

# Analisador Léxico - Objetivo

Gerar sequência de *tokens* a partir do código fonte.

Agrupar caracteres classificando-os como tokens.



# Analisador Léxico - Tokens

Menores unidades de informação que constituem a linguagem.

Usam o conjunto de terminais da gramática para serem definidos.

Podem ser:

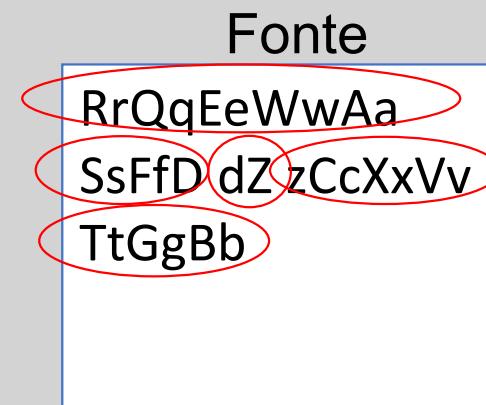
- Palavras reservadas ( ex.: **if, for, while, else, const, int** )
- Caracteres especiais ( ex.: { , ( , = , == , > )
- Identificadores (ex.: variáveis, tipos do usuário, funções )

# Analisador Léxico - Lexema

Conjunto de caracteres no programa-fonte que será classificado como um token.

## Separadores

Caracteres usados para delimitar os lexemas.



## Lexemas:

- RrQqEeWwAa
- SsFfD
- dZ
- zCcXxVv
- TtGgBb

## Mais comuns:

Espaço, Tabulação, Quebra de linha etc...

# Analisador Léxico - Padrão

Definidas as possíveis classificações dos tokens.

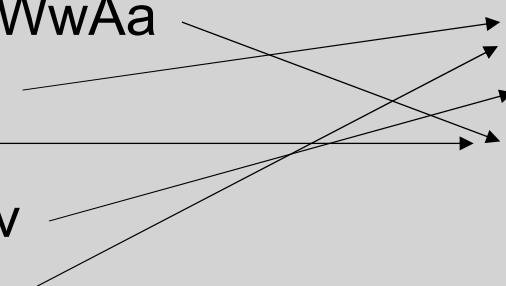
Cada classificação possui um padrão de sequência.

## Lexemas:

- RrQqEeWwAa
- SsFfD
- dZ
- zCcXxVv
- TtGgBb

## Classificações:

- Classificação 1
- Classificação 2
- Classificação 3



# Analisador Léxico - Padrão - Prioridade

Mesmo lexema pode encaixar em mais de um padrão.

Regras conflitantes devem ter prioridades definidas.

## Lexema:

- RrQqEeWwAa

## Classificações:

- - Classificação 1 – Prioridade 2
- - Classificação 2
- - Classificação 3 – Prioridade 1

# Analisador Léxico - Padrão - Classificação

Sequência de caracteres pode ser dividida em vários lexemas se isso permitir a classificação.

**Lexema:**

- zCcXxVv

?

**Lexemas:**

- zCcX

- xVv

**Classificações:**

- Classificação 1
- Classificação 2
- Classificação 3

# Analisador Léxico - Padrão - Exemplo

**Números** – Sequência de dígitos podendo conter um ponto.

**Classificados como Número:**

- 123
- 12.3
- 12.45
- .789
- 345.

**Não classificados como número:**

- 12a3
- 12.3.4
- a234

# Analisador Léxico - Padrão - Exemplo

**Identificador** – Sequência de letras, podendo conter dígitos e ‘\_’, não iniciada por dígito.

**Classificados como Identificador:**

- temp
- i
- \_variavel
- Funcao\_Teste
- var2
- const\_3\_virgula\_14

**Não classificados como identificador:**

- 1temp
- temp=1
- a+b

# Analisador Léxico - Padrão - Conflito

## **Exemplo de conflito:**

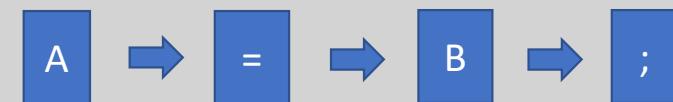
**Palavras Reservadas** – Sequência específica de caracteres.  
while – Sequência de caracteres ‘w’, ‘h’, ‘i’, ‘l’, ‘e’.

As regras das palavras reservadas conflitam com a de Identificador.

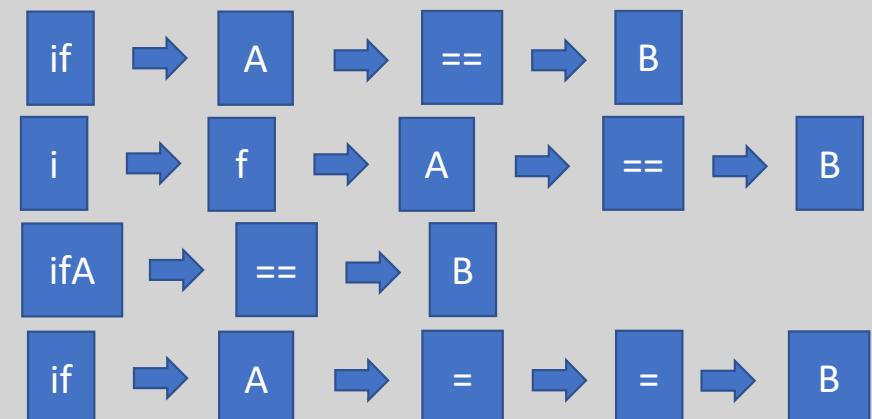
Palavras reservadas tem prioridade sobre identificadores.

# Analisador Léxico - Lexema - Exemplo

A = B ;  
A = B ;  
A=B;



if A == B  
if A == B  
ifA == B  
if A == B



# Analisador Léxico - Exemplo

```
const pi = 3.1415927;
```

Token	Lexema	Padrão
const	const	Caractere ‘c’ seguido de ‘o’, ‘n’, ‘s’ e ‘t’
identificador	pi	letra seguida por letras e/ou dígitos
relacional	=	< ou > ou = ou <> ou <= ou >=
numero	3.1415927	Qualquer constante numérica
ponto-e-virgula	;	Caractere ;

# Analisador Léxico - Características

Deve eliminar comentários.

Pode ser uma sub-rotina ou co-rotina do analisador sintático.

Atenção com performance devido ao acesso a disco.

Relaciona tokens com número de linha para mensagem de erro.

# Analisador Léxico - Tabela de Símbolos

Pode criar entrada na tabela de símbolo para Identificadores.

Estrutura de dados contendo um registro para cada identificador usado no programa.

Armazena informação dos identificadores para ajudar a compilação.

Pode criar entrada ao classificar token como Identificador.

# Analisador Léxico - Erros

Detectável na fase de análise léxica quando não consegue classificar o token.

Maioria é de difícil detecção na fase de análise léxica.

- Erro de digitação
- Falta de separador
- Inclusão indevida de separador

Em geral serão detectados na análise sintática ou semântica.

# Analisador Léxico - Erros - Exemplo

## Exemplo:

while (a > 25)

Token	Lexema	Padrão
Repetição	while	Letra 'w' seguida das letras 'h', 'i', 'l' e 'e'
Abreparentese	(	Caractere (
Identificador	a	Letra seguida por letras e/ou dígitos
relacional	>	< ou > ou == ou != ou <= ou >=
numero	25	Qualquer constante numérica
fecharentese	)	Caractere )

# Analisador Léxico - Erros - Exemplo

**Exemplo:**

while (a > 25)

**Erro de digitação:**

wile (a > 25)

Token	Lexema	Padrão
Identificador	wile	Letra seguida por letras e/ou dígitos
Abreparentese	(	Caractere (
Identificador	a	Letra seguida por letras e/ou dígitos
relacional	>	< ou > ou == ou != ou <= ou >=
numero	25	Qualquer constante numérica
fecharentese	)	)

# Analisador Léxico - Erros - Exemplo

## Exemplo:

return b;

Token	Lexema	Padrão
Retorno	return	Letra 'r' seguida da letra 'e', 't', 'u', 'r' e 'n'
Identificador	b	Letra seguida por letras e/ou dígitos
Pontovirgula	;	Caractere ;

# Analisador Léxico - Erros - Exemplo

**Exemplo:**

return b;

**Erro de falta de separador:**

returnb;

Token	Lexema	Padrão
Identificador	returnb	Letra seguida por letras e/ou dígitos
Pontovirgula	;	Caractere ;

# Analisador Léxico - Erros - Exemplo

## Exemplo:

if (a == 25)

Token	Lexema	Padrão
Condicional	If	Letra 'i' seguida da letra 'f'
Abreparentese	(	Caractere (
Identificador	a	Letra seguida por letras e/ou digito
Relacional	==	> ou < ou >= ou <= ou != ou ==
Numeral	25	Sequencia de dígitos podendo conter 1 ponto
Fechaparentese	)	Caractere )

# Analisador Léxico - Erros - Exemplo

**Exemplo:**

if (a == 25)

**Erro de excesso de separador:**

if (a = = 25)

Token	Lexema	Padrão
Condicional	If	Letra ‘i’ seguida da letra ‘f’
Abreparentese	(	Caractere (
Identificador	a	Letra seguida por letras e/ou digito
atribuição	=	Caractere =
atribuição	=	Caractere =
Numeral	25	Sequencia de dígitos podendo conter 1 ponto
Fechaparentese	)	Caractere )

# Analisador Léxico - Tratamento de Erros

Quando um lexema não consegue ser classificado não deve parar a análise léxica.

Definir uma classificação ERRO e criar um token com essa classificação.

O token será completo com lexema e número da linha.

# Analisador Léxico - Erros - Exemplo

**Exemplo de erro detectável pelo analisador léxico:**

if (a =@ 25)

Token	Lexema	Padrão
Condicional	If	Letra 'i' seguida da letra 'f'
Abreparentese	(	Caractere (
Identificador	a	Letra seguida por letras e/ou digito
atribuição	=	Caractere =
erro	@	Nenhum padrão conhecido
Numeral	25	Sequencia de dígitos podendo conter 1 ponto
Fechaparentese	)	Caractere )