

# Processamento de Linguagens - TP1a

Carlos Pereira (A61887)

João Barreira (A73831)

Rafael Costa (A61799)

Março 2017

## 1 Processador de transações da Via Verde

### 1.0.1 Expressões regulares

### 1.0.2 Ações semânticas

### 1.0.3 Estruturas de dados globais

### 1.0.4 Filtro de texto

## 2 Álbum fotográfico em HTML

## 3 Autores musicais

Existe uma diretoria com vários ficheiros de extensão *'lyr'*, que contêm a letra de canções precedida de 2 ou mais campos de meta-informação (1 por linha). Esta informação pode ser: título da canção, autor da letra, cantor, etc. A letra da música e a meta-informação estão separadas por uma linha em branco.

Foi feito um Processador de Texto com o auxílio do *GAWK* para ler todos os ficheiros *'lyr'* e obter informações sobre os cantores, autores e títulos de canções.

### 3.1 Total de *cantores* e a lista com os nomes

Aqui foi feito um processador que recebe ficheiros de letra de música como *input* e escreve o total e os nomes de todos os cantores.

#### 3.1.1 Expressões regulares

Após a análise de alguns possíveis ficheiros de entrada, verificou-se que as linhas de informações relativas aos nomes dos cantores começam por *"singer:"*,

onde os nomes estão separados por espaços em branco, dois pontos, vírgulas ou por pontos e vírgulas, daí os termos utilizado como *Field Separator*.

A informação que queremos retirar está em linhas que começam por "*singer:*".

### 3.1.2 Ações semânticas

Sempre que o processador encontra um registo iniciado por "*singer:*", percorrem-se todos os campos desse registo. Para cada um destes campos, retiram-se os caracteres desnecessários. Depois, caso o registo não exista no *array* dos nomes dos cantores, é adicionado a esse *array* e o contador de todos os cantores é incrementado.

No fim, antes de se imprimirem os resultados, o *array* com os nomes dos cantores é ordenado.

### 3.1.3 Estruturas de dados globais

Foram utilizadas duas estruturas de dados globais: um *array singers*, em que o índice são os nomes de todos os cantores e uma variável *total* que é utilizada como contador de cantores.

### 3.1.4 Filtro de texto

```
BEGIN {
    FS = " *[:;,] *"
}

/singer:/ {
    for (i = 2; i <= NF; i++) {
        sub("[ ?()]+$", "", $i);

        if (!($i in singers) && ($i != "")) {
            count++;
            singers[$i] = $i;
        }
    }
}

END {
    n = asort(singers);

    for (i = 1; i <= n; i++) {
        print singers[i];
    }

    print "Total: " count;
```

```
}
```

## 3.2 Todas as canções do mesmo *autor*

Fez-se um processador que recebe ficheiros de letra de música como *input* e calcula o número de canções de cada autor.

### 3.2.1 Expressões regulares

Verificou-se que as linhas de informações relativas aos nomes dos autores começam por "*author:*". Tal como no caso dos cantores, os nomes estão separados por espaços em branco, dois pontos, vírgulas ou por pontos e vírgulas, daí os termos utilizado como *Field Separator*.

As informações que pretendemos são retiradas de registos que começam por "*author:*".

### 3.2.2 Ações semânticas

Ao encontrar um registo iniciado por "*author:*", o processador percorre todos os campos desse registo. Para cada um destes campos, retiram-se os caracteres desnecessários. Depois, caso o registo não seja vazio, ou seja, conhecem-se os autores da música, a posição correspondente ao nome do autor no *array* de autores é incrementado. Caso contrário, incrementa-se a posição correspondente ao "Autor Desconhecido".

No fim, imprimem-se os resultados da seguinte forma: nome do autor seguido do número de músicas associadas a ele.

### 3.2.3 Estruturas de dados globais

Temos como estrutura global um *array authors*, em que o índice é o nome de um autor e o valor correspondente ao número de músicas associado a esse autor.

### 3.2.4 Filtro de texto

```
BEGIN {  
    FS = " *[:; ,] *"  
}  
  
/author:/ {  
    for (i = 2; i <= NF; i++) {  
        sub("[ ?()\\t]+$", "", $i);  
  
        if ($i != "") {  
            songs[$i]++;  
        }  
    }  
}
```

```

        else {
            songs["Autor desconhecido"]++;
        }
    }
}

END {
    for (i in songs) {
        print i " - " songs[i];
    }
}

```

### 3.3 Escrever o nome de cada *autor* seguido do título das suas canções

O processador recebe ficheiros de letra de música como *input* e escreve, para cada autor, o título das suas canções.

#### 3.3.1 Expressões regulares

Aqui, as linhas de informação relativas ao nome de uma canção começam por *"title:"* e as dos nomes dos autores começam por *"author:"*.

Da mesma maneira que se fez nos exercícios anteriores, os nomes estão separados por espaços em branco, dois pontos, vírgulas ou por pontos e vírgulas, daí os termos utilizado como *Field Separator*.

As informações que pretendemos são retiradas de registos que começam por *"title:"* e por *"author:"*.

#### 3.3.2 Ações semânticas

O processador, a cada registo que comece por *"title:"*, atribui esse campo à variável global *song*. Aqui fica guardado o título da canção. Depois retira os caracteres que são considerados desnecessários.

Aos registos que sejam iniciados em *author:*, começa a percorrê-lo a partir do seu segundo campos (visto que ser *author:*). Cada um destes campos é o nome de um autor. Depois, a cada um destes que não seja vazio, retiram-se os caracteres desnecessários e é adicionado à variável global *songs*, na linha correspondente ao nome do autor, o conteúdo na variável *song* que se traduz no título da canção.

Caso não se tenha o nome do cantor, ou seja, o campo é vazio, adiciona-se o título da canção à linha do "Autor Desconhecido".

No fim, para cada uma das linhas da matriz *songs*, imprime-se para cada linha, o seu índice e os conteúdos de cada uma das suas colunas.

### 3.3.3 Estruturas de dados globais

Utilizaram-se 2 variáveis globais: a variável *song* que guarda o título de uma canção e uma matriz *songs*. Nesta matriz, as linhas correspondem a nome de autores e as respectivas colunas correspondem aos títulos das canções associadas ao autor dessa linha.

### 3.3.4 Filtro de texto

```
BEGIN {
    FS = " *[:;,] *"
}

/title: / {
    song = $2;
    sub("\\(\\?\\)", "", song);
    sub("^[ *]=+", "", song);
}

/author:/ {
    for (i = 2; i <= NF; i++) {
        sub("[ ?()\\t]+$", "", $i);

        if ($i != "") {
            authors[$i][song] = song;
        }
        else {
            authors["Autor desconhecido"][song] = song;
        }
    }
}

END {
    for (a in authors) {
        printf("%s: ", a);

        flag = 0;

        for (j in authors[a]) {
            if (flag == 0) {
                printf("%s", j);
            }
            else {
                printf(", %s", j);
            }
        }
    }
}
```

```
        flag++;  
    }  
  
    printf("\n");  
}  
}
```

## 4 Dicionauro

4.0.5 Expressões regulares

4.0.6 Ações semânticas

4.0.7 Estruturas de dados globais

4.0.8 Filtro de texto