## Especificación de Código

|  |  |
| --- | --- |
| **Función de Código** | **Plantillas de Código** |
| run : Programa → Instruccion\* | run [[ Programa  →  bloques:Bloque\* ]] =  #SOURCE {file}  call main  HALT  define[[bloquesi]] |
|  |  |
| define : Bloque → Instruccion\* | define[[ funcionDef → name:String parametros:parametro\*  type:type variables:varDef\* sentencias:sentencia\*]] =  #func {name}  {name}:  Si ∑variablesi.tamaño>0  enter {∑variablesi.tamaño}  ejecuta[[sentenciasi]]  ret {type.tamaño}, {∑variablesi.type.tamaño }, {∑parametrosi.type.tamaño }  define[[**varDef** → name:String type:type ]] =    define[[**estructuraDef** → nombre:String atributos:atributo\* ]] = |
|  |  |
| ejecuta : Sentencia → Instruccion\* |  |
|  | ejecuta [[ assign  →  left:Expresion  right:Expresion ]] =  #LINE {end.line}  dir[[left]]  valor[[right]]  STORE |
|  | ejecuta [[ print  →  operacion:String value:Expresion ]] =  #LINE {end.line}  valor[[expr]]  OUT<value.type>  **Si** operacion==”println”  PUSHB 10  OUTB  **Si** operacion==”printsp”  PUSHB 32  OUTB  ejecuta[[**read**  →  *value*:expresion ]] =  #LINE {end.line}  dir[[value]]  IN<value.type>  STORE<value.type>  ejecuta[[**whileloop** → condicion:expresion sentencias:sentencia\* ]] =  while{contadorBucles} :  valor[[condicion]]  JZ fin{ contadorBucles }  ejecuta[[sentenciasi]]  JMP while{contadorBucles }  fin{contadorBucles }  ejecuta[[**ifDef**  →  *condicion*:expresion  *iftrue*:sentencia\*  *iffalse*:sentencia\* ]] =  valor[[condición]]  JZ iffalse{contadorBucles}  ejecuta[[iftruei]]  JMP fin{contadorBucles}  iffalse{contadorBucles}:  ejecuta[[iffalsei]]  fin{contadorBucles}:  ejecuta[[**returnSent**  →  *retorno*:expresion ]] =  valor[[retorno]]  ejecuta[[**llamadaS**  →  *invocacion*:llamada ]] =  valor[[invocacion.parametrosi]]  call {invocacion.nombre}  **Si** invocación.definition.type != null  POP<invocacion.definition.type> |
|  |  |
| valor: Expresion → Instruccion\* | valor[[variable → name:String ]] =  dir[[variable]]  LOAD<variable.type>    valor[[accesoStruct → left:expresion right:String]] =  dir[[left]]  PUSH {left.atributos[atributoi.name==right].address}  ADD  LOAD<right.type>    valor[[integral → valor:String ]] =  PUSH {valor}    valor[[real → valor:String ]] =  PUSHF {valor}    valor[[caracter → valor:String ]] =  PUSHB {valor}    valor[[accesoArray → var:expresion dimensiones:array ]] =  dir[[var]]  valor[[dimensiones.pos]]  PUSH {var.type.type.tamaño}  MUL  ADD  LOAD<var.type.type>    valor[[parentesis → contenido:expresion ]] =  valor[[contenido]]    valor[[casting → tipo:type contenido:expresion ]] =  valor[[contenido]]  {contenido.type.suffix}2{tipo.suffix}    valor[[aritmetica → left:expresion operando:String right:expresion ]] =  valor[[left]]  valor[[right]]  **si** operando == “+”  ADD<aritmetica.type>  **si** operando == “-”  SUB<aritmetica.type>  **si** operando == “\*”  MUL<aritmetica.type>  **si** operando == “/”  DIV<aritmetica.type>    valor[[comparativa → left:expresion operando:String right:expresion ]] =  valor[[left]]  valor[[right]]  **si** operando == “>”  GT<right.type>  **si** operando == “<”  LT<right.type>  **si** operando == “>=”  GE<right.type>  **si** operando == “<=”  LE< right.type>  **si** operando == “!=”  NE< right.type>  **si** operando == “==”  EQ< right.type>    valor[[logica → left:expresion operando:String right:expresion ]] =  **Si** operando != ‘!’  valor[[left]]  valor[[right]]  **Si** operando == “&&”  AND  **Si** operando == “||”  OR  **Si no**  valor[[right]]  NOT    valor[[llamadaE → invocacion:llamada ]] =  valor[[invocacion.parametrosi]]  call {invocacion.nombre} |
| dir: Expresion → Instruccion\* | dir[[variable → name:String ]] =  **Si** {!variable.definition.localInFunc} **AND** {!node.definition.isParameter}  PUSHA {variable.definition.address}  **Si no**  PUSHA BP  **Si** {node.definition.isParameter}  PUSH {variable.definition.parameter.address}  **Si no**  PUSH {variable.definition.address}  ADD  dir[[accesoStruct → left:expresion right:String]] =  dir[[left]]  PUSH {left.atributos[atributoi.name==right].address}  ADD  dir[[accesoArray → var:expresion dimensiones:array ]] =  dir[[var]]  valor[[dimensiones.pos]]  PUSH {var.type.type.tamaño}  MUL  ADD |

Tabla de Atributos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nodo/Categoría** | **Nombre** | **Tipo Java** | **H/S** | **Descripción** |
| VarDef | localInFunc | Boolean | Heredado | Especifica si la variable es una variable local de una función o no |