PRODUCCIONES s ::= intrucciones	SEMANTICO
s ::= intrucciones instruccion ::= select2 PCOMA	instrucciones=instrucciones;instrucciones.append(instruccion) instruccion = select2 select2 = Select(2, Felex, Name, select, list, subgroups, inner, and other, one, Onder, limit, Name, select 2
select2 ::= SELECT select_list FROM subquery inner orderby opc_Order limit limit ::= orderby ::=	select2 = Select(3, False, None, select_list, subquery, inner, orderby, opc_Order, limit, No limit = orderby =
inner ::= compSelect compSelect ::= list list ::= columna	inner = compSelect compSelect = list list = [columna]
columna ::= prim prim ::= var	columna = prim prim = var
E ::= valores var ::= ID subquery ::=	E = valores $var = Id(ID)$ $subquery =$
select_list ::= MULT s ::= intrucciones	select_list =MULT instrucciones=instrucciones;instrucciones.append(instruccion)
instruccion ::= DROP FUNCTION IF EXISTS ID PTCOMA s ::= intrucciones instruccion :: = INSERT INTO ID VALUES value PTCOMA	instruccion = DropFunction(ID) instrucciones=instrucciones;instrucciones.append(instruccion) instruccion = InsertInto(2,ID, None, value)
value :: = PARIZQ listaExpresiones PARDR listaExpresiones :: = listaExpresiones COMA E	value = listaExpresiones listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores valores ::= ENTERO listo Expressiones ::= listo Expressiones COMA E	E = valores valores = Primitivo(ENTERO) listaExpresiones = listaExpresiones; listaExpresiones.append(E)
listaExpresiones :: = listaExpresiones COMA E E ::= ID PARIZQ listaExpresiones PARDR listaExpresiones :: = listaExpresiones COMA E	E = Funcion(ID, listaExpresiones) listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores valores ::= ENTERO	E = valores valores = Primitivo(ENTERO)
listaExpresiones :: = E E ::= valores valores ::= ENTERO	listaExpresiones.append(E) E = valores valores = Primitivo(ENTERO)
listaExpresiones ::= listaExpresiones COMA E E ::= valores	listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = valores
valores ::= CADENA listaExpresiones ::= E E ::= valores	valores = Primitivo(CADENA) listaExpresiones.append(E) $E = valores$
valores ::= ENTERO s ::= intrucciones	valores = Primitivo(ENTERO) instrucciones=instrucciones;instrucciones.append(instruccion)
instruccion :: = INSERT INTO ID VALUES value PTCOMA value :: = PARIZQ listaExpresiones PARDR listaExpresiones :: = listaExpresiones COMA E	instruccion = InsertInto(2,ID, None, value) value = listaExpresiones listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores valores ::= ENTERO	E = valores valores = Primitivo(ENTERO)
listaExpresiones :: = listaExpresiones COMA E E ::= ID PARIZQ listaExpresiones PARDR	listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = Funcion(ID,listaExpresiones)
listaExpresiones ::= listaExpresiones COMA E E ::= valores valores ::= ENTERO	listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = valores valores = Primitivo(ENTERO)
listaExpresiones :: = E E ::= valores	listaExpresiones.append(E) E = valores
valores ::= ENTERO listaExpresiones ::= listaExpresiones COMA E E ::= valores	valores = Primitivo(ENTERO) listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = valores
valores ::= CADENA listaExpresiones ::= E	valores = Primitivo(CADENA) listaExpresiones.append(E)
E ::= valores valores ::= ENTERO s ::= intrucciones	E = valores valores = Primitivo(ENTERO) instrucciones=instrucciones;instrucciones.append(instruccion)
instruccion :: = INSERT INTO ID VALUES value PTCOMA value :: = PARIZQ listaExpresiones PARDR	instruccion = InsertInto(2,ID, None, value) value = listaExpresiones
listaExpresiones ::= listaExpresiones COMA E E::= valores	listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = valores
valores ::= ENTERO listaExpresiones ::= listaExpresiones COMA E E ::= ID PARIZQ listaExpresiones PARDR	valores = Primitivo(ENTERO) listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = Funcion(ID,listaExpresiones)
listaExpresiones :: = listaExpresiones COMA E E ::= valores	listaExpresiones = listaExpresiones; listaExpresiones.append(E) E = valores
valores ::= ENTERO listaExpresiones ::= E E ::= valores	valores = Primitivo(ENTERO) listaExpresiones.append(E) $E = valores$
valores ::= ENTERO listaExpresiones ::= listaExpresiones COMA E	valores = Primitivo(ENTERO) listaExpresiones = listaExpresiones; listaExpresiones.append(E)
E ::= valores valores ::= CADENA listaExpresiones ::= E	E = valores valores = Primitivo(CADENA) listaExpresiones.append(E)
E ::= valores valores ::= ENTERO	E = valores valores = Primitivo(ENTERO)
s ::= intrucciones ruccion ::= CREATE FUNCTION ID PARIZQ l_param PARDR RETURNS plreturns AS DOLAR DOLAR blodecla blobegin DOLAR DOLAR LANGUAGE PLPGSQL PTC blobegin ::= BEGIN l_plsen END PTCOMA	instrucciones=instrucciones;instrucciones.append(instruccion) OMA instruccion = CreateFunction(ID,l_param,plreturns,blodecla,blobegin blobegin = l_plsen
l_plsen ::= l_plsen plsen plsen ::= plinstruccion	l_plsen.append(plsen); l_plsen = plsen plsen = plinstruccion
plretu ::= RETURN E PTCOMA	
E ::= valores operando ::= E MAS E	plretu = Return(E) E = valores operando = Expresion(E, E, MAS)
operando ::= E MAS E E ::= valores var ::= ID	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID)
operando ::= E MAS E E ::= valores var ::= ID E ::= valores operando ::= E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS)
operando ::= E MAS E E ::= valores var ::= ID E ::= valores	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores
operando := E MAS E E := valores var := ID E := valores operando := E MAS E operando := E MAS E E := valores var := ID E := valores var := ID L plsen := plsen	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) L = valores var = Id(ID) E = valores var = Id(ID)
operando ::= E MAS E E ::= valores var ::= ID E ::= valores operando ::= E MAS E E ::= valores tan := ID E ::= valores var ::= ID E ::= valores var ::= ID	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID)
operando ::= E MAS E E ::= valores var ::= ID E ::= valores operando ::= E MAS E E ::= valores in the second one of the second one	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) E = valores var = Id(ID) [_plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO)
Description	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [_plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= l_pldeclare l_pldeclare.append(pldecla); l_pldeclare = pldecla
Operando ::= E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [_plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= l_pldeclare l_pldeclare.append(pldecla); l_pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores
Operando := E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) Lplsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID, pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= 1_pldeclare Lpldeclare.append(pldecla); 1_pldeclare = pldecla pldecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None)
Operando := E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= l_pldeclare l_pldeclare.append(pldecla); l_pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE)
Operando := E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [D
operando == E MAS E E == valores var := ID E == valores operando == E MAS E E == valores var := ID E == valores var := ID L == valores var := ID L pisen := pisen plasignacion := ID pasigvalor PTCOMA pasigvalor := DOSPT IGUAL E E == valores valores valores == DECLARE L pikeclare L pikeclare := pidecla pidecla := DECLARE L pikeclare L pikeclare := pidecla pidecla := TRUE tipo == RODLEAN 1 pikeclare := pidecla pidecla := ID tipo pisig PTCOMA plasig := DOSPT IGUAL E E := valores D tipo DIST IGUAL E D tipo DIST IGUAL E D tipo DIST IGUAL E D tipo	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [E = valores var = Id(ID) [I_plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= I_pldeclare I_pldeclare.append(pldecla); I_pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) I_pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores valores = Primitivo(ENTERO)
Operando := E.MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) [E = valores var = Id(ID) [I plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= [pldeclare I pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOCLEAN.upper(), None) I pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores valores = Primitivo(TRUE) tipo = Tipo(BOELEAN.upper(), None) I placelare = pldecla placela = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) plreturns = typeparam typeparam = tipo tipo = Tipo(INTEGER.upper(), None)
operando := E MAS E E := valores var := ID E := valores operando := E MAS E E := valores var := ID E := valores var := ID Lpken := plasen plast := plinstruccion plastignacion := ID pastignalor PTCOMA pastignacion := ID pastignalor PTCOMA pastignacion := DOSPT IGUAL E E := valores valores := ENTEGEA plate := DECLARE L platetare lpkelen := Dispo plastig PTCOMA plastig := DOSPT IGUAL E E := valores valores := ENTEGEA Dideclare := picteta plate := DECLARE L platetare lptice lare := Dispo plastig PTCOMA plastig := DOSPT IGUAL E E := valores booken := TRUE tipo := BOOLEAN lptice lare := picteta plateta := ID tipo plastig PTCOMA plastig := IOULE E := valores booken := TRUE tipo := BOOLEAN lptice lare := picteta plateta := ID tipo plastig PTCOMA plastig := IGUAL E E := valores valores := ENTERO tipo := ENTEGER preturns := typo pararam typo param := typo tipo := INTEGER	E = valores
Operando = EMAS E	E = valores operando = Expresion(F, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) E = valores var = Id(ID) Lpisen = [pisen] placen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= Lpideclare Lpideclare.append(pidecla); Lpideclare = pidecla placel = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) Lpideclare = pidecla placel = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) Lipo = Tipo(INTEGER.upper(), None) Lparam.apped(param): Lparam = Lparam param = Parametro(1, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(1, ID, typeparam)
Operando = EMAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) I_pleen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla :: I_pldeclare I_pldeclare.append(pldecla); I_pldeclare = pldecla pldecla = Declaracion(ID,Fascupo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOCLEAN.upper(), None) I_pldeclare = pldecla pldecla = Declaracion(ID,Fascupo,None,False,plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(EDECLEAN.upper(), None) I_plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) I_param.apped(param); I_param = I_param param = Parametro(I,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) I_param = param param = Parametro(I,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) I_param = Parametro(I,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None)
Committee Comm	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) Lpken = [pken] pken = plinstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla := Lpldeclare l_pldeclare.append(pldecla); l_pldeclare = pldecla pldecla = Declaracion(ID,Fase,tipo,None,False,plasig) plasig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) Lpldeclare = pldecla pldecla = Declaracion(ID,Fase,tipo,None,False,plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) preturns = typeparam typeparam = tipo tipo = Tipo(INTEGER.upper(), None) L param.apped(param); L param = L param param = Parametro(1,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) L param = param param = Parametro(1,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) l_param = Parametro(1,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) l_param = Parametro(1,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) instruccion:= CreateTable(ID,campos,None) campos = campos; campos.append(campo)
Operando = E-MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) Lesca = [pken] pken = [pken] pken = [pken] pken = planstruccion plasignacion = Plasignacion(ID,pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= L pldeclare L pldeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores bookean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) Lpdeclare = pldecla pldecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = F E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) Lparum.apped(param); Lparam = Lparam param = Parametro(,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) L param = param param = Parametro(,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) 1 param = param param = Parametro(,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) 1 param = param param = Parametro(,ID,typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) instrucciones-instrucciones; instrucciones.append(instruccion) instruccion :: C reate l'able(ID,campos,None) campos = campos; campos,appendicampo) campos = campos; campos,appendicampo) campos = campos; campos,appendicampo)
Operando = E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) Le = valores var = Id(ID) Lpken = [pken] pken = plinstruccion plasignacion = Plasignacion(ID, pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) blodecla ::= Lpkleclare Lpkleclare = pldecla pklecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores bookan = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) 1 pkleclare = pklecla pklecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER.upper(), None) preturns = typeparam typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam.apped(param); Lparam = Lparam param = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) lparam = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) lparam = Parametro(1,ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) instruccion :: = Create Table(ID, campos, None) campos = campos; campos, append(campo) campos = campos; campos, append(campo)
Operation F. MASE	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) E = valores var = Id(ID) I = I = Valores plasignacion = Plasignacion(ID, plasigvalor) pasigvalor = E E = Valores valores = Primitivo(ENTERO) blodecla := I_pkleclare = pklecla pkleclare = ppkleclare I_pkleclare = ppkleclare I_pkleclare = pkleclare I_pkleclare = pkleclare pklecla = Declaracion(ID,False,tipo,None,False,plasig) plasig = E E = valores bookan = Primitivo(ENTERO) tipo = Tipo(BOLEAN,upper(), None) I_pkleclare = pklecla pklecla = Declaracion(ID,False,tipo,None,False,plasig) phsig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(INTEGER,upper(), None) Iperum = Iperum lypeparum = tipo upo = Tipo(INTEGER,upper(), None) I = parum = parum parum = Parametro(I,ID,typeparum) lypeparum = tipo upo = Tipo(INTEGER,upper(), None) I = parum = parum parum = Parametro(I,ID,typeparum) lypeparum = tipo upo = Tipo(INTEGER,upper(), None) atipo = Tipo(INTEGER,upper(), None) instruccions = instrucciones; instrucciones, append(instruccion) instruccion = CreateTable(ID,campos,None) campos = campos; campos, append(campo)
Sprando = E MAS E	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) Lpken = [pbsen] plsen = plinstruccion plasignacion Plasignacion(ID, pasigvalor) pasigvalor = B E = valores valores = Primitivo(ENTERO) blodecla := Lpldeclare Lpldeclare append(pldecla); Lpldeclare = pldecla pldecla = Declaracion(ID, False, tipo, None, False, plasig) plisig = E E = valores boolean = Primitivo(TRUE) tipo = Tipo(BOOLEAN.upper(), None) Lpdeclare = pldecla pldecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores valores = Primitivo(FNTERO) tipo = Tipo(INTEGER.upper(), None) Lparam.apped(param); Lparam = Lparam puram = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam.apped(param); Lparam = Lparam param = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) aparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) Lparam = param param = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) aparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) aparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) aparam = Parametro(, ID, typeparam) typeparam = tipo tipo = Tipo(INTEGER.upper(), None) aparam = Parametro(, ID, typeparam, None, None) campo = Campo (, ID, type, None, None, None, None) acompaniamiento, None, None, None, None) acompaniamiento, None, None, None, None)
Open	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) E = valores var = Id(ID) I plem = [plem] plem = plinstruccion plasignacion = Plasignacion(D, pasigvalor) pasigvalor = E E = valores valores = Primitivo(ENTERO) bloadeda := 1.placelare 1.placelare = placela placela = Declaracion(ID, False, plasig) plasig = E E = valores bookan = Primitivo(ENTERO) tipo = Tipo(BOOLEAN, tapper(), None) 1.placelare = placela placela = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores bookan = Primitivo(ENTERO) tipo = Tipo(MOTEGER, upper(), None) 1.placelare = placela placela = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores valores = Primitivo(ENTERO) tipo = Tipo(NTEGER, upper(), None) 1.param = type tipo = Tipo(NTEGER, upper(), None) 1.param = type tipo = Tipo(NTEGER, upper(), None) 1.param = param param = Parametro(1, ID, typeparam) typeparam = tipo tipo = Tipo(NTEGER, upper(), None) 1.param = param param = Parametro(1, ID, typeparam) typeparam = tipo tipo = Tipo(NTEGER, upper(), None) 1.param = param param = Parametro(1, ID, typeparam) typeparam = tipo tipo = Tipo(NTEGER, upper(), None) acom = Campo(1, ID, Tipo, None, None, None, None, None, None) campo = Campo(1, ID, Tipo, None, None, None, None, None) campo = Campo(1, ID, Tipo, Aone, None, None, None, None) acom = Acompaniamiento(NOTNULL, None) isou = Tipo(tonge, upper(), None) campo = Campo(1, ID, Tipo, acompaniamiento, None, None, None, None) acompaniamiento(notnull, None) isou = Tipo(tonge, upper(), None) campo = Campo(1, ID, Tipo, acompaniamiento, None, None, None, None, None, None) acompaniamiento(notnull, None) isou = Tipo(tonge, upper(), None) campo = Campo(1, ID, Tipo, acompaniamiento, None, None, None, None, None, None, None, None, None, None) acompaniamiento(notnull, Lone) isou = Tipo(tonge, upper(),
openado - B MAS B Reservible Reservible Services Openado - F MAS R Bernotoria Openado - F MAS R Bernotoria Very ED Reservible Reservible Openado - F MAS R Bernotoria Very ED Reservible Openado - F MAS R	E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores operando = Expresion(E, E, MAS) E = valores operando = Expresion(E, E, MAS) E = valores var = Id(ID) E = valores var = Id(ID) E = valores var = Id(ID) I = E = valores var = Id(ID) I = Valores valores = Valores I = Valores Debechar = Opelaracion(ID, Falec,tipo, None, False, plasig) plasig = E E = valores booleun = Printitvo(TRUT) tipo = Tipo(IBOOLEAN upper(I), None) I = Valores valores = Printitvo(TRUR) tipo = Tipo(INTEGER, upper(I), None) I = Valores valores = Valores valore
epomodo - B MAS E E = WASE The STATE The Motors operation - B MAS E E = Wastes operation - B MAS E E = Wastes war = ID I = Wastes A	E = valores operando = Expression(E, E, MAS) E = valores vur = Id(ID) E = valores operando = Expression(E, E, MAS) E = valores operando = Expression(E, E, MAS) E = valores vur = Id(ID) E = valores vur = Id(ID) E = valores vur = Id(ID) I _plsen = [plsen] plsen = plinstruccion plasignacion = Plasignacion(ID, pasigvalor) pasigvalor = E E = valores valores = Primitvo(ENTERO) blodecla = I _pldeclare I _pldeclare = pldecla pldecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores bolockar = Primitvo(ENTERE) tipo = Tipo(BOOLEAN) plasig = E E = valores pldecla = Declaracion(ID, False, tipo, None, False, plasig) plasig = E E = valores valores = Primitvo(ENTERO) tipo = Tipo(SNTEGER, upper(), None) I _pldeclare = placela valores = Primitvo(ENTERO) tipo = Tipo(NTEGER, upper(), None) _pranum = pranum = tipo tipo = Tipo(NTEGER, upper(), None) _parame = parame purame = parame p
Open	E - valores
Special Color	E - valores