

Práctica 2:

Analizador Léxico de Matrículas

Para esta práctica sólo hemos tenido en cuenta el sistema de matrículas general utilizado dentro de la Unión Europea y de España, es decir, en nuestro dominio entran:

- Matrículas europeas de cuatriciclos a motor (coches).
- Matrículas de vehículos de los cuerpos de seguridad del Estado.
- Ciclomotores.
- Matrículas de vehículos diplomáticos.
- Remolques.
- Cuatriciclos a motor (Quads).
- Matrículas provinciales.
- Matrículas provisionales.
- Matrículas turísticas.
- Matrículas históricas.

Para implementar el analizador léxico hemos utilizado flex, dividiendo el programa en tres partes: Declaración de variables y funciones en C, declaración de variables y analizadores en flex y la función main de C para arrancar nuestro programa.

Primera parte

```
analizadorLexico.l
1  %{
2  /* se necesita esto para la llamada a atof() más abajo */
3  #include <math.h>
4  #include <stdio.h>
5  #include <string.h>
6
7  int matriculas = 0;
8
9  int compruebaSintaxis(char * matricula) {
10     int correcto = 0;
11
12     if (matricula[strlen(matricula) - 1] != '\0')
13         matricula[strlen(matricula) - 1] = '\0';
14
15     if (matricula[-1] == ' ' || matricula[-1] == '\b' || matricula[-1] == '"') {
16         correcto = 1;
17         matriculas++;
18     }
19
20     return correcto;
21 }
22
23 %}
```

Segunda parte

```
25  /*[^A-Za-z0-9]? */
26  DIGITOS    [0-9]{4}
27  CONSONANTESEUROPEAS [B-DF-HJ-NPR-TV-Z]{3}
28  CONSONANTESDIPLOMATICAS ("CD"|"CC"|"TA"|OI)
29  FINALFUERZASDELESTADO [B-DF-HJ-NPR-TV-Z]
30  CABECERAFUERZASDELESTADO ("PGC"|"CNP"|"ET"|"EA"|"FN"|"E"|"CGPC")
31  CONSONANTESPROVISIONALES ("P"|"S"|"B")
32  ID         [a-z][a-z0-9]*
33  FIN_PALABRA ([^A-Za-z0-9]|\0)
34  FINAL      {DIGITOS}[ -]*{CONSONANTESEUROPEAS}{FIN_PALABRA}
35
36  PROVINCIAL [A-Z][A-Z]?{DIGITOS}[A-Z]{2}{FIN_PALABRA}
37  EUROPEAS  (\b)?{FINAL}
38  FUERZAESTADO (\b)?{CABECERAFUERZASDELESTADO}[ -]*{DIGITOS}[ -]*{FINALFUERZASDELESTADO}{FIN_PALABRA}
39  CICLOMOTOR (\b)?"C"[ -]*{FINAL}
40  DIPLOMATICAS (\b)?{CONSONANTESDIPLOMATICAS}[ -]*[0-9]{2}[ -]*[0-9]{3}{FIN_PALABRA}
41  REMOLQUES  (\b)?"R"[ -]*{FINAL}
42  QUADS      (\b)?"E"[ -]*{FINAL}
43  PROVISIONAL (\b)?{CONSONANTESPROVISIONALES}[ -]*{FINAL}
44  TURISTICAS (\b)?"T"[ -]*{FINAL}
45  HISTORICO  (\b)?"H"[ -]*{FINAL}
```

```
45  {EUROPEAS} {
46      char * x ;
47      x = yytext;
48
49      if(compruebaSintaxis(x) == 1)
50          printf("MATRICULA EUROPEA MODERNA: %s\n", x);
51  }
52
53  {FUERZAESTADO} {
54      char * x ;
55      x = yytext;
56
57      if(compruebaSintaxis(x) == 1)
58          printf("MATRICULA FUERZAS DEL ESTADO: %s\n", x);
59  }
60
61  {CICLOMOTOR} {
62      char * x ;
63      x = yytext;
64
65      if(compruebaSintaxis(x) == 1)
66          printf("MATRICULA CICLOMOTOR: %s\n", x);
67  }
68
69  {DIPLOMATICAS} {
70      char * x ;
71      x = yytext;
72
73      if(compruebaSintaxis(x) == 1)
74          printf("MATRICULA DIPLOMATICA: %s\n", x);
75  }
```

```

77 {REMOLQUES} {
78     char * x ;
79     x = yytext;
80
81     if(compruebaSintaxis(x) == 1)
82     |         printf("MATRICULA REMOLQUES: %s\n", x);
83 }
84
85 {QUADS} {
86     char * x ;
87     x = yytext;
88
89     if(compruebaSintaxis(x) == 1)
90     |         printf("MATRICULA QUAD: %s\n", x);
91 }
92 }
93
94 {PROVINCIAL} {
95     char * x ;
96     x = yytext;
97
98     if(compruebaSintaxis(x) == 1)
99     |         printf("MATRICULA PROVINCIAL: %s\n", x);
00 }
01
02 {PROVISIONAL} {
03     char * x ;
04     x = yytext;
05
06     if(compruebaSintaxis(x) == 1)
07     |         printf("MATRICULA PROVISIONAL: %s\n", x);
08 }

```

```

109
110 {TURISTICAS} {
111     char * x ;
112     x = yytext;
113
114     if(compruebaSintaxis(x) == 1)
115     |         printf("MATRICULA TURISTICA: %s\n", x);
116 }
117
118 {HISTORICO} {
119     char * x ;
120     x = yytext;
121
122     if(compruebaSintaxis(x) == 1)
123     |         printf("MATRICULA HISTORICO: %s\n", x);
124 }
125 . {}
126 %%
127

```

Tercera parte

```
132  int main( int argc, char ** argv ) {
133      ++argv, --argc; /* se salta el nombre del programa */
134
135      if ( argc > 0 )
136          yyin = fopen( argv[0], "r" );
137      else
138          yyin = stdin;
139      yylex();
140
141      printf("Encontre %i matriculas\n", matriculas);
142  }
```

También hemos llamado un fichero "hola.txt" con algunos ejemplos de matrículas en un texto para comprobar en tiempo real la eficiencia de nuestro programa. Hemos tenido que retocar varios sectores en la declaración de variables de flex para que leyera las matrículas íntegras en el texto de manera que no cogiese, por ejemplo, de una palabra 2002 CHRISTIAN, la matrícula 2002CHR, que podría ser válida dentro de "matrículas europeas".

Realizado por:

- Javier González Peregrín.
- Ismael Tengo Rodríguez.
- Ismael Díaz Díaz.