|  |  |
| --- | --- |
| Token | Pattern |
| start | “start” |
| id | ([A-Z]|[a-z])( [A-Z]|[a-z]|\-|\\_|[0-9])\* |
| ; | “;” |
| { | “{“ |
| } | “}” |
| , | “,” |
| . | “.” |
| : | “:” |
| < | “<” |
| > | “>” |
| class | “class” |
| print | “print” |
| cte.string | \” ([^\”] | \\\”)\* \” |
| ( | “(“ |
| ) | “)” |
| + | “+” |
| - | “-“ |
| \* | “\*” |
| / | “/” |
| var | “var” |
| for | “for” |
| function | “function” |
| cte.i | [0-9]+ |
| cte.f | [0-9]+ \.? [0-9]\* |

A01657055 A00821805

Sofia Recinos Ulrik Alberto Nuño Tapia

TAREA 3.1

**Expresiones regulares:**

**Diagramas:**

**Diagrama

Descripción generada automáticamente**

**Gramática:**

PROGRAMA-> start PROGVARS

PROGVARS-> PROGVAR PROGVARS | ε

PROGVAR-> DECFUNCION | INSTRUCCION | CLASE

INSTRUCCION-> VARS | CONDICION | USOFUNCION | LOOP | BLOQUE | RETURN | ASIGNACION

CLASE-> class id BLOQUEC

BLOQUEC -> { CBLOQUEC }

CBLOQUEC -> CBLOQUEC2 CBLOQUEC | ε

CBLOQUEC2 -> DECFUNCION | VARS

VARS-> TIPO id ARR IDS ;

ARR -> [EXPRESION] ARR | ε

IDS-> , id ARR IDS| ε

RETURN-> return EXPRESION ;

DECFUNCION-> function id ( FVARS ) BLOQUE

USOFUNCION-> function id ( FVARSU ) ;

FVARS -> id ARR IDS| ε

FVARSU -> EXPRESION FVARSUS | ε

FVARSUS -> , EXPRESION FVARSUS | ε

LOOP-> for ( ASIGNACION EXPRESION ; EXPRESION ) BLOQUE

BLOQUE-> { INSTRUCCIONES }

INSTRUCCIONES-> INSTRUCCION INSTRUCCIONES | ε

ASIGNACION-> id = EXPRESION ;

ESCRITURA-> print ( P\_ESCRITURA M\_ESCRITURA ) ;

P\_ESCRITURA-> EXPRESION | cte.string

M\_SCRITURA -> , P\_ESCRITURA M\_ESCRITURA | ε

EXPRESION-> EXP POSEXP

POSEXP-> SYMEXP EXP | ε

SYMEXP-> > | < MAY

MAY-> > | ε

EXP-> TERMINO MEXP

MEXP-> SUMRES EXP | ε

SUMRES-> + | -

TERMINO-> FACTOR MTERMINO

MTERMINO-> MULTDIV TERMINO | ε

MULTDIV-> \* | /

FACTOR-> ( EXPRESION ) | SUMRESVAC VAR\_CTE

SUMRESVAC-> SUMRES | ε

CONDICION-> if ( EXPRESION ) BLOQUE ELSE

ELSE-> else BLOQUE | ε

VAR\_CTE-> USOID | cte.i | cte.f | USOFUNCION

USOID -> id USOID2

USOID2 -> USOID3 |ARR

USOID3 -> . id USOID3 USOID2

USOID4 -> ( FVARSU ) | ε

TIPO -> INT | FLOAT | CHAR | STRING | BOOL