|  |  |
| --- | --- |
| Función | Plantillas de Código |
| run[[program]] | run[[program  →  definicion:definicion\* ]] =  #SOURCE {file}  CALL main  HALT  define[[definicion]] |
|  |  |
| ƒ2[[parametros]] | |  |  | | --- | --- | |  | ƒ2[[parametros  →  nombre:String  tipo:tipo ]] = | |  |  | |
|  |  |
| ƒ3[[variableStruct]] | |  |  | | --- | --- | |  | ƒ3[[variableStruct  →  nombre:String  tipo:tipo ]] = |   push {variableStruct.address} |
|  |  |
| define[[definicion]] | ƒ4[[defVariable  →  tipo:tipo  nombre:String ambito:String ]] =  #GLOBAL {end.line} |
| ƒ4[[defStruct  →  nombre:String  definicion:definicion\* ]] =  #TYPE {end.line} |
| ƒ4[[defFuncion  →  nombre:String  parametros:parametros\*       tipo:tipo  defvariable:defVariable\*  sentencia:sentencia\* ]] =  {name}:  ENTER {defVariable.type.size}  ejecuta[[sentencia]]  IF tipo == null  RET 0, { ∑defVariable.tipo.size},{∑parametros.tipo.size} |
|  |  |
| ƒ5[[tipo]] | |  |  | | --- | --- | |  | ƒ5[[intTipo  → λ ]] = | |  |  | |
| |  |  | | --- | --- | |  | ƒ5[[realTipo  → λ ]] = | |
| ƒ5[[charTipo  → λ ]] = |
| |  |  | | --- | --- | |  | ƒ5[[arrayTipo  →  posicion:String  tipo:tipo ]] = | |
| ƒ5[[structTipo  →  nombre:String ]] = |
|  |  |
| ejecuta[[sentencia]] | ƒ6[[asignacion  →  izquierda:expresion  derecha:expresion ]] =  #LINE {end.line}  address[[izquierda]]  valor[[derecha]]  STORE |
| ƒ6[[print  →  print:expresion  printTipo:String ]] =  #LINE {end.line}  valor[[print]]  OUT |
| ƒ6[[read  →  read:expresion ]] =  address[[read]]  IN  STORE |
| ƒ6[[funcionLlamada  →  nombre:String  expresion:expresion\* ]] =  valor[[expression]]  CALL {name}  IF funcionLlamada.definicion.tipo != null  POP |
| ƒ6[[if  →  condicion:expresion  if\_true:sentencia\*  if\_false:sentencia\* ]] =  valor[[condicion]]  jz else  ejecuta[[if\_true]]  jmp fin\_else  else  ejecuta[[if\_false]]  fin\_else |
| ƒ6[[while  →  condicion:expresion  sentencia:sentencia\* ]] =  while  valor[[condicion]]  jz fin\_while  ejecuta[[sentencia]]  jmp while  fin\_while |
| ƒ6[[return  →  retorno:expresion ]] =  IF retorno == null  RET 0, {∑ funcion.defvariable.tipo.size}, {∑ funcion.parametros.tipo.size}  ELSE  RET {expresion.tipo.size}, {∑ funcion.defvariable.tipo.size}, {∑ funcion.parametros.tipo.size} |
|  |  |
| valor[[expresion]] | valor[[expresionAritmetica  →  izquierda:expresion  operador:String  derecha:expresion ]] =  valor[[derecha]]  valor[[izquierda]]  IF operador == ‘+’  ADD  IF operador == ‘-’  SUB  IF operador == ‘\*’  MUL  IF operador == ‘/’  DIV |
| valor[[expresionLogica  →  izquierda:expresion  operador:String  derecha:expresion ]] =  valor[[izquierda]]  valor[[derecha]]  IF operador == ‘<’  LT  IF operador == ‘>’  GT  IF operador == ‘<=’  LE  IF operador == ‘>=’  GE  IF operador == ‘!=’  NE  IF operador == ‘==’  EQ |
| valor[[expresionLogicaAndOr  →  izquierda:expresion  operador:String  derecha:expresion ]] =  valor[[izquierda]]  valor[[derecha]]  IF operador == ‘&&’  AND  IF operador == ‘||’  OR |
| valor[[expresionDistinto  →  not:expresion ]] =  valor[[not]]  NOT |
| valor[[variable  →  nombre:String ]] = |
| valor[[ident  →  valor:String ]] =  address[[ident]]  LOAD |
| valor[[litEnt  →  valor:String ]] = |
| valor[[litReal  →  valor:String ]] = |
| valor[[litChar  →  valor:String ]] = |
| valor[[cast  →  tipo:tipo  valor:expresion ]] =  valor[[valor]]  IF tipo == intTipo && expresion.tipo == charTipo  b2i  IF tipo == intTipo && expresion.tipo == realTipo  f2i  IF tipo == charTipo && expresion.tipo == intTipo  i2b  IF tipo == realTipo && expresion.tipo == intTipo  i2f |
| valor[[array  →  nombre:expresion  valor:expresion ]] =  address[[nombre]]  LOAD |
| valor[[struct  →  nombre:expresion  campos:String ]] =  address[[nombre]]  LOAD |
| valor[[expresionLlamada  →  nombre:String  expresion:expresion\* ]] =  valor[[expresión]]  CALL{nombre} |
|  |  |
| address[[expresion]] | address[[ ident → valor:String ]] =  IF definición.ambito == GLOBAL  PUSHA BP  PUSH {definición.address}  IF definición.ambito == PARAM  PUSH BP  PUSH {definición.parametros.address}  ADD  IF definicion.ambito == LOCAL  PUSHA {definicion{dir}} |
|  | address[[struct  →  nombre:expresion  campos:String ]] =  address[[nombre]]  PUSH {nombre.tipo.definicion.defVariable[campos].address}  ADD |
|  | address[[array  →  nombre:expresion  valor:expresion ]] =  address[[nombre]]  PUSH {nombre.tipo.size}  valor[[valor]]  MUL  ADD |
|  |  |