

# Relatório do Compilador (LL(1) + SLR)

## === TABELA DE TOKENS ===

Tipo	Lexema
FUNCAO	funcao
TIPO_VAR	int
IDENT	soma
LPAREN	(
TIPO_VAR	int
IDENT	a
VIRGULA	,
TIPO_VAR	int
IDENT	b
RPAREN	)
LCHAVE	{
RETORNO	retornar
IDENT	a
OPER_ARIT	+
IDENT	b
PONTOVIRG	;
RCHAVE	}
PRINCIPAL	principal
LCHAVE	{
TIPO_VAR	int
IDENT	x
ATRIB	=
NUMERO_INT	10
PONTOVIRG	;
TIPO_VAR	int
IDENT	y

## Relatório do Compilador (LL(1) + SLR)

ATRIB	=
NUMERO_INT	20
PONTOVIRG	;
TIPO_VAR	int
IDENT	resultado
PONTOVIRG	;
IDENT	resultado
ATRIB	=
IDENT	soma
LPAREN	(
IDENT	x
VIRGULA	,
IDENT	y
RPAREN	)
PONTOVIRG	;
SE	se
LPAREN	(
IDENT	resultado
COMPAR	>
NUMERO_INT	20
RPAREN	)
LCHAVE	{
IDENT	resultado
ATRIB	=
IDENT	resultado
OPER_ARIT	-
NUMERO_INT	5
PONTOVIRG	;
RCHAVE	}

## Relatório do Compilador (LL(1) + SLR)

SENAO	senao
LHAVE	{
IDENT	resultado
ATRIB	=
IDENT	resultado
OPER_ARIT	+
NUMERO_INT	5
PONTOVIRG	;
RHAVE	}
ENQUANTO	enquanto
LPAREN	(
IDENT	resultado
COMPAR	<
NUMERO_INT	50
RPAREN	)
LHAVE	{
IDENT	resultado
ATRIB	=
IDENT	resultado
OPER_ARIT	+
NUMERO_INT	1
PONTOVIRG	;
RHAVE	}
FACA	faca
LHAVE	{
IDENT	resultado
ATRIB	=
IDENT	resultado
OPER_ARIT	*

## Relatório do Compilador (LL(1) + SLR)

NUMERO_INT	2
PONTOVIRG	;
RCHAVE	}
ENQUANTO	enquanto
LPAREN	(
IDENT	resultado
COMPAR	<
NUMERO_INT	200
RPAREN	)
PONTOVIRG	;
PARA	para
LPAREN	(
TIPO_VAR	int
IDENT	i
ATRIB	=
NUMERO_INT	0
PONTOVIRG	;
IDENT	i
COMPAR	<
NUMERO_INT	10
PONTOVIRG	;
IDENT	i
ATRIB	=
IDENT	i
OPER_ARIT	+
NUMERO_INT	1
RPAREN	)
LCHAVE	{
IDENT	resultado

## Relatório do Compilador (LL(1) + SLR)

ATRIB	=
IDENT	resultado
OPER_ARIT	+
IDENT	i
PONTOVIRG	;
RCHAVE	}
TIPO_VAR	cadeia
IDENT	texto
ATRIB	=
PALAVRA	"Resultado final: "
PONTOVIRG	;
IDENT	escrever
LPAREN	(
IDENT	texto
VIRGULA	,
IDENT	resultado
RPAREN	)
PONTOVIRG	;
RCHAVE	}
EOF	

## Relatório do Compilador (LL(1) + SLR)

### === ÚLTIMOS 10 PASSOS DO LL(1) ===

Passo	Pilha	Entrada	Ação
471	EXPR_ARITMETICA_RESTO_G EXPR_COMPAR_RESTO_G EXPR_LOGICA_RESTO_G ARGUMENTOS_RESTO_G RPAREN PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	expandir TERMO_RESTO_G $\rightarrow \epsilon$
472	EXPR_COMPAR_RESTO_G EXPR_LOGICA_RESTO_G ARGUMENTOS_RESTO_G RPAREN PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	expandir EXPR_ARITMETICA_RESTO_G $\rightarrow$
473	EXPR_LOGICA_RESTO_G ARGUMENTOS_RESTO_G RPAREN PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	expandir EXPR_COMPAR_RESTO_G $\rightarrow \epsilon$
474	ARGUMENTOS_RESTO_G RPAREN PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	expandir EXPR_LOGICA_RESTO_G $\rightarrow \epsilon$
475	RPAREN PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	expandir ARGUMENTOS_RESTO_G $\rightarrow \epsilon$
476	PONTOVIRG COMANDOS_G RCHAVE EOF	RPAREN	casar 'RPAREN'
477	COMANDOS_G RCHAVE EOF	PONTOVIRG	casar 'PONTOVIRG'
478	RCHAVE EOF	RCHAVE	expandir COMANDOS_G $\rightarrow \epsilon$
479	EOF	RCHAVE	casar 'RCHAVE'
480		EOF	casar 'EOF'

# Relatório do Compilador (LL(1) + SLR)

## === GRAMÁTICA CONVERTIDA PARA LR(0) ===

PROGRAMA\_G:

- DECL\_FUNCOES\_G PRINCIPAL\_G

DECL\_FUNCOES\_G:

- FUNCAO TIPO\_VAR IDENT LPAREN PARAMS\_G RPAREN LCHAVE COMANDOS\_G RCHAVE

DECL\_FUNCOES\_G

- $\epsilon$

CHAM\_FUNCOES\_G:

- IDENT LPAREN ARGUMENTOS\_G RPAREN

PRINCIPAL\_G:

- PRINCIPAL LCHAVE COMANDOS\_G RCHAVE

PARAMS\_G:

- TIPO\_VAR IDENT PARAMS\_RESTO\_G

- $\epsilon$

PARAMS\_RESTO\_G:

- VIRGULA TIPO\_VAR IDENT PARAMS\_RESTO\_G

- $\epsilon$

COMANDOS\_G:

- COMANDO\_G COMANDOS\_G

- $\epsilon$

COMANDO\_G:

- DECLARACOES\_G PONTOVIRG
- IDENT ELEMENTO\_IDENT\_G PONTOVIRG
- SE LPAREN EXPRESSAO\_G RPAREN LCHAVE COMANDOS\_G RCHAVE SENAO\_G
- ENQUANTO LPAREN EXPRESSAO\_G RPAREN LCHAVE COMANDOS\_G RCHAVE
- FACA LCHAVE COMANDOS\_G RCHAVE ENQUANTO LPAREN EXPRESSAO\_G RPAREN PONTOVIRG
  - PARA LPAREN DECL\_OU\_ATRIB\_G PONTOVIRG EXPRESSAO\_G PONTOVIRG ATRIBUICAO\_G RPAREN

LCHAVE COMANDOS\_G RCHAVE

- RETORNO EXPRESSAO\_G PONTOVIRG

DECL\_OU\_ATRIB\_G:

- DECLARACOES\_G
- ATRIBUICAO\_G

DECLARACOES\_G:

- TIPO\_VAR IDENT DECLARACOES\_ATRIB\_G DECLARACOES\_RESTO\_G

DECLARACOES\_RESTO\_G:

- VIRGULA IDENT DECLARACOES\_ATRIB\_G DECLARACOES\_RESTO\_G
- $\epsilon$

DECLARACOES\_ATRIB\_G:

- ATRIB EXPRESSAO\_G

# Relatório do Compilador (LL(1) + SLR)

- $\epsilon$

ATRIBUICAO\_G:

- IDENT ATRIB EXPRESSAO\_G

ELEMENTO\_IDENT\_G:

- ATRIB EXPRESSAO\_G
- LPAREN ARGUMENTOS\_G RPAREN

SENAO\_G:

- SENAO LCHAVE COMANDOS\_G RCHAVE
- $\epsilon$

ARGUMENTOS\_G:

- EXPRESSAO\_G ARGUMENTOS\_RESTO\_G
- $\epsilon$

ARGUMENTOS\_RESTO\_G:

- VIRGULA EXPRESSAO\_G ARGUMENTOS\_RESTO\_G
- $\epsilon$

EXPRESSAO\_G:

- EXPR\_LOGICA\_G

EXPR\_LOGICA\_G:

- EXPR\_COMPAR\_G EXPR\_LOGICA\_RESTO\_G

EXPR\_LOGICA\_RESTO\_G:

- OPER\_LOGI\_BIN EXPR\_COMPAR\_G EXPR\_LOGICA\_RESTO\_G
- $\epsilon$

EXPR\_COMPAR\_G:

- EXPR\_ARITMETICA\_G EXPR\_COMPAR\_RESTO\_G

EXPR\_COMPAR\_RESTO\_G:

- COMPAR EXPR\_ARITMETICA\_G EXPR\_COMPAR\_RESTO\_G
- $\epsilon$

EXPR\_ARITMETICA\_G:

- TERMO\_G EXPR\_ARITMETICA\_RESTO\_G

EXPR\_ARITMETICA\_RESTO\_G:

- OPER\_ARIT TERMO\_G EXPR\_ARITMETICA\_RESTO\_G
- $\epsilon$

TERMO\_G:

- FATOR\_G TERMO\_RESTO\_G

TERMO\_RESTO\_G:

- OPER\_ARIT FATOR\_G TERMO\_RESTO\_G
- $\epsilon$

FATOR\_G:



## **Relatório do Compilador (LL(1) + SLR)**

- LPAREN EXPRESSAO\_G RPAREN
- OPER\_LOGI\_UN FATOR\_G
- IDENT FATOR\_IDENT\_G
- NUMERO\_INT
- NUMERO\_REAL
- PALAVRA
- CARACTERE
- BOOLEANO

FATOR\_IDENT\_G:

- LPAREN ARGUMENTOS\_G RPAREN
- $\epsilon$

# Relatório do Compilador (LL(1) + SLR)

## === ÚLTIMOS 25 PASSOS DO SLR(1) ===

Passo	Estados	Símbolos	Entrada	Ação
442	... [16, 15, 41, 66, 86, 28] EXPRESSAO_G VIRGULA EXPR_COMPAR_G	... COMANDO_G IDENT LPAREN	RPAREN	reduc
443	... [15, 41, 66, 86, 28, 54] <del>EXPRESSAO_G VIRGULA EXPR_COMPAR_G</del> EXPRESSAO_G VIRGULA EXPR_LOGICA_RESTO_G	... IDENT LPAREN EXPRESSAO_G	RPAREN	reduc
444	... [16, 15, 41, 66, 86, 25] EXPRESSAO_G VIRGULA EXPR_LOGICA_G	... COMANDO_G IDENT LPAREN	RPAREN	reduc
445	... [16, 15, 41, 66, 86, 103] EXPRESSAO_G VIRGULA EXPRESSAO_G	... COMANDO_G IDENT LPAREN	RPAREN	reduc
446	... [15, 41, 66, 86, 103, 112] <del>EXPRESSAO_G VIRGULA EXPRESSAO_G</del> EXPRESSAO_G VIRGULA ARGUMENTOS_RESTO_G	... IDENT LPAREN EXPRESSAO_G	RPAREN	reduc
447	... [16, 16, 15, 41, 66, 87] <del>ARGUMENTOS_RESTO_G</del> ARGUMENTOS_RESTO_G	... COMANDO_G COMANDO_G IDENT	RPAREN	reduc
448	... [16, 16, 16, 15, 41, 67] IDENT LPAREN ARGUMENTOS_G	... COMANDO_G COMANDO_G COMANDO_G	RPAREN	shift
449	... [16, 16, 15, 41, 67, 88] <del>ARGUMENTOS_RESTO_G</del> ARGUMENTOS_RESTO_G	... COMANDO_G COMANDO_G IDENT		reduc
450	... [16, 16, 16, 16, 15, 39] <del>COMANDO_G</del> COMANDO_G IDENT ELEMENTO_IDENT_G	... COMANDO_G COMANDO_G COMANDO_G		shift
451	... [16, 16, 16, 15, 39, 64] <del>ELEMENTO_IDENT_G</del> ELEMENTO_IDENT_G PONTO_VIRG	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	reduc
452	... [16, 16, 16, 16, 16, 16] COMANDO_G COMANDO_G COMANDO_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	reduc
453	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	reduc
454	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	reduc
455	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	reduc

## Relatório do Compilador (LL(1) + SLR)

456	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	redu
457	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	redu
458	... [16, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... COMANDO_G COMANDO_G COMANDO_G	RCHAVE	redu
459	... [7, 16, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... LCHAVE COMANDO_G COMANDO_G	RCHAVE	redu
460	... [4, 7, 16, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	... PRINCIPAL LCHAVE COMANDO_G	RCHAVE	redu
461	... [1, 4, 7, 16, 16, 42] COMANDOS_G COMANDO_G COMANDOS_G	DECL_FUNCOES_G PRINCIPAL LCHAVE	RCHAVE	redu
462	[0, 1, 4, 7, 16, 42] COMANDOS_G COMANDOS_G	DECL_FUNCOES_G PRINCIPAL LCHAVE	RCHAVE	redu
463	[0, 1, 4, 7, 11] COMANDOS_G	DECL_FUNCOES_G PRINCIPAL LCHAVE	RCHAVE	shift
464	[0, 1, 4, 7, 11, 35] COMANDOS_G COMANDOS_G LCHAVE RCHAVE	DECL_FUNCOES_G PRINCIPAL LCHAVE	EOF	redu
465	[0, 1, 5] PRINCIPAL_G	DECL_FUNCOES_G PRINCIPAL_G	EOF	redu
466	[0, 2]	PROGRAMA_G	EOF	accep