## Especificación de Código

|  |  |
| --- | --- |
| **Función de Código** | **Plantillas de Código** |
| run : Programa → Instruccion\* | run [[ Programa  →  definiciones:Definicion\* ]] =  define[[definicionesi]] |
|  |  |
| define : Definicion → Instruccion\* | define[[ DefinicionFuncion  →  nombre:String  parametros:DefinicionVariable\*   retorno:Tipo  locales:DefinicionVariable\*  sentencias:Sentencia\* ]] =  ejecuta[[sentenciasi]] |
|  |  |
| ejecuta : Sentencia → Instruccion\* | // Hacer solo Asignación y Write |
|  | ejecuta [[ Asignacion  →  left:Expresion  right:Expresion ]] =  #LINE {end.line}  dirección[[left]]  Valor[[right]]  STORE |
|  | ejecuta [[ Write  →  expr:Expresion ]] =  #LINE {end.line}  valor[[expr]]  OUT |
|  |  |
| valor: Expresion → Instruccion\* | Valor[[lintchar  →  string:String ]] =  PUSH {string}  Valor[[lintreal   →  string:String ]] =  PUSHF {string}  Valor[[lintent   →  string:String ]] =  PUSHB {string}  Valor[[var  →  string:String ]] =  Dirección[[var]]  LOAD  Valor[[expresionNumerica  →  izq:expr  string:String  der:expr ]] =  Valor[[izq]]  Valor[[der]]  Si string==’+’  ADD  Sino si string ==’-’  SUB  Sino si string ==’\*’  MUL  Sino si string ==’/’  DIV  Sino si string ==’<’  LT  Sino si string ==’>’  GT  Sino si string ==’==’  EQ  Sino si string ==’!=’  NE  Sino si string ==’>=’  GE  Sino si string ==’<=’  LE  Valor[[expresionLogica  →  izq:expr  string:String  der:expr ]] =  Valor[[izq]]  Valor[[der]]  Si string==’&&’  AND  Sino si string ==’||’  OR |
| dirección: **Expresion** -> Instrucción\* | dirección[[var  →  string:String ]] =  PUSHA {definición.direccion} |
|  |  |

NOTA: Lo que está en naranja es temporal para esta clase y habrá que ampliarlo en la siguiente.