

Especificación de Código

Función	Plantillas de Código
run[[programa]]	run[[programa → <i>definicion:definicion*</i>]] = CALL main HALT define[[<i>definicion_i</i>]]
defineCuerpo[[cuerpo]]	defineCuerpo [[cuerpo → <i>defvariable:defVariable* sentencia:sentencia*</i>]] = ENTER { \sum defVariable.tipo.size } ejecuta[[<i>sentencia_i</i>]]
define[[definicion]]	define[[defVariable → <i>nombre:String tipo:tipo ambito:int</i>]] = <i>si { ambito } == GLOBAL {</i> <i> '#GLOBAL' { nombre } ':' { tipo }</i> <i>}</i> define [[defFuncion → <i>nombre:String parametros:defVariable*</i> <i>defReturn:defReturn cuerpo:cuerpo</i>]] = <i>'#FUNC' { nombre }</i> <i>'#PARAM' { parametros, nombre } ':' { parametros, tipo }</i> <i>'#RET' { defReturn.tipo }</i> <i>'#LOCAL' { cuerpo.defVariable, nombre } ':' { cuerpo.defVariable, tipo }</i> { nombre }: defineCuerpo[[<i>cuerpo</i>]] si defReturn == VOID RET 0, { \sum cuerpo.defVariable.tipo.size }, { \sum parametros.tipo.size } define [[defEstructura → <i>nombre:String defcampo:defCampo*</i>]] = <i>'#TYPE' { nombre } '{'</i> <i> { defCampo, nombre } ':' { defCampo, tipo }</i> <i>'}'</i>
ejecuta[[sentencia]]	ejecuta [[while → <i>condicion:expresion cierto:sentencia*</i>]] = while _x : valor[[<i>condicion</i>]] jz finWhile _x ejecuta[[<i>cierto_i</i>]] jmp while _x finWhile _x : ejecuta [[ifElse → <i>condicion:expresion cierto:sentencia* falso:sentencia*</i>]] = valor[[<i>condicion</i>]] jz else _x ejecuta[[<i>cierto_i</i>]]

```

    jmp finIfx
  elsex:
    ejecuta[[falsoi]]
  finIfx:

```

```

ejecuta [[return → expresion:expresion ]] =
  valor [[expresion]]
  RET expresion.tipo.size, {Σ return.definicion.cuerpo.defVariablei.tipo.size }, {Σ
  return.definicion.parametrosi.tipo.size }

```

```

ejecuta [[print → expresion:expresion ]] =
  valor [[expresion]]
  OUT

```

```

ejecuta [[printLN → expresion:expresion ]] =
  valor [[expresion]]
  OUT
  PUSHB 10
  OUTB

```

```

ejecuta [[printSP → expresion:expresion ]] =
  valor [[expresion]]
  OUT
  PUSHB 32
  OUTB

```

```

ejecuta [[read → expresion:expresion ]] =
  dirección [[expresion]]
  IN
  STORE

```

```

ejecuta [[asignacion → left:expresion right:expresion ]] =
  direccion [[left]]
  valor [[right]]
  STORE

```

```

ejecuta[[InvocacionProcedimiento → nombre:String argumentos:expresion* ]] =
  valor[[argumentosi]]
  CALL {nombre}
  Si InvocacionProcedimiento.definicion.defReturn.tipo ≠ VOID
    POP

```

valor[[**expresion**]]

```

  valor[[expresionAritmética → left:expresion operador:String right:expresion ]] =
    valor[[left]]
    valor [[right]]
    <operador>

```

```

  valor[[expresionComparacion → left:expresion operador:String right:expresion ]] =
    valor[[left]]
    valor [[right]]

```

	<operador>
	valor [[invocacionFuncion → <i>nombre:String argumentos:expresion*</i>]] = valor [[<i>argumentos</i>]] CALL { <i>nombre</i> }
	valor [[variable → <i>nombre:String</i>]] = direccion [[<i>nombre</i>]] LOAD< <i>variable.tipo</i> >
	valor [[literalEntero → <i>valor:String</i>]] = PUSH { <i>valor</i> }
	valor [[literalReal → <i>valor:String</i>]] = PUSHF { <i>valor</i> }
	valor [[literalChar → <i>valor:String</i>]] = PUSHB { <i>valor</i> }
	valor [[accesoArray → <i>identificacion:expresion posicion:expresion</i>]] = direccion [[<i>accesoArray</i>]] LOAD< <i>accesoArray.tipo</i> >
	valor [[cast → <i>tipo:tipo expresion:expresion</i>]] = valor [[<i>expresion</i>]] < <i>expresion.tipo</i> >2< <i>tipo</i> >
	valor [[accesoCampo → <i>expresion:expresion nombre:String</i>]] = direccion [[<i>accesoCampo</i>]] LOAD< <i>accesoCampotipo</i> >
	valor [[negacion → <i>expresion:expresion</i>]] = valor [[<i>expresion</i>]] NOT
	valor [[menosUnario → <i>expresion:expresion</i>]] = PUSHi 0 valor [[<i>expresion</i>]] SUB
direccion [[expresion]]	direccion [[expresionAritmética → <i>left:expresion operador:String right:expresion</i>]] = direccion [[expresionComparacion → <i>left:expresion operador:String right:expresion</i>]] = direccion [[invocacionFuncion → <i>nombre:String argumentos:expresion*</i>]] = direccion [[variable → <i>nombre:String</i>]] = si <i>variable.definicion.ambito</i> == GLOBAL PUSHA { <i>variable.definicion.direccion</i> } sino PUSH BP PUSH { <i>variable.definicion.direccion</i> } ADD

dirección [[**literalEntero** → *valor:String*]] =

dirección [[**literalReal** → *valor:String*]] =

dirección [[**literalChar** → *valor:String*]] =

dirección [[**accesoArray** → *identificacion:expresion posicion:expresion*]] =
 dirección[[*identificación*]]
 valor[[*posición*]]
 PUSH *identificación.tipo.tipo.size*
 MUL
 ADD

dirección [[**cast** → *tipo:tipo expresion:expresion*]] =

dirección [[**accesoCampo** → *expresion:expresion nombre:String*]] =
 dirección[[*expresión*]]
 PUSH { *expresión.definición.campos[nombre].dirección* }
 ADD

dirección [[**negacion** → *expresion:expresion*]] =

dirección [[**menosUnario** → *expresion:expresion*]] =