Palabras: main { } ; ( ) , = if then else while return for to do ++ @ $ ! -- + - # ? \* / < <= > >= == and or xor % \*\* list of int double boolean char const\_int const\_double const\_boolean const\_char const\_list\_int const\_list\_double const\_list\_boolean const\_list\_char list\_int list\_double list\_boolean list\_char list\_id id << >>

Tokens: {main}, {{} {}} {;} {(} {)} {,} {=} {if} {then} {else} {while} {return} {for} {to} {do} {++} {@} {$} {!, #, ?} {+, -} {--} {\*, /, <, <=, >, >=, ==, and, or, xor, %, \*\*} {<<, >>} {list of} {int, double, boolean, char} {const\_int} {const\_double} {const\_boolean} {const\_char} {const\_list\_int} {const\_list\_double} {const\_list\_boolean} {const\_list\_char} {list\_int} {list\_double} {list\_boolean} {list\_char} {list\_id} {id}

|  |  |  |  |
| --- | --- | --- | --- |
| Token | Ident | Attribute | Regular expression |
| MAIN | 257 | 0:main | {“main”} |
| LEFT\_KEY | 258 | 0: { | {“{“} |
| RIGHT\_KEY | 259 | 0: } | {“}“} |
| SEMICOLON | 260 | 0: ; | {“;”} |
| LEFT\_PARENTHESIS | 261 | 0: ( | {“(“} |
| RIGHT\_PARENTHESIS | 262 | 0: ) | {“)”} |
| COMA | 263 | 0: , | {“,”} |
| EQUALS | 264 | 0: = | {“=”} |
| IF | 265 | 0: if | {“if”} |
| THEN | 266 | 0: then | {“then”} |
| ELSE | 267 | 0: else | {“else”} |
| WHILE | 268 | 0: while | {“while”} |
| RETURN | 269 | 0: return | {“return”} |
| FOR | 270 | 0: for | {“for”} |
| TO | 271 | 0: to | {“to”} |
| DO | 272 | 0: do | {“do”} |
| PLUSPLUS | 273 | 0: ++ | {“++”} |
| AT | 274 | 0: @ | {“@”} |
| DOLLAR | 275 | 0: $ | {“$”} |
| NEG\_COUNT\_QUESTION | 276 | 0: ! 1: # 2: ? | {“!” | “#” | “?”} |
| SYMBOL\_OP | 277 | 0: + 1: - | {“+” | “-”} |
| MINUS\_MINUS | 278 | 0: -- | {“--”} |
| BINARY\_OP | 279 | 0:\* 1:/ 2:< 3:<= 4:> 5:>= 6:== 7:and 8:or 9:xor 10:% 11:\*\* | {“\*” | “/” | “<” “<=” “>” | “>=” | “==” | “and” | “or” | “xor” | “%” | “\*\*”} |
| LIST\_OP | 280 | 0: << 1: >> | {“<<” | “>>”} |
| LIST\_OF | 281 | 0: list of | {“list of”} |
| BASIC\_TYPES | 282 | 0:int 1:double 2:boolean 3:char | {“int” | “double” | “boolean” | “char”} |
| CONST\_INT | 283 | 0: const\_int | [0-9]+ |
| CONST\_DOUBLE | 284 | 0: const\_double | [0-9]\*.[0-9]+ |
| CONST\_BOOLEAN | 285 | 0: const\_boolean | {“true”|”false”} |
| CONST\_CHAR | 286 | 0: const\_char | \”[^\“]\” |
| ID | 287 | 0: id | {“id”} |
| CADENA | 288 | 0: ascii | \”[^\“]\*\” |

/\*----- Sección de Declaraciones --------------\*/

%{

%}

MAIN main  
LEFT\_KEY {  
RIGHT\_KEY }  
SEMICOLON ;  
LEFT\_PARENTHESIS \(  
RIGHT\_PARENTHESIS \)  
COMA ,  
EQUALS =  
IF if  
THEN then  
ELSE else  
WHILE while  
RETURN return  
FOR for  
TO to  
DO do  
PLUSPLUS ++  
AT @  
DOLLAR $  
NEG\_COUNT\_QUESTION !|#|\?  
SYMBOL\_OP \+|\-  
MINUS\_MINUS --  
BINARY\_OP \\*|\/|<|<=|>|>=|==|and|or|xor|%|\\*\\*  
LIST\_OP <<|>>  
LIST\_OF list of  
BASIC\_TYPES int|double|boolean|char  
CONST\_INT [0-9]+  
CONST\_DOUBLE [0-9]\*.[0-9]+  
CONST\_BOOLEAN true|false  
CONST\_CHAR \”[^\“]\”  
ID id  
CADENA \”[^\“]\*\”

%%

/\*----- Sección de Reglas ----------------\*/

%%

/\*----- Sección de Procedimientos --------\*/