

Isabella Vieira Ferreira  
Mônica Neli de Resende

# Analizador Léxico e Sintático de um Compilador para a Linguagem C

Projeto apresentado à disciplina de  
Compiladores do curso de Ciência  
da Computação da Universidade  
Federal de São João del-Rei.

Orientador: Prof Flávio Luiz  
Schiavoni

São João del-Rei  
2015

---

# Sumário

<b>1</b>	<b>Objetivo do trabalho</b>	<b>3</b>
<b>2</b>	<b>Analizador Léxico e Sintático</b>	<b>3</b>
2.1	Analizador Léxico . . . . .	3
2.1.1	Descrição dos tokens e do arquivo <i>arquivoTokens.txt</i> . . . . .	3
2.1.2	Descrição do arquivo <i>arquivoConstantes.txt</i> . . . . .	6
2.1.3	Descrição do arquivo <i>arquivoIdentificadores.txt</i> . . . . .	7
2.1.4	Descrição do arquivo <i>arquivoLiterais.txt</i> . . . . .	8
2.1.5	Descrição do arquivo <i>arquivoErrosLexicos.txt</i> . . . . .	9
2.2	Analizador Sintático . . . . .	9
2.2.1	Restrições da gramática . . . . .	12
2.2.2	Descrição do arquivo <i>arquivoErrosSintaticos.txt</i> . . . . .	12
<b>3</b>	<b>Exemplo</b>	<b>12</b>
3.1	Arquivo <i>arquivoTokens.txt</i> . . . . .	13
3.1.1	Arquivo <i>arquivoLiterais.txt</i> . . . . .	14
3.1.2	Arquivo <i>arquivoIdentificadores.txt</i> . . . . .	14
3.1.3	Arquivo <i>arquivoConstantes.txt</i> . . . . .	14
3.1.4	Arquivo <i>arquivoErrosLexicos.txt</i> . . . . .	14
3.1.5	Arquivo <i>arquivoErrosSintaticos.txt</i> . . . . .	15

---

## Lista de Figuras

1	Exemplo do arquivo <i>arquivoTokens.txt</i> . . . . .	6
2	Exemplo do arquivo <i>arquivoConstantes.txt</i> . . . . .	7
3	Exemplo do arquivo <i>arquivoIdentificadores.txt</i> . . . . .	8
4	Exemplo do arquivo <i>arquivoLiterais.txt</i> . . . . .	8
5	Exemplo do arquivo <i>arquivoErrosLexicos.txt</i> . . . . .	9
6	Exemplo do arquivo <i>arquivoErrosSintaticos.txt</i> . . . . .	12
7	Código-fonte . . . . .	13

## Lista de Tabelas

1	Comentário . . . . .	4
2	Atribuição . . . . .	4
3	Operadores . . . . .	4
4	Palavras Reservadas . . . . .	5
5	Literais . . . . .	5
6	Semicolon . . . . .	5
7	Delimitadores . . . . .	6

---

# 1 Objetivo do trabalho

Este trabalho visa desenvolver um analisador léxico e um analisador sintático de um compilador para a Linguagem C.

## 2 Analisador Léxico e Sintático

Os analisadores léxico e sintático foram desenvolvidos na linguagem de programação JavaScript. O código desenvolvido apresenta para o usuário as tabelas de tokens, constantes, identificadores e literais, e os erros léxicos e sintáticos encontrados no código-fonte. Além disso, é possível fazer o download do arquivo .txt referente a cada tabela descrita anteriormente.

### 2.1 Analisador Léxico

#### 2.1.1 Descrição dos tokens e do arquivo *arquivoTokens.txt*

Os tokens serão representados no arquivo *arquivoTokens.txt*. A estrutura de tal arquivo é:

<id\_tabela\_token> <id\_token> <linha> <coluna> <id\_referencia\_token>

Onde:

- <id\_tabela\_token> é um identificador gerado automaticamente à medida que novos tokens são inseridos na tabela de tokens. Cada linha do arquivo possui um identificador único.
- <id\_token> é o identificador de cada token lido do arquivo. Tais identificadores serão apresentados logo abaixo. Caso o token seja um identificador, id\_token será -1. Caso o token seja uma constante, id\_token será -2.
- <linha> é o número da linha do arquivo de entrada em que tal token foi encontrado.
- <coluna> é o número da coluna do arquivo de entrada em que tal token foi encontrado.
- <id\_referencia\_token> é o identificador que referencia os arquivos *arquivoConstantes.txt*, *arquivoIdentificadores.txt* ou *arquivoLiterais.txt*, caso o token lido seja uma constante, um identificador, ou um literal. Caso o token lido não se encaixe nas três categorias mencionadas anteriormente, o <id\_referencia\_token> será -1.

Os identificadores de cada token lido do arquivo de entrada são:

---

Token	Identificador
\\	1

Tabela 1: Comentário

Token	Identificador
=	2

Tabela 2: Atribuição

Token	Identificador
==	3
>	4
<	5
>=	6
<=	7
!=	8
&&	9
	10
!	11
%	12
+	13
-	14
/	15
*	16
^	17

Tabela 3: Operadores

---

<b>Token</b>	<b>Identificador</b>
if	18
else	19
continue	20
break	21
while	22
print	23
read	24
int	25
float	26
string	27
char	28

Tabela 4: Palavras Reservadas

<b>Token</b>	<b>Identificador</b>
”	29
,	30

Tabela 5: Literais

<b>Token</b>	<b>Identificador</b>
;	31

Tabela 6: Semicolon

---

Token	Identificador
{	32
{	33
(	34
)	35
,	36
.	37

Tabela 7: Delimitadores

Exemplo do arquivo *arquivoTokens.txt*.

id	Codigo	Linha	Coluna	id Tabela
0	23	0	0	-1
1	29	0	6	0
2	31	0	33	-1
3	25	1	0	-1
4	-1	1	4	0
5	2	1	6	-1

Figura 1: Exemplo do arquivo *arquivoTokens.txt*

### 2.1.2 Descrição do arquivo *arquivoConstantes.txt*

O arquivo *arquivoConstantes.txt* armazena os valores numéricos. O arquivo utiliza a seguinte estrutura:

<id\_tabela\_constantes> <constante> <linha> <coluna>

Onde:

- <id\_tabela\_constantes> é um identificador gerado automaticamente à medida que novas constantes são inseridas na tabela de constantes. Cada linha do arquivo possui um identificador único.
- <constante> é o valor numérico.

- 
- **<linha>** é o número da linha do arquivo de entrada em que tal token foi encontrado.
  - **<coluna>** é o número da coluna do arquivo de entrada em que tal token foi encontrado.

Exemplo do arquivo *arquivoConstantes.txt*.

id	Constante	Linha	Coluna
0	1	0	5
1	1.2	1	0
2	100	2	0

Figura 2: Exemplo do arquivo *arquivoConstantes.txt*

### 2.1.3 Descrição do arquivo *arquivoIdentificadores.txt*

O arquivo *arquivoIdentificadores.txt* armazena textos presentes no código fonte. O arquivo utiliza a seguinte estrutura:

**<id\_tabela\_identificadores>** **<identificador>** **<linha>** **<coluna>**

Onde:

- **<id\_tabela\_identificadores>** é um identificador gerado automaticamente à medida que novas variáveis (identificadores) são inseridas na tabela de identificadores. Cada linha do arquivo possui um identificador único.
- **<identificador>** é o nome do identificador (variável) presente no código-fonte.
- **<linha>** é o número da linha do arquivo de entrada em que tal token foi encontrado.
- **<coluna>** é o número da coluna do arquivo de entrada em que tal token foi encontrado.

Exemplo do arquivo *arquivoIdentificadores.txt*.



---

id	Identificador	Linha	Coluna
0	a	1	4
1	b	3	4

Figura 3: Exemplo do arquivo *arquivoIdentificadores.txt*

#### 2.1.4 Descrição do arquivo *arquivoLiterais.txt*

O arquivo *arquivoLiterais.txt* armazena textos presentes no código fonte. O arquivo utiliza a seguinte estrutura:

<id\_tabela\_literais> <literal> <linha> <coluna>

Onde:

- <id\_tabela\_literais> é um identificador gerado automaticamente à medida que novas literais são inseridas na tabela de literais. Cada linha do arquivo possui um identificador único.
- <literal> é o texto presente no código fonte.
- <linha> é o número da linha do arquivo de entrada em que tal token foi encontrado.
- <coluna> é o número da coluna do arquivo de entrada em que tal token foi encontrado.

Exemplo do arquivo *arquivoLiterais.txt*.

id	Literal	Linha	Coluna
0	Entre o primeiro valor	0	6
1	Entre o segundo valor	2	6
2	O maior valor e:	4	6

Figura 4: Exemplo do arquivo *arquivoLiterais.txt*

---

### 2.1.5 Descrição do arquivo *arquivoErrosLexicos.txt*

O arquivo *arquivoErrosLexicos.txt* tem a finalidade de mostrar os erros léxicos encontrados no código fonte.

<id\_tabela\_erros> <erro> <linha> <coluna>

Onde:

- <id\_tabela\_erros> é um identificador gerado automaticamente à medida que novos erros são inseridos na tabela de erros. Cada linha do arquivo possui um identificador único.
- <erro> é a descrição do erro encontrado
- <linha> é o número da linha do arquivo de entrada em que tal erro foi encontrado.
- <coluna> é o número da coluna do arquivo de entrada em que tal erro foi encontrado.

Exemplo do arquivo *arquivoErrosLexicos.txt*.

id	Erro	Linha	Coluna
0	Número inválido! Você acha que número e letra é a mesma coisa!?	3	0

Figura 5: Exemplo do arquivo *arquivoErrosLexicos.txt*

## 2.2 Analisador Sintático

O analisador sintático analisa se a sintaxe da linguagem está correta mediante uma gramática. Segue abaixo gramática definida para nosso compilador.

Ressaltamos que utilizamos a gramática do trabalho da dupla Arthur e Ramon do ano de 2014 como base. Além disso, fizemos a elaboração da gramática adaptada juntamente com a dupla do Lucas Carvalho e Massilon.

1. statement  $\rightarrow$  declaration
2. statement  $\rightarrow$  assignment\_expression
3. statement  $\rightarrow$  if\_statement
4. statement  $\rightarrow$  while\_statement
5. statement  $\rightarrow$  print\_statement

- 
6.  $\text{statement} \rightarrow \text{read\_statement}$
  7.  $\text{statement} \rightarrow \text{continue\_statement}$
  8.  $\text{statement} \rightarrow \text{break\_statement}$
  9.  $\text{declaration} \rightarrow \text{type\_specifier id ' ;'}$
  10.  $\text{type\_specifier} \rightarrow \text{"char"}$
  11.  $\text{type\_specifier} \rightarrow \text{"int"}$
  12.  $\text{type\_specifier} \rightarrow \text{"float"}$
  13.  $\text{type\_specifier} \rightarrow \text{"string"}$
  14.  $\text{assignment\_expression} \rightarrow \text{id '=' assignment\_expression\_prime}$
  15.  $\text{assignment\_expression\_prime} \rightarrow \text{arithmetic\_operation}$
  16.  $\text{assignment\_expression\_prime} \rightarrow \text{read\_statement}$
  17.  $\text{assignment\_expression\_prime} \rightarrow \text{literal ' ;'}$
  18.  $\text{assignment\_expression\_prime} \rightarrow \text{expression ' ;'}$
  19.  $\text{if\_statement} \rightarrow \text{"if"(' expression ') "statement" if\_statement\_linha}$
  20.  $\text{if\_statement\_linha} \rightarrow \text{"else" withElse}$
  21.  $\text{if\_statement\_linha} \rightarrow \epsilon$
  22.  $\text{withElse} \rightarrow \text{"if"(' expression ') "statement" if\_statement\_linha}$
  23.  $\text{withElse} \rightarrow \text{' statement '}$
  24.  $\text{print\_statement\_linha} \rightarrow \text{id ' ;'}$
  25.  $\text{expression} \rightarrow \text{id compare\_operator expression\_prime}$
  26.  $\text{expression} \rightarrow \text{num compare\_operator expression\_prime}$
  27.  $\text{expression\_prime} \rightarrow \text{id}$
  28.  $\text{expression\_prime} \rightarrow \text{num}$
  29.  $\text{compare\_operator} \rightarrow \text{"<"}$
  30.  $\text{compare\_operator} \rightarrow \text{">"}$
-

- 
31. `compare_operator`  $\rightarrow$  “<=”
  32. `compare_operator`  $\rightarrow$  “>=”
  33. `compare_operator`  $\rightarrow$  “==”
  34. `compare_operator`  $\rightarrow$  “!=”
  35. `compare_operator`  $\rightarrow$  “!”
  36. `compare_operator`  $\rightarrow$  “”
  37. `compare_operator`  $\rightarrow$  “||”
  38. `print_statement`  $\rightarrow$  “print” `print_statement_linha`
  39. `print_statement_linha`  $\rightarrow$  literal ‘;’
  40. `print_statement_linha`  $\rightarrow$  id ‘;’
  41. `read_statement`  $\rightarrow$  “read” ‘(’ ‘)’ ‘;’
  42. `continue_statement`  $\rightarrow$  “continue” ‘;’
  43. `break_statement`  $\rightarrow$  “break” ‘;’
  44. `while_statement`  $\rightarrow$  “while” ‘(’ `assignment_expression` ‘)’ ‘ ’ `statement` “
  45. `arithmetic_operation`  $\rightarrow$  `arithmetic_operand` `arithmetic_operation_prime` ‘;’
  46. `arithmetic_operation_prime`  $\rightarrow$  `arithmetic_operator` `arithmetic_operand` `arithmetic_operation_prime`
  47. `arithmetic_operation_prime`  $\rightarrow$   $\epsilon$
  48. `arithmetic_operand`  $\rightarrow$  id
  49. `arithmetic_operand`  $\rightarrow$  num
  50. `arithmetic_operator`  $\rightarrow$  ‘+’
  51. `arithmetic_operator`  $\rightarrow$  ‘-’
  52. `arithmetic_operator`  $\rightarrow$  ‘\*’
  53. `arithmetic_operator`  $\rightarrow$  ‘/’
  54. `arithmetic_operator`  $\rightarrow$  ‘^’
  55. `arithmetic_operator`  $\rightarrow$  ‘%’
-

---

### 2.2.1 Restrições da gramática

- Não é permitido ter expressões do tipo  $((a\&\&b) \parallel (a\parallel b))$ . É permitido somente expressões simples como  $(a\&\&b)$
- Um bloco de código deve estar entre chaves
- Não é possível utilizar operações como:  $++id$  ou  $id++$  ou  $id+=num$

### 2.2.2 Descrição do arquivo *arquivoErrosSintaticos.txt*

O arquivo *arquivoErrosSintaticos.txt* tem a finalidade de mostrar os erros sintáticos encontrados no código fonte.

`<id_tabela_erros> <erro> <linha> <coluna>`

Onde:

- `<id_tabela_erros>` é um identificador gerado automaticamente à medida que novos erros são inseridos na tabela de erros. Cada linha do arquivo possui um identificador único.
- `<erro>` é a descrição do erro encontrado
- `<linha>` é o número da linha do arquivo de entrada em que tal erro foi encontrado.
- `<coluna>` é o número da coluna do arquivo de entrada em que tal erro foi encontrado.

Exemplo do arquivo *arquivoErrosSintaticos.txt*.

id	Erro	Linha	Coluna
0	Deve haver um fecha chave depois de um bloco de código.	11	14
1	Deve haver um fecha chave depois de um bloco de código.	4	15

Figura 6: Exemplo do arquivo *arquivoErrosSintaticos.txt*

## 3 Exemplo

O nosso analisador léxico e sintático avaliará o seguinte código-fonte:

---

Linha	Código
0	<code>print " Entre o primeiro valor " ;</code>
1	<code>int a = read ;</code>
2	<code>print " Entre o segundo valor " ;</code>
3	<code>int b = read ;</code>
4	<code>print " O maior valor e: " ;</code>
5	<code>if ( a &gt; b ) {</code>
6	<code>print a ;</code>
7	<code>}</code>
8	<code>print b ;</code>
9	<code>}</code>

Figura 7: Código-fonte

### 3.1 Arquivo *arquivoTokens.txt*

```
0 23 0 0 -1
1 29 0 6 0
2 31 0 33 -1
3 25 1 0 -1
4 -1 1 4 0
5 2 1 6 -1
6 24 1 8 -1
7 31 1 13 -1
8 23 2 0 -1
9 29 2 6 1
10 31 2 32 -1
11 25 3 0 -1
12 -1 3 4 1
13 2 3 6 -1
14 24 3 8 -1
15 31 3 13 -1
16 23 4 0 -1
```

---

```
17 29 4 6 2
18 31 4 27 -1
19 18 5 0 -1
20 34 5 3 -1
21 -1 5 5 0
22 4 5 7 -1
23 -1 5 9 1
24 35 5 11 -1
25 32 5 13 -1
26 23 6 1 -1
27 -1 6 7 0
28 31 6 9 -1
29 33 7 0 -1
30 23 8 1 -1
31 -1 8 7 1
32 31 8 9 -1
33 33 9 0 -1
```

### 3.1.1 Arquivo *arquivoLiterais.txt*

0 Entre o primeiro valor 0 6  
1 Entre o segundo valor 2 6  
2 O maior valor e: 4 6

### 3.1.2 Arquivo *arquivoIdentificadores.txt*

0 a 1 4  
1 b 3 4

### 3.1.3 Arquivo *arquivoConstantes.txt*

Como não há constantes no código-fonte apresentado, tal arquivo não será gerado.

### 3.1.4 Arquivo *arquivoErrosLexicos.txt*

Como não há erros no código-fonte apresentado, tal arquivo não será gerado.

---

### 3.1.5 Arquivo *arquivoErrosSintaticos.txt*

- 0 Não é permitido atribuições na linha de declaração 1 6
- 1 Não é permitido atribuições na linha de declaração 3 6