

PRODUCCIONES	SEMANTICO
s ::= instrucciones	.
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= select2 PCOMA	instruccion = select2
select2 ::= SELECT select_list FROM subquery inner orderby opc_Order limit	select2 = Select(3, False, None, select_list, subquery, inner, orderby, opc_Order, limit, None)
limit ::=	limit =
orderby ::=	orderby =
inner ::= compSelect	inner = compSelect
compSelect ::= list	compSelect = list
list ::= columna	list = [columna]
columna ::= prim	columna = prim
prim ::= var	prim = var
E ::= valores	E = valores
var ::= ID	var = Id(ID)
subquery ::=	subquery =
select_list ::= MULT	select_list =MULT
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= DROP FUNCTION IF EXISTS ID PTCOMA	instruccion = DropFunction(ID)
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= INSERT INTO ID VALUES value PTCOMA	instruccion = InsertInto(2,ID, None, value)
value ::= PARIZQ listaExpresiones PARDR	value = listaExpresiones
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= ID PARIZQ listaExpresiones PARDR	E = Funcion(ID,listaExpresiones)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= CADENA	valores = Primitivo(CADENA)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
s ::= intrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= INSERT INTO ID VALUES value PTCOMA	instruccion = InsertInto(2,ID, None, value)
value ::= PARIZQ listaExpresiones PARDR	value = listaExpresiones
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= ID PARIZQ listaExpresiones PARDR	E = Funcion(ID,listaExpresiones)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= CADENA	valores = Primitivo(CADENA)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= INSERT INTO ID VALUES value PTCOMA	instruccion = InsertInto(2,ID, None, value)
value ::= PARIZQ listaExpresiones PARDR	value = listaExpresiones
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= ID PARIZQ listaExpresiones PARDR	E = Funcion(ID,listaExpresiones)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E	listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= CADENA	valores = Primitivo(CADENA)
listaExpresiones ::= E	listaExpresiones.append(E)
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= CREATE FUNCTION ID PARIZQ l_param PARDR RETURNS preturns AS DOLAR DOLAR blodecla blobegin DOLAR DOLAR LANGUAGE PLPGSQL PTCOMA	instruccion = CreateFunction(ID,l_param,preturns,blodecla,blobegin
blobegin ::= BEGIN l_plsen END PTCOMA	blobegin = l_plsen
l_plsen ::= l_plsen plsen	l_plsen.append(plsen); l_plsen = plsen
plsen ::= plinstruccion	plsen = plinstruccion
plretu ::= RETURN E PTCOMA	plretu = Return(E)
E ::= valores	E = valores
operando ::= E MAS E	operando = Expresion(E, E, MAS)
E ::= valores	E = valores
var ::= ID	var = Id(ID)
E ::= valores	E = valores
operando ::= E MAS E	operando = Expresion(E, E, MAS)
E ::= valores	E = valores
var ::= ID	var = Id(ID)
E ::= valores	E = valores
var ::= ID	var = Id(ID)
l_plsen ::= plsen	l_plsen = [plsen]
plsen ::= plinstruccion	plsen = plinstruccion
plasnacion ::= ID pasigvalor PTCOMA	plasnacion = Plasnacion(ID,pasigvalor)
pasigvalor ::= DOSPT IGUAL E	pasigvalor = E
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
blodecla ::= DECLARE l_pldeclare	blodecla ::= l_pldeclare
l_pldeclare ::= pldecla	l_pldeclare.append(pldecla); l_pldeclare = pldecla
pldecla ::= ID tipo plasig PTCOMA	pldecla = Declaracion(ID,False,tipo,None,False,plasig)
plasig ::= DOSPT IGUAL E	plasig = E
E ::= valores	E = valores
boolean ::= TRUE	boolean = Primitivo(TRUE)
tipo ::= BOOLEAN	tipo = Tipo(BOOLEAN.upper(), None)
l_pldeclare ::= pldecla	l_pldeclare = pldecla
pldecla ::= ID tipo plasig PTCOMA	pldecla = Declaracion(ID,False,tipo,None,False,plasig)
plasig ::= IGUAL E	plasig = E
E ::= valores	E = valores
valores ::= ENTERO	valores = Primitivo(ENTERO)
tipo ::= INTEGER	tipo = Tipo(INTEGER.upper(), None)
preturns ::= typeparam	preturns = typeparam
typeparam ::= tipo	typeparam = tipo
tipo ::= INTEGER	tipo = Tipo(INTEGER.upper(), None)
l_param ::= l_param COMA param	l_param.append(param); l_param = l_param
param ::= ID typeparam	param = Parametro(l.ID,typeparam)
typeparam ::= tipo	typeparam = tipo
tipo ::= INTEGER	tipo = Tipo(INTEGER.upper(), None)
l_param ::= param	l_param = param
param ::= ID typeparam	param = Parametro(l.ID,typeparam)
typeparam ::= tipo	typeparam = tipo
tipo ::= INTEGER	tipo = Tipo(INTEGER.upper(), None)
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= CREATE TABLE ID PARIZQ campos PARDR PTCOMA	instruccion ::= CreateTable(ID,campos,None)
campos ::= campos COMA campo	campos = campos; campos.append(campo)
campo ::= ID tipo	campo = Campo(l.ID,Tipo,None,None,None,None)
tipo ::= integer	tipo = Tipo(integer.upper(), None)
campos ::= campos COMA campo	campos = campos; campos.append(campo)
campo ::= ID tipo acompaniamiento	campo = Campo(l.ID,tipo,acompaniamiento,None,None,None)
acompaniamiento ::= acom	acompaniamiento.append([acom])
acom ::= NOT NULL	acom = Acompaniamiento(NOTNULL, None)
tipo ::= integer	tipo = Tipo(integer.upper(), None)
campos ::= campos COMA campo	campos = campos; campos.append(campo)
campo ::= ID tipo acompaniamiento	campo = Campo(l.ID,tipo,acompaniamiento,None,None,None)
acompaniamiento ::= acom	acompaniamiento.append([acom])
acom ::= NOT NULL	acom = Acompaniamiento(NOTNULL, None)
tipo ::= varchar(150)	tipo = Tipo(varchar- ENTERO,PRIMITIVO(150)
campos ::= campo	campos = campo
campo ::= ID tipo acompaniamiento	campo = Campo(l.ID,tipo,acompaniamiento,None,None,None)
acompaniamiento ::= acompaniamiento acom	acompaniamiento = acompaniamiento; acompaniamiento.append(acom)
acom ::= PRIMARY KEY	acom = Acompaniamiento(PRIMARYKEY, None)
acompaniamiento ::= acom	acompaniamiento.append([acom])
acom ::= NOT NULL	acom = Acompaniamiento(NOTNULL, None)
tipo ::= integer	tipo = Tipo(integer.upper(), None)
s ::= instrucciones	instrucciones=instruccion;instrucciones.append(instruccion)
instruccion ::= USE ID PTCOMA	instruccion = Use(ID)
instrucciones ::= instrucciones instruccion	intrucciones=[instruccion]
instruccion ::= opcionCR ID PTCOMA	instruccion = CreateReplace(opcionR, False, ID, None)
opcionCR ::= CREATE DATABASE	opcionCR = l