

Compiladores

Aula 1: Apresentacao da Disciplina

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

CEFET-MG - Campus Timoteo
Dep. Engenharia de Computação

Fevereiro de 2026

- 1 Introdução
- 2 A Disciplina
- 3 Bibliografia

- Compreender o funcionamento e regras da disciplina.
- Conhecer o **Projeto Prático: Mini-Pascal**.
- Conectar Compiladores com:
 - Estruturas de Dados
 - Linguagens de Programacao
 - Teoria da Computacao

Fundamentacao Teorica (60%)

Base matematica e algoritmica.

- Automatos e Expressoes Regulares
- Gramaticas Livres de Contexto
- Sistemas de Tipos
- Otimizacao deCodigo

Projeto Pratico (40%)

Compilador Mini-Pascal

- Construcao incremental.
- Alvo: JVM (Java Bytecode).
- Ferramentas: Java 21, ANTLR4.

Mini-Pascal: Etapas 1-4 (Front-end)

1. **Infraestrutura e AST**: Definir classes da Arvore Sintatica Abstrata.
2. **Analise Lexica**: Implementar Scanner Manual (Char -> Token).
3. **Analise Sintatica**: Implementar Parser Descendente Recursivo.
4. **Ferramentas (ANTLR4)**: Migrar para gerador de parser automatizado.

5. **Semantica (Escopo)**: Tabelas de Simbolos e resolucao de nomes.
6. **Semantica (Tipos)**: Verificacao de compatibilidade (Type Checking).
7. **Geracao deCodigo**: Traducao da AST para Bytecode JVM (ASM).

1. Estruturas de Dados (AED2)

O compilador transforma estruturas de dados complexas.

- **Arvores (AST):** Estrutura central da representacao do codigo.
- **Tabelas Hash:** Base da Tabela de Simbolos ($O(1)$).
- **Grafos:**
 - Fluxo de Controle (Loops, ifs).
 - Interferencias (Alocacao de Registradores).

2. Linguagens de Programacao

Entendendo “sob o capô”:

- **Escopo:** Como funciona ‘ int x; ‘? (Pilha de Tabelas).
- **Tipos:** Estatico vs Dinamico. Coercao.
- **Runtime:** Stack (Chamadas de funcao) vs Heap (Objetos dinâmicos).

3. Linguagens Formais

A matemática garante a correção.

- **Expressões Regulares** - Scanners eficientes e robustos.
- **Gramáticas Livres de Contexto** - Parsers que entendem aninhamento.

- **Git**: Versionamento obrigatorio (Commits atomicos).
- **Java 21**: Linguagem moderna (Records, Pattern Matching).
- **ANTLR4**: Gerador de Parser industrial.
- **IDE**: IntelliJ ou VS Code.

- Aho, et al. *Compilers: Principles, Techniques, and Tools*.
- Cooper & Torczon. *Engineering a Compiler*.

Duvidas? **Email:** alessio@cefetmg.br