

# Code Specification

Función de Código	Plantillas de Código
run[[ <b>programa</b> ]]	run[[ <b>programa</b> → <i>definiciones: defVariable* sentencias: sentencia*</i> ]] = #SOURCE {ficheroEntrada} metadatos[[definiciones <sub>i</sub> ]] ejecuta[[sentencias <sub>i</sub> ]] HALT
metadatos[[ <b>defVariable</b> ]]	metadatos[[ <b>defVariable</b> → <i>tipo: tipo nombre: String</i> ]] = #GLOBAL {nombre}: {tipoMAPL(tipo)}
ejecuta[[ <b>sentencia</b> ]]	ejecuta[[ <b>print</b> → <i>expresion: expresion</i> ]] = #LINE {end.line} valor[[expresion]] OUT<expresion.tipo>  ejecuta[[ <b>asigna</b> → <i>left: expresion right: expresion</i> ]] = #LINE {end.line} direccion[[left]] valor[[right]] STORE<left.tipo>
valor[[ <b>expresion</b> ]]	valor[[ <b>exprAritmetica</b> → <i>left: expresion operador: String right: expresion</i> ]] = valor[[left]] valor[[right]] si operador == "+" ADD<exprAritmetica.tipo> si operador == "-" SUB<exprAritmetica.tipo> si operador == "*" MUL<exprAritmetica.tipo> si operador == "/" DIV<exprAritmetica.tipo>  valor[[ <b>variable</b> → <i>nombre: String</i> ]] = direccion[[variable]] LOAD<variable.tipo>  valor[[ <b>literalInt</b> → <i>valor: String</i> ]] = PUSH {valor}  valor[[ <b>literalReal</b> → <i>valor: String</i> ]] = PUSHF {valor}
direccion[[ <b>expresion</b> ]]	direccion[[ <b>variable</b> → <i>nombre: String</i> ]] = PUSHA {variable.definicion.direccion}

## Nota:

La notación *Instruccion*<expresión de tipo> representa a la versión adecuada de la instrucción para el tipo indicado.

## Ejemplos:

LOAD<int> → LOADI  
LOAD<real> → LOADF

**Metadatos:**

Todas las líneas con metadatos (prefijadas con el símbolo #) son opcionales (ver el tutorial de MAPL para más información).

**Funciones Auxiliares**

<b>Función Auxiliar</b>	<b>Definición</b>
tipoMAPL(tipo)	Si tipo == intType tipoMAPL = "int" Si tipo == realType tipoMAPL = "float"