

# Processamento de Linguagens I

## LESI + LMCC (3º ano)

2ª Ficha Prática

Ano Lectivo de 01/02

### 1 Objectivos

Esta ficha prática contém exercícios para serem resolvidos nas aulas teórico-práticas com vista a sedimentar os conhecimentos relativos a:

- Sedimentar a utilização do lex;
- Utilização do lex para implementação de autómatos complexos (a dois níveis);
- Introdução aos parsers Recursivos Descendentes;

### 2 Enunciados

#### Somador de números

Construa um programa em Lex que some todos os números dum texto e que imprima a sua soma sempre que detectar o carácter '+' ou o carácter '='.

#### Comprimento de palavras e comentários

Considerando que uma palavra é definida da seguinte forma:

```
palavra = digito* letra (letra|digito|-)*
```

construa um programa Lex que ignorando comentários (todas as palavras entre "/\*" e "\*/") apresente o comprimento de todas as palavras.

#### Palavras, Números e número de linha

Construa um programa em Lex que apresente todas as palavras e números do texto fonte, indicando o número da linha onde se encontram (numa segunda versão indique também em que coluna da linha é iniciada a palavra ou o número).

#### Aprovados

Considere um texto com o seguinte formato:

```
Nome: Paulo Anacleto; Num: 4567; Nota: 15  
Nome: Maria Palmira; Num: 10635; Nota: 9  
...
```

Construa um programa em lex que imprima o nome das pessoas com nota maior ou igual a 10.

## O programador da iluminação

Um programador para controlo dum sistema de iluminação de jardim, depois de ligado à corrente, pode ser de novo desligado, ou então aceita dois comandos: `manual` ou `automatico`.

Em modo `manual` pode ser colocado em `on` (as luzes ficam a partir daí e até novo comando ligadas), ou em `off` (as luzes são desligadas e assim permanecem).

Em modo `automático` requer, ainda, mais duas indicações, a hora (0 a 24) a que deve ser ligado e o número de horas que deve permanecer ligado; fica, então, em `stand-by` até que receba um sinal para acender as luzes, mantendo as luzes acesas até ter recebido um número de impulsos de relógio igual ao número de horas para que foi programado.

Considerando que o sistema a modelar pode ser encarado como uma máquina de Transição de Estados, desenhe o *autómato determinista reactivo* que modela o sistema e utilize a sua imaginação para o implementar em C com a ajuda do Lex.

## Listas de Elementos

Considere a seguinte gramática para a qual se pretende escrever um *parser* recursivo descendente:

```
S          --> '<' Lista '>'
Lista     --> &
Lista     --> Elemento Lista
Elemento  --> dig
Elemento  --> pal
```

Utilize o Lex para implementar o respectivo analisador léxico.