

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 -> 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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

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

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 -> variable

{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 -> variable

{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 -> conversion

{t[0]= t[1] }

conversion -> PARIZQ

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

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 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 -> CADENA

{t[0] =crear_hoja('cadena',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 -> leer_valor

{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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] =crear_hoja('cadena',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_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 -> CADENA

{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 -> 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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> 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_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_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 -> CADENA

{t[0] =crear_hoja('cadena',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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] =crear_hoja('cadena',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 -> variable

{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) }

valorp -> variable

{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 -> variable

{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 -> 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 -> variable

{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) }

valorp -> variable

{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 -> 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 -> variable

{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) }

valorp -> variable

{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 -> 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 -> variable

{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) }

valorp -> variable

{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 -> 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_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_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 -> variable

{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 -> variable

{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 -> variable

{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 -> variable

{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 -> conversion

{t[0]= t[1] }

conversion -> PARIZQ

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

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 -> CADENA

{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 -> 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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] = crear_hoja('cadena', 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 -> leer_valor

{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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] =crear_hoja('cadena',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_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 -> CADENA

{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 -> 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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] = crear_hoja('cadena',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 -> leer_valor

{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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> 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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] = crear_hoja('cadena',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 -> leer_valor

{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_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 -> CADENA

{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 -> expresion_num

{t[0]= t[1] }

expresion_num -> valorp

{ t[0]=t[1]}

valorp -> CADENA

{t[0] =crear_hoja('cadena',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_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 -> CADENA

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

listainstrucciones -> lista

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

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 -> CADENA

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