

REPORTE GRAMATICAL

inicio -> instruccion

{ t[0]= t[1] }

instruccion -> MAIN DOSPUNTOSlistainstrucciones

{ t[0]=t[3] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',''); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',''); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,'');t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,'');t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,'');t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',''); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,'');t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_if

{t[0]=t[1] }

inst_if -> IFPARIZQ expresion PARDERGOTO ID PUNTOCOMA

{t[0] = crear_hoja('sentenciaif','') }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',""); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_if

{t[0]=t[1] }

inst_if -> IFPARIZQ expresion PARDERGOTO ID PUNTOCOMA

{t[0] = crear_hoja('sentenciaif',") }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_if

{t[0]=t[1] }

inst_if -> IFPARIZQ expresion PARDERGOTO ID PUNTOCOMA

{t[0] = crear_hoja('sentenciaif','') }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_exit

{t[0]=t[1] }

inst_exit -> EXIT PUNTOCOMA

{t[0] = crear_hoja('exit','') }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',''); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,'');t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> RESTA valorp

{ t[0] = crear_hoja('negativo',"");t[0] = agregar_hijo(t[0],t[2]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> inst_array

{t[0]= t[1] }

inst_array -> ARRAY PARIZQPARDER

{t[0] = crear_hoja('exit',"") }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir',"");t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir',"");t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir',''); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir',''); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir',''); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir',''); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> inst_array

{t[0]= t[1] }

inst_array -> ARRAY PARIZQPARDER

{t[0] = crear_hoja('exit','') }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_goto

{t[0]=t[1] }

inst_goto -> GOTO ID PUNTOCOMA

{ t[0] = crear_hoja('goto',''); hijo = crear_hoja('label',t[2]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var', t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable', ""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0], hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0], t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir', ""); t[0] = agregar_hijo(t[0], t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena', t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0], t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] = crear_hoja('imprimir',''); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_if

{t[0]=t[1] }

inst_if -> IFPARIZQ expresion PARDERGOTO ID PUNTOCOMA

{t[0] = crear_hoja('sentenciaif','') }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp op valorp

{t[0]=crear_hoja(t[2].valor,"");t[0]=agregar_hijo(t[0],t[1]);t[0]=agregar_hijo(t[0],t[2]);t[0]=agregar_hijo(t[0],t[3]) }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

valorp -> VAR

{t[0] = crear_hoja('var',t[1]) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> etiqueta

{t[0]=t[1] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero', t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable', ""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0], hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0], t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion', ""); t[0] = agregar_hijo(t[0], t[1]); t[0] = agregar_hijo(t[0], t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero', t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable', ""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0], hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0], t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion', ""); t[0] = agregar_hijo(t[0], t[1]); t[0] = agregar_hijo(t[0], t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] = crear_hoja('cadena', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero', t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable', ""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0], hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0], t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion', ""); t[0] = agregar_hijo(t[0], t[1]); t[0] = agregar_hijo(t[0], t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero', t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',""); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENADOBLE

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> CADENA

{t[0] =crear_hoja('cadena',t[1]) }

variable -> variable

{t[0] = t[1] }

var_arreglo -> LLAVEIZQ valorp LLAVEDER

{t[0]=t[2] }

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> lista

{ t[0] = crear_hoja('lista_inst',''); t[0] = agregar_hijo(t[0],t[1]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}

expresion -> inst_array

{t[0]= t[1] }

inst_array -> ARRAY PARIZQPARDER

{t[0] = crear_hoja('exit','') }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

inicio -> instruccion

{ t[0]= t[1] }

instruccion -> MAIN DOSPUNTOSlistainstrucciones

{ t[0]=t[3] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir','');t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> lista

{ t[0] = crear_hoja('lista_inst',''); t[0] = agregar_hijo(t[0],t[1]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3]) }

expresion -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> ENTERO

{t[0]=crear_hoja('entero',t[1]) }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

inicio -> instruccion

{ t[0]= t[1] }

instruccion ->MAIN DOSPUNTOSlistainstrucciones

{ t[0]=t[3] }

listainstrucciones -> listainstrucciones

{t[0]=t[1]; T[0]=agregar_hijo(t[0],t[2]) }

lista -> inst_imprimir

{t[0]=t[1] }

inst_imprimir -> IMPRIMIR PARIZQ expresion PARDER PUNTOCOMA

{t[0] =crear_hoja('imprimir',"");t[0] = agregar_hijo(t[0],t[3]) }

expresion -> variable

{t[0]= t[1] }

variable -> VAR

{t[0] = crear_hoja('variable',""); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }

listainstrucciones -> lista

{ t[0] = crear_hoja('lista_inst',""); t[0] = agregar_hijo(t[0],t[1]) }

lista -> inst_asignacion

{t[0]=t[1] }

inst_asignacion -> variable IGUAL expresion PUNTOCOMA

```
{ t[0] = crear_hoja('asignacion',''); t[0] = agregar_hijo(t[0],t[1]); t[0] = agregar_hijo(t[0],t[3])}
```

expresion -> expresion_num

```
{t[0]= t[1] }
```

expresion_num -> valorp

```
{ t[0]=t[1]}
```

valorp -> ENTERO

```
{t[0]=crear_hoja('entero',t[1]) }
```

variable -> VAR

```
{t[0] = crear_hoja('variable',''); hijo = crear_hoja('var', t[1]); t[0] = agregar_hijo(t[0],hijo) }
```