|  |  |  |
| --- | --- | --- |
| **2ª Fase**  **Analizador Léxico con FLEX** | 2010 | |
|  | |  |

**Autores**

Pablo Eduardo Ojeda Vasco

Roberto Marco Sánchez

Joaquín Ocón Ojeda

# A Índice

# 1 Expresiones regulares del lenguaje ………………………... *Página 3*

# 1 Expresiones regulares del lenguaje

* **digito:**  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
* **digito-decimal:** 1 | .. | 9
* **entero:** digito{digito}
* **real:** entero . [entero] [(e|E) [+ | -] entero] | . entero [(e|E) [+ | -] entero] |

entero (e|E) [+ | -] entero

* **letra:** a | ... | z | A | ... | Z
* **ascii:** digito | letra | ¿ | ¡ | # | : | \* | / | ...
* **caracter:** ' ascii ' | ' carácter-esc '
* **carácter-esc:**  \n | \t | \” | \r | \’ | \digito-octal [digito-octal] [digito-octal] |

\x digito-hexadecimal [digito-hexadecimal]

* **ristra:** “ {ascii | carácter-esc} “
* **identificador:** letra {letra | digito}
* **comentarios:** // {ascii}\r | /\* {ascii - (/\* | \*/) } \*/
* **palabras reservadas:** main | function | procedure | return | if | else | for | while | switch | case | default | break | char | scanf | printf | int | float | const
* **separadores:** = | , | ; | ( | )
* **operadores unarios:** + | - | ~ | !
* **operadores binarios**: + | - | \* | / | && | || | < | > | <= | >= | == | & | ‘|’ | % | != | ^ | << | >>

# 2 Analizador Léxico con Flex

Un analizador léxico es un programa que reconoce patrones en un texto. Pueden ser escritos a mano o generados automáticamente por un generador de analizador léxico a partir de la descripción de los patrones que se tienen que reconocer. Las descripciones se muestran en forma de expresiones regulares.

Lex es un generador de analizadores léxicos desarrollado por M. E. Lesk y E. Schmidt (AT&T Bell Laboratories). La entrada de Lex es un fichero que contiene tokens definidos a través de expresiones regulares.

Lex crea un fichero conteniendo la función yylex(), la cual devuelve un entero denotando el token reconocido.

*C and scanner declarations*

*%%*

*Token definitions and actions*

*%%*

*C subroutines*

La primera sección del archivo contiene la declaración en C con inclusión de ficheros. En ella también se detallan definiciones usadas mediante expresiones regulares.

%{

#include ”Simple.tab.h” /\* The tokens \*/

%}

DIGIT [0-9]

ID [a-z][a-z0-9]\*

%%

Token definitions and actions

%% C subroutines