# Attribute Grammar

|  |  |  |
| --- | --- | --- |
| Nodo | Predicados | Reglas Semánticas |
| program → *bloque*:bloque\* |  |  |
|  |  |  |
| definicion\_campo\_struct → *nombre*:String *tipo*:tipo | campos\_struct[nombre]== ∅ | campos\_struct [nombre] = definicion\_variable\_struct |
| definicion\_variable\_local → *nombre*:String *tipo*:tipo | variables\_funcion[nombre]== ∅  parametros[nombre]== ∅ | variables\_funcion[nombre]=definicion\_variable\_funcion |
| parametro → *nombre*:String *tipo*:tipo | parametros[nombre]== ∅ | parametros[nombre]= parametro |
|  |  |  |
| definicion\_variable\_global:bloque → *nombre*:String *tipo*:tipo | variables[nombre]== ∅ | variables[nombre]=definicion\_vairable |
| struct:bloque → *nombre*:String *definicion\_campo\_struct*:definicion\_campo\_struct\* | estructuras[nombre]== ∅ | estructuras[nombre]=struct  {  campos\_struct.set()  visit(definicion\_variable\_struct i)  campos\_struct.reset()  } |
| funcion:bloque → *nombre*:String *parametros*:parametro\* *retorno*:tipo\* *locales*:definicion\_variable\_local\* *sentencias*:sentencia\* | funcion[nombre] == ∅ | funcion [nombre] = funcion  {  parametros.set()  variables\_locales.set()  visit(parametro i)  variables\_locales.reset()  parametros.reset()  } |
|  |  |  |
| sentencia\_asignacion:sentencia → *izquierda*:expr *derecha*:expr |  |  |
| sentencia\_print:sentencia → *expresiones*:expr |  |  |
| sentencia\_read:sentencia → *expresiones*:expr |  |  |
| sentencia\_if:sentencia → *condicion*:expr *sentencias*:sentencia\* *sino*:sentencia\* |  |  |
| sentencia\_while:sentencia → *condicion*:expr *sentencias*:sentencia\* |  |  |
| sentencia\_llamada\_funcion:sentencia → *nombre*:String *parametros*:expr\* | funciones [nombre] ≠ ∅ | funciones.definition = funciones[nombre] |
| sentencia\_return:sentencia → *expresion*:expr |  |  |
|  |  |  |
| tipoInt:tipo → λ |  |  |
| tipoFloat:tipo → λ |  |  |
| tipoChar:tipo → λ |  |  |
| tipoStruct:tipo → *string*:String | estructuras[nombre] ≠ ∅ | tipoVar.definicion = estructuras[nombre] |
| tipoArray:tipo → *tamanio*:String *tipo*:tipo |  |  |
|  |  |  |
| expr\_int:expr → *string*:String |  |  |
| expr\_real:expr → *string*:String |  |  |
| expr\_char:expr → *string*:String |  |  |
| expr\_ident:expr → *string*:String | variables[nombre]= ∅  variables\_funcion[nombre]≠ ∅  parametros[nombre]≠ ∅ | expr\_ident.definicion = variables[nombre]  expr\_ident.definicion = variables\_funcion [nombre]  expr\_ident.definicion = parametros [nombre] |
| expr\_binaria:expr → *izquierda*:expr *operador*:operador *derecha*:expr |  |  |
| expr\_vector:expr → *fuera*:expr *dentro*:expr |  |  |
| expr\_punto:expr → *izquierda*:expr *derecha*:expr |  |  |
| expr\_parentesis:expr → *expr*:expr |  |  |
| expr\_cast:expr → *tipo\_convertido*:tipo *expr*:expr |  |  |
| expr\_llamada\_funcion:expr → *nombre*:String *parametros*:expr\* | funciones [nombre] ≠ ∅ | funciones.definition = funciones[nombre] |
|  |  |  |
| operador\_aritmetico:operador → *string*:String |  |  |
| operador\_logico:operador → *string*:String |  |  |
| operador\_booleano:operador → *string*:String |  |  |
|  |  |  |

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

# Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoría Sintáctica** | **Nombre del atributo** | **Tipo Java** | **Heredado/Sintetizado** | **Descripción** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |