

# UNIT

Universidade Tiradentes

Alan Reis Anjos, Anthony Ramos dos Santos, José Henrique Oliveira de Carvalho.

Analisador Léxico e Sintático  
Aracaju, Setembro de 2025

Aracaju, Setembro de 2025

Relatório realizado na disciplina de Compiladores orientado pela professora Layse como complemento para a nota da 1<sup>a</sup> Unidade

## 1. Introdução

A linguagem QuizLang foi projetada para definir questionários de forma estruturada, possuindo regras léxicas e sintáticas próprias. Nesta atividade, desenvolvemos o analisador léxico e sintático para essa linguagem, identificando tokens, corrigindo problemas de gramática e garantindo que a linguagem possa ser processada corretamente por um compilador.

## 2. Identificação e Classificação dos Tokens da QuizLang

Com base na gramática fornecida, os tokens da QuizLang podem ser classificados da seguinte forma:

Categoria	Token	Exemplo
-----------	-------	---------

Palavra-chave	'quiz', 'titulo', 'tempo', 'secao', 'mcq' 'pergunta', 'opcoes', 'resposta', 'discursiva', 'palavras', 'numerica', 'intervalo'	QUIZ
Identificadores	ID	q1, pergunta1
Literais String	STRING	"Qual a capital do Brasil?"
Literais Numéricos	NUMBER	1, 2025
Operadores	=, :	QUESTION = "..."
Delimitadores	{, }, ;	}
Comentários	// ...	// comentário

## 3. Funcionamento do Analisador Léxico para a QuizLang

O analisador léxico da QuizLang percorre o código fonte caractere por caractere e agrupa os símbolos em tokens válidos de acordo com os padrões definidos.

### Exemplo de Regras no Lexer:

```
QUIZ : 'QUIZ'; QUESTION : 'QUESTION'; ANSWER : 'ANSWER'; CORRECT : 'CORRECT'; ID : [a-zA-Z_][a-zA-Z0-9_]*; STRING : """.*?"""; NUMBER : [0-9]+; WS : [\t\r\n]+ -> skip; COMMENT : '/\/*[^\r\n]*' -> skip;
```

## 4. Análise da gramática original e seus problemas

A gramática fornecida como ponto de partida para a atividade apresentava deficiências que dificultavam a sua implementação direta em um analisador. Durante a análise, foram identificados os seguintes problemas principais:

1. Ambiguidade na Notação de Repetição ({...}): O uso de chaves para indicar a repetição de regras (como em **secoes** = { **secao** }) não é uma notação padrão em formalismos como EBNF, gerando dúvida se a repetição seria de "zero ou mais" (\*) ou "um ou mais" (+) elementos.
2. Possibilidade de Estruturas Vazias: A ambiguidade anterior poderia permitir a criação de um quiz sem seções ou uma seção sem questões, o que não é logicamente válido para o domínio da aplicação.

3. Notação de Lista Não Convencional: A regra para *opcoes* era funcional, mas utilizava uma estrutura complexa e pouco intuitiva (*{ OPCAO ',' }* OPCAO), dificultando a leitura e a implementação de um parser.

#### Gramática Corrigida:

```
quiz      : 'quiz' ID '{' metadados secoes '}';
metadados : 'titulo' STRING 'tempo' NUMBER;
secoes   : secao+;
secao    : 'secao' STRING '{' item+ '}';
item     : mcq | discursiva | numerica;
mcq      : 'mcq' ID ':' pergunta opcoes 'resposta' OPCAO;
pergunta : 'pergunta' STRING;
opcoes   : 'opcoes' '[' OPCAO (',' OPCAO)* ']';
discursiva: 'discursiva' ID ':' pergunta 'palavras' NUMBER;
numerica : 'numerica' ID ':' pergunta 'intervalo' NUMBER '-' NUMBER;
OPCAO    : STRING;
```

## 5. Explicação dos Erros Encontrados e Soluções

Durante o processo de análise, identificamos os seguintes problemas e respectivas soluções:

#### Erros e Soluções:

1. Solução para Ambiguidade de Repetição: A notação {...} foi substituída por operadores padrão. Em secoes: secao+; e secao: ... item+;, o operador + ("um ou mais") foi utilizado para garantir que um quiz sempre tenha ao menos uma seção e que cada seção contenha ao menos uma questão.
2. Solução para a Notação de Lista: A regra de opcoes foi reescrita para opcoes: 'opcoes' '[' OPCAO (',' OPCAO)\* ']'. Esta estrutura define uma lista que começa com uma OPCAO e é seguida por "zero ou mais" ocorrências de ',' OPCAO, um padrão claro para listas separadas por vírgula.

## 6. Conclusão

Com a implementação do analisador léxico e sintático da QuizLang conseguimos:

- identificar e classificar corretamente os tokens;
- especificar o funcionamento do léxico; reescrever a gramática corrigida;
- explicar os erros e soluções aplicadas.

## 7. Referência

AHO, Alfred V.; LAM, Monica S.; SETHI, Ravi; ULLMAN, Jeffrey D. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo: Pearson, 2009.