Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
0	X : * CompilationUnit CompilationUnit : * ImportClauseList TintoDecl ImportClauseList : * ImportClauseList ImportClause ImportClauseList : *	CompilationUnit → 1 ImportClauseList → 2 R3
1	X : CompilationUnit *	R0
2	CompilationUnit : ImportClauseList * TintoDecl ImportClauseList : ImportClauseList * ImportClause ImportClause : * **import identifier semicolon** TintoDecl : * LibraryDecl TintoDecl : * NativeDecl LibraryDecl : * **library identifier lbrace** FunctionList **rbrace** NativeDecl : * **native identifier lbrace** NativeFunctionList **rbrace**	TintoDecl → 3 ImportClause → 4 **import** → 5 LibraryDecl → 6 NativeDecl → 7 **library** → 8 **native** → 9
3	CompilationUnit : ImportClauseList TintoDecl *	R1
4	ImportClauseList : ImportClauseList ImportClause *	R2
5	ImportClause : **import** * **identifier semicolon**	**identifier** →10
6	TintoDecl : LibraryDecl *	R5
7	TintoDecl : NativeDecl *	R6
8	LibraryDecl : **library** * **identifier lbrace** FunctionList **rbrace**	**identifier** → 11
9	NativeDecl : **native** * **identifier lbrace** NativeFunctionList **rbrace**	**identifier** → 12
10	ImportClause : **import identifier** * **semicolon**	**semicolon** → 13
11	LibraryDecl : **library identifier** * **lbrace** FunctionList **rbrace**	**lbrace** → 14
12	NativeDecl : **native identifier** * **lbrace** NativeFunctionList **rbrace**	**lbrace** → 15
13	ImportClause : **import identifier semicolon** *	R4
14	LibraryDecl : **library identifier lbrace** * FunctionList **rbrace** FunctionList : * FunctionList FunctionDecl FunctionList : *	FunctionList → 16 R9

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
15	NativeDecl : **native identifier lbrace** * NativeFunctionList **rbrace** NativeFunctionList : * NativeFunctionList NativeFunctionDecl NativeFunctionList : *	NativeFunctionList → 17 R13
16	LibraryDecl : **library identifier lbrace** FunctionList * **rbrace** FunctionList : FunctionList * FunctionDecl FunctionDecl : * Access FunctionType **identifier** ArgumentDecl FunctionBody Access : * **public** Access : * **private**	**rbrace** → 18 FunctionDecl → 19 Access → 20 **public** → 21 **private** → 22
17	NativeDecl : **native identifier lbrace** NativeFunctionList * **rbrace** NativeFunctionList : NativeFunctionList * NativeFunctionDecl NativeFunctionDecl : * Access FunctionType **identifier** ArgumentDecl **semicolon** Access : * **public** Access : * **private**	**rbrace** → 23 NativeFunctionDecl → 24 Access → 25 **public** → 21 **private** → 22
18	LibraryDecl : **library identifier lbrace** FunctionList **rbrace** *	R7
19	FunctionList : FunctionList FunctionDecl *	R8
20	FunctionDecl : Access * FunctionType **identifier** ArgumentDecl FunctionBody FunctionType : * Type FunctionType : * **void** Type : * **int** Type : * **char** Type : * **boolean**	FunctionType → 26 Type → 27 **void** → 28 **int** → 29 **char** → 30 **boolean** → 31
21	Access : **public** *	R15
22	Access : **private** *	R16
23	NativeDecl : **native identifier lbrace** NativeFunctionList **rbrace** *	R11
24	NativeFunctionList : NativeFunctionList NativeFunctionDecl *	R12

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
25	NativeFunctionDecl : Access * FunctionType **identifier** ArgumentDecl **semicolon** FunctionType : * Type FunctionType : * **void** Type : * **int** Type : * **char** Type : * **boolean**	FunctionType → 32 Type → 27 **void** → 28 **int** → 29 **char** → 30 **boolean** → 31
26	FunctionDecl : Access FunctionType * **identifier** ArgumentDecl FunctionBody	**identifier** → 33
27	FunctionType : Type *	R17
28	FunctionType : **void** *	R18
29	Type : **int** *	R19
30	Type : **char** *	R20
31	Type : **boolean** *	R21
32	NativeFunctionDecl : Access FunctionType * **identifier** ArgumentDecl **semicolon**	**identifier** → 34
33	FunctionDecl : Access FunctionType **identifier** * ArgumentDecl FunctionBody ArgumentDecl : * **lparen rparen** ArgumentDecl : * **lparen** ArgumentList **rparen**	ArgumentDecl → 35 **lparen** → 36
34	NativeFunctionDecl : Access FunctionType **identifier** * ArgumentDecl **semicolon** ArgumentDecl : * **lparen rparen** ArgumentDecl : * **lparen** ArgumentList **rparen**	ArgumentDecl → 37 **lparen** → 36
35	FunctionDecl : Access FunctionType **identifier** ArgumentDecl * FunctionBody FunctionBody : * **lbrace** StatementList **rbrace**	FunctionBody → 38 **lbrace** → 39
36	ArgumentDecl : **lparen** * **rparen** ArgumentDecl : **lparen** * ArgumentList **rparen** ArgumentList : * Argument ArgumentList : * ArgumentList **comma** Argument Argument : * Type **identifier** Type : * **int** Type : * **char** Type : * **boolean**	**rparen** → 40 ArgumentList → 41 Argument → 42 Type → 43 **int** → 29 **char** → 30 **boolean** → 31

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
37	NativeFunctionDecl : Access FunctionType **identifier** ArgumentDecl * **semicolon**	**semicolon** → 44
38	FunctionDecl : Access FunctionType **identifier** ArgumentDecl FunctionBody *	R10
39	FunctionBody : **lbrace** * StatementList **rbrace** StatementList : * StatementList Statement StatementList : *	StatementList → 45 R29
40	ArgumentDecl : **lparen** **rparen** *	R22
41	ArgumentDecl : **lparen** ArgumentList * **rparen** ArgumentList : ArgumentList * **comma** Argument	**rparen** → 46 **comma** → 47
42	ArgumentList : Argument *	R24
43	Argument : Type * **identifier**	**identifier** -> 48
44	NativeFunctionDecl : Access FunctionType **identifier** ArgumentDecl **semicolon** *	R14
45	FunctionBody : **lbrace** StatementList * **rbrace** StatementList : StatementList * Statement Statement : * Decl Statement : * IdStm Statement : * IfStm Statement : * WhileStm Statement : * ReturnStm Statement : * NoStm Statement : * BlockStm Decl : * Type IdList **semicolon** Type : * **int** Type : * **char** Type : * **boolean** IfStm : * **if** **lparen** Expr **rparen** Statement IfStm : * **if** **lparen** Expr **rparen** Statement **else** Statement WhileStm : * **while** **lparen** Expr **rparen** Statement ReturnStm : * **return** Expr **semicolon** ReturnStm : * **return** **semicolon** NoStm : * **semicolon** IdStm : * **identifier** **assign** Expr **semicolon** IdStm : * **identifier** FunctionCall **semicolon** IdStm : * **identifier** **dot** **identifier** FunctionCall **semicolon** BlockStm : * **lbrace** StatementList **rbrace**	**rbrace** → 49 Statement → 50 Decl → 51 IdStm → 52 IfStm → 53 WhileStm → 54 ReturnStm → 55 NoStm → 56 BlockStm → 57 Type → 58 **int** → 29 **char** → 30 **boolean** → 31 **if** → 59 **while** → 60 **return** → 61 **semicolon** → 62 **identifier** → 63 **lbrace** → 64

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
46	ArgumentDecl : **lparen** ArgumentList **rparen** *	R23
47	ArgumentList : ArgumentList **comma** * Argument Argument : * Type **identifier** Type : * **int** Type : * **char** Type : * **boolean**	Argument → 65 Type → 43 **int** → 29 **char** → 30 **boolean** → 31
48	Argument : Type **identifier** *	R26
49	FunctionBody : **lbrace** StatementList **rbrace** *	R27
50	StatementList : StatementList Statement *	R28
51	Statement : Decl *	R30
52	Statement : IdStm *	R31
53	Statement : IfStm *	R32
54	Statement : WhileStm *	R33
55	Statement : ReturnStm *	R34
56	Statement : NoStm *	R35
57	Statement : BlockStm *	R36
58	Decl : Type * IdList **semicolon** IdList : * **identifier** IdList : * **identifier assign** Expr IdList : * IdList **comma identifier** IdList : * IdList **comma identifier assign** Expr	IdList → 66 **identifier** → 67
59	IfStm : **if** * **lparen** Expr **rparen** Statement IfStm : **if** * **lparen** Expr **rparen** Statement **else** Statement	**lparen** → 68
60	WhileStm : **while** * **lparen** Expr **rparen** Statement	**lparen** → 69

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
61	ReturnStm : **return** * Expr **semicolon** ReturnStm : **return** * **semicolon** Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Expr → 70 **semicolon** → 71 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
62	NoStm : **semicolon** *	R47
63	IdStm : **identifier** * **assign** Expr **semicolon** IdStm : **identifier** * FunctionCall **semicolon** IdStm : **identifier** * **dot identifier** FunctionCall **semicolon** FunctionCall : * **lparen rparen** FunctionCall : * **lparen** ExprList **rparen**	**assign** → 88 FunctionCall → 89 **dot** → 90 **lparen** → 91

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
64	BlockStm : **lbrace** * StatementList **rbrace** StatementList : * StatementList Statement StatementList : *	StatementList → 92 R29
65	ArgumentList : ArgumentList **comma** Argument *	R25
66	Decl : Type IdList * **semicolon** IdList : IdList * **comma identifier** IdList : IdList * **comma identifier assign** Expr	**semicolon** → 93 **comma** → 94
67	IdList : **identifier** * IdList : **identifier** * **assign** Expr	R38 **assign** → 95

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
68	IfStm : **if** **lparen** * Expr **rparen** Statement IfStm : **if** **lparen** * Expr **rparen** Statement **else** Statement Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier** **dot** **identifier** FunctionCall	Expr → 96 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
69	WhileStm : **while lparen** * Expr **rparen** Statement Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Expr → 97 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
70	ReturnStm : **return** Expr * **semicolon** Expr : Expr * **or** AndExpr	**semicolon** → 98 **or** → 99
71	ReturnStm : **return semicolon** *	R46
72	Expr : AndExpr * AndExpr : AndExpr * **and** RelExpr	R52 **and** → 100
73	AndExpr : RelExpr *	R54

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
74	RelExpr : SumExpr *	R56
	RelExpr : SumExpr * **eq** SumExpr	**eq** → 101
	RelExpr : SumExpr * **ne** SumExpr	**ne** → 102
	RelExpr : SumExpr * **gt** SumExpr	**gt** → 103
	RelExpr : SumExpr * **ge** SumExpr	**ge** → 104
	RelExpr : SumExpr * **lt** SumExpr	**lt** → 105
	RelExpr : SumExpr * **le** SumExpr	**le** → 106
	SumExpr : SumExpr * **minus** ProdExpr	**minus** → 107
	SumExpr : SumExpr * **plus** ProdExpr	**plus** → 108
75	SumExpr : **not** * ProdExpr	ProdExpr → 109
	ProdExpr : * Factor	Factor → 79
	ProdExpr : * ProdExpr **prod** Factor	
	ProdExpr : * ProdExpr **div** Factor	
	ProdExpr : * ProdExpr **mod** Factor	
	Factor : * Literal	Literal → 80
	Factor : * Reference	Reference → 81
	Factor : * **lparen** Expr **rparen**	**lparen** → 82
	Literal : * **integer_literal**	**integer_literal** →83
	Literal : * **char_literal**	**char_literal** → 84
	Literal : * **true**	**true** → 85
	Literal : * **false**	**false** → 86
	Reference : * **identifier**	**identifier** → 87
	Reference : * **identifier** FunctionCall	
	Reference : * **identifier dot identifier** FunctionCall	
76	SumExpr : **minus** * ProdExpr	ProdExpr → 110
	ProdExpr : * Factor	Factor → 79
	ProdExpr : * ProdExpr **prod** Factor	
	ProdExpr : * ProdExpr **div** Factor	
	ProdExpr : * ProdExpr **mod** Factor	
	Factor : * Literal	Literal → 80
	Factor : * Reference	Reference → 81
	Factor : * **lparen** Expr **rparen**	**lparen** → 82
	Literal : * **integer_literal**	**integer_literal** →83
	Literal : * **char_literal**	**char_literal** → 84
	Literal : * **true**	**true** → 85
	Literal : * **false**	**false** → 86
	Reference : * **identifier**	**identifier** → 87
	Reference : * **identifier** FunctionCall	
	Reference : * **identifier dot identifier** FunctionCall	

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
77	SumExpr : **plus** * ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	ProdExpr → 111 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
78	SumExpr : ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R66 **prod** → 112 **div** → 113 **mod** → 114
79	ProdExpr : Factor *	R69
80	Factor : Literal *	R73
81	Factor : Reference *	R74

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
82	Factor : **lparen** * Expr **rparen** Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Expr → 115 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
83	Literal : **integer_literal** *	R76
84	Literal : **char_literal** *	R77
85	Literal : **true** *	R78
86	Literal : **false** *	R79

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
87	Reference : **identifier** * Reference : **identifier** * FunctionCall Reference : **identifier** * **dot identifier** FunctionCall FunctionCall : * **lparen** **rparen** FunctionCall : * **lparen** ExprList **rparen**	R80 FunctionCall → 116 **dot** → 117 **lparen** → 91
88	IdStm : **identifier** **assign** * Expr **semicolon** Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier** **dot** **identifier** FunctionCall	Expr → 118 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
89	IdStm : **identifier** FunctionCall * **semicolon**	**semicolon** → 119
90	IdStm : **identifier** **dot** * **identifier** FunctionCall **semicolon**	**identifier** → 120

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
91	FunctionCall : **lparen** * **rparen** FunctionCall : **lparen** * ExprList **rparen** ExprList : * Expr ExprList : * ExprList comma Expr Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	**rparen** -> 121 ExprList -> 122 Expr -> 123 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
92	BlockStm : **lbrace** StatementList * **rbrace** StatementList : StatementList * Statement Statement : * Decl Statement : * IdStm Statement : * IfStm Statement : * WhileStm Statement : * ReturnStm Statement : * NoStm Statement : * BlockStm Decl : * Type IdList **semicolon** Type : * **int** Type : * **char** Type : * **boolean** IfStm : * **if lparen** Expr **rparen** Statement IfStm : * **if lparen** Expr **rparen** Statement **else** Statement WhileStm : * **while lparen** Expr **rparen** Statement ReturnStm : * **return** Expr **semicolon** ReturnStm : * **return semicolon** NoStm : * **semicolon** IdStm : * **identifier assign** Expr **semicolon** IdStm : * **identifier** FunctionCall **semicolon** IdStm : * **identifier dot identifier** FunctionCall **semicolon** BlockStm : * **lbrace** StatementList **rbrace**	**rbrace** → 124 Statement → 50 Decl → 51 IdStm → 52 IfStm → 53 WhileStm → 54 ReturnStm → 55 NoStm → 56 BlockStm → 57 Type → 58 **int** → 29 **char** → 30 **boolean** → 31 **if** → 59 **while** → 60 **return** → 61 **semicolon** → 62 **identifier** → 63 **lbrace** → 64
93	Decl : Type IdList **semicolon** *	R37
94	IdList : IdList **comma** * **identifier** IdList : IdList **comma** * **identifier assign** Expr	**identifier** → 125

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
95	IdList : **identifier assign** * Expr Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Expr → 126 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
96	IfStm : **if lparen** Expr * **rparen** Statement IfStm : **if lparen** Expr * **rparen** Statement **else** Statement Expr : Expr * **or** AndExpr	**rparen** → 127 **or** → 99
97	WhileStm : **while lparen** Expr * **rparen** Statement Expr : Expr * **or** AndExpr	**rparen** → 128 **or** → 99
98	ReturnStm : **return** Expr **semicolon** *	R45

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
99	Expr : Expr **or** * AndExpr	AndExpr → 129
	AndExpr : * RelExpr	RelExpr → 73
	AndExpr : * AndExpr **and** RelExpr	
	RelExpr : * SumExpr	SumExpr → 74
	RelExpr : * SumExpr **eq** SumExpr	
	RelExpr : * SumExpr **ne** SumExpr	
	RelExpr : * SumExpr **gt** SumExpr	
	RelExpr : * SumExpr **ge** SumExpr	
	RelExpr : * SumExpr **lt** SumExpr	
	RelExpr : * SumExpr **le** SumExpr	
	SumExpr : * **not** ProdExpr	**not** → 75
	SumExpr : * **minus** ProdExpr	**minus** → 76
	SumExpr : * **plus** ProdExpr	**plus** → 77
	SumExpr : * ProdExpr	ProdExpr → 78
	SumExpr : * SumExpr **minus** ProdExpr	
	SumExpr : * SumExpr **plus** ProdExpr	
	ProdExpr : * Factor	Factor → 79
	ProdExpr : * ProdExpr **prod** Factor	
	ProdExpr : * ProdExpr **div** Factor	
	ProdExpr : * ProdExpr **mod** Factor	
	Factor : * Literal	Literal → 80
	Factor : * Reference	Reference → 81
	Factor : * **lparen** Expr **rparen**	**lparen** → 82
	Literal : * **integer_literal**	**integer_literal** →83
	Literal : * **char_literal**	**char_literal** → 84
	Literal : * **true**	**true** → 85
	Literal : * **false**	**false** → 86
	Reference : * **identifier**	**identifier** → 87
	Reference : * **identifier** FunctionCall	
	Reference : * **identifier dot identifier** FunctionCall	

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
100	AndExpr : AndExpr **and** * RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	RelExpr → 130 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
101	RelExpr : SumExpr **eq** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 131 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
102	RelExpr : SumExpr **ne** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 132 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
103	RelExpr : SumExpr **gt** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 133 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
104	RelExpr : SumExpr **ge** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 134 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
105	RelExpr : SumExpr **lt** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 135 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
106	RelExpr : SumExpr **le** * SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	SumExpr → 136 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
107	SumExpr : SumExpr **minus** * ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	ProdExpr → 137 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
108	SumExpr : SumExpr **plus** * ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	ProdExpr → 138 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
109	SumExpr : **not** ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R63 **prod** → 112 **div** → 113 **mod** → 114
110	SumExpr : **minus** ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R64 **prod** → 112 **div** → 113 **mod** → 114

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
111	SumExpr : **plus** ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R65 **prod** → 112 **div** → 113 **mod** → 114
112	ProdExpr : ProdExpr **prod** * Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Factor → 139 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
113	ProdExpr : ProdExpr **div** * Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Factor → 140 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
114	ProdExpr : ProdExpr **mod** * Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Factor → 141 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
115	Factor : **lparen** Expr * **rparen** Expr : Expr * **or** AndExpr	**rparen** → 142 **or** → 99

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
116	Reference : **identifier** FunctionCall *	R81
117	Reference : **identifier dot** * **identifier** FunctionCall	**identifier** → 143
118	IdStm : **identifier assign** Expr * **semicolon** Expr : Expr * **or** AndExpr	**semicolon** → 144 **or** → 99
119	IdStm : **identifier** FunctionCall **semicolon** *	R49
120	IdStm : **identifier dot identifier** * FunctionCall **semicolon** FunctionCall : * **lparen rparen** FunctionCall : * **lparen** ExprList **rparen**	FunctionCall → 145 **lparen** → 91
121	FunctionCall : **lparen rparen** *	R83
122	FunctionCall : **lparen** ExprList * **rparen** ExprList : ExprList * **comma** Expr	**rparen** →146 **comma** → 147
123	ExprList : Expr * Expr : Expr * **or** AndExpr	R85 **or** → 99
124	BlockStm : **lbrace** StatementList **rbrace** *	R51
125	IdList : IdList **comma identifier** * IdList : IdList **comma identifier** * **assign** Expr	R40 **assign** → 148
126	IdList : **identifier assign** Expr * Expr : Expr * **or** AndExpr	R39 **or** → 99

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
127	IfStm : **if** **lparen** Expr **rparen** * Statement IfStm : **if** **lparen** Expr **rparen** * Statement **else** Statement Statement : * Decl Statement : * IdStm Statement : * IfStm Statement : * WhileStm Statement : * ReturnStm Statement : * NoStm Statement : * BlockStm Decl : * Type IdList **semicolon** Type : * **int** Type : * **char** Type : * **boolean** IfStm : * **if** **lparen** Expr **rparen** Statement IfStm : * **if** **lparen** Expr **rparen** Statement **else** Statement WhileStm : * **while** **lparen** Expr **rparen** Statement ReturnStm : * **return** Expr **semicolon** ReturnStm : * **return** **semicolon** NoStm : * **semicolon** IdStm : * **identifier** **assign** Expr **semicolon** IdStm : * **identifier** FunctionCall **semicolon** IdStm : * **identifier** **dot** **identifier** FunctionCall **semicolon** BlockStm : * **lbrace** StatementList **rbrace**	Statement → 149 Decl → 51 IdStm → 52 IfStm → 53 WhileStm → 54 ReturnStm → 55 NoStm → 56 BlockStm → 57 Type → 58 **int** → 29 **char** → 30 **boolean** → 31 **if** → 59 **while** → 60 **return** → 61 **semicolon** → 62 **identifier** → 63 **lbrace** → 64

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
128	WhileStm : **while lparen** Expr **rparen** * Statement Statement : * Decl Statement : * IdStm Statement : * IfStm Statement : * WhileStm Statement : * ReturnStm Statement : * NoStm Statement : * BlockStm Decl : * Type IdList **semicolon** Type : * **int** Type : * **char** Type : * **boolean** IfStm : * **if lparen** Expr **rparen** Statement IfStm : * **if lparen** Expr **rparen** Statement **else** Statement WhileStm : * **while lparen** Expr **rparen** Statement ReturnStm : * **return** Expr **semicolon** ReturnStm : * **return semicolon** NoStm : * **semicolon** IdStm : * **identifier assign** Expr **semicolon** IdStm : * **identifier** FunctionCall **semicolon** IdStm : * **identifier dot identifier** FunctionCall **semicolon** BlockStm : * **lbrace** StatementList **rbrace**	Statement → 150 Decl → 51 IdStm → 52 IfStm → 53 WhileStm → 54 ReturnStm → 55 NoStm → 56 BlockStm → 57 Type → 58 **int** → 29 **char** → 30 **boolean** → 31 **if** → 59 **while** → 60 **return** → 61 **semicolon** → 62 **identifier** → 63 **lbrace** → 64
129	Expr : Expr **or** AndExpr * AndExpr : AndExpr * **and** RelExpr	R53 **and** → 100
130	AndExpr : AndExpr **and** RelExpr *	R55
131	RelExpr : SumExpr **eq** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R57 **minus** → 107 **plus** → 108
132	RelExpr : SumExpr **ne** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R58 **minus** → 107 **plus** → 108
133	RelExpr : SumExpr **gt** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R59 **minus** → 107 **plus** → 108
134	RelExpr : SumExpr **ge** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R60 **minus** → 107 **plus** → 108

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
135	RelExpr : SumExpr **lt** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R61 **minus** → 107 **plus** → 108
136	RelExpr : SumExpr **le** SumExpr * SumExpr : SumExpr * **minus** ProdExpr SumExpr : SumExpr * **plus** ProdExpr	R62 **minus** → 107 **plus** → 108
137	SumExpr : SumExpr **minus** ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R67 **prod** → 112 **div** → 113 **mod** → 114
138	SumExpr : SumExpr **plus** ProdExpr * ProdExpr : ProdExpr * **prod** Factor ProdExpr : ProdExpr * **div** Factor ProdExpr : ProdExpr * **mod** Factor	R68 **prod** → 112 **div** → 113 **mod** → 114
139	ProdExpr : ProdExpr **prod** Factor *	R70
140	ProdExpr : ProdExpr **div** Factor *	R71
141	ProdExpr : ProdExpr **mod** Factor *	R72
142	Factor : **lparen** Expr **rparen** *	R75
143	Reference : **identifier dot identifier** * FunctionCall FunctionCall : * **lparen rparen** FunctionCall : * **lparen** ExprList **rparen**	FunctionCall → 151 **lparen** → 91
144	IdStm : **identifier assign** Expr **semicolon** *	R48
145	IdStm : **identifier dot identifier** FunctionCall * **semicolon**	**semicolon** → 152
146	FunctionCall : **lparen** ExprList **rparen** *	R84

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
147	ExprList : ExprList **comma** * Expr	Expr → 153
	Expr : * AndExpr	AndExpr → 72
	Expr : * Expr **or** AndExpr	
	AndExpr : * RelExpr	RelExpr → 73
	AndExpr : * AndExpr **and** RelExpr	
	RelExpr : * SumExpr	SumExpr → 74
	RelExpr : * SumExpr **eq** SumExpr	
	RelExpr : * SumExpr **ne** SumExpr	
	RelExpr : * SumExpr **gt** SumExpr	
	RelExpr : * SumExpr **ge** SumExpr	
	RelExpr : * SumExpr **lt** SumExpr	
	RelExpr : * SumExpr **le** SumExpr	
	SumExpr : * **not** ProdExpr	**not** → 75
	SumExpr : * **minus** ProdExpr	**minus** → 76
	SumExpr : * **plus** ProdExpr	**plus** → 77
	SumExpr : * ProdExpr	ProdExpr → 78
	SumExpr : * SumExpr **minus** ProdExpr	
	SumExpr : * SumExpr **plus** ProdExpr	
	ProdExpr : * Factor	Factor → 79
	ProdExpr : * ProdExpr **prod** Factor	
	ProdExpr : * ProdExpr **div** Factor	
	ProdExpr : * ProdExpr **mod** Factor	
	Factor : * Literal	Literal → 80
	Factor : * Reference	Reference → 81
	Factor : * **lparen** Expr **rparen**	**lparen** → 82
	Literal : * **integer_literal**	**integer_literal** →83
	Literal : * **char_literal**	**char_literal** → 84
	Literal : * **true**	**true** → 85
	Literal : * **false**	**false** → 86
	Reference : * **identifier**	**identifier** → 87
	Reference : * **identifier** FunctionCall	
	Reference : * **identifier dot identifier** FunctionCall	

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
148	IdList : IdList **comma identifier assign** * Expr Expr : * AndExpr Expr : * Expr **or** AndExpr AndExpr : * RelExpr AndExpr : * AndExpr **and** RelExpr RelExpr : * SumExpr RelExpr : * SumExpr **eq** SumExpr RelExpr : * SumExpr **ne** SumExpr RelExpr : * SumExpr **gt** SumExpr RelExpr : * SumExpr **ge** SumExpr RelExpr : * SumExpr **lt** SumExpr RelExpr : * SumExpr **le** SumExpr SumExpr : * **not** ProdExpr SumExpr : * **minus** ProdExpr SumExpr : * **plus** ProdExpr SumExpr : * ProdExpr SumExpr : * SumExpr **minus** ProdExpr SumExpr : * SumExpr **plus** ProdExpr ProdExpr : * Factor ProdExpr : * ProdExpr **prod** Factor ProdExpr : * ProdExpr **div** Factor ProdExpr : * ProdExpr **mod** Factor Factor : * Literal Factor : * Reference Factor : * **lparen** Expr **rparen** Literal : * **integer_literal** Literal : * **char_literal** Literal : * **true** Literal : * **false** Reference : * **identifier** Reference : * **identifier** FunctionCall Reference : * **identifier dot identifier** FunctionCall	Expr → 154 AndExpr → 72 RelExpr → 73 SumExpr → 74 **not** → 75 **minus** → 76 **plus** → 77 ProdExpr → 78 Factor → 79 Literal → 80 Reference → 81 **lparen** → 82 **integer_literal** →83 **char_literal** → 84 **true** → 85 **false** → 86 **identifier** → 87
149	IfStm : **if lparen** Expr **rparen** Statement * IfStm : **if lparen** Expr **rparen** Statement ***else** Statement	R42 **else** → 155
150	WhileStm : **while lparen** Expr **rparen** Statement *	R44
151	Reference : **identifier dot identifier** FunctionCall *	R82
152	IdStm : **identifier dot identifier** FunctionCall **semicolon** *	R50
153	ExprList : ExprList **comma** Expr * Expr : Expr * **or** AndExpr	R86 **or** → 99

Análisis SLR de la gramática de Tinto

Estado	Reglas	Transiciones
154	IdList : IdList **comma identifier assign** Expr * Expr : Expr * **or** AndExpr	R41 **or** → 99
155	IfStm : **if lparen** Expr **rparen** Statement **else** * Statement Statement : * Decl Statement : * IdStm Statement : * IfStm Statement : * WhileStm Statement : * ReturnStm Statement : * NoStm Statement : * BlockStm Decl : * Type IdList **semicolon** Type : * **int** Type : * **char** Type : * **boolean** IfStm : * **if lparen** Expr **rparen** Statement IfStm : * **if lparen** Expr **rparen** Statement **else** Statement WhileStm : * **while lparen** Expr **rparen** Statement ReturnStm : * **return** Expr **semicolon** ReturnStm : * **return semicolon** NoStm : * **semicolon** IdStm : * **identifier assign** Expr **semicolon** IdStm : * **identifier** FunctionCall **semicolon** IdStm : * **identifier dot identifier** FunctionCall **semicolon** BlockStm : * **lbrace** StatementList **rbrace**	Statement → 156 Decl → 51 IdStm → 52 IfStm → 53 WhileStm → 54 ReturnStm → 55 NoStm → 56 BlockStm → 57 Type → 58 **int** → 29 **char** → 30 **boolean** → 31 **if** → 59 **while** → 60 **return** → 61 **semicolon** → 62 **identifier** → 63 **lbrace** → 64
156	IfStm : **if lparen** Expr **rparen** Statement **else** Statement *	R43