## Gramática Atribuida - Identificación

|  |  |  |
| --- | --- | --- |
| **Nodo** | **Predicados** | **Reglas Semánticas** |
| programa → definicion:definicion\* |  |  |
| retorno → tipo:tipo |  |  |
| cuerpo → defvariable:defVariable\* sentencia:sentencia\* |  |  |
| defCampo → nombre:String tipo:tipo | campos[nombre] == null | campos[nombre] = defCampo |
|  |  |  |
| defVariable:definicion → nombre:String tipo:tipo ambito:int | variables.buscarActual(nombre) == null | variables[nombre]=defVariable |
| defFuncion:definicion → nombre:String parametros:defVariable\* retorno:retorno cuerpo:cuerpo | funciones[nombre] == null | funciones[nombre]  {  variables.set()  visit(parámetros i)  visit(cuerpo)  variable.reset()  } |
| defEstructura:definicion → nombre:String defcampo:defCampo\* | estructuras[nombre] == null | estructuras[nombre]  {  visit(defCampo i)  } |
|  |  |  |
| intType:tipo → λ |  |  |
| realType:tipo → λ |  |  |
| charType:tipo → λ |  |  |
| identType:tipo → nombre:String | estructuras[nombre] ≠ null | identType.definicion = estructuras[nombre] |
| arrayType:tipo → dimension:literalInt tipo:tipo |  |  |
|  |  |  |
| while:sentencia → condicion:expresion cierto:sentencia\* |  |  |
| ifElse:sentencia → condicion:expresion cierto:sentencia\* falso:sentencia\* |  |  |
| return:sentencia → expresion:expresion |  | return.definicion = defFuncion |
| print:sentencia → expresion:expresion |  |  |
| println:sentencia → expresion:expresion |  |  |
| printsp:sentencia → expresion:expresion |  |  |
| read:sentencia → expresion:expresion |  |  |
| asigna:sentencia → left:expresion right:expresion |  |  |
| llamadaFuncionSentencia:sentencia → nombre:String argumentos:expresion\* | funciones[nombre] ≠ null | llamadaFuncionSentencia.definicion = funciones[nombre] |
|  |  |  |
| expresionAritmetica:expresion → left:expresion operador:String right:expresion |  |  |
| expresionBooleana:expresion → left:expresion operador:String right:expresion |  |  |
| invocacion:expresion → nombre:String argumentos:expresion\* | funciones[nombre] ≠ null | invocacion.definicion = funciones[nombre] |
| variable:expresion → nombre:String | variables.buscar(nombre) ≠ null | variable.definición = variablesbuscar(nombre) |
| literalInt:expresion → valor:String |  |  |
| literalReal:expresion → valor:String |  |  |
| literalChar:expresion → valor:String |  |  |
| varArray:expresion → identificacion:expresion posicion:expresion |  |  |
| cast:expresion → tipo:tipo expresion:expresion |  |  |
| navega:expresion → expresion:expresion nombre:String |  |  |
| conParentesis:expresion → expresion:expresion |  |  |
| negacion:expresion → expresion:expresion |  |  |
| menosUnario:expresion → expresion:expresion |  |  |
|  |  |  |

Recordatorio de operadores (para cortar y pegar): ⇒ ⇔ ≠ ∅ ∈ ∉ ∪ ∩ ⊂ ⊄ ∑ ∃ ∀

### Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoría Sintáctica** | **Nombre del atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
| variable | definicion | DefVariable | Sintetizado |  |
| llamadaFuncionSentencia | definicion | DefFuncion | Sintetizado |  |
| invocacion | definicion | DefFuncion | Sintetizado |  |
| campo | definicion | DefCampo | Sintetizado |  |
| return | definicion | DefFuncion | Sintetizado¿? |  |

Conjuntos auxiliares:

variables, funciones y estructuras. Existe otro conjunto, campos, pero es local de cada estructura.