

ANÁLISIS SINTÁCTICO					
>> Gramática formal <<					
G = {NT, T, RP, So}					
No Terminales [NT]	Terminales [T]	Reglas de Producción [RP]			
		No.	Regla	Estado Inicial [So]	
inicio	EOF	1	inicio : clase EOF		
clase	SANGRIA	2	clase : nl class_elements		
class_elements	IMPORT		SANGRIA nl class_elements		
header	ID		class_elements		
imports	CRL	3	class_elements : header content		
import	INCERTEZA		content		
incerteza	INT		header : imports incerteza		
content	DOUBLE	4	imports		
sentences	STRING		incerteza		
class_content	BOOLEAN		imports : imports import		
class_content_elements	CHAR	5	import		
declaracion_var_global	VOID	6	import : IMPORT ID ':' CRL nl		
declaracion_var	MAIN	7	incerteza : INCERTEZA expression nl		
creacion_vars	UMINUS	8	content : content sentences		
var_list	INTEGER		sentences		
content_type	DECIMAL	9	sentences : class_content		
declaracion_funcion	CADENA		function_content		
params	CHARACTER	10	class_content: class_content_elements nl		
params_list	TRUE	11	class_content_elements : declaracion_var_global		
param	FALSE		declaracion_funcion		
function_content	MOSTRAR	12	declaracion_var_global : declaracion_var		
function_sentence	CADENA	13	declaracion_var : content_type creacion_vars asignacion_value		
only_sentence	DRAW_AST		content_type creacion_vars		
asignacion_var	DRAW_EXP	14	creacion_vars : var_list		
asignacion_value	DRAW_TS	15	var_list : var_list ',' ID		
expression	RETORNO		ID		
expr	CONTINUAR	16	content_type : INT		
expr2	DETENER		DOUBLE		
expr3	PARA		STRING		
expr4	INCREMENTO		BOOLEAN		
expr5	MIENTRAS		CHAR		
expr6	SI	17	declaracion_funcion : content_type ID '(' params ')' ':'		
expr7	SINO		VOID ID		
expr8	NEW_LINE		VOID MAIN		
expr9		18	params : params_list		
expr10		19	params_list : params_list ',' param		
booleano			param		
contenido_var		20	param : content_type ID		
invocacion		21	function_content : SANGRIA function_sentence nl		
contenido_asignacion		22	function_sentence : only_sentence		
argumentos			loop_sentence		
dibujar			control_sentence		
breakpoints			only_sentence : declaracion_var		
loop_sentence			asignacion_var		
for_var		23	invocacion		

		control_sentence		43	mostrar
		nl			dibujar
					breakpoints
				24	asignacion_var : ID asignacion_value
				25	asignacion_value : '=' expression
				26	expression : expr
					expr : expr '+' expr2
				27	expr '-' expr2
					expr2
					expr2 : expr2 '*' expr3
				28	expr2 '/' expr3
					expr2 '%' expr3
					expr3
				29	expr3 : expr4 '^' expr3
					expr4
				30	expr4 : ':' expr5 %prec UMINUS
					expr5
					expr5 : expr5 '=' expr6
				31	expr5 '!' expr6
					expr5 '<' expr6
					expr5 '>' expr6
					expr5 '<=' expr6
					expr5 '>=' expr6
					expr5 '-' expr6
					expr6
				32	expr6 : expr6 OR expr7
					expr7
				33	expr7 : expr7 XOR expr8
					expr8
				34	expr8 : expr8 '&&' expr9
					expr9
				35	expr9 : '!' expr10
					expr10
				36	expr10 : INTEGER
					DECIMAL
					CADENA
					booleano
					CHARACTER
					contenido_var
					'(' expr ')'
				37	booleano : TRUE
					FALSE
				38	contenido_var : ID
					invocacion
				39	invocacion : ID '(' argumentos ')'
					ID '(' ')'
				40	mostrar : MOSTRAR '(' CADENA contenido_asignacion ')'
				41	contenido_asignacion : ':' argumentos
				42	argumentos : argumentos ',' expression
					expression
				43	dibujar : DRAW_AST '(' ID ')'
					DRAW_EXP '(' expression ')'
					DRAW_TS '(' ')'

inicio

				44	breakpoints: RETORNO
					RETORNO expression
					CONTINUAR
					DETENER
				45	loop_sentence: PARA '(' for_var ':' expression ':' INCREMENTO ')' ':'
					MIENTRAS '(' expression ')' ':'
				46	for_var: INT ID '=' INTEGER
				47	control_sentence: SI '(' expression ')' ':'
					SINO ':'
				48	nl: nl NEW_LINE
					NEW_LINE

ANÁLISIS SINTÁCTICO					
		>> Gramática formal <<			
		G = {NT, T, RP, So}			
		Reglas de Producción [RP]			
		No Terminales [NT]	Terminales [T]	No.	Regla
		INICIO	<u>import</u>		
		IMPORTS	<u>class</u>	1	INICIO -> IMPORTS CLASE CLASE
		IIMPORT	<u>this</u>	2	IMPORTS -> <u>import</u> IIMPORT IIMPORT
		DIRECCION_IMPORT		3	IIMPORT -> <u>import</u> DIRECCION_IMPORT ;
		DIRECCION	.		
		CLASE	;		
		VISIBILIDAD	*	4	IMPORT -> DIRECCION , * DIRECCION
		DEF_CLASE	{		
		BLOQUE_CLASE	}	5	DIRECCION -> DIRECCION , <u>nombre</u> <u>nombre</u> , <u>nombre</u>
		SENTENCIAS_CLASE	;		
		DEF_VARIABLE	.	6	CLASE -> VISIBILIDAD DEF_CLASE DEF_CLASE
		DEF_TIPO_VARIABLE	<u>public</u>	7	
		CREACION_VARIABLE_NUMERO	<u>protected</u>	8	VISIBILIDAD -> <u>public</u> <u>protected</u> . <u>private</u> <u>final</u>
		VARIABLE_NUMERO	<u>private</u>		
		CREACION_VARIABLE_STRING	<u>final</u>		
		VARIABLE_STRING	<u>nombre</u>	9	DEF_CLASE -> class objeto { BLOQUE_CLASE } class objeto { }
		CREACION_VARIABLE_CHAR			
		VARIABLE_CHAR	<u>new</u>	10	BLOQUE_CLASE -> BLOQUE_CLASE SENTENCIAS_CLASE SENTENCIAS_CLASE
		CREACION_VARIABLE_BOOLEAN	<u>int</u>		
		VARIABLE_BOOLEAN	<u>boolean</u>	11	
		CREACION_VARIABLE_OBJECT	<u>String</u>	12	SENTENCIAS_CLASE -> DEF_VARIABLE DEF_METODO
		VARIABLE_OBJECT	<u>char</u>	13	
		CREACION_VARIABLE_OBJETO	<u>double</u>	14	DEF_VARIABLE -> VISIBILIDAD DEF_TIPO_VARIABLE ; DEF_TIPO_VARIABLE ;
		VARIABLE_OBJETO	<u>Object</u>	15	DEF_TIPO_VARIABLE -> int CREACION_VARIABLE.:NUMERO double CREACION_VARIABLE.:NUMERO string CREACION_VARIABLE_STRING char CREACION_VARIABLE_CHAR boolean CREACION_VARIABLE_BOOLEAN object CREACION_VARIABLE_OBJECT objeto CREACION_VARIABLE_OBJETO
		DEF_METODO	<u>Objeto</u> <u>!si es que harás la revisión de la lista</u>	16	CREACION_VARIABLE_NUMERO -> CREACION_VARIABLE_NUMERO , VARIABLE_NUMERO VARIABLE_NUMERO
		CUERPO_DEF_METODO	<u>void</u>	17	VARIABLE_NUMERO -> nombre = CONTENIDO_NUMERO nombre = CONTENIDO_CHAR nombre
		DEF_METODO_CONSTRUCTOR	<u>cadena</u>	18	CREACION_VARIABLE_STRING -> CREACION_VARIABLE_STRING , VARIABLE_STRING VARIABLE_STRING
		CUERPO_METODO_CONSTRUCTOR	<u>valor</u>	19	VARIABLE_STRING -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_STRING nombre
		DEF_METODO_TIPO	<u>numero</u>	20	CREACION_VARIABLE_CHAR -> CREACION_VARIABLE_CHAR , VARIABLE_CHAR VARIABLE_CHAR
		CUERPO_METODO_NUMERO	<u>±</u>	21	VARIABLE_CHAR -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_CHAR nombre

		CUERPO_METODO_STRING	=	22	CREACION_VARIABLE_BOOLEAN -> CREACION_VARIABLE_BOOLEAN , VARIABLE_BOOLEAN VARIABLE_BOOLEAN
		CUERPO_METODO_CHAR	/	23	VARIABLE_BOOLEAN -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_BOOLEAN nombre
		CUERPO_METODO_BOOLEAN	=	24	CREACION_VARIABLE_OBJECT -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJECT VARIABLE_OBJECT
		CUERPO_METODO_OBJECT	{	25	VARIABLE_OBJECT -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_OBJECT nombre = CONTENIDO_OBJETO nombre
		CUERPO_METODO_OBJETO	}	26	CREACION_VARIABLE_OBJECT -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJECT VARIABLE_OBJECT
		DEF_METODO_VOID	for	27	VARIABLE_OBJETO -> nombre = CONTENIDO_VARIABLE nombre = CONTENIDO_OBJETO nombre
		CUERPO_METODO_VOID	while	28	DEF_METODO -> VISIBILIDAD CUERPO_DEF_METODO CUERPO_DEF_METODO
		PARAMETROS	do	29	CUERPO_DEF_METODO -> DEF_METODO_CONSTRUCTOR DEF_METODO_TIPO DEF_METODO_VOID
		PARAMETRO	if	30	DEF_METODO_CONSTRUCTOR -> objeto (PARAMETROS) CUERPO_METODO_CONSTRUCTOR
		TIPO	else	31	CUERPO_METODO_CONSTRUCTOR -> { BLOQUE } { }
		BLOQUE	switch		
		SENTENCIAS	case	32	DEF_METODO_TIPO -> int nombre (PARAMETROS) CUERPO_METODO_NUMERICO double nombre (PARAMETROS) CUERPO_METODO_NUMERICO string nombre (PARAMETROS) CUERPO_METODO_STRING char nombre (PARAMETROS) CUERPO_METODO_CHAR boolean nombre (PARAMETROS) CUERPO_METODO_NUMERICO object nombre (PARAMETROS) CUERPO_METODO_NUMERICO objeto nombre (PARAMETROS) CUERPO_METODO_NUMERICO
		ASIGNACIÓN	default	33	CUERPO_METODO_NUMERO -> { BLOQUE_RETORNO_NUMERO } { RETORNO_NUMERO } { BLOQUE_RETONO_CHAR } { RETONO_CHAR }
		TIPOS_ASIGNACIÓN	!		
		CONTENIDO_VARIABLE	&		
		CONTENIDO_NUMERO		34	CUERPO_METODO_STRING -> { BLOQUE_RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE_RETONO_STRING } { RETONO_STRING }
		CONTENIDO_STRING	≤	35	CUERPO_METODO_CHAR -> { BLOQUE_RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE_RETONO_CHAR } { RETONO_CHAR }
		PERMITIDOS	≥	36	CUERPO_METODO_BOOLEAN -> { BLOQUE_RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE_RETONO_BOOLEAN } { RETONO_BOOLEAN }
		CUALQUIER_CONTENIDO	!=	37	CUERPO_METODO_OBJECT -> { BLOQUE_RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE_RETONO_OBJECT } { RETONO_OBJECT }
		OPCIONES_CUALQUIER_CONTENIDO	&&	38	CUERPO_METODO_OBJETO -> { BLOQUE_RETORNO_VARIABLE } { RETORNO_VARIABLE } { BLOQUE_RETONO_OBJETO } { RETONO_OBJETO }
		STRINGS		39	DEF_METODO_VOID -> void nombre CUERPO_METODO_VOID
		CONTENIDO_CHAR	≤=	40	CUERPO_METODO_VOID -> { PARAMETROS } { BLOQUE_RETURN ; } { PARAMETROS } { BLOQUE } { PARAMETROS } { RETURN ; } { PARAMETROS } { }
		CONTENIDO_BOOLEAN	≥=	41	PARAMETROS -> PARAMETROS , PARAMETRO PARAMETRO λ

		CONTENIDO_OBJECT	<u>return</u>	42	PARAMETRO -> TIPO <u>nombre</u>
		CONTENIDO_OBJETO	<u>break</u>	43	BLOQUE -> BLOQUE SENTENCIAS SENTENCIAS λ
		OPERACION_UNARIA	<u>break</u>		
		INCREMENTO	<u>break</u>		
		AUTOINCREMENTO		44	TIPO -> <u>int</u> <u>boolean</u> <u>String</u> <u>char</u> <u>double</u> <u>Object</u> <u>Objeto</u>
		OPCIONES_AUTOINCREMENTO			
		OPCIONES_AUTOINCREMENTAS			
		CICLO			
		CICLO_FOR		45	BLOQUE -> BLOQUE SENTENCIAS SENTENCIAS
		ASIGNACION_FOR			
		ASIGNACION_COMPLETA_FOR			
		CONDICIONES_FOR		46	SENTENCIAS -> DEFINICION_VAR ASIGNACION CICLO INVOCACION SENTENCIA_CONTROL OPERACION_UNARIA;
		INCREMENTO_FOR			
		INVOCACION			
		CUERPO_INVOCACION_VAR			
		CUERPO_INVOCACION_METODO			
		ARGUMENTACIÓN		47	ASIGNACION -> CUERPO_INVOCACION_VAR = TIPOS_ASIGNACION ; this . CUERPO_INVOCACION_VAR = TIPOS_ASIGNACION ;
		ARGUMENTOS			
		ARGUMENTO		48	TIPOS_ASIGNACION -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_BOOLEAN CONTENIDO_OBJECT CONTENIDO_OBJETO this . CUERPO_INVOCACION_VAR CUERPO_INVOCACION_METODO this . CUERPO_INVOCACION_METODO
		SENTENCIA_CONTROL		49	CONTENIDO_VARIABLE -> (CONTENIDO_VARIABLE) nombre
		SENTENCIA_IF		50	CONTENIDO_NUMERO -> CONTENIDO_NUMERO + CONTENIDO_NUMERO CONTENIDO_NUMERO - CONTENIDO_NUMERO CONTENIDO_NUMERO * CONTENIDO_NUMERO CONTENIDO_NUMERO / CONTENIDO_NUMERO - CONTENIDO_NUMERO (CONTENIDO_NUMERO) CONTENIDO_NUMERO NOMBRE
		CONTENIDO_BOOLEAN		51	CONTENIDO_STRING -> CONTENIDO_BOOLEAN + CUALQUIER_CONTENIDO PERMITIDOS
		CONDICION		52	PERMITIDOS -> CONTENIDO_NUMERO + PERMITIDOS CONTENIDO_CHAR + PERMITIDOS CUALQUIER_CONTENIDO
		OPCION_CONDICION			
		CONTENIDO_CONDICION		53	CUALQUIER_CONTENIDO -> + CONTENIDO_NUMERO + CONTENIDO_CHAR + CONTENIDO_BOOLEAN + STRINGS
		CONTENIDO_BOOL			
		NEGACIONES			
		SENTENCIA_SWITCH		54	STRINGS -> (STRINGS) CADENA
		CUERPO_SWITCH			
		CASOS_NORMALES		55	CONTENIDO_CHAR -> (CONTENIDO_CHAR) CARACTER
		CASOS_NORMALES		56	CONTENIDO_OBJECT -> (CONTENIDO_OBJECT) new OBJECT ()
		CASO_DEFAULT		57	CONTENIDO_OBJETO -> (CONTENIDO_OBJETO) new OBJETO ()
		CONTENIDO_SWITCH			OPERACION_UNARIA -> INCREMENTO AUTOINCREMENTO
		CUERPO_ESTRUCTURAS			

		BLOQUE_GENERAL		58	
		RETORNO			
		RETORNO_COMPUUESTO			INCREMENTO -> nombre ++
		RETORNO_VARIABLE			nombre --
		RETORNO_NUMERO		59	++ nombre
		RETORNO_STRING		60	-- nombre
					AUTOINCREMENTO -> nombre OPCIONES_AUTOINCREMENTO
		RETORNO_CHAR		61	OPCIONES_AUTOINCREMENTO -> OPCIONES_AUTOINCREMAS - = CONTENIDO_NUMERO * = CONTENIDO_NUMERO / = CONTENIDO_NUMERO
		RETORNO_BOOLEAN		62	OPCIONES_AUTOINCREMAS -> + = CONTENIDO_NUMERO + = CONTENIDO_STRING
		RETORNO_OBJECT		63	CICLO -> CICLO_FOR while (CONTENIDO_BOOLEAN) CUERPO_ESTRUCTURAS while (CONTENIDO_VARIABLE) CUERPO_ESTRUCTURAS do CUERPO_ESTRUCTURAS while (CONTENIDO_BOOLEAN); do CUERPO_ESTRUCTURAS while (CONTENIDO_VARIABLE);
		RETORNO_OBJETO		64	CICLO_FOR -> for (ASIGNACION_FOR CONDICIONES_FOR INCREMENTO_FOR) CUERPO_ESTRUCTURAS
		PARADA		65	ASIGNACION_FOR -> ASIGNACION_COMPLETA_FOR ASIGNACION ;
				66	ASIGNACION_COMPLETA_FOR -> int nombre = CONTENIDO_NUMERO ; double nombre = CONTENIDO_NUMERO ; string nombre = CONTENIDO_STRING ; string nombre = CONTENIDO_VARIABLE ; char nombre = CONTENIDO_CHAR ; char nombre = CONTENIDO_VARIABLE ; boolean nombre = CONTENIDO_BOOLEAN ; boolean nombre = CONTENIDO_VARIABLE ; Object nombre = CONTENIDO_OBJECT ; Object nombre = CONTENIDO_VARIABLE ; Objeto nombre = CONTENIDO_OBJETO ; Objeto nombre = CONTENIDO_VARIABLE ;
				67	CONDICIONES_FOR -> CONTENIDO_BOOLEAN ; CONTENIDO_VARIABLE ; ;
				68	INCREMENTO_FOR -> OPERACION_UNARIA λ
				69	INVOCACION -> CUERPO_INVOCACION_METODO ; this . CUERPO_INVOCACION_METODO ;
				70	CUERPO_INVOCACION_VAR -> nombre . CUERPO_INVOOCACION_VAR nombre (ARGUMENTACION) . CUERPO_INVOCACION_VAR nombre
				71	CUERPO_INVOCACION_METODO -> nombre . CUERPO_INVOOCACION_METODO nombre (ARGUMENCION) . CUERPO_INVOCACION_MET nombre
				72	ARGUMENTACION -> ARGUMENTOS λ
				73	ARGUMENTOS -> ARGUMENTOS , ARGUMENTO ARGUMENTO
				74	ARGUMENTO -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_BOOLEAN CONTENIDO_OBJECT CONTENIDO_OBJETO

				75	SENTENCIA_CONTROL -> SENTENCIA_IF SENTENCIA_IF else CUERPO_ESTRUCTURAS SENTENCIA_SWITCH
				76	SENTENCIA_IF -> if (CONTENIDO_BOOLEAN) CUERPO_ESTRUCTURAS if (CONTENIDO_VARIABLE) CUERPO_ESTRUCTURAS
				77	CONTENIDO_BOOLEAN -> NEGACIONES (CONTENIDO_BOOLEAN) (CONTENIDO_BOOLEAN) CONTENIDO_BOOLEAN operador_logico CONTENIDO_BOOLEAN CONDICION
				78	CONDICION -> CONTENIDO_CONDICION operador_relacional OPCION_CONDICION CONTENIDO_BOOL operador_relacional OPCION_CONDICION CONTENIDO_BOOL
				79	OPCION_CONDICION -> CONTENIDO_BOOL CONTENIDO_CONDICION
				80	CONTENIDO_CONDICION -> CONTENIDO_NUMERO CONTENIDO_STRING CONTENIDO_CHAR CONTENIDO_OBJECT CONTENIDO_OBJETO
				81	CONTENIDO_BOOL -> (CONTENIDO_BOOL) NEGACIONES CONTENIDO_BOOL NEGACIONES CONTENIDO_VARIABLE booleano
				82	NEGACIONES -> NEGACIONES no no
				83	SENTENCIA_SWITCH -> switch (CONTENIDO_SWITCH) { CUERPO_SWITCH }
				84	CUERPO_SWITCH -> CASOS_NORMALES CASO_DEFAULT CASO_DEFAULT
				85	CASOS_NORMALES -> CASOS_NORMALES CASO CASO
				86	CASO -> case CONTENIDO_SWITCH : BLOQUE_GENERAL case CONTENIDO_SWITCH :
				87	CASO_DEFAULT -> default : BLOQUE_GENERAL default :
				88	CONTENIDO_SWITCH -> CONTENIDO_STRING CONTENIDO_NUMERO CONTENIDO_CHAR this . CUERPO_INVOCACION_VAR nombre . CUERPO_INVOCACION_VAR CUERPO_INVOCACION_METODO
				89	CUERPO_ESTRUCTURAS -> { BLOQUE_GENERAL } { }
				90	BLOQUE_GENERAL -> BLOQUE_RETORNO BLOQUE RETORNO
				91	RETORNO -> RETORNO_COMPUUESTO parada ;
				92	RETORNO_COMPUUESTO -> RETORNO_NUMERO RETORNO_STRING RETORNO_CHAR RETORNO_BOOLEAN RETORNO_OBJECT RETORNO_OBJETO return INVOCACION return nombre . CUERPO_INVOCACION_VAR ; return this . CUERPO_INVOCACION_VAR ;
				93	RETORNO_VARIABLE -> return CONTENIDO_VARIABLE ;
				94	RETORNO_NUMERO -> return CONTENIDO_NUMERO ;
				95	RETORNO_STRING -> return CONTENIDO_STRING ;
				96	RETORNO_CHAR -> return CONTENIDO_CHAR ;

				97	RETORNO_BOOLEAN -> return CONTENIDO_BOOLEAN ;	INICIO
				98	RETORNO_OBJECT -> return CONTENIDO_OBJECT ;	
				99	RETORNO_OBJETO -> return CONTENIDO_OBJETO ;	
				100	PARADA -> return break	