Regla	Producción	Acción semántica
Programa	Lista_sentencias PUNTO	
Lista_sentencia	Lista_sentencias Sentencia	
Lista_sentencia		
Sentencia	Sentencia PTOCOMA  VAL ID = Exp Rel	
Sentencia		
	Exp_Rel FUN MatchingFunc	
MatchingFunc	ID ID CurryingList	
iviatoringi unc	ID TuplePattern IGUAL	
	Exp Rel	
CurryingList	ID CurryingList	
	IGUAL Exp_Rel	
	PARABRE TuplePattern2	
TuplePattern	PARCIERRA	
TuplePattern2	TuplePattern2 COMA ID	
		Attrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		if (esTipoTextOrNum(t)){
		RESULT = new Attrs("boolean");
		}else{
		throw new Exception("String or num expected.");
	Exp Rel:e1 OPREL	<pre>}else{ throw new Exception("String or num expected.");</pre>
Exp_Rel	Exp_Ref.eT OFREL	It is the expected of the state
LXP_IXCI	EXP_00113.02	Attrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		t = tipoMasEspecifico(t,crearPolitipo("politipo con
		igualdad"));
		if (t!=null && !t.getType().equals("fun")){
		RESULT = new Attrs("boolean");
		}else{
		throw new Exception("Function does not admit
		equality.");
		} 
		}else{
	Exp_Rel:e1 DISTINTO	throw new Exception("Types are not compatibles.");
	Exp_cons:e2	tompatibles. ),
		Attrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		t = tipoMasEspecifico(t,crearPolitipo("politipo con
		igualdad"));
		if (t!=null && !t.getType().equals("fun")){
		RESULT= new Attrs("boolean");
		}
		else{
		throw new Exception("Function does not admit
		equality.");
		}  }else{
		throw new Exception("Types are not
	Exp_Rel:e1 IGUAL	compatibles.");
	Exp_cons:e2	<b> </b>
	Exp_cons	RESULT = e;
	1	1

Regla	Producción	Acción semántica
		Attrs t = tipoMasEspecifico(new Attrs("list",e1),e2); if (t!=null){ RESULT = t;
Exp_cons	Exp_add:e1 CONS Exp_cons:e2	<pre>} else{ throw new Exception("The list has elements of different types."); }</pre>
· -	Exp_add:e	RESULT = e;
		Attrs t = tipoMasEspecifico(e1,e2); if (t!=null){   t = tipoMasEspecifico(t,crearPolitipo("politipo numerico"));   if (t!=null){     RESULT = t; }else{
		throw new Exception("Integer or real expected."); }
Exp_add	Exp_add:e1 OPADD Exp_mul:e2	<pre>}else{ throw new Exception("Integer or real expected."); }</pre>
		Attrs t = tipoMasEspecifico(e1,e2); if (t!=null){ t = tipoMasEspecifico(t,new Attrs ("boolean")); if (t!=null){ RESULT = t; }else{
	Exp_add:e1 ORELSE Exp_mul:e2	throw new Exception("Boolean expected."); } else{ throw new Exception("Boolean expected."); }
		Attrs t = tipoMasEspecifico(e1,e2); if (t!=null){   t = tipoMasEspecifico(t,new Attrs ("string"));   if (t!=null){     RESULT = t;   }else{     throw new Exception("String expected."); }
	Exp_add:e1 CONCAT Exp_mul:e2	}else{ throw new Exception("String expected."); }
	Exp_mul:e	RESULT = e;
		Attrs t = tipoMasEspecifico(e1,e2); if (t!=null){     t = tipoMasEspecifico(t,crearPolitipo("politipo     numerico")); if (t!=null){     RESULT = t; }else{     throw new Exception("Integer or real expected."); }
Exp_mul	Exp_mul:e1 MUL Exp_un: e2	<pre>} }else{ throw new Exception("Integer or real expected."); }</pre>

Regla	Producción	Acción semántica
rtogia	1 Toddoolon	Attrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		t = tipoMasEspecifico(t,new Attrs("real"));
		if (t!=null){ RESULT = t;
		}else{
		throw new Exception("Real expected.");
		}
		}else{
	Exp_mul:e1 DIVREAL Exp_un:e2	throw new Exception("Real expected."); }
		Attrs t = tipoMasEspecifico(e1,e2);
		<pre>if (t!=null){   t = tipoMasEspecifico(t,new Attrs("int"));</pre>
		if (t!=null){
		RESULT = t;
		}else{
		throw new Exception("Integer expected.");
		}  }else{
	Exp_mul:e1 DIVINT	throw new Exception("Integer expected.");
	Exp_un:e2	
		ttrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		<pre>t = tipoMasEspecifico(t,new Attrs("int")); if (t!=null){</pre>
		(t!=  ti  ){  RESULT = t;
		}else{
		throw new Exception("Integer expected.");
		<b> </b> }
	Fire resulted MOD Fire view	}else{
	, · =	throw new Exception("Integer expected.");
		Attrs t = tipoMasEspecifico(e1,e2);
		if (t!=null){
		t = tipoMasEspecifico(t,new Attrs("boolean"));
		}
		}else{
		throw new Exception("Boolean expected.");
	l .	RESULT = e:
		Attrs t = tipoMasEspecifico(fun, new Attrs
		("boolean"));
		if (t!=null){
Exp un	NOT Exp fun:fun	}
Exp_un	Exp_mul:e1 MOD Exp_un: e2  Exp_mul:e1 ANDALSO Exp_un:e2 Exp_un:e  NOT Exp_fun:fun	throw new Exception("Integer expected."); } Attrs t = tipoMasEspecifico(e1,e2); if (t!=null){ t = tipoMasEspecifico(t,new Attrs("boolean")); if (t!=null){ RESULT = t; }else{ throw new Exception("Boolean expected."); } }else{ throw new Exception("Boolean expected."); } RESULT = e; Attrs t = tipoMasEspecifico(fun, new Attrs ("boolean"));

Regla Producción Acción semántica  ttrs t = tipoMasEspecifico(fun, numerico")); if (t!= null){     RESULT = t; }else{     throw new Exception("Integer } } Exp_fun:e RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){     RESULT = aux.getRange();	or real expected");
ttrs t = tipoMasEspecifico(fun, numerico")); if (t != null){ RESULT = t; }else{ throw new Exception("Integer MINUS Exp_fun:fun } Exp_fun:e  RESULT = e; Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	or real expected");
numerico")); if (t != null){     RESULT = t; }else{     throw new Exception("Integer } Exp_fun:e RESULT = e; Attrs aux = tipoMasEspecifico     atom,crearPolitipo("politipo"))) if (aux!=null){	or real expected");
if (t!= null){     RESULT = t;     }else{         throw new Exception("Integer     }     Exp_fun:e RESULT = e;  Attrs aux = tipoMasEspecifico     atom,crearPolitipo("politipo")))     if (aux!=null){	
RESULT = t; }else{ throw new Exception("Integer } Exp_fun:e  RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	
}else{ throw new Exception("Integer }  Exp_fun:e  RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	
throw new Exception("Integer  MINUS Exp_fun:fun }  Exp_fun:e RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	
MINUS Exp_fun:fun }  Exp_fun:e RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo")))  if (aux!=null){	
Exp_fun:e RESULT = e;  Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	(f.new Attrs("fun".
Attrs aux = tipoMasEspecifico atom,crearPolitipo("politipo"))) if (aux!=null){	(f.new Attrs("fun".
atom,crearPolitipo("politipo"))) if (aux!=null){	(f.new Attrs("fun".
if (aux!=null){	
if (aux!=null){	);
i ikezul i = aux detkande().	
else{	
throw new Exception("Not a vi	alid function.");
Exp_fun   Exp_fun:f Exp_atom:atom   }	
Exp_atom:atom RESULT = atom;	
Exp_atom INT   RESULT = new Attrs("int");	
REAL   RESULT = new Attrs("real")	
STRING   RESULT = new Attrs("string")	
Attrs aux = obtenerTipo(id.get	Lexeme());
if (aux!=null){	
if (!esTipoSimple(aux)){	
aux.agregarID(id.getLexeme()	));
	,
RESULT = aux;	
}	
else	
throw new Exception("Undecla	ared identifier (" + id.
ID	
RESULT = new Attrs("list",cre	arPolitipo("politipo"))
	1 (1 1 //
PARABRE Exp_Rel	
PARCIERRA   RESULT = e;	
,	
Tuple   RESULT = t;	
List RESULT = I;	
PARABRE Exp_Rel:e t2.addTupleTypeBegin(e);	
COMA Tuple2:t2   corregirTupla(t2);	
Tuple PARCIERRA RESULT = t2;	
t.addTupleTypeFinal(e);	
Tuple2 uple2:t COMA Exp_Rel:e RESULT = t;	
Attrs t = new Attrs("tuple", new	v Arrayl ict()):
	V ArrayList()),
t.addTupleTypeFinal(e);	
Exp_Rel:e RESULT = t;	
CORABRE List2:I	
List CORCIERRA RESULT = I;	
Attrs t = tipoMasEspecifico(list	t.getListType().exp):
if (t!=null)	51 (//- 1-//
RESULT = tipoMasEspecifico	(list new Attre/"list"
	(ווסנ,וופשי הננוס( ווסנ ,
(t));	
else	
List2:list COMA Exp_Rel:   throw new Exception("The list	has elements of
List2 exp different types.");	
Exp_Rel:exp RESULT = new Attrs("list",exp	o):