

Introdução

o analisador léxico pega algo como:

```
final = (nota1 + nota2) / 2;
```

e classifica para algo como:

```
id1 = (id2 + id3) / 2
```

criando uma **Tabela de símbolos** que classifica os elementos do código, dando um id para ele e identificando seu tipo, ex:

Tabela de Símbolos

id1 | final | double | linha×coluna onde aparece

id2 | nota1 | double | linha×coluna onde aparece

id3 | nota2 | double | linha×coluna onde aparece

depois disso vai para o analisador sintático, que cria uma árvore sintática

E da árvore sintática vai para o analisador semântico, que identifica os tipos dos tokens

E finaliza no gerador de código, que traduz o código escrito para um código mais amigável para a máquina

Análise léxica

analisa o programa caractere por caractere que classifica em conjuntos:

- palavras chave (if, else, for, ...)
- operadores (+, -, *, ...)
- constantes (1, 1.0, 'a', ...)
- literais ("hello world")
- símbolos de pontuação (";", {}, ...)
- labels

constanteInt pode ser visto como: digito digito*

constanteDouble pode ser visto como: digito digito*.digito*

em que digito pertence à {0, 2, 3, 4, 5, 6, 7, 8, 9}

quando for armazenar na tabela de símbolos ele identifica que 01 e 1 são iguais

o programa fonte é lido pelo analisador léxico, que manda o token para o analisador sintático, este pede outro token para o analisador léxico. O analisador sintático gera a tabela de símbolos, que passa para o analisador semântico.

