

Gramatica	Reglas Semanticas
INICIO ::= INFO:info FUENTE:fuelle EXPRESIONES:ex separador SIMBOLOS:sim separador GRAMATICA:gram	ConstruirLenguaje(info,fuelle,ex,sim,gram)
INFO ::= INFO_NO_OBL nombre dosPuntos CADENA_ESPACIOS:cadena puntoYComa INFO_NO_OBL	info.nombre=cadena
INFO_NO_OBL ::= version dosPuntos NumeroVersion:cadena puntoYComa INFO_NO_OBL	info.version=cadena
autor dosPuntos CADENA_ESPACIOS:cadena puntoYComa INFO_N	info.autor=cadena
lanzamiento dosPuntos Entero:cadena puntoYComa INFO_NO_OBL	info.lanzamiento=cadena
extension dosPuntos Cadena:cadena puntoYComa INFO_NO_OBL	info.extension=cadena
;	
CADENA_ESPACIOS ::= Cadena:cad CADENA_ESPACIOS:cads	cadena.val=cad+cads
Cadena:cad	cadena.val=cad
FUENTE ::= Fuente;	
EXPRESIONES ::= Cadena:cad igual EXP puntoYComa EXPRESIONES:ex	Construir expresion(cad, ex)
ampersand igual EXP puntoYComa EXPRESIONES:ex	Construir expresion(&, ex)
	saltos.val=" "
saltoDeLinea	saltos.val="\t"
tab	
EXPRESION_OP ::= EXPRESION_OP:estado2 EXPRESION:estado1	estado.valor.concatenar(estado1,estado2)
EXPRESION_OP:estado2 op EXPRESION:estado1	estado.valor.union(estado1,estado2)
EXPRESION ::= parentesisA EXPRESION_OP:estado1 parentesisC EX:ex	estado.valor.crearExpresion(ex,estado1)
numeros EX:ex	estado.valor.crearExpresion(ex,numero)
letras EX:ex	estado.valor.crearExpresion(ex,letras)
Cadena:cad	estado.valor.crearExpresion(cad)
corcheteA STRS:strs corcheteC	for(each:strs){ estado.valor.union(crear Estado(strs)) }
STRS ::= STR2:st STRS:sts	strs.val= st + sts
STR2:st	strs.val= st
STR2 := Cadena:cad	str2.val=cad
SALTOS:sal	str2.val=sal
asterisco	str2.val=*
qMark	str2.val=?
suma	str2.val=+
EX ::= asterisco	ex.val=1
qMark	ex.val=2
suma	ex.val=3
;	
SIMBOLOS ::= sTerminal CADENAS:cads puntoYComa SIMBOLOS	simbolos.val.crearSimbolos(terminal, cads)
no sTerminal CADENAS:cads puntoYComa SIMBOLOS	simbolos.val.crearSimbolos(Notterminal, cads)

;	
CADENAS ::= Cadena:nom coma CADENAS:cads	cadenas.val=nom+cads
GRAMATICA ::= Cadena:noterminal dosPuntos dosPuntos PRODS:prod CODIGO:code GRAMATICA: gram	gramatica.val+=prod+gram
Cadena:noterminal dosPuntos dosPuntos PRODS:prod CODIGO:code	gramatica.val=prod
PRODS ::= Cadena:nombre dosPuntos Cadena:id PRODS:prod	prods.val.crearProduccion(nombre,id)+prod
Cadena:nombre PRODS:prod	prods.val.crearProduccion(nombre)+prod