		ANÁLISI	S LÉXICO					
>> Análisis del lenguaje <<								
Tipo	Concepto general	Representación	Contenido	Trabajo que desempeña	Objetivo del trabajo			
Alfabeto	Conjunto de los caracteres que permitien formar las cadenas y palabras aceptadas en el lenguaje actual	Σ	(Importor, Incerteza, Principal, Int, Double, String, Boolean,Char,Void,Si,Sino,Para,Mientras,Detener, Continuar,Mostrar,DibujarAST, DibujarEXP, DibujarTS, true, false, ++, -, +, -, *, -, *, -, *, -, -, -, -, -, -, -, -, -, -, -, -, -,	Permitir que se formen las palbras aceptadas que el lenguje requiere para funcionar	Servir como "tienda" para escoger elementos aceptados para generar agrupaciones aceptadas			
Palabra reservada	Elementos que poseen un único lexema con el cual coincidir, utilizadas para funciones especiales y específicas del lenguaje	Importar   Incerteza   Principal   Int   Double   String   Boolean   Char   Void   Si   Sino   Para   Mientras   Detener   Continuar   Mostrar   DibujarAST   D	Importor Incerteza Principal Int Double String Boolean Char Void Si Sino Para Mientras Detener Continuar Mostrar DibujarAST DibujarEXP DibujarEXP ibujarEXP ibujarEx  true false ++	Permitir que el lenguaje posea polabras con una función específica	Permitir que el lenguaje pueda tener un listado de elementos con funciones específicas reservadas para los elementos de dicho listado			
Expresión regular	Reglas que permiten la generación de codenas aceptadas en el lenguaje	cadena identidicador decimal entero salto de línea Sangría	(\"[\"]\"\") ("_" \" \" \" \"\"\"\"\"\"\"\"\"\"\"\"\"\	Permitir que puedan ser generados elementos de los conjuntos de palabras aceptadas, definidas por medio de la regla en cuestión	Permitir que se generen elementos de los grupos de palobras aceptadas			
		espacio en blanco comentario de línea comentario multilínea Todo lo demás (error)	[.] "[n/^]!!" ('\/)K[*]K///)					

## **ANÁLISIS SINTÁCTICO** >> Gramática formal << $G = \{NT, T, RP, So\}$ Reglas de Producción [RP] Estado Inicial [So] No Terminales [NT] Terminales [T] No. Regla EOF 1 inicio inicio : clase EOF SANGRIA clase clase : nl class\_elements IMPORT class\_elements 2 | SANGRIA nl class\_elements ID header | class\_elements imports CRL class\_elements: header content 3 INCERTEZA Content import incerteza INT header: imports incerteza DOUBLE content 4 |imports STRING incerteza sentences class\_content BOOLEAN imports: imports import 5 class\_content\_elements CHAR import declaracion\_var\_global VOID 6 import: IMPORT ID '.' CRL nl declaracion\_var MAIN 7 incerteza: INCERTEZA expression nl UMINUS creacion\_vars content : content sentences 8 INTEGER var\_list sentences DECIMAL content\_type sentences : class\_content 9 CADENA | function\_content declaracion\_funcion CHARACTER 10 class\_content: class\_content\_elements nl params TRUE class\_content\_elements : declaracion\_var\_global params\_list 11 FALSE | declaracion\_funcion param MOSTRAR 12 declaracion\_var\_global : declaracion\_var function\_content CADENA function\_sentence declaracion\_var: content\_type creacion\_vars asignation\_value 13 DRAW\_AST | content\_type creacion\_vars only\_sentence asignacion\_var DRAW\_EXP 14 creacion\_vars : var\_list DRAW\_TS var\_list: var\_list',' ID asignation\_value 15 RETORNO | ID expression expr CONTINUAR content\_type: INT DOUBLE DETENER expr2 expr3 PARA 16 STRING expr4 INCREMENTO | BOOLEAN MIENTRAS LCHAR expr5 ехргб SI declaracion\_funcion : content\_type ID '(' params ')' ':' 17 |VOID ID SINO expr7 expr8 NEW\_LINE VOID MAIN expr9 18 params : params\_list expr10 params\_list : params\_list ',' param 19 booleano param 20 contenido\_var param:content\_type ID 21 function\_content : SANGRIA function\_sentence nl invocacion contenido\_asignacion function\_sentence : only\_sentence 22 argumentos |loop\_sentence dibujar |control\_sentence breakpoints only\_sentence : declaracion\_var loop\_sentence asignacion\_var invocacion for\_var 23

control_sentence	23	mostrar
nl		ldibujar
nı .		
	2.	breakpoints
	24	asignacion_var: ID asignation_value
	25	asignation_value: '=' expression
	26	expression: expr
		expr:expr'+'expr2
	27	expr'-'expr2
		l expr2
		expr2:expr2'*' expr3
		expr2'/'expr3
	28	expr2 '%' expr3
		lexpr3
		expr3 : expr4 '^' expr3
	29	expr4
	30	expr4:'-'expr5 %prec UMINUS
		[expr5
		expr5 : expr5 '==' expr6
		expr5 '!=' expr6
		expr5 '<' expr6
	31	expr5 '>' expr6
	31	l expr5 '<=' expr6
		expr5 '>=' expr6
		expr5 '~' expr6
		exprб
		expr6 : expr6 OR expr7
	32	lexpr7
		expr7 : expr7 XOR expr8
	33	l expr8
		expr8 : expr8 '&&' expr9
	34	lexpr9
		expr9:"l' expr10
	35	expr10
		expr10 : INTEGER
		DECIMAL
		CADENA
	96	
	36	booleano
		CHARACTER
		contenido_var
		'(' expr ')'
	37	booleano: TRUE
	**	FALSE
	38	contenido_var: ID
	 	linvocacion
	39	invocacion: ID '(' argumentos ')'
	39	ID '(' ')'
	40	mostrar: MOSTRAR '(' CADENA contenido_asignacion ')'
	41	contenido_asignacion:','argumentos
		argumentos: argumentos', expression
	42	expression
		dibujar: DRAW_AST '(' ID ')'
	43	DRAW_EXP '(' expression ')'
		DRAW_TS (' ')'
		DIGW_10 ( )

inicio

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

DIRE  DIRE  BL  SEN'  D  DEF  CREACION  VAR  CREACIO  VAR  CREACIO  VA  VA  CREACIO  VA  VA  CREACIO  VA  CREACIO  VA  VA  VA  CREACIO  VA  VA  VA  CREACIO  VA  VA  VA  VA  VA  VA  VA  VA  VA  V	Terminales [NT]  INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE  BLOQUE_CLASE  VIENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	Terminales [T]  import class  this	No.  1 2 3	>> Gramática formal << G = {NT, T, RP, So}  Reglas de Producción [RP]  Regla  INICIO -> IMPORTS CLASE   CLASE   CLASE  IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION _*   DIRECCION - nombre	Estado Inicial [So]
DIRE  DIRE  SEN'  DEF  CREACION  VAR	INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NITENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	import class this  this	1 2 3	G = {NT, T, RP, So}  Reglas de Producción [RP]  Regla  INICIO -> IMPORTS CLASE   CLASE   CLASE   IMPORTS -> impart IIMPORT   IIMPORT   IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*   DIRECCION	Estado Inicial [So]
DIRE  DIRE  SEN'  DEF  CREACION  VAR	INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NITENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	import class this  this	1 2 3	Reglas de Producción [RP]  Regla  INICIO -> IMPORTS CLASE   CLASE   CLASE  IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*_   DIRECCION	Estado Inicial [So]
DIRE  DIRE  SEN'  DEF  CREACION  VAR	INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NITENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	import class this  this	1 2 3	Reglas de Producción [RP]  Regla  INICIO -> IMPORTS CLASE   CLASE   CLASE  IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*_   DIRECCION	Estado Inicial [So]
DIRE  DIRE  SEN'  DEF  CREACION  VAR  CREACION	INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NITENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	import class this  this	1 2 3	Regla  INICIO -> IMPORTS CLASE   CLASE   CLASE  IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*   DIRECCION	Estado Inicial [So]
DIRE  DIRE  SEN'  DEF  CREACION  VAR	INICIO IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NITENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	import class this  this	1 2 3	INICIO -> IMPORTS CLASE   CLASE   MPORTS -> import IIMPORT   IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*   DIRECCION	Estado Inicial [So]
BL SEN' DDF_ CREACION VAR CREACIO VAR CREACIO VAR CREACIO VAR CREACIO VAR CREACIO VAR CREACION VAR	IMPORTS  IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	class this  this  this	2 3	CLASE  IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*_   DIRECCION	
BL SEN' DDF_ CREACION VAR	IIMPORT  ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	this	2 3	IMPORTS -> import IIMPORT   IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*   DIRECCION	_
BL SEN' DDEF_ CREACION VAR	ECCION_IMPORT  DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	· :	3	IIMPORT  IIMPORT -> import DIRECCION_IMPORT;  IMPORT -> DIRECCION .*_   DIRECCION	
BL SEN' DDEF_ CREACION VAR	DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	· :	4	IMPORT -> DIRECCION .* _   DIRECCION	
BL SEN' DDF_ CREACION VAR CREACIO  VAR CREACIO  VAR CREACIO  VAR CREACIO  VAR CREACIO VAR CREACION VAR CREACION VAR CREACION VAR	DIRECCION  CLASE  VISIBILIDAD  DEF_CLASE BLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	<u>.</u> <u>.</u> 1		IMPORT -> DIRECCION .* _   DIRECCION	
BL SEN' DDF_ CREACION VAR CREACIO  VAR CREACIO  VAR CREACIO  VAR CREACIO VAR CREACIO VAR CREACION VAR CREACION VAR	CLASE VISIBILIDAD DEF_CLASE BLOQUE_CLASE NTENCIAS_CLASE DEF_VARIABLE F_TIPO_VARIABLE	<u>.</u> <u>.</u> 1		DIRECCION	
BL SEN' DD DEF_ CREACION VAR CREACIOI  VAF CREACIOI  VAF CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR VAR	VISIBILIDAD  DEF_CLASE BLOOUE_CLASE NTENCIAS_CLASE DEF_VARIABLE F_TIPO_VARIABLE	<u>.</u> <u>.</u> 1		DIRECCION	
BL SEN' DD DEF_ CREACION VAR CREACIOI  VAF CREACIOI  VAF CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR VAR	DEF_CLASE BLOQUE_CLASE NTENCIAS_CLASE DEF_VARIABLE F_TIPO_VARIABLE	1			
BL SEN' DD DEF_ CREACION VAR CREACIOI  VAF CREACIOI  VAF CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR CREACIOI  VAR VAR	SLOQUE_CLASE  NTENCIAS_CLASE  DEF_VARIABLE  F_TIPO_VARIABLE	1		DIRECCION -> DIRECCION : nombre	$\dashv$
SEN' D DEF_ CREACION VAR CREACIOI  VAF CREACIO  VAF CREACIO  VAR CREACIO  VAR CREACION VAR CREACION VAR VAR VAR	NTENCIAS_CLASE DEF_VARIABLE F_TIPO_VARIABLE	1			
DEF_ CREACION VAR CREACION  VAF CREACIO  VAR  CREACIO  VAR  CREACION  VAR  CREACION  VAR  VAR	DEF_VARIABLE F_TIPO_VARIABLE		5	nombre , nombre	_
DEF_ CREACION VAR CREACION  VAF CREACION  VAR  CREACION  VAR  CREACION  VAR  CREACION  VAR  VAR  VAR	DEF_VARIABLE F_TIPO_VARIABLE			CLASE -> VISIBILIDAD DEF_CLASE   DEF_CLASE	
DEF. CREACION VAR CREACIOI  VAF CREACIO  VAF CREACIO  VA CREACIOI  VA CREACIOI  VAR VAR VAR	F_TIPO_VARIABLE	1	6	1 201_001 001	
CREACION VAR CREACIOI  VAA CREACIO  VA CREACIOI  VA CREACIOI  VA CREACIOI  VARI CREACIOI  VARI		public public	7		
VAR CREACIOI  VAF CREACIO  VA CREACION  VAR  CREACION  VARI  VAR	N_VARIABLE_NUMERO		8	VISIBILIDAD -> public	
CREACIOI  VAF  CREACIC  VA  CREACION  VARI  CREACION  VARI  VARI  VARI	RIABLE_NUMERO	protected private	0	protected .	
VAF CREACIO VA CREACION VARI CREACION VARI	ON_VARIABLE_STRING	final		private   final	
CREACIO VAR CREACION VARI CREACION VAR	JIN_VARIABLE_STRING	indi		DEF_CLASE -> class objeto { BLOQUE_CLASE }	-
VA CREACION VARI CREACIOI	RIABLE_STRING	nombre	9	class objeto {}	
CREACION VARI CREACIOI VAR	ION_VARIABLE_CHAR				
VARI CREACIOI VAR	'ARIABLE_CHAR	new	10	BLOQUE_CLASE -> BLOQUE_CLASE SENTENCIAS_CLASE	
CREACIOI VAR	N_VARIABLE_BOOLEAN	int		SENTENCIAS_CLASE	
VAR	RIABLE_BOOLEAN	boolean	11		
	ON_VARIABLE_OBJECT	String	12	SENTENCIAS_CLASE -> DEF_VARIABLE	
	RIABLE_OBJECT	<u>char</u>	13	DEF_METODO	
				DEF_VARIABLE -> VISIBILIDAD DEF_TIPO_VARIABLE ;	
CREACIO	ON_VARIABLE_OBJETO	double	14	DEF_TIPO_VARIABLE ;	
VAF	.RIABLE_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_DBJECT   object CREACION_VARIABLE_DBJECT	
		Objeto [si es que harás la revisión		CREACION_VARIABLE_NUMERO -> CREACION_VARIABLE_NUMERO , VARIABLE_NUMERO	
	DEF_METODO	de la lista	16	VARIABLE_NUMERO	
				VARIABLE_NUMERO -> nombre = CONTENIDO_NUMERO	
CLIFR	RPO_DEF_METODO	void	17	nombre = CONTENIDO_CHAR   nombre	
OCEN		1515		CREACION_VARIABLE_STRING -> CREACION_VARIABLE_STRING , VARIABLE_STRING	_
DEF_MET	TODO_CONSTRUCTOR	cadena	18	VARIABLE_STRING	
				VARIABLE_STRING -> nombre = CONTENIDO_VARIABLE   nombre = CONTENIDO_STRING	
CUERPO_M	METODO_CONSTRUCTOR	valor	19	nombre	
DEF	METODO_CONSTRUCTOR	<u>numero</u>	20	CREACION_VARIABLE_CHAR -> CREACION_VARIABLE_CHAR , VARIABLE_CHAR   VARIABLE_CHAR	
CUERPO	F_METODO_TIPO	±	21	VARIABLE_CHAR -> nombre = CONTENIDO_VARIABLE   nombre = CONTENIDO_CHAR   nombre	

		1	
CUERPO_METODO_STRING	=	22	CREACION_VARIABLE_BOOLEAN -> CREACION_VARIABLE_BOOLEAN , VARIABLE_BOOLEAN   VARIABLE_BOOLEAN
			VARIABLE_BOOLEAN -> nombre = CONTENIDO_VARIABLE
	,		nombre = CONTENIDO_BOOLEAN
CUERPO_METODO_CHAR	<u></u>	23	nombre
CUERPO_METODO_BOOLEAN	Ξ.	24	CREACION_VARIABLE_OBJECT -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJECT   VARIABLE_OBJECT
			VARIABLE_OBJECT -> nombre = CONTENIDO_VARIABLE
			nombre = CONTENIDO_OBJECT
CUERPO_METODO_OBJECT	(	25	nombre = CONTENIDO_OBJETO   nombre
COEM CIMETODO COBSECT	7	23	CREACION_VARIABLE_OBJECT -> CREACION_VARIABLE_OBJECT , VARIABLE_OBJECT
CUERPO_METODO_OBJETO	1	26	VARIABLE_OBJECT
			VARIABLE_OBJETO -> nombre = CONTENIDO_VARIABLE
DEF_METODO_VOID	for	27	nombre = CONTENIDO_OBJETO   nombre
DEI_INIETODO_VOID	for	21	DEF_METODO -> VISIBILIDAD CUERPO_DEF_METODO
CUERPO_METODO_VOID	<u>while</u>	28	CUERPO_DEF_METODO
			CUERPO_DEF_METODO -> DEF_METODO_CONSTRUCTOR
PARAMETROS	do.	29	DEF_METODO_TIPO   DEF_METODO_VOID
PARAMETRO	<u>do.</u> if	30	DEF_METODO_CONSTRUCTOR -> objeto ( PARAMETROS ) CUERPO_METODO_CONSTRUCTOR
TIPO		30	
	else	24	CUERPO_METODO_CONSTRUCTOR -> { BLOQUE }
BLOQUE	<u>switch</u>	31	{}
			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
SENTENCIAS	0000	32	object nombre ( PARAMETROS ) CUERPO_METODO_NUMERICO bjeto nombre ( PARAMETROS ) CUERPO_METODO_NUMERICO
ASIGNACIÓN	case	32	CUERPO_METODO_NUMERO -> { BLOQUE RETORNO_NUMERO }
	default		{ RETORNO_NUMERO }
TIPOS_ASIGNACIÓN	<u>l</u>	22	{ BLOQUE RETONO_CHAR }
CONTENIDO_VARIABLE	<u>&amp;</u>	33	{ RETONO_CHAR}
			CUERPO_METODO_STRING -> { BLOQUE RETORNO_VARIABLE }  { RETORNO_VARIABLE }
			{ BLOQUE RETONO_STRING}
CONTENIDO_NUMERO	I	34	{ RETONO_STRING}
			CUERPO_METODO_CHAR -> { BLOQUE RETORNO_VARIABLE }
			{ RETORNO_VARIABLE }
CONTENIDO_STRING	<	35	{ BLOQUE RETONO_CHAR }  { RETONO_CHAR }
GOTTE TADO_STITUTO		33	CUERPO_METODO_BOOLEAN -> { BLOQUE RETORNO_VARIABLE }
			{ RETORNO_VARIABLE }
			{ BLOQUE RETONO_BOOLEAN }
PERMITIDOS	<u>&gt;</u>	36	{ RETONO_BOOLEAN }
			CUERPO_METODO_OBJECT -> { BLOQUE RETORNO_VARIABLE }
			{ RETORNO_VARIABLE }   { BLOQUE RETONO_OBJECT }
CUALQUIER_CONTENIDO	<u>!=</u>	37	{ RETONO_OBJECT }
		<u> </u>	CUERPO_METODO_OBJETO -> { BLOQUE RETORNO_VARIABLE }
			{ RETORNO_VARIABLE }
			{ BLOQUE RETONO_OBJETO }
OPCIONES_CUALUIER_CONTENIDO	8.8	38	{ RETONO_OBJETO }
STRINGS		39	DEF_METODO_VOID -> void nombre CUERPO_METODO_VOID
			CUERPO_METODO_VOID -> ( PARAMETROS) { BLOQUE RETURN; }
			( PARAMETROS) { BLOQUE }   ( PARAMETROS) { RETURN ; }
CONTENIDO_CHAR	< <u>=</u>	40	( PARAMETROS) { RETURN; }
	<del>_</del>	1-2	PARAMETROS -> PARAMETROS , PARAMETRO
			PARAMETRO
CONTENIDO_BOOLEAN	<u>&gt;=</u>	41	λ

CRITERIOLOGRAFIC   blook   CRITERIOLOGRAFIC				
DORADON_UNMANA   brask   40   MIDULES SMITHULES			42	PARAMETRO -> TIPO nombre
CONTENDED   CONTENDED CO		<u>break</u>		BLOQUE -> BLOQUE SENTENCIAS
MATORISCHANTON				
OFCIONES_ALTOINCREMENTO		<u>break</u>	43	λ
OFCIONES_AUTONICEMENTO   SETUD   Close   Clo	AUTOINCREMENTO			TIPO -> <u>int</u>
Lobe				
DOCIONES_AUTOMOREMAS   1 Soulide	OPCIONES_AUTOINCREMENTO			
Disable				
CICLO   44     Cibble   CICLO FOR     ASSIGNACION FOR     ASSIGNACION ELP-OP     ASSIGNACION ELP-OP     CONDICIONS FOR     NECESSATION     NECESSATION     NECESSATION     NECESSATION     CURPO, INVOCACION     ASSIGNACION     ASS	OPCIONES_AUTOINCREMAS			
ASIGNACION COMPITATION	CICLO		44	
ASSUNCTION COMPLETAL FOR	CICLO_FOR			
ASSEMUCION, CONDITION, CONTENDO, CONTENDO, CONTENDO, CONTENDO, SOLUENO   CONTENDO, CONTENDO, CONTENDO   CONTENDO, CONTENDO, CONTENDO, CONTENDO   CONTENDO, CONTENDO, CONTENDO, CONTENDO, CONTENDO   CONTENDO, CONTENDO, CONTENDO, CONTENDO   CONTENDO, CONTENDO, CONTENDO, CONTENDO   CONTENDO, CONTENDO   CONTENDO, CONTENDO   CONTENDO, CONTENDO   CONTEND	ASIGNACION_FOR			BLOQUE -> BLOQUE SENTENCIAS
NICEBERTO FOR   NICECATION   NICEDATION	ASIGNACION_COMPLETA_FOR		45	
ASIGNACION   CUERPO_INVOCACION_LYRR   CUERPO_INVOCACION_LYRR   TIPOS_ASIGNACION   CONTENDO_LYRR   TIPOS_ASIGNACION   CONTENDO_LYRR   TIPOS_ASIGNACION   CONTENDO_LYRR   CONTENDO_LYRR   CONTENDO_LYRR   CONTENDO_LYRR   CONTENDO_LYRR   CONTENDO_LYRR   CUERPO_INVOCACION_LYRR   CUERPO_LYRR   CUERPO_LYR   CUERPO_LYRR   CUERPO_LYR	CONDICIONES_FOR			
ASIGNACION     ASIGNACION   CUERPO, INVOCACION, METODO   CUERPO, INVOCACION, METODO, METODO   CONTENIDO, ANDREO   CONTENIDO, CARRA + FEMITIDOS   CONTENI	INCREMENTO_FOR			SENTENCIAS -> DEFINICION VAR
CUERPO_NNOCACION_VARE   CUERPO_NNOCACION_MICTODO				
CURROLANDICACION_METODO				
ARGUMENTACIÓN   46			•	
ARGUMENTOS  47  ASIGNACION - CUERRO, NVOCACION, LAR - TIPOS, ASIGNACION:  TIPOS, ASIGNACION - CONTENIDO, STRING  (CONTENIDO, STRING)  (CONTENIDO, STRING)  (CONTENIDO, STRING)  (CONTENIDO, SOLICAN)  (CONTENIDO, SOLICAN)  (CONTENIDO, BOLICAN)  (CONTENIDO, BOLICAN)  (CONTENIDO, BOLICAN)  (CONTENIDO, BOLICAN)  (CONTENIDO, AND CACION, LARGE TO)  (Inia, CUERRO, LANOCACION, LARGE TO)  (Inia, CUERRO, LANGCACION, LARGE TO)  (Inia, CUERRO, LARGE TO, LARGE TO)  (Inia, CUERRO, LARGE TO, LARGE TO)  (Inia, CUERRO, LARGE TO, LARGE			46	
ARGUMENTOS	ARGOMENTACION		70	
TIPOS_ASIGNACION - CONTENDO, JUMERO   CONTENDO STRING   CONTENDO STRING   CONTENDO STRING   CONTENDO STRING   CONTENDO STRING   CONTENDO STRING   CONTENDO SALEAN   CONTENDO	ARGUMENTOS		47	
CONTENIDO_STRING   CONTENIDO_GOLEAN   CONTENIDO_MUMERO   CONTENIDO MUMERO   CONTENIDO M				<u> </u>
CONTENDO_BOCLEAN   CONTENDO_DOBLECT   CONTENDO_DOBLECT   CONTENDO_DOBLECT   CONTENDO_DOBLECT   CONTENDO_DOBLECT   CONTENDO_DOBLECT   CONTENDO_NOCACION_WAR   CUERPO_INVOCACION_METODO				
CONTENIDO_OBJECT   CONTENIDO_OBJECT   CONTENIDO_OBJECT   His. CUEPRO_INVOCACION_VAR   CUEPRO_INVOCACION_METODO   His. CUEPRO_INVOCACION_METODO   His. CUEPRO_INVOCACION_METODO   His. CUEPRO_INVOCACION_METODO   His. CUEPRO_INVOCACION_METODO   His. CUEPRO_INVOCACION_METODO				
CONTENDO_OBJETO				
This.CUERPO_INVOCACION_VAR   CUERPO_INVOCACION_METODO   This.CUERPO_INVOCACION_METODO   This				
ARGUMENTO				
ARGUMENTO  ARGUMENTO  SENTENCIA, CONTROL  49  CONTENIDO, VARIABLE - (CONTENIDO, NUMERO - CONTENIDO, NUMERO - PERMITIDOS - CONTENIDO, SEO - CONTENIDO, CONTENIDO, CONTENIDO, SEO - CONTE				
SENTENCIA_CONTROL	ARGUMENTO		48	
CONTENIDO_NUMERO -> CONTENIDO_NUMERO   CONTENIDO_NUMERO -CONTENIDO_NUMERO   CONTENIDO_NUMERO -CONTENIDO_NUMERO   CONTENIDO_NUMERO -CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_SOULEAN   CONTENIDO_SOULEAN   CONTENIDO_SOULEAN   CONTENIDO_CONDICION   CONTENIDO_CO				CONTENIDO_VARIABLE -> ( CONTENIDO_VARIABLE )
CONTENIDO_NUMERO - CONTENIDO_NUMERO   CONTENIDO_SOCIEAN + CUALQUIER_CONTENIDO   PERMITIDOS   CONTENIDO_CONDICION   PERMITIDOS   CONTENIDO_NUMERO + PERMITIDOS   CONTENIDO_CONDICION   PERMITIDOS   CONTENIDO_CONDICION   CONTENIDO_CONDICION   S2   CUALQUIER_CONTENIDO   CONTENIDO_CONDICION   CONTENIDO_CONDIC	SENTENCIA_CONTROL		49	nombre
CONTENIDO_NUMERO 'CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   (CONTENIDO_NUMERO )   (CONTENIDO_SOCIEAN + CUALQUIER_CONTENIDO )   (CONTENIDO_SOCIEAN + CUALQUIER_CONTENIDO )   (CONTENIDO_NUMERO + PERMITIDOS )   (CONTENIDO_NUMERO + PERMITIDOS )   (CONTENIDO_NUMERO + PERMITIDOS )   (CONTENIDO_NUMERO + PERMITIDOS )   (CONTENIDO_CONDICION )   (CONTENIDO_CONTE				
CONTENIDO_NUMERO / CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   NOMBRE				
- CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   CONTENIDO_NUMERO   NOMBRE				
SENTENCIA_IF  50  CONTENIDO_NUMERO   CONTENIDO_NUMERO   NOMBRE  CONTENIDO_BOOLEAN				
CONTENIDO_NUMERO   NOMBRE				
SENTENCIA_IF   50				
CONTENIDO_BOOLEAN  CONDICION  OPCION_CONDICION  CONTENIDO_CONDICION  CONTENIDO_CONDICION  CONTENIDO_CONDICION  CONTENIDO_BOOL  NEGACIONES  SENTENCIA_SWITCH  CASOS_NORMALES  CONTENIDO_BOOL  CARACTER  CONTENIDO_CHAR + PERMITIDOS    CONTENIDO_NUMERO + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CULALQUIER_CONTENIDO   CONTENIDO_NUMERO   + CONTENIDO_NUMERO   + CONTENIDO_CHAR   + CONTENIDO_CHAR   + CONTENIDO_BOOLEAN   + STRINGS   STRINGS -> (STRINGS)   CADENA  CONTENIDO_CHAR -> (CONTENIDO_CHAR)   CARACTER   CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)	SENTENCIA_IF		50	
CONDICION OPCION_CONDICION OPCION_CONDICION CONTENIDO_CONDICION 52 CUALQUIER_CONTENIDO_NUMERO + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CUALQUIER_CONTENIDO     CUALQUIER_CONTENIDO     CUALQUIER_CONTENIDO     CONTENIDO_CHAR     + CONTENIDO_CHAR     + CONTENIDO_CHAR     + CONTENIDO_BOOLEAN     + STRINGS     + STRINGS     CABOS_NORMALES  CONTENIDO_CHAR -> (CONTENIDO_CHAR)     CARACTER     CARACTER     CONTENIDO_CHAR -> (CONTENIDO_CHAR)     CARACTER     CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)				
PERMITIDOS -> CONTENIDO_NOMERO + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CONTENIDO_CHAR + PERMITIDOS   CUALQUIER_CONTENIDO - PERMITIDOS   CUALQUIER_CONTENIDO - PERMITIDOS   CUALQUIER_CONTENIDO -> + CONTENIDO_NOMERO   + CONTENIDO_CHAR   + CONTENIDO_CHAR   + CONTENIDO_CHAR   + CONTENIDO_CHAR   + CONTENIDO_BOOLEAN   + STRINGS -> (STRINGS)   CUERPO_SWITCH   54   CAPACA   CAPACA   CAPACA   CAPACA   CONTENIDO_CHAR   + C			51	PERMITIDOS
OPCION_CONDICION  CONTENIDO_CONDICION  CONTENIDO_BOOL  NEGACIONES  SENTENCIA_SWITCH  CASOS_NORMALES  OPCION_CONDICION  52    CONTENIDO_CHAR + PERMITIDOS   CUALQUIER_CONTENIDO     CUALQUIER_CONTENIDO_NUMERO   + CONTENIDO_CHAR   + CONTENIDO_CHAR   + CONTENIDO_BOOLEAN   + STRINGS     STRINGS -> (STRINGS -> (	CONDICION			PERMITIDOS -> CONTENIDO NUMERO + PERMITIDOS
CONTENIDO_BOOL	OPCION_CONDICION			
+ CONTENIDO_CHAR	CONTENIDO_CONDICION		52	CUALQUIER_CONTENIDO
CONTENIDO_OBJECT -> (CONTENIDO_OBJECT )   CONTENIDO_OBJECT -> (CONTENIDO_OBJECT )   CONTENIDO_OBJECT -> (CONTENIDO_OBJECT -> (CONTENI	CONTENIDO_BOOL			
SENTENCIA_SWITCH         53           + STRINGS           CUERPO_SWITCH         54         STRINGS -> ( STRINGS )   CADENA           CASOS_NORMALES         CONTENIDO_CHAR -> ( CONTENIDO_CHAR )   CARACTER           CONTENIDO_OBJECT -> ( CONTENIDO_OBJECT )         CONTENIDO_OBJECT -> ( CONTENIDO_OBJECT )	NEGACIONES			
CUERPO_SWITCH  54  STRINGS -> (STRINGS)   CADENA  CONTENIDO_ CHAR -> (CONTENIDO_CHAR)   CARACTER  CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)	SENTENCIA_SWITCH		53	
CUERPO_SWITCH         54           CADENA           CONTENIDO_ CHAR -> (CONTENIDO_CHAR)         CONTENIDO_ CHAR -> (CONTENIDO_CHAR)           CASOS_NORMALES         55           CARACTER           CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)         CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)				STRINGS -> (STRINGS)
CASOS_NORMALES         55           CARACTER           CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)           CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)	CUERPO_SWITCH		54	
CONTENIDO_OBJECT -> (CONTENIDO_OBJECT)	CASOS NORMALES		55	
	CASOS_INORIVIALES		J.	
	CASOS_NORMALES		56	
CASO_DEFAULT 57 CONTENIDO_OBJETO -> (CONTENIDO_OBJETO )   new OBJETO ()	CASO DEFAULT		57	
			3/	Hew Objeto ()
CONTENIDO_SWITCH  OPERACION_UNARIA -> INCREMENTO				OPERACION_UNARIA -> INCREMENTO
CUERPO_ESTRUCTURAS   AUTOINCREMENTO	CUERPO_ESTRUCTURAS			AUTOINCREMENTO

	1	
BLOQUE_GENERAL	58	
RETORNO		INCREMENTO
RETORNO_COMPUESTO		INCREMENTO -> nombre ++   nombre
RETORNO_VARIABLE		++ nombre
RETORNO_NUMERO	59	nombre
RETORNO_STRING	60	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
NETGINIO_BEGLETIN	02	CICLO -> CICLO_FOR
RETORNO_OBJECT	63	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
		ASIGNACION_FOR -> ASIGNACION_COMPLETA_FOR   ASIGNACION
PARADA	65	i;
	66	ASIGNACION_COMPLETA_FOR -> int nombre = CONTENIDO_NUMERO;   double nombre = CONTENIDO_NUMERO;   string nombre = CONTENIDO_STRING;   string nombre = CONTENIDO_VARIABLE;   char nombre = CONTENIDO_VARIABLE;   char nombre = CONTENIDO_VARIABLE;   boolean nombre = CONTENIDO_BOOLEAN;   boolean nombre = CONTENIDO_VARIABLE;   Object nombre = CONTENIDO_VARIABLE;   Object nombre = CONTENIDO_VARIABLE;   Object nombre = CONTENIDO_OBJECT;   Object nombre = CONTENIDO_OBJECT;
		CONDICIONES_FOR -> CONTENIDO_BOOLEAN ;   CONTENIDO_VARIABLE ;
	67	i;
	68	INCREMENTO_FOR -> OPERACION_UNARIA   λ
		INVOCACION -> CUERPO_INVOCACION_METODO;
	69	this.CUERPO_INVOCACION_METODO;
		CUERPO_INVOCACION_VAR -> nombre . CUERPO_INVOOCACION_VAR
	70	nombre(ARGUMENTACION).CUERPO_INVOCACION_VAR
	/0	nombre  CUERPO_INVOCACION_METODO -> nombre . CUERPO_INVOOCACION_METODO
		nombre (ARGUMENCION). CUERPO_INVOCACION_MET
	71	nombre
	72	$ \begin{array}{c} \text{ARGUMENTACION} \rightarrow \text{ARGUMENTOS} \\ \mid \lambda \end{array} $
	73	ARGUMENTOS -> ARGUMENTOS , ARGUMENTO   ARGUMENTO
	74	ARGUMENTO -> CONTENIDO_NUMERO   CONTENIDO_STRING   CONTENIDO_CHAR   CONTENIDO_BOOLEAN   CONTENIDO_BOBJECT   CONTENIDO_OBJECT

	75	SENTENCIA_CONTROL -> SENTENCIA_IF   SENTENCIA_IF else CUERPO_ESTRUCTURAS   SENTENCIA_SWITCH
	75	SENTENCIA_SWITCH   SENTENCIA_IF -> if (CONTENIDO_BOOLEAN) CUERPO_ESTRUCTURAS
	76	if (CONTENIDO_BOOLEAN PODEN O_ESTNOCTORAS
		CONTENIDO_BOOLEAN -> NEGACIONES (CONTENIDO_BOOLEAN )
		(CONTENIDO_BOOLEAN)
	77	CONTENIDO_BOOLEAN operador_logico CONTENIDO_BOOLEAN   CONDICION
		CONDICION -> CONTENIDO_CONDICION operador_relacional OPCION_CONDICION
	78	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
		CONTENIDO_BOOL -> ( CONTENIDO_BOOL )   NEGACIONES CONTENIDO_BOOL   NEGACIONES CONTENIDO_VARIABLE
	81	booleano   NEGACIONES -> NEGACIONES no
	82	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
	00	CUERPO_ESTRUCTURAS -> { BLOQUE_GENERAL }
	89	1.0
	00	BLOQUE_GENERAL -> BLOQUE_RETORNO   BLOQUE   BLOQUE
	90	RETORNO   RETORNO_COMPUESTO
	91	parada ;   RETORNO_COMPUESTO -> RETORNO_NUMERO
		RETORNO_STRING   RETORNO_CHAR   RETORNO_BOOLEAN   RETORNO_OBJECT   RETORNO_OBJETO   return_INVOCACION
	92	return nombre . CUERPO_INVOCACION_VAR; return this . CUEPRO_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 ;	
	98	RETORNO_OBJECT -> return CONTENIDO_OBJECT ;	
	99	RETORNO_OBJETO -> return CONTENIDO_OBJETO ;	
		PARADA -> return	
	100	break	INICIO