```
init \rightarrow program
program → main
main → MAIN PAR_IZQ PAR_DER function_block
function block → BRA IZQ instructions BRA DER
instructions → instructions instruction
instruction \rightarrow if
if → IF PAR_IZQ expression PAR_DER function_block ELSE function_block
function block → BRA IZQ instructions BRA DER
instruction \rightarrow instruction
instruction → expression PUNTO_COMA
expression \rightarrow function_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression list \rightarrow expression
expression \rightarrow CADENA
function block → BRA IZQ instructions BRA DER
instructions \rightarrow instructions instruction
instruction → goto PUNTO COMA
goto \rightarrow GOTO ID
instructions \rightarrow instruction
instruction → expression PUNTO_COMA
expression → function_call
function_call \rightarrow ID param_val
param_val → PAR IZQ expression list PAR DER
expression list \rightarrow expression
expression \rightarrow CADENA
expression \rightarrow expression OR expression
expression → expression ES IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
expression → expression ES_IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
instructions \rightarrow instructions instruction
instruction → declaration PUNTO COMA
declaration → primitive_type declaration_list
declaration_list → ID IGUAL expression
expression → function_call
function_call \rightarrow ID param_val
param val → PAR IZQ PAR DER
primitive_type \rightarrow CHAR
instructions \rightarrow instructions instruction
instruction → expression PUNTO_COMA
expression → function_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression list \rightarrow expression
expression \rightarrow CADENA
instructions → instructions instruction
instruction \rightarrow if
if \rightarrow IF PAR IZQ expression PAR DER function block ELSE if
if → IF PAR_IZQ expression PAR_DER function_block ELSE if
if → IF PAR_IZQ expression PAR_DER function_block ELSE if
if \rightarrow IF PAR IZQ expression PAR DER function block
function_block → BRA_IZQ instructions BRA_DER
instructions \rightarrow instruction
instruction → expression PUNTO COMA
expression \rightarrow function_call
function_call \rightarrow ID param_val
param val → PAR IZQ expression list PAR DER
expression_list → expression_list COMA expression
expression → expression DIV expression
expression \rightarrow identifier
identifier \rightarrow ID
expression \rightarrow identifier
identifier \rightarrow ID
expression_list \rightarrow expression
expression \rightarrow CADENA
expression → expression ES_IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
function block → BRA IZQ instructions BRA DER
instruction \rightarrow instruction
instruction → expression PUNTO_COMA
expression \rightarrow function\_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression_list → expression_list COMA expression
expression \rightarrow expression MULT expression
expression \rightarrow identifier
identifier \rightarrow ID
expression \rightarrow identifier
identifier \rightarrow ID
expression list \rightarrow expression
expression \rightarrow CADENA
expression → expression ES IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
function_block → BRA_IZQ instructions BRA_DER
instructions → instruction
instruction \rightarrow expression PUNTO_COMA
expression → function_call
function_call \rightarrow ID param_val
param\_val \rightarrow PAR\_IZQ \ expression\_list \ PAR\_DER
expression_list → expression_list COMA expression
expression \rightarrow expression RESTA expression
expression \rightarrow identifier
identifier \rightarrow ID
expression \rightarrow identifier
identifier \rightarrow ID
expression_list \rightarrow expression
expression \rightarrow CADENA
expression → expression ES_IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
function block → BRA IZQ instructions BRA_DER
instructions \rightarrow instruction
instruction \rightarrow expression PUNTO_COMA
expression → function_call
function_call \rightarrow ID param_val
param\_val \rightarrow PAR\_IZQ \ expression\_list \ PAR\_DER
expression_list → expression_list COMA expression
expression \rightarrow expression SUMA expression
expression \rightarrow identifier
identifier \rightarrow ID
expression \rightarrow identifier
identifier \rightarrow ID
expression_list \rightarrow expression
expression \rightarrow CADENA
expression → expression ES_IGUAL expression
expression \rightarrow CADENA
expression \rightarrow identifier
identifier \rightarrow ID
instructions → instructions instruction
instruction → expression PUNTO_COMA
expression \rightarrow function_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression_list \rightarrow expression
expression \rightarrow CADENA
instructions → instructions instruction
instruction → declaration PUNTO_COMA
declaration → primitive type declaration list
declaration_list \rightarrow ID IGUAL expression
expression \rightarrow function call
function_call \rightarrow ID param_val
param_val → PAR_IZQ PAR_DER
primitive_type \rightarrow CHAR
instructions \rightarrow instructions instruction
instruction → expression PUNTO_COMA
expression \rightarrow function_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression_list \rightarrow expression
expression \rightarrow CADENA
instructions → instructions instruction
instruction \rightarrow expression PUNTO_COMA
expression → function_call
function_call \rightarrow ID param_val
param val → PAR IZQ expression list PAR DER
expression list \rightarrow expression
expression \rightarrow CADENA
instructions \rightarrow instructions instruction
instruction → expression PUNTO_COMA
expression → function_call
function call \rightarrow ID param val
param_val → PAR_IZQ expression_list PAR_DER
expression list \rightarrow expression
expression \rightarrow CADENA
instructions → instructions instruction
instruction → declaration PUNTO COMA
declaration → primitive_type declaration_list
declaration\_list \rightarrow ID IGUAL expression
expression → function_call
function_call → ID param val
param_val → PAR_IZQ PAR_DER
primitive type \rightarrow INT
instructions \rightarrow instructions instruction
instruction → expression PUNTO_COMA
expression → function_call
function call \rightarrow ID param val
param val → PAR IZQ expression list PAR DER
expression_list \rightarrow expression
expression \rightarrow CADENA
instructions → instructions instruction
instruction → expression PUNTO COMA
expression \rightarrow function_call
function_call \rightarrow ID param_val
param_val → PAR_IZQ expression_list PAR_DER
expression_list \rightarrow expression
expression \rightarrow CADENA
instructions → instruction instruction
instruction → declaration PUNTO COMA
declaration → primitive_type declaration_list
declaration_list → ID IGUAL expression
expression \rightarrow function_call
function call \rightarrow ID param val
param val \rightarrow PAR IZQ PAR DER
primitive type \rightarrow INT
instructions → instructions instruction
instruction → expression PUNTO COMA
expression \rightarrow function call
function_call \rightarrow ID param_val
param val \rightarrow PAR IZQ expression list PAR DER
expression list \rightarrow expression
expression \rightarrow CADENA
instruction \rightarrow instruction
instruction \rightarrow label
```

 $label \rightarrow ID DOS PUNTOS$