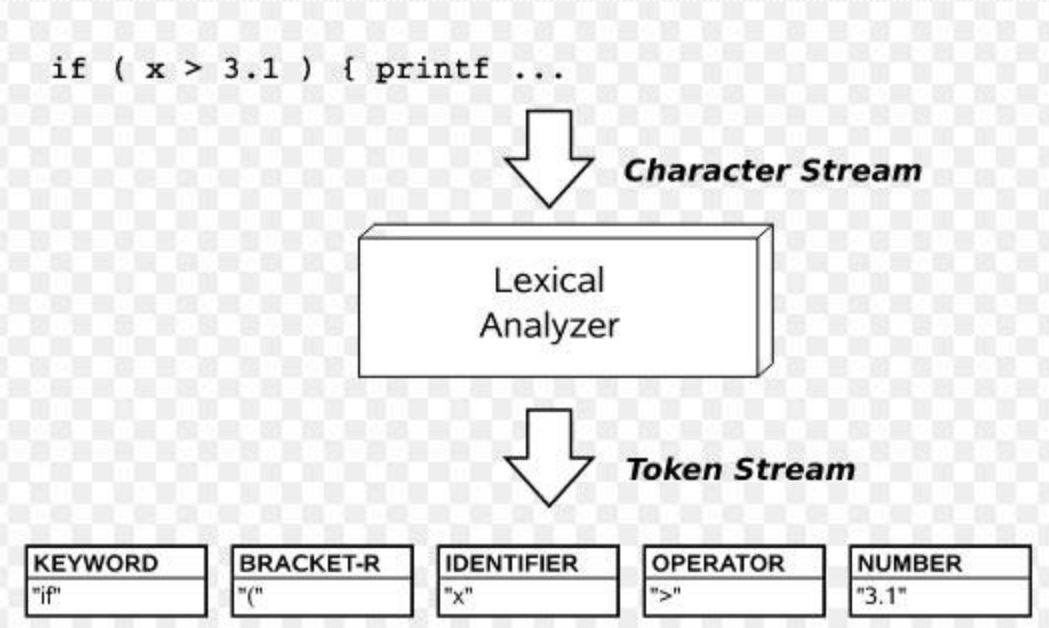
Analisador Léxico

Análise léxica

É o processo de analisar a entrada de linhas de caracteres (tal como o código-fonte de um programa de computador) e produzir uma sequência de símbolos chamados ***tokens***.



Analisador léxico

Também chamado de ***scanner***, faz a varredura do programa fonte caractere por caractere. É nesta fase que são reconhecidos:

- Palavras reservadas
- Identificadores
- Constantes
- E outras palavras que pertencem ao alfabeto da linguagem de programação.

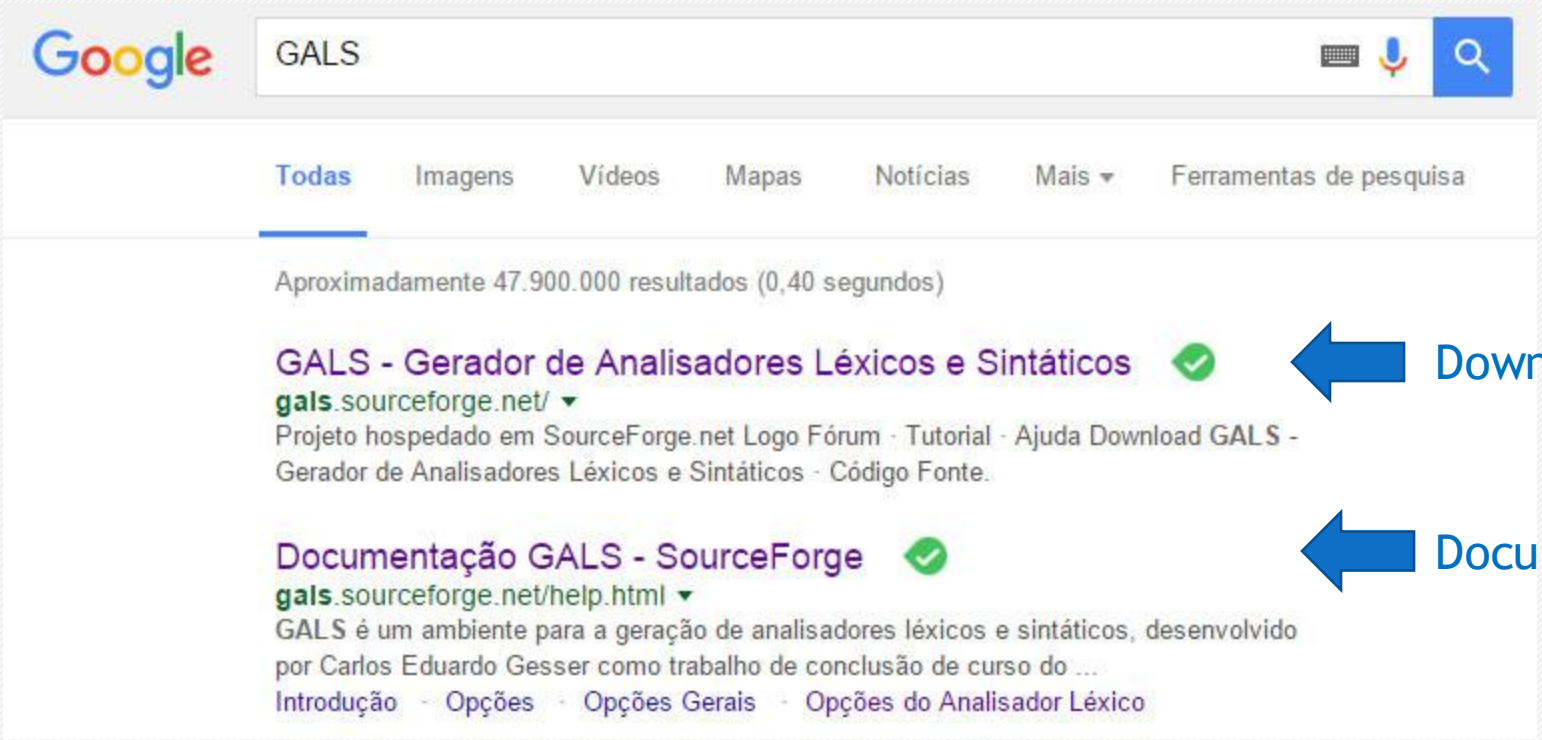
Analisador léxico

Exemplo:

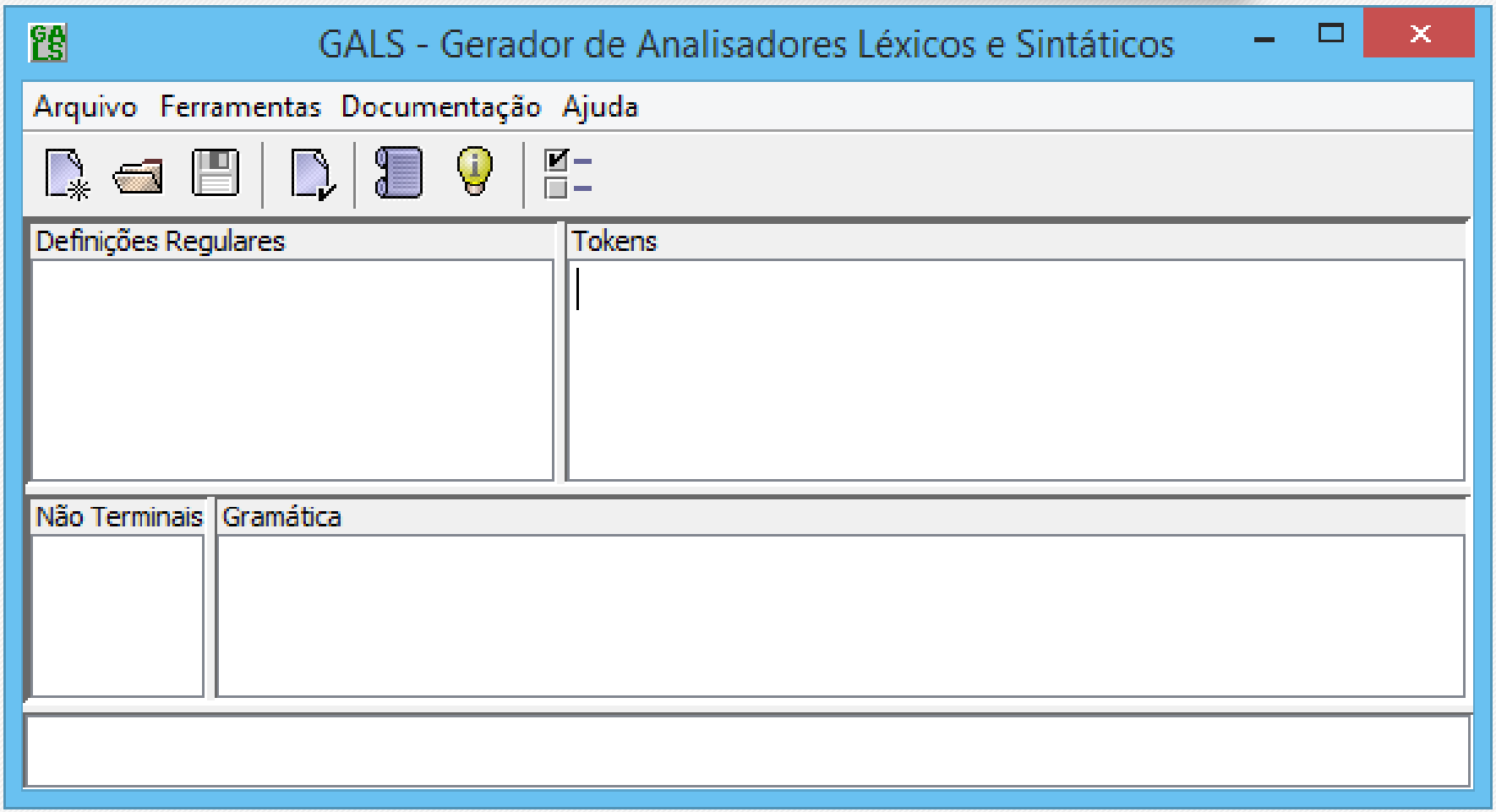
vetor[10]

Token	ID	LSB	INT	RSB
Lexema	vetor	[	10	]

GALS


Google
GALS
Todas
Imagens
Vídeos
Mapas
Notícias
Mais
Ferramentas de pesquisa
Aproximadamente 47.900.000 resultados (0,40 segundos)
GALS - Gerador de Analisadores Léxicos e Sintáticos
gals.sourceforge.net/
Projeto hospedado em SourceForge.net Logo Fórum · Tutorial · Ajuda Download GALS - Gerador de Analisadores Léxicos e Sintáticos · Código Fonte.
Documentação GALS - SourceForge
gals.sourceforge.net/help.html
GALS é um ambiente para a geração de analisadores léxicos e sintáticos, desenvolvido por Carlos Eduardo Gesser como trabalho de conclusão de curso do ...
Introdução · Opções · Opções Gerais · Opções do Analisador Léxico
Download
Documentação

GALS


GALS - Gerador de Analisadores Léxicos e Sintáticos
Arquivo
Ferramentas
Documentação
Ajuda
Definições Regulares
Tokens
Não Terminais
Gramática

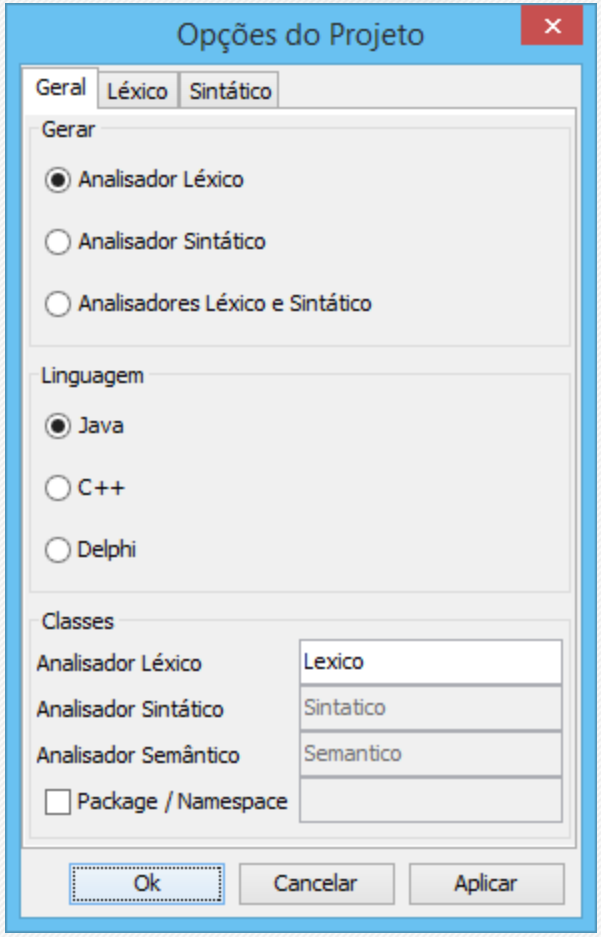
GALS

Gera:

- Analisador Léxico
- Analisador Sintático
- Analisadores Léxico e Sintático
  (e semântico vazio)

Nas linguagens:

- Java
- C++
- Delphi



GALS

Definindo tokens a partir do identificador

Pode-se declarar um token apenas declarando um identificador para ele. Esta é a forma mais simples de se declarar um token, porém, a menos flexível.

- Identificadores normais
- Qualquer sequência de caracteres entre aspas (")

begin	identifica	begin
"!="	identifica	!=

GALS

Definindo tokens a partir de uma expressão regular

Esta é a forma mais genérica de se definir um token.

[identificador] : [expressão regular]

Sempre que o analisador identifica a expressão regular ele produz o *token* correspondente.

INT : [0-9]+ identifica 0;1;2;3;...;9;10;11;12;...

GALS

Definindo *tokens* como caso especial de outro *token*

Nestes casos, sempre que o analisador identifica o token base, ele procura pelo valor do token em uma lista de casos especiais. Se for encontrado, o caso especial é retornado, senão é produzido o token base.

ID : [a-zA-Z][a-z A-Z0-9]*

BEGIN = ID : "begin"
END = ID : "end"
WHILE = ID : "while"

GALS

Definições Regulares

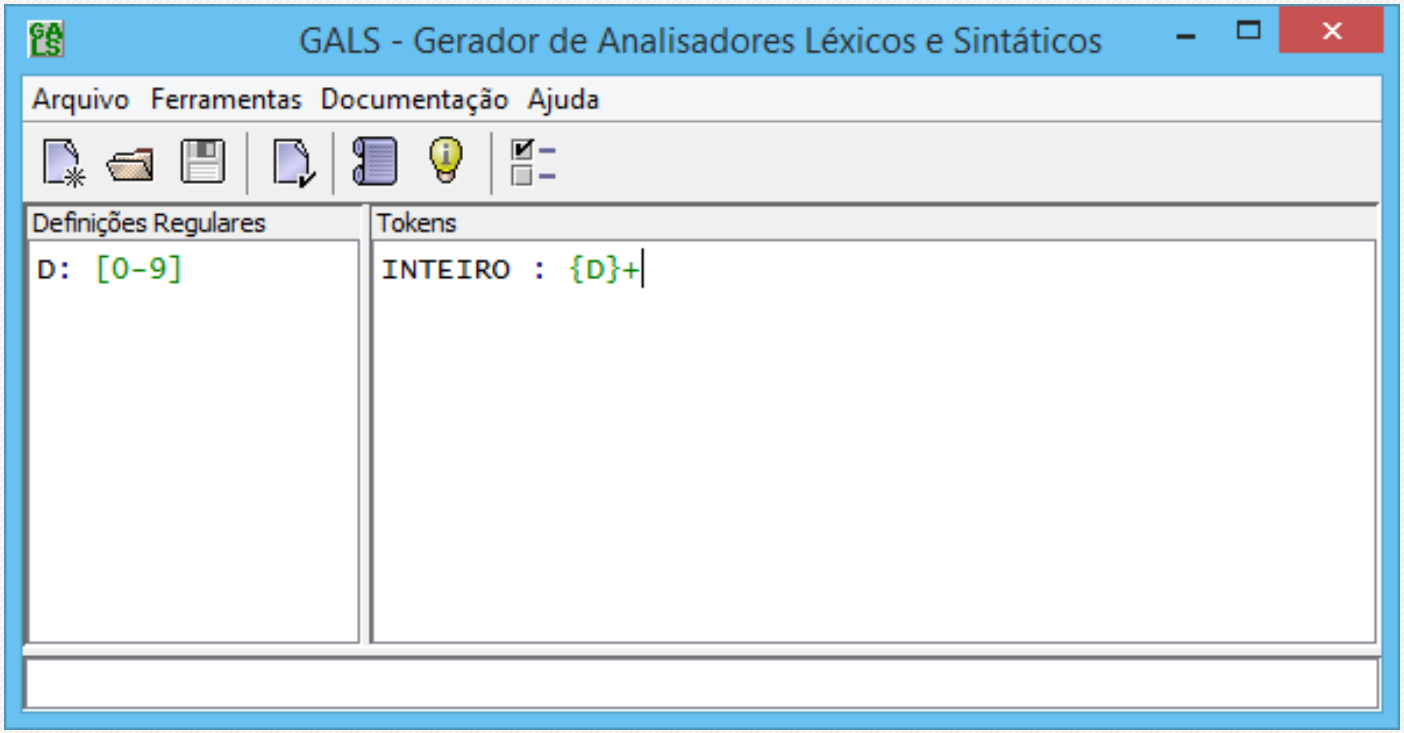
Podem ser utilizadas em outras expressões regulares, utilizando o identificador { }. Exemplo:

D : [0-9] < - Definição regular

INTEIRO : {D}+ < - Token

GALS

Exemplo



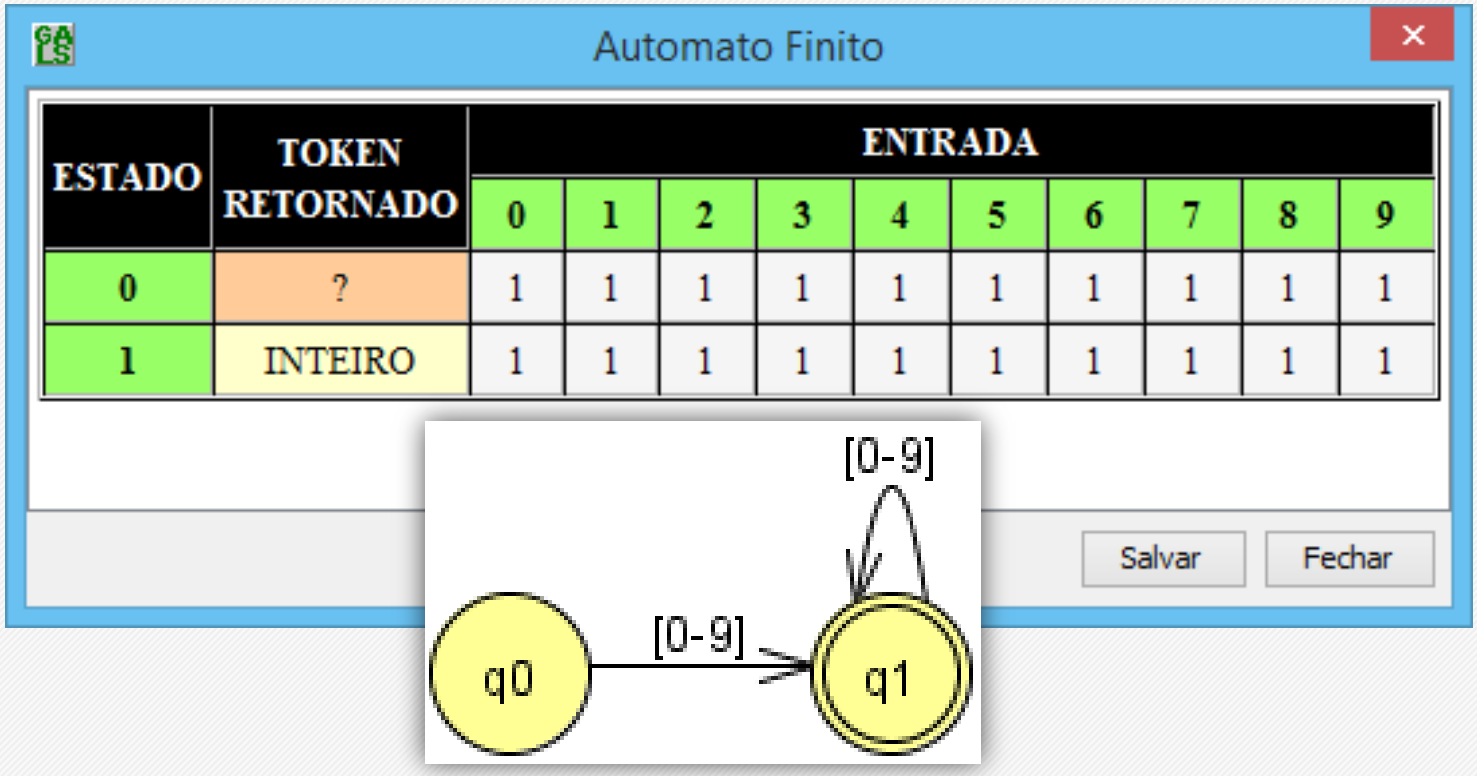
GALS

Tabela de análise léxica (autômato)

ESTADO	TOKEN RETORNADO	ENTRADA									
		0	1	2	3	4	5	6	7	8	9
0	?	1	1	1	1	1	1	1	1	1	1
1	INTEIRO	1	1	1	1	1	1	1	1	1	1

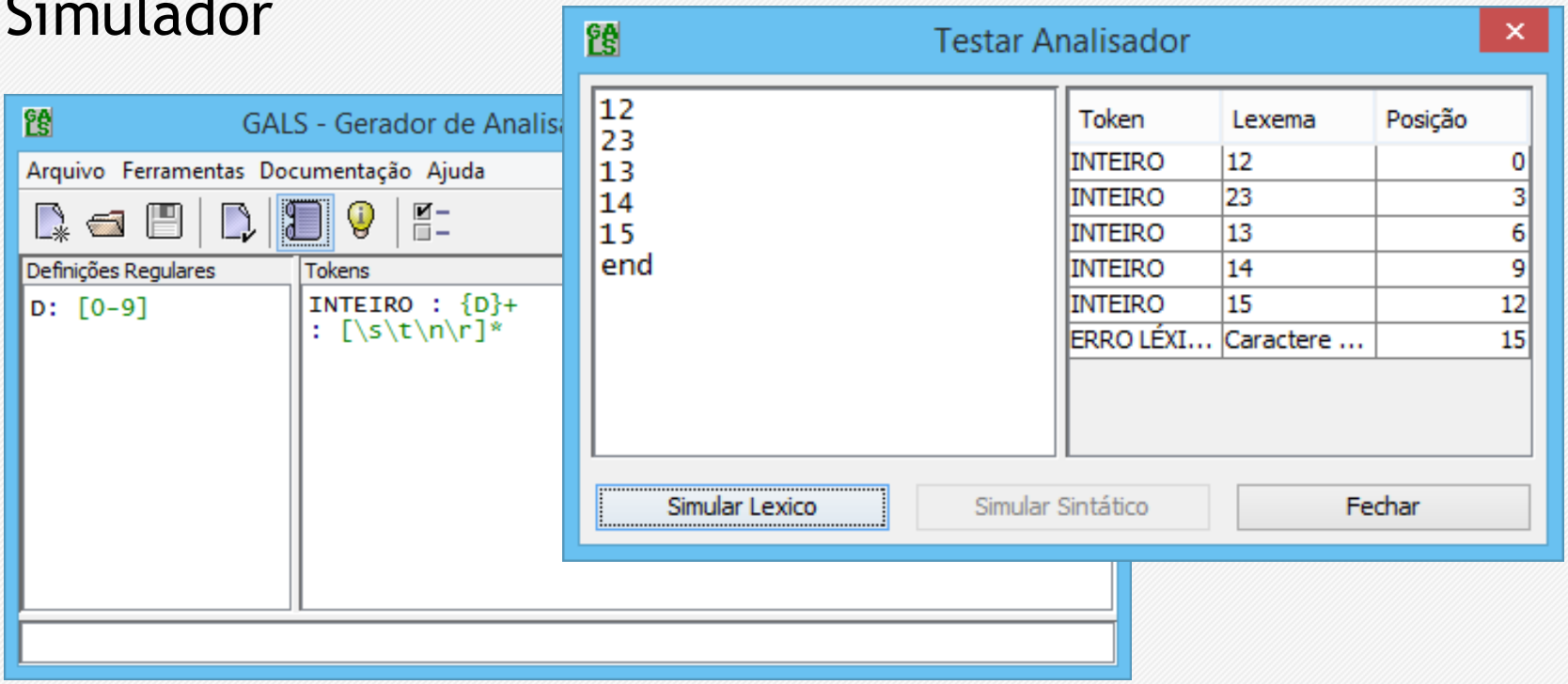
GALS

Tabela de análise léxica (autômato)



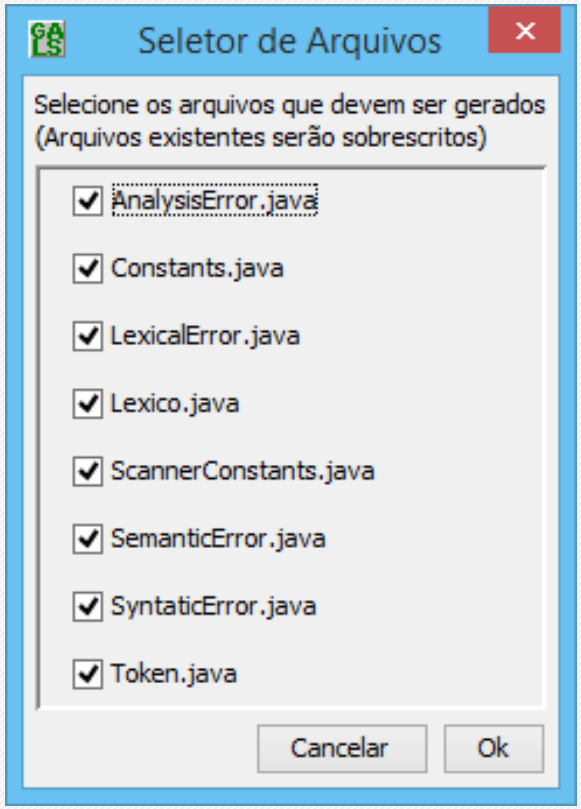
GALS

Simulador



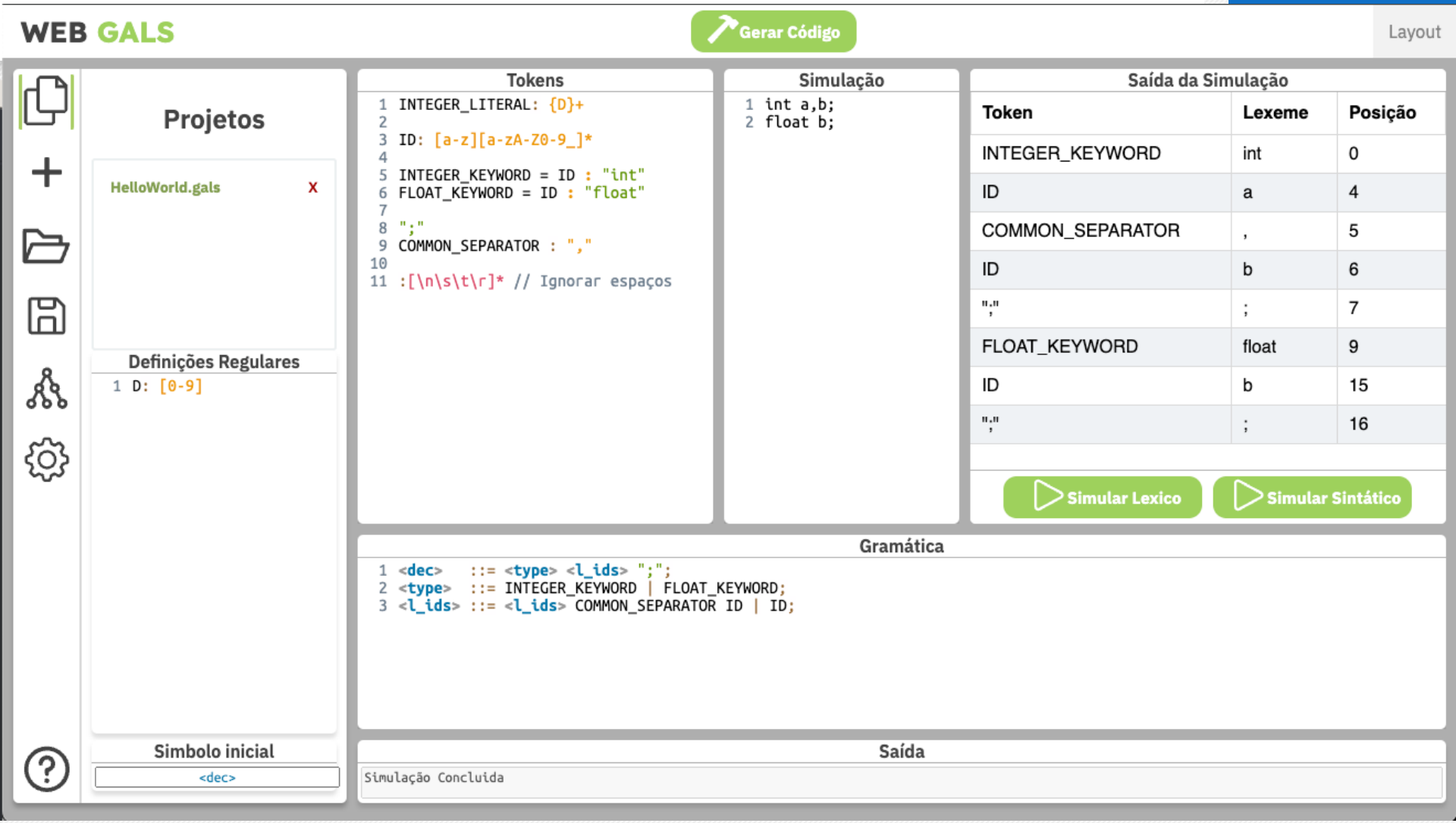
GALS

Geração de código



https://lia-univali.github.io/Web-GALS/

NOVA VERSÃO - EM TESTES



Referências

- AHO, Alfred V.; VIEIRA, Daniel. Compiladores: princípios, técnicas e ferramentas. 2. ed. São Paulo, SP: Pearson/Addison Wesley, c2007, c2008. x, 634 p. ISBN 9788588639249.
- DELAMARO, Márcio Eduardo. Como construir um compilador utilizando ferramentas Java. São Paulo: Novatec, c2004. 308 p. ISBN 8575220551