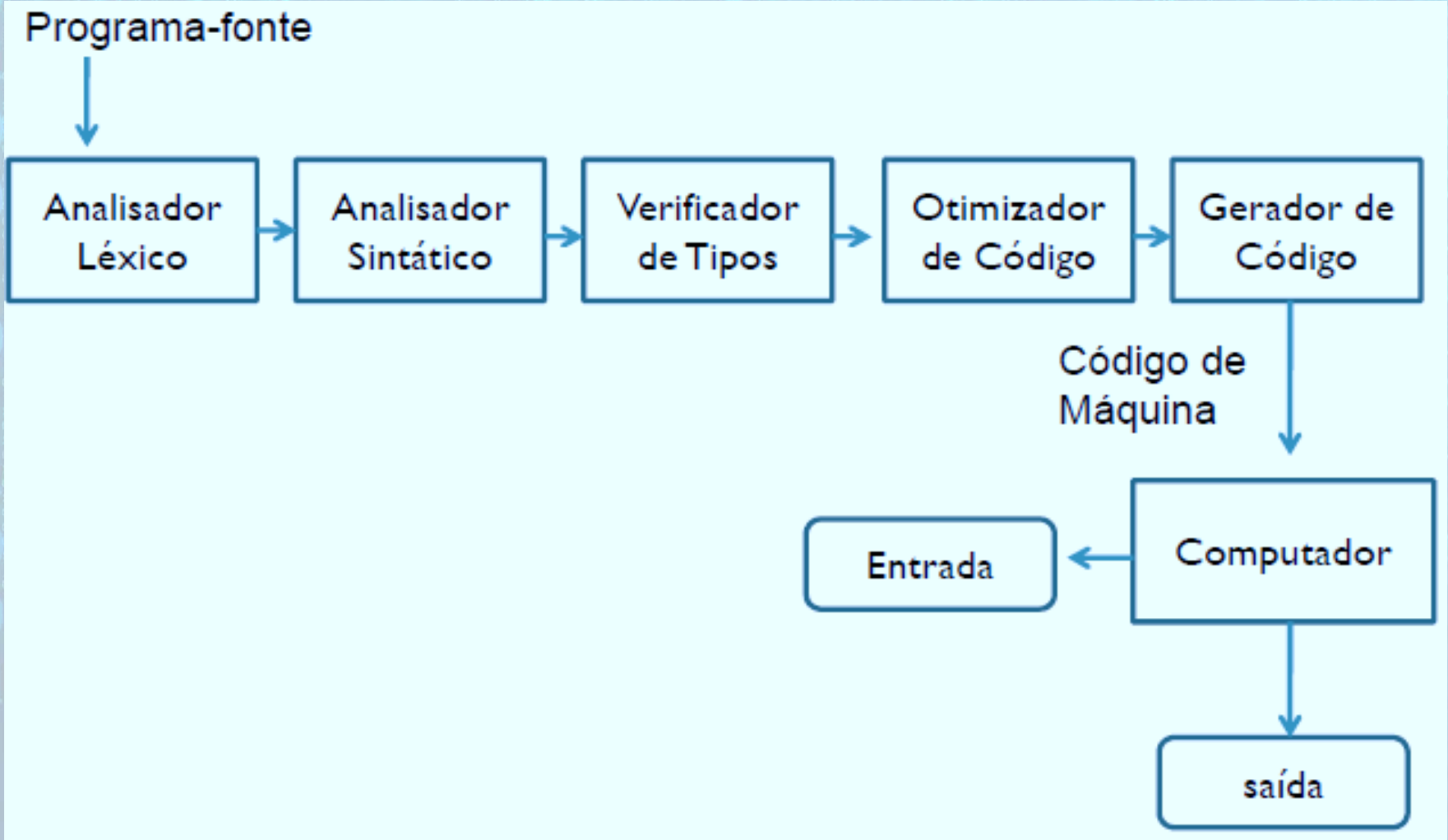


# Compiladores

- O processo de compilação traduz um programa-fonte para a linguagem de um computador (linguagem de máquina);
- Após o processo o código de máquina resultante pode ser executado no computador.



# Processo de Compilação e Execução



# Máquinas Virtuais e Interpretadores

Programa-fonte

