

## Gramática Atribuida - Identificación

Nodo	Predicados	Reglas Semánticas
<b>programa</b> → <i>definiciones:definicion*</i>		
<b>defCampo</b> → <i>nombre:String tipo:tipo</i>	campos[nombre] == null	campos[nombre] = defCampo
<b>defReturn</b> → <i>tipo:tipo</i>		
<b>cuerpo</b> → <i>definicionesVariables:defVariable* sentencias:sentencia*</i>		
<b>defVariable:definicion</b> → <i>nombre:String tipo:tipo ambito:String</i>	variables.buscarActual(nombre) == null	variables[nombre]=defVariable
<b>defFuncion:definicion</b> → <i>nombre:String parametros:defVariable* retorno:defReturn cuerpo:cuerpo</i>	funciones[nombre] == null	funciones[nombre]=defFuncion { variables.set() visit(parámetros i) visit(cuerpo, funciones[nombre]) variable.reset() }
<b>defEstructura:definicion</b> → <i>nombre:String campos:defCampo*</i>	estructuras[nombre] == null	estructuras[nombre]=defEstructura { visit(defCampo i) }
<b>tipoEntero:tipo</b> → λ		
<b>tipoReal:tipo</b> → λ		
<b>tipoChar:tipo</b> → λ		
<b>tipoIdent:tipo</b> → <i>nombre:String</i>	estructuras[nombre] ≠ null	tipoEntero.definicion = estructuras[nombre]
<b>tipoArray:tipo</b> → <i>dimension:literalEntero tipo:tipo</i>		
<b>asignacion:sentencia</b> → <i>left:expresion right:expresion</i>		
<b>print:sentencia</b> → <i>salida:expresion</i>		
<b>printLN:sentencia</b> → <i>salida:expresion</i>		
<b>printSP:sentencia</b> → <i>salida:expresion</i>		

<b>read:</b> sentencia → <i>entrada:</i> expresion		
<b>ifElse:</b> sentencia → <i>condicion:</i> expresion <i>correcto:</i> sentencia* <i>incorrecto:</i> sentencia*		
<b>while:</b> sentencia → <i>condicion:</i> expresion <i>correcto:</i> sentencia*		
<b>return:</b> sentencia → <i>devolucion:</i> expresion		return.definicion = defFuncion
<b>invocacionProcedimiento:</b> sentencia → <i>nombre:</i> String <i>argumentos:</i> expresion*	funciones[nombre] ≠ null	invocacionProcedimiento.definicion = funciones[nombre]
<b>literalEntero:</b> expresion → <i>valor:</i> String		
<b>literalReal:</b> expresion → <i>valor:</i> String		
<b>literalChar:</b> expresion → <i>valor:</i> String		
<b>variable:</b> expresion → <i>nombre:</i> String	variables.buscar(nombre) ≠ null	variable.definición = variable.buscar(nombre)
<b>expresionAritmetica:</b> expresion → <i>left:</i> expresion <i>operador:</i> String <i>right:</i> expresion		
<b>expresionComparacion:</b> expresion → <i>left:</i> expresion <i>operador:</i> String <i>right:</i> expresion		
<b>negacion:</b> expresion → <i>expresion:</i> expresion		
<b>invocacionFuncion:</b> expresion → <i>nombre:</i> String <i>argumentos:</i> expresion*	funciones[nombre] ≠ null	invocacionFuncion.definicion = funciones[nombre]
<b>cast:</b> expresion → <i>tipo:</i> tipo <i>expresion:</i> expresion		
<b>accesoArray:</b> expresion → <i>identificador:</i> expresion <i>posicion:</i> expresion		
<b>accesoCampo:</b> expresion → <i>expresion:</i> expresion <i>campo:</i> String		
<b>menosUnario:</b> expresion → <i>expresion:</i> expresion		

Recordatorio de operadores (para cortar y pegar):  $\Rightarrow \Leftrightarrow \neq \emptyset \in \notin \cup \cap \subset \not\subset \sum \exists \forall$

Atributos

Categoría Sintáctica	Nombre del atributo	Tipo Java	Heredado/Sintetizado	Descripción
variable	definicion	DefVariable	Sintetizado	
invocacionProcedimiento	definicion	DefFuncion	Heredado	
invocacionFuncion	definicion	DefFuncion	Heredado	
campo	definicion	DefCampo	Sintetizado	
return	definicion	DefFuncion	Heredado	