GRAMÁTICA

<INI> ::= <INSTRUCTIONSBODY> EOF

<INSTRUCTIONSBODY> ::= <INSTRUCTIONSBODY> <BODY>

| <BODY>

<BODY> ::= <DEC_VAR> ';' | <ASIG_VAR> ';' | <METHODS> | <MAIN>

| <DEC_STRUCT> | <SET_STRUCT> | error ';'

<METHODS> ::= Rvoid id '(' ')' '{' <INSTRUCTIONS> '}'

| Rvoid id '(' <PARAMS> ')' '{' <INSTRUCTIONS> '}'

| <TYPE> id '(' ')' '{' <INSTRUCTIONS> '}'

| <TYPE> id '(' <PARAMS> ')' '{' <INSTRUCTIONS> '}'

<PARAMS> ::= <PARAMS> ',' <PARAM> | <PARAM>

<PARAM> ::= <TYPE> id

<CALL> ::= id '(' ')' | id '(' <PARAMS_CALL> ')'

<MAIN> ::= Rmain id '(' ')' ';' | Rmain id '(' <PARAMS_CALL> ')' ';'

<PARAMS_CALL> ::= <PARAMS_CALL> ',' <EXPRESSION> | <EXPRESSION>

<DEC_VAR> ::= <TYPE> id | <TYPE> id same <EXPRESSION>

<ASIG_VAR> ::= id same <EXPRESSION> | id inc | id dec

<TYPE> ::= Rint | Rdouble | Rchar | Rboolean | Rstring

<INSTRUCTIONS> ::= <INSTRUCTIONS> <INSTRUCTION> | <INSTRUCTION>

<INSTRUCTION> ::= <DEC_VAR> | <ASIG_VAR> | <DEC_STRUCT>

| <SET_STRUCT> | <PRINT> | <IF> | <SWITCH> | <WHILE> | <FOR>

| <DO_WHILE> | <CALL> | Rbreak ';' | Rcontinue ';' | <RETURN> | error ';'

<PRINT> ::= Rprint '(' <EXPRESSION> ')' ';'

<IF> ::= Rif '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}'

| Rif '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}' Relse '{' <INSTRUCTIONS> '}'

| Rif '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}' <ELSEIF>

| Rif '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}' <ELSEIF> Relse '{' <INSTRUCTIONS> '}'

<ELSEIF> ::= <ELSEIF> <EIF> | <EIF>

<EIF> ::= Relse Rif '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}'

<SWITCH> ::= Rswitch '(' <EXPRESSION> ')' '{' <CASES> <DEFAULT> '}'

| Rswitch '(' <EXPRESSION> ')' '{' <CASES> '}'

| Rswitch '(' <EXPRESSION> ')' '{' <DEFAULT> '}'

<CASES> ::= <CASES> <CASE> | <CASE>

<CASE> ::= Rcase <EXPRESSION> ':' <INSTRUCTIONS>

<DEFAULT>: Rdefault ':' <INSTRUCTIONS>

<WHILE> ::= Rwhile '(' <EXPRESSION> ')' '{' <INSTRUCTIONS> '}'

<FOR> ::= Rfor '(' <DEC_VAR> ';' <EXPRESSION> ';' <ASIG_VAR> ')' '{'

<INSTRUCTIONS> '}'

| Rfor '(' <ASIG_VAR> ';' <EXPRESSION> ';' <ASIG_VAR> ')' '{' <INSTRUCTIONS> '}'

<DO_WHILE> ::= Rdo '{' <INSTRUCTIONS> '}' Rwhile '(' <EXPRESSION> ')' ';'

<RETURN> ::= Rreturn <EXPRESSION> ';' | Rreturn ';'

<DEC_STRUCT> ::= <TYPE> '[' ']' id same Rnew <TYPE> '[' <EXPRESSION> ']' ';'

| <TYPE> '[' ']' id same '{' <LIST_VALUES> '}' ';'

|Rlist '<' <TYPE> '>' id same Rnew Rlist '<' <TYPE> '>' ';'

|Rlist '<' <TYPE> '>' id same <EXPRESSION> ';'

<LIST_VALUES> ::= <LIST_VALUES> ',' <EXPRESSION> | <EXPRESSION>

<SET_STRUCT> ::= id '[' <EXPRESSION> ']' same <EXPRESSION> ';'

| id '.' Radd '(' <EXPRESSION> ')' ';'

| id '[[' <EXPRESSION> ']]' same <EXPRESSION> ';'

<EXPRESSION> ::= <EXPRESSION> '?' <EXPRESSION> ':' <EXPRESSION>

| <EXPRESSION> '+' <EXPRESSION>

| <EXPRESSION> '-' <EXPRESSION>

| <EXPRESSION> '*' <EXPRESSION>

| <EXPRESSION> '/' <EXPRESSION>

| <EXPRESSION> '^' <EXPRESSION>

| <EXPRESSION> '%' <EXPRESSION>

| <EXPRESSION> '<' <EXPRESSION>

| <EXPRESSION> '>' <EXPRESSION>

| <EXPRESSION> '<=' <EXPRESSION>

| <EXPRESSION> '>=' <EXPRESSION>

| <EXPRESSION> '==' <EXPRESSION>

| <EXPRESSION> '!=' <EXPRESSION>

| <EXPRESSION> '&&' <EXPRESSION>

| <EXPRESSION> '||' <EXPRESSION>

| '(' <TYPE> ')' <EXPRESSION>

| RtoLower '(' <EXPRESSION> ')'

| RtoUpper '(' <EXPRESSION> ')'

| Rlength '(' <EXPRESSION> ')'

| Rtruncate '(' <EXPRESSION> ')'

| Rround '(' <EXPRESSION> ')'

| Rtypeof '(' <EXPRESSION> ')'

| RtoString '(' <EXPRESSION> ')'

| RtoCharArray '(' <EXPRESSION> ')'

| '!' <EXPRESSION>

| '-' <EXPRESSION>

| '(' <EXPRESSION> ')'

| <CALL>

| id '[' <EXPRESSION> ']'

| id '[[' <EXPRESSION> ']]'

| double

| int

| Rtrue

| Rfalse

| string

| id

| char