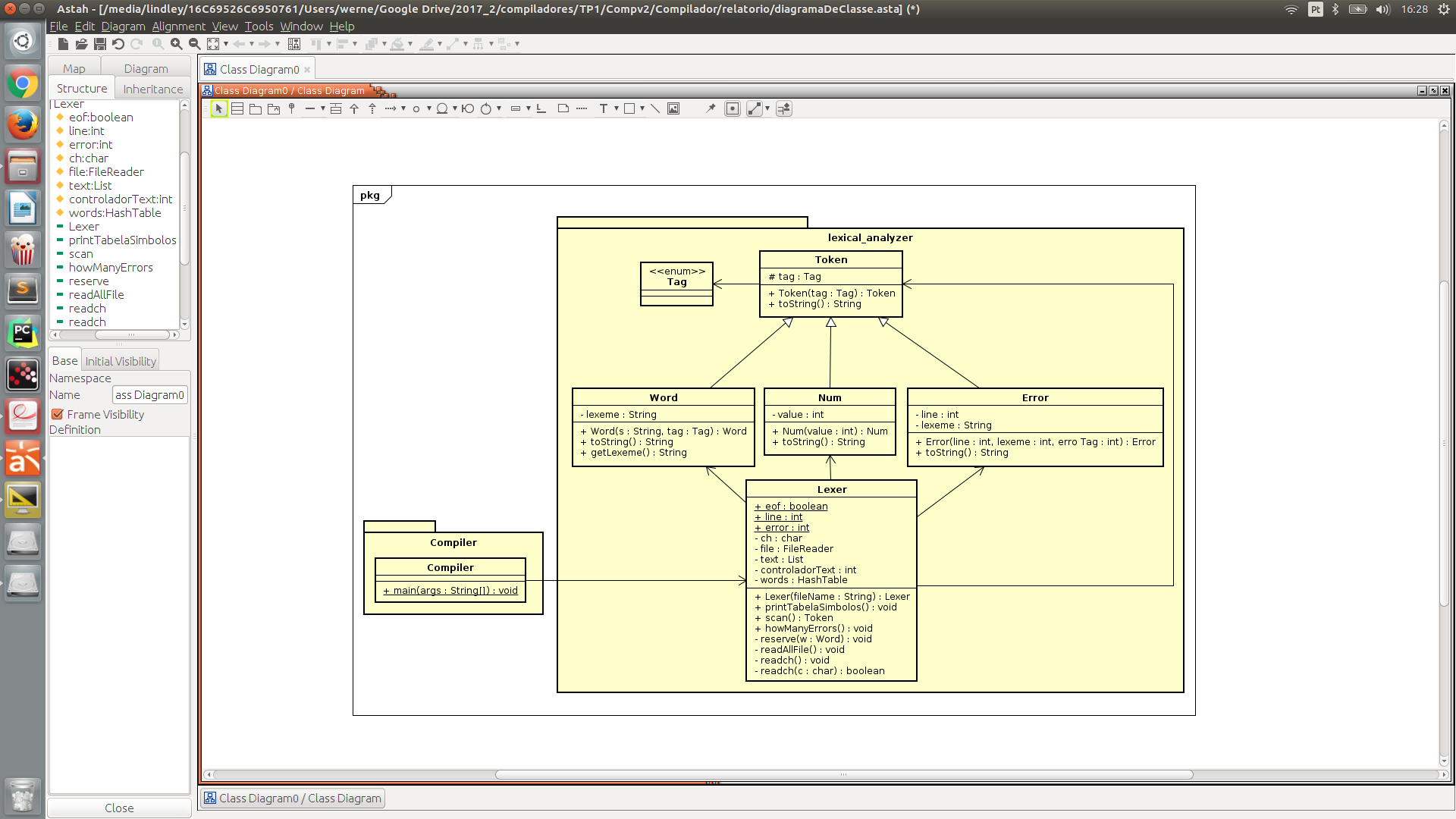
**Relatório Analisador Léxico**

Alunos: Lindley Werner, Nícolas Maduro.

**Implementação**

O Analisador Léxico foi feito em Java. Utilizando a IDE Netbeans. A implementação está descrita no diagrama de Classes abaixo.



**Casos de testes**

A seguir são mostrados os resultados dos testes feitos.

Todos eles seguem o seguinte padrão:

* testeX.txt (proposto pela professora)
* testeXok.txt (código corrigido)

Cada um dos testes mostra o **Fluxo de tokens, quantidade de erros léxicos** e **tabela de símbolos**, nesta ordem.

* **Teste 1**

**Código original (teste1.txt)**

Digite o nome do arquivo a ser compilado:

teste1.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("b")>

<DOT\_COMMA (;)>

<INT ("int")>

<ID ("result")>

<DOT\_COMMA (;)>

<ID ("float")>

<ID ("a")>

<COMMA (,)>

<ID ("x")>

<COMMA (,)>

<ID ("total")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<NUM ("2")>

<DOT\_COMMA (;)>

<ID ("x")>

<ASSIGN (=)>

<ERROR\_CARACTER\_INVALIDO (linha: 7, caracter inválido: ".")>

<NUM ("1")>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("y")>

<CLOSE\_PAREN ())>

<ID ("result")>

<ASSIGN (=)>

<OPEN\_PAREN (()>

<ID ("a")>

<MULT (\*)>

<ID ("b")>

<PLUS (+)>

<PLUS (+)>

<NUM ("1")>

<CLOSE\_PAREN ())>

<DIV (/)>

<NUM ("2")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<STRING ("Resultado: ")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("result")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Total: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("total")>

<ASSIGN (=)>

<ID ("y")>

<DIV (/)>

<ID ("x")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Total: ")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("total")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

1 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<PROGRAM ("program")>

<ID ("y")>

<ID ("x")>

<ID ("result")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ELSE ("else")>

<THEN ("then")>

<ID ("b")>

<ID ("a")>

<ID ("float")>

<ID ("total")>

**Código corrigido (teste1ok.txt)**

Digite o nome do arquivo a ser compilado:

teste1ok.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("b")>

<DOT\_COMMA (;)>

<INT ("int")>

<ID ("result")>

<DOT\_COMMA (;)>

<ID ("float")>

<ID ("a")>

<COMMA (,)>

<ID ("x")>

<COMMA (,)>

<ID ("total")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<NUM ("2")>

<DOT\_COMMA (;)>

<ID ("x")>

<ASSIGN (=)>

<NUM ("1")>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("y")>

<CLOSE\_PAREN ())>

<ID ("result")>

<ASSIGN (=)>

<OPEN\_PAREN (()>

<ID ("a")>

<MULT (\*)>

<ID ("b")>

<PLUS (+)>

<PLUS (+)>

<NUM ("1")>

<CLOSE\_PAREN ())>

<DIV (/)>

<NUM ("2")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<STRING ("Resultado: ")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("result")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Total: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("total")>

<ASSIGN (=)>

<ID ("y")>

<DIV (/)>

<ID ("x")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Total: ")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("total")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

0 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<PROGRAM ("program")>

<ID ("y")>

<ID ("x")>

<ID ("result")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ELSE ("else")>

<THEN ("then")>

<ID ("b")>

<ID ("a")>

<ID ("float")>

<ID ("total")>

* **Teste 2**

**Código original (teste2.txt)**

Digite o nome do arquivo a ser compilado:

teste2.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ERROR\_CARACTER\_INVALIDO (linha: 3, caracter inválido: ":")>

<ID ("a")>

<COMMA (,)>

<ID ("c")>

<DOT\_COMMA (;)>

<ID ("float")>

<ID ("d")>

<COMMA (,)>

<ERROR\_CARACTER\_INVALIDO (linha: 4, caracter inválido: "\_")>

<ID ("e")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<NUM ("0")>

<DOT\_COMMA (;)>

<ID ("d")>

<ASSIGN (=)>

<NUM ("3")>

<ERROR\_CARACTER\_INVALIDO (linha: 6, caracter inválido: ".")>

<NUM ("5")>

<ID ("c")>

<ASSIGN (=)>

<ID ("d")>

<DIV (/)>

<NUM ("1")>

<ERROR\_CARACTER\_INVALIDO (linha: 7, caracter inválido: ".")>

<NUM ("2")>

<DOT\_COMMA (;)>

<ID ("Scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("Scan")>

<OPEN\_PAREN (()>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("b")>

<ASSIGN (=)>

<ID ("a")>

<MULT (\*)>

<ID ("a")>

<DOT\_COMMA (;)>

<ID ("c")>

<ASSIGN (=)>

<ID ("b")>

<PLUS (+)>

<ID ("a")>

<MULT (\*)>

<OPEN\_PAREN (()>

<NUM ("1")>

<PLUS (+)>

<ID ("a")>

<MULT (\*)>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Resultado: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<ID ("c")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<ID ("b")>

<PLUS (+)>

<ID ("c")>

<PLUS (+)>

<ID ("d")>

<CLOSE\_PAREN ())>

<DIV (/)>

<NUM ("22")>

<DOT\_COMMA (;)>

<ID ("e")>

<ASSIGN (=)>

<ID ("val")>

<PLUS (+)>

<ID ("c")>

<PLUS (+)>

<ID ("a")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("E: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("e")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

4 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<PROGRAM ("program")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ELSE ("else")>

<ID ("Scan")>

<ID ("e")>

<THEN ("then")>

<ID ("d")>

<ID ("c")>

<ID ("b")>

<ID ("a")>

<ID ("float")>

<ID ("val")>

**Código corrigido (teste2ok.txt)**

Digite o nome do arquivo a ser compilado:

teste2ok.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("c")>

<DOT\_COMMA (;)>

<ID ("float")>

<ID ("d")>

<COMMA (,)>

<ID ("e")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<NUM ("0")>

<DOT\_COMMA (;)>

<ID ("d")>

<ASSIGN (=)>

<NUM ("35")>

<ID ("c")>

<ASSIGN (=)>

<ID ("d")>

<DIV (/)>

<NUM ("12")>

<DOT\_COMMA (;)>

<ID ("Scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("Scan")>

<OPEN\_PAREN (()>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("b")>

<ASSIGN (=)>

<ID ("a")>

<MULT (\*)>

<ID ("a")>

<DOT\_COMMA (;)>

<ID ("c")>

<ASSIGN (=)>

<ID ("b")>

<PLUS (+)>

<ID ("a")>

<MULT (\*)>

<OPEN\_PAREN (()>

<NUM ("1")>

<PLUS (+)>

<ID ("a")>

<MULT (\*)>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Resultado: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<ID ("c")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<ID ("b")>

<PLUS (+)>

<ID ("c")>

<PLUS (+)>

<ID ("d")>

<CLOSE\_PAREN ())>

<DIV (/)>

<NUM ("22")>

<DOT\_COMMA (;)>

<ID ("e")>

<ASSIGN (=)>

<ID ("val")>

<PLUS (+)>

<ID ("c")>

<PLUS (+)>

<ID ("a")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("E: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("e")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

0 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<PROGRAM ("program")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ELSE ("else")>

<ID ("Scan")>

<ID ("e")>

<THEN ("then")>

<ID ("d")>

<ID ("c")>

<ID ("b")>

<ID ("a")>

<ID ("float")>

<ID ("val")>

* **Teste 3**

**Código original (teste3.txt)**

Digite o nome do arquivo a ser compilado:

teste3.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("pontuacao")>

<COMMA (,)>

<ID ("pontuacaoMaxima")>

<COMMA (,)>

<ID ("disponibilidade")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("pontuacaoMinima")>

<DOT\_COMMA (;)>

<ID ("disponibilidade")>

<ASSIGN (=)>

<STRING ("Sim")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMinima")>

<ASSIGN (=)>

<NUM ("50")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMaxima")>

<ASSIGN (=)>

<NUM ("100")>

<DOT\_COMMA (;)>

<ERROR\_CARACTER\_INESPERADO (linha: 9, esperado: "\*/")>

1 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<ID ("pontuacao")>

<ID ("disponibilidade")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<STRING ("string")>

<INT ("int")>

<ID ("pontuacaoMaxima")>

<ELSE ("else")>

<ID ("pontuacaoMinima")>

<SCAN ("scan")>

**Código corrigido (teste3ok.txt)**

Digite o nome do arquivo a ser compilado:

teste3ok.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("pontuacao")>

<COMMA (,)>

<ID ("pontuacaoMaxima")>

<COMMA (,)>

<ID ("disponibilidade")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("pontuacaoMinima")>

<DOT\_COMMA (;)>

<ID ("disponibilidade")>

<ASSIGN (=)>

<STRING ("Sim")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMinima")>

<ASSIGN (=)>

<NUM ("50")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMaxima")>

<ASSIGN (=)>

<NUM ("100")>

<DOT\_COMMA (;)>

<BLOCK\_COMMENT (/\*\*/)>

<DO ("do")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Pontuacao Candidato: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("pontuacao")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Disponibilidade Candidato: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<ID ("and")>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ELSE ("else")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato reprovado")>

<CLOSE\_PAREN ())>

<END ("end")>

<WHILE ("while")>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GE (>=)>

<NUM ("0")>

<CLOSE\_PAREN ())>

<END ("end")>

<END ("end")>

0 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<ID ("pontuacao")>

<ID ("disponibilidade")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<ID ("and")>

<STRING ("string")>

<INT ("int")>

<ID ("pontuação")>

<ID ("pontuacaoMaxima")>

<ELSE ("else")>

<ID ("pontuacaoMinima")>

<SCAN ("scan")>

* **Teste 4**

**Código original (teste4.txt)**

Digite o nome do arquivo a ser compilado:

teste4.txt

Fluxo de Tokens

<INT ("int")>

<ERROR\_CARACTER\_INVALIDO (linha: 1, caracter inválido: ":")>

<ID ("a")>

<COMMA (,)>

<ID ("aux")>

<ERROR\_CARACTER\_INVALIDO (linha: 1, caracter inválido: "$")>

<COMMA (,)>

<ID ("b")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("nome")>

<COMMA (,)>

<ID ("sobrenome")>

<COMMA (,)>

<ID ("msg")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("Nome")>

<ERROR\_CARACTER\_INVALIDO (linha: 4, caracter inválido: ":")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("nome")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Sobrenome: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("sobrenome")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("msg")>

<ASSIGN (=)>

<STRING ("Ola, ")>

<PLUS (+)>

<ID ("nome")>

<PLUS (+)>

<STRING (" ")>

<PLUS (+)>

<ID ("sobrenome")>

<PLUS (+)>

<STRING ("!")>

<DOT\_COMMA (;)>

<ID ("msg")>

<ASSIGN (=)>

<ID ("msg")>

<PLUS (+)>

<NUM ("1")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("msg")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<IF ("if")>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("b")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<ID ("aux")>

<ASSIGN (=)>

<ID ("b")>

<DOT\_COMMA (;)>

<ID ("b")>

<ASSIGN (=)>

<ID ("a")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<ID ("aux")>

<DOT\_COMMA (;)>

<END ("end")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Apos a troca: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("out")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("out")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<END ("end")>

3 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<ID ("msg")>

<PROGRAM ("program")>

<ID ("sobrenome")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ID ("nome")>

<ID ("out")>

<ELSE ("else")>

<THEN ("then")>

<ID ("b")>

<ID ("a")>

<ID ("aux")>

<ID ("Nome")>

**Código corrigido (teste4ok.txt)**

Digite o nome do arquivo a ser compilado:

teste4ok.txt

Fluxo de Tokens

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("aux")>

<COMMA (,)>

<ID ("b")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("nome")>

<COMMA (,)>

<ID ("sobrenome")>

<COMMA (,)>

<ID ("msg")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("Nome")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("nome")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Sobrenome: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("sobrenome")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("msg")>

<ASSIGN (=)>

<STRING ("Ola, ")>

<PLUS (+)>

<ID ("nome")>

<PLUS (+)>

<STRING (" ")>

<PLUS (+)>

<ID ("sobrenome")>

<PLUS (+)>

<STRING ("!")>

<DOT\_COMMA (;)>

<ID ("msg")>

<ASSIGN (=)>

<ID ("msg")>

<PLUS (+)>

<NUM ("1")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("msg")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<IF ("if")>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("b")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<ID ("aux")>

<ASSIGN (=)>

<ID ("b")>

<DOT\_COMMA (;)>

<ID ("b")>

<ASSIGN (=)>

<ID ("a")>

<DOT\_COMMA (;)>

<ID ("a")>

<ASSIGN (=)>

<ID ("aux")>

<DOT\_COMMA (;)>

<END ("end")>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Apos a troca: ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("out")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ID ("out")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<END ("end")>

0 erro(s) léxico(s)

Tabela de Simbolos

<STRING ("string")>

<INT ("int")>

<PRINT ("print")>

<END ("end")>

<ID ("msg")>

<PROGRAM ("program")>

<ID ("sobrenome")>

<SCAN ("scan")>

<IF ("if")>

<WHILE ("while")>

<DO ("do")>

<ID ("nome")>

<ID ("out")>

<ELSE ("else")>

<THEN ("then")>

<ID ("b")>

<ID ("a")>

<ID ("aux")>

<ID ("Nome")>

* **Teste 5**

**Código original (teste5.txt)**

Digite o nome do arquivo a ser compilado:

teste5.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("b")>

<COMMA (,)>

<ID ("c")>

<COMMA (,)>

<ID ("maior")>

<COMMA (,)>

<ID ("outro")>

<DOT\_COMMA (;)>

<DO ("do")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("A")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("B")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("C")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<LINE\_COMMENT (//)>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("b")>

<CLOSE\_PAREN ())>

<AND (&&)>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("c")>

<CLOSE\_PAREN ())>

<CLOSE\_PAREN ())>

<ID ("maior")>

<ASSIGN (=)>

<ID ("a")>

<ELSE ("else")>

<IF ("if")>

<OPEN\_PAREN (()>

<ID ("b")>

<GT (>)>

<ID ("c")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<ID ("maior")>

<ASSIGN (=)>

<ID ("b")>

<DOT\_COMMA (;)>

<ELSE ("else")>

<ID ("maior")>

<ASSIGN (=)>

<ID ("c")>

<DOT\_COMMA (;)>

<END ("end")>

<END ("end")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Maior valor:")>

<ERROR\_CARACTER\_INESPERADO (linha: 24, esperado: """)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("maior")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Outro? ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("outro")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<WHILE ("while")>

<OPEN\_PAREN (()>

<ID ("outro")>

<GE (>=)>

<NUM ("0")>

<CLOSE\_PAREN ())>

<END ("end")>

1 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<ID ("maior")>

<STRING ("string")>

<INT ("int")>

<ID ("outro")>

<ID ("c")>

<ID ("b")>

<ELSE ("else")>

<ID ("a")>

<SCAN ("scan")>

**Código corrigido (teste5ok.txt)**

Digite o nome do arquivo a ser compilado:

teste5ok.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("a")>

<COMMA (,)>

<ID ("b")>

<COMMA (,)>

<ID ("c")>

<COMMA (,)>

<ID ("maior")>

<COMMA (,)>

<ID ("outro")>

<DOT\_COMMA (;)>

<DO ("do")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("A")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("a")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("B")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("b")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("C")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("c")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<LINE\_COMMENT (//)>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("b")>

<CLOSE\_PAREN ())>

<AND (&&)>

<OPEN\_PAREN (()>

<ID ("a")>

<GT (>)>

<ID ("c")>

<CLOSE\_PAREN ())>

<CLOSE\_PAREN ())>

<ID ("maior")>

<ASSIGN (=)>

<ID ("a")>

<ELSE ("else")>

<IF ("if")>

<OPEN\_PAREN (()>

<ID ("b")>

<GT (>)>

<ID ("c")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<ID ("maior")>

<ASSIGN (=)>

<ID ("b")>

<DOT\_COMMA (;)>

<ELSE ("else")>

<ID ("maior")>

<ASSIGN (=)>

<ID ("c")>

<DOT\_COMMA (;)>

<END ("end")>

<END ("end")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Maior valor:")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<ID ("maior")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Outro? ")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<SCAN ("scan")>

<OPEN\_PAREN (()>

<ID ("outro")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<WHILE ("while")>

<OPEN\_PAREN (()>

<ID ("outro")>

<GE (>=)>

<NUM ("0")>

<CLOSE\_PAREN ())>

<END ("end")>

0 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<ID ("maior")>

<STRING ("string")>

<INT ("int")>

<ID ("outro")>

<ID ("c")>

<ID ("b")>

<ELSE ("else")>

<ID ("a")>

<SCAN ("scan")>

* **Teste 6**

**Código do teste6.txt**

program

int @ pontuacao, pontuacaoMaxima, disponibilidade;

string pontuacaoMinima;

disponibilidade = "Sim";

pontuacaoMinima = 50;

pontuacaoMaxima = 100;

if ((pontuação > pontuacaoMinima) & (disponibilidade=="Sim") then

print("Candidato aprovado");

else

if ((pontuação > pontuacaoMinima) | (disponibilidade=="Sim") then

print("Candidato aprovado");

end

if ((pontuação > pontuacaoMinima) || (disponibilidade=="Sim") then

print("Candidato aprovado");

end

end

end

**Código original (teste6.txt)**

Digite o nome do arquivo a ser compilado:

teste6.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ERROR\_CARACTER\_INVALIDO (linha: 2, caracter inválido: "@")>

<ID ("pontuacao")>

<COMMA (,)>

<ID ("pontuacaoMaxima")>

<COMMA (,)>

<ID ("disponibilidade")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("pontuacaoMinima")>

<DOT\_COMMA (;)>

<ID ("disponibilidade")>

<ASSIGN (=)>

<STRING ("Sim")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMinima")>

<ASSIGN (=)>

<NUM ("50")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMaxima")>

<ASSIGN (=)>

<NUM ("100")>

<DOT\_COMMA (;)>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<ERROR\_CARACTER\_INESPERADO (linha: 9, esperado: "&")>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ELSE ("else")>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<ERROR\_CARACTER\_INESPERADO (linha: 12, esperado: "|")>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<OR (||)>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

<END ("end")>

<END ("end")>

3 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<ID ("pontuacao")>

<ID ("disponibilidade")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<STRING ("string")>

<INT ("int")>

<ID ("pontuação")>

<ID ("pontuacaoMaxima")>

<ELSE ("else")>

<ID ("pontuacaoMinima")>

<SCAN ("scan")>

**Código corrigido (teste6ok.txt)**

Digite o nome do arquivo a ser compilado:

teste6ok.txt

Fluxo de Tokens

<PROGRAM ("program")>

<INT ("int")>

<ID ("pontuacao")>

<COMMA (,)>

<ID ("pontuacaoMaxima")>

<COMMA (,)>

<ID ("disponibilidade")>

<DOT\_COMMA (;)>

<STRING ("string")>

<ID ("pontuacaoMinima")>

<DOT\_COMMA (;)>

<ID ("disponibilidade")>

<ASSIGN (=)>

<STRING ("Sim")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMinima")>

<ASSIGN (=)>

<NUM ("50")>

<DOT\_COMMA (;)>

<ID ("pontuacaoMaxima")>

<ASSIGN (=)>

<NUM ("100")>

<DOT\_COMMA (;)>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<AND (&&)>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<ELSE ("else")>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<OR (||)>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

<IF ("if")>

<OPEN\_PAREN (()>

<OPEN\_PAREN (()>

<ID ("pontuação")>

<GT (>)>

<ID ("pontuacaoMinima")>

<CLOSE\_PAREN ())>

<OR (||)>

<OPEN\_PAREN (()>

<ID ("disponibilidade")>

<EQ (==)>

<STRING ("Sim")>

<CLOSE\_PAREN ())>

<THEN ("then")>

<PRINT ("print")>

<OPEN\_PAREN (()>

<STRING ("Candidato aprovado")>

<CLOSE\_PAREN ())>

<DOT\_COMMA (;)>

<END ("end")>

<END ("end")>

<END ("end")>

0 erro(s) léxico(s)

Tabela de Simbolos

<PRINT ("print")>

<IF ("if")>

<ID ("pontuacao")>

<ID ("disponibilidade")>

<END ("end")>

<WHILE ("while")>

<DO ("do")>

<THEN ("then")>

<PROGRAM ("program")>

<STRING ("string")>

<INT ("int")>

<ID ("pontuação")>

<ID ("pontuacaoMaxima")>

<ELSE ("else")>

<ID ("pontuacaoMinima")>

<SCAN ("scan")>