

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

3º ANO, 2º SEMESTRE, 2018/2019

Processamento de Linguagens

Trabalho Prático - GAWK



Ana Pereira
A81712



Maria Dias
A81611

April 28, 2019

Índice

1	Introdução	2
2	Especificação do Problema	3
2.1	Enunciado - Processador de Processos de Formação	3
2.2	Descrição do Problema	3
2.2.1	Alínea a)	3
2.2.2	Alínea b)	3
2.2.3	Alínea c)	4
2.2.4	Alínea d)	4
2.2.5	Extra: Contar o número de processos de cada instituição . . .	4
3	Desenho e Implementação da Solução	5
3.1	Limpeza do ficheiro	5
3.2	Registos com código numérico	5
3.3	Contagem dos tipos de processos	6
3.4	Grafo para ações de formação	6
3.5	Extra: Contagem de processos associados a instituições	7
4	Resultados Obtidos	8
4.1	Alínea a)	8
4.2	Alínea b)	9
4.3	Alínea c)	9
4.4	Alínea d)	10
4.5	Alínea extra	12
5	Guia de Utilização	13
6	Conclusão	14

1 Introdução

No âmbito da Unidade Curricular de Processamento de Linguagens, foi proposto o desenvolvimento de filtros de texto em C utilizando a ferramenta **GAWK**. Para tal, aplicamos os conhecimentos adquiridos nas aulas da unidade curricular sobre gramática e expressões regulares.

Neste relatório é apresentado o enunciado do problema, bem como todas as escolhas efetuadas pelo grupo para a formulação da solução do problema proposto. De seguida, são expostos alguns exemplos de utilização dos filtros implementados.

De acordo com o critério de atribuição de enunciados, coube ao nosso grupo resolver o enunciado número 2, **Processador de Processos de Formação**.

2 Especificação do Problema

2.1 Enunciado - Processador de Processos de Formação

Dado o ficheiro `natura.di.uminho.pt/~jj/pl-19/TP2/formacao.csv`, contendo informação detalhada sobre a criação e gestão de cursos de formação profissional, pretende-se construir um ou mais programas **Awk** que permitam:

1. Limpar o ficheiro dado (criar um segundo ficheiro pre-processado) retirando linhas vazias extra e linhas cujos campos estão todos vazios; sempre que o 'Estado' esteja vazio, esse campo deve tomar o valor 'NIL'. Este ficheiro pré-processado deve ser usado nas alíneas seguintes!
2. Contar o número de registos que apresentam um Código numérico e mostrar para esses, num formato legível, o dito código da ação de formação, o seu título, descrição e notas.
3. Identificar os tipos diferentes e calcular o número de processos por tipo.
4. Desenhar um grafo (em DOT) que relacione cada ação de formação (identificada pelo seu código) com todos os Diplomas jurídico-administrativos usados.

2.2 Descrição do Problema

O ficheiro fornecido, `formacao.csv`, contém uma tabela com registos de processos intervalados por linhas vazias e, no final, com um conjunto de linhas, também elas sem conteúdo relevante.

Assim, é necessário implementar filtros de texto de modo a obter uma tabela "limpa" de todo o conteúdo que possa ser dispensado. Depois disso, teremos um ficheiro que podemos mais facilmente manipular, de forma a extrair informações como o número de registos que apresentam código numérico e as informações relevantes relativas a esses mesmos registos, ou até representar informações pertinentes em grafos.

A apresentação dos dados será efetuada no formato `.csv`, uma vez que o grupo considera que este formato pode ser facilmente visualizado e interpretado, não havendo assim necessidade de o traduzir para um formato diferente.

2.2.1 Alínea a)

Este exercício tem como objetivo eliminar todas as linhas vazias contidas no ficheiro original, e inserir "NIL" no campo **Estado** sempre que este estiver vazio. O ficheiro resultante desta alínea passa a ser o ficheiro a ser processado nas seguintes.

2.2.2 Alínea b)

Nesta alínea pretende-se identificar todos os registos em que exista o campo **Código** e, para cada um desses, apresentar os campos relativos ao código da ação de formação, título, descrição e notas. Deste exercício podemos considerar que resulta uma versão

condensada do ficheiro relativo, com apenas as informações mais relevantes acerca de cada processo.

2.2.3 Alínea c)

Para cada linha do ficheiro fornecido, pretende-se retirar da mesma o valor do campo **Tipo**, fazendo uma contagem de processos para cada um dos tipos identificados, e apresentando-os num formato legível.

2.2.4 Alínea d)

Nesta alínea pretende-se desenhar um grafo que relacione cada código de processo com os diferentes diplomas jurídico-administrativos usados nesse processo. Para isso, será necessário, em cada processo, consultar o campo relativo a estes diplomas, e representá-los, um a um, no formato correto para ser representado num ficheiro .dot. O formato usado para definir o grafo será o seguinte:

```
graph {  
  "codigo_processo" -- "diploma_juridico";  
  "codigo_processo" -- "diploma_juridico";  
  ...  
}
```

2.2.5 Extra: Contar o número de processos de cada instituição

Esta alínea consiste em verificar o campo que identifica os detentores de cada processo de formação, contando o número total de processos possuídos pelas várias instituições.

3 Desenho e Implementação da Solução

3.1 Limpeza do ficheiro

Ao realizar esta primeira alínea do trabalho, o grupo notou logo à partida algumas anomalias no ficheiro apresentado, sendo que não tínhamos o caso típico de um registo por linha, como estávamos habituados. A principal preocupação passou, então, por tentar identificar o separador de registos e o separador de campos. Para identificar as diferentes colunas das tabelas, embora as mesmas sejam efetivamente separadas pelo caractere `;`, não poderia servir como separador de campos por si só, uma vez que, entre campos, existem muitas vezes quebras de linha. Assim sendo, precisamos também de definir o que era, no nosso ficheiro, um registo, uma vez que não nos servia o `\n` pré-definido em `Awk`.

Analisando o ficheiro fornecido atentamente, era notório que entre cada registo significativo (excluindo as linhas em branco) existiam duas quebras de linha. No entanto, verificamos também que, entre as quebras de linha, em muitos casos, podíamos encontrar espaços ou caracteres como `\r` ou `\t`. Após esta análise, que foi acompanhada também pelo método de "tentativa e erro", acabamos com os seguintes separadores:

Separador de Registos: `RS = \n[\r\t]*\n`
Separador de Campos: `FS = ";"`

Tendo estes dois separadores definidos, o nosso trabalho ficou simplificado. Para a primeira linha, que consiste no título de cada campo, apenas temos que a reproduzir para o novo ficheiro. Nas seguintes, temos apenas que garantir que, quando o primeiro campo (campo `Estado`) é vazio, escrevemos "NIL" nesse campo e que não se trata de uma das linhas vazias que se pode ver no final do ficheiro fornecido, que contém todos os campos vazios, ou seja, 34 repetições do caractere `;`.

```
1 BEGIN { RS="\n[ \r\t]*\n"; FS=";" }
2 NR == 1 { print $0 > "processado.csv" }
3 NR>1 && $1 == "" && $0 !~ /[;]{34}/ {gsub(/\n/, " "); print "NIL" $0 > "processado.csv"}
4 NR>1 && $1 && $0 !~ /[;]{34}/ {gsub(/\n/, " "); print $0 > "processado.csv"}
5 END { }
```

Figura 1: Processador de texto para a alínea a)

3.2 Registos com código numérico

Visto que utilizaremos nesta e nas seguintes alíneas o ficheiro "limpo" obtido na alínea anterior, é necessário identificar o novo separador de registos, uma vez que continua a ser possível observar várias quebras de linha ao longo do ficheiro que não identificam o fim do registo. Como a primeira coluna do ficheiro tem sempre o valor `NIL`, podemos aproveitar este facto para separar cada registo por `\nNIL`, uma vez que sabemos que estamos num novo registo sempre que lemos este valor. O separador de registos mantém-se.

Para obter os registos que apresentam código numérico imprimimos o valor dos campos \$2, \$3, \$4 e \$5 que correspondem respetivamente ao código do processo, ao título, à sua descrição e às notas de aplicação do mesmo, no ficheiro **registos.csv**. Para além disso, criamos um campo **#Registo**, que permite contar o número de registos, numerando cada um deles de 1 até ao seu número total.

```
1 BEGIN          { RS="\nNIL"; FS=";"}
2 NR==1          {print "#Registo;" $2 ";" $3 ";" $4 ";" $5 > "registos.csv"; i++}
3 NR>1           {print i ";" $2 ";" $3 ";" $4 ";" $5 > "registos.csv"; i++}
4 END            { }
```

Figura 2: Processador de texto para a alínea b)

3.3 Contagem dos tipos de processos

Para este exercício, definimos os separadores de registos e de campos exatamente da mesma forma que na alínea anterior.

De modo a contabilizar todos os tipos de processos presentes no ficheiro processado, inserimos num array cada entrada encontrada no campo \$10, relativo ao tipo de processo, criando uma nova entrada no array se se tratar de um novo tipo, ou incrementando o valor contido no índice do tipo, se este já existir. Ou seja, cada índice do array é uma string que representa o nome do tipo e, em cada índice, encontra-se guardado o número de ocorrências desse tipo.

No final, escrevemos cada tipo encontrado e o respetivo número de processos encontrados no ficheiro **tipos.csv**.

```
1 BEGIN          { RS="\nNIL"; FS=";"; print "Tipo de processo; Número de Registos" > "tipos.csv" }
2 NR>1 && $10 != "" { array[$10]++; }
3 END            { for(i in array) print i ";" array[i] > "tipos.csv" }
```

Figura 3: Processador de texto para a alínea c)

3.4 Grafo para ações de formação

Tal como na alínea anterior, nesta voltamos logicamente a usar os mesmos separadores de campos e de registos. Analisando o ficheiro **processado.csv**, podemos ver que temos dois campos relativos a diplomas jurídico-administrativos, os campos \$8 e \$9. Então, para cada processo, vamos buscar estes dois campos. Estes dois campos consistem em listas de diplomas, contendo um diploma por linha, sendo que cada linha termina com o caractere # que é depois procedido de uma quebra de linha e/ou outros caracteres como \r e \t. Então, aproveitamos este padrão para definir o *split* que coloca num array cada diploma individual da lista de diplomas jurídico-administrativos.

Tendo agora acesso a cada diploma individualmente, primeiro processou-se cada um deles de forma a retirar aspas que apareciam desnecessariamente. Depois disso,

imprimiu-se cada um deles e o processo a que estavam associados, no formato que já foi referido na secção anterior.

```

1 BEGIN      { RS="\nNIL"; FS=";"; print "graph {" > "grafo.dot" }
2 NR > 1      { split($8, array, /[#][\n\t\r]+/);
3              for(i in array) {
4                  gsub(/\"/,"",array[i]);
5                  if(array[i]!="")
6                      print "\x22 $2 \x22 -- \x22 array[i] \x22;" > "grafo.dot";
7              }
8              split($9, array2, /[#][\n\t\r]+/);
9              for(i in array2){
10                  gsub(/\"/,"",array2[i]);
11                  if(array2[i]!="")
12                      print "\x22 $2 \x22 -- \x22 array2[i] \x22;" > "grafo.dot";
13              }
14          }
15 END        { print "}" > "grafo.dot" }

```

Figura 4: Processador de texto para a alínea d)

3.5 Extra: Contagem de processos associados a instituições

Seguindo as alíneas anteriores, iniciamos o processo por identificar os separadores de registos e de campos, como acima indicado.

No campo \$12 encontram-se as instituições portadoras de cada processo. Visto que um processo pode ter vários donos, é necessário aplicar a função **split** de modo a separar as várias instituições presentes nesse campo num único registo. Estas encontram-se separadas por um #.

Efetuada a inserção das instituições num array, resultado do split, contamos agora as ocorrências de cada uma delas e imprimimos o respetivo número de processos que pertencem a cada instituição no ficheiro **donos.csv**.

```

1 BEGIN      { RS="\nNIL"; FS=";"; print "Instituições; Número de Processos" > "donos.csv"}
2 NR>1      { split($12,array,/#[\n\t\r]*/);
3              for(i in array) {
4                  gsub(/\"/,"",array[i]);
5                  array2[array[i]]++
6              }
7          }
8 END        { for(i in array2) if(i != "") print i ";" array2[i] > "donos.csv";}

```

Figura 5: Processador de texto para a alínea extra

4 Resultados Obtidos

4.1 Alínea a)

Desta alínea, resulta um ficheiro `processado.csv`, que é em conteúdo praticamente igual ao fornecido, mas sem linhas em branco ou linhas extra, e com o campo "Estado" identificado como "NIL" em caso de ser vazio. De seguida temos dois excertos que representam a mudança do ficheiro original para o ficheiro `processado.csv`.

Estado	Código	Título
	750	PRESTAÇÃO DE SERVIÇOS DE ENSINO E FORMAÇÃO
	750.10	Gestão do aluno/formando
	750.10.001	Seleção e seriação para ingresso no ensino ou formação

Figura 6: Excerto do ficheiro `formacao.csv`

Estado	Código	Título
NIL	750	PRESTAÇÃO DE SERVIÇOS DE ENSINO E FORMAÇÃO
NIL	750.10	Gestão do aluno/formando
NIL	750.10.001	Seleção e seriação para ingresso no ensino ou formação

Figura 7: Excerto do ficheiro `processado.csv`

4.2 Alínea b)

O ficheiro registos.csv é o resultado do processador de texto explícito em b.gawk e consiste numa tabela onde surgem os códigos dos processos numerados, e informação pertinente acerca dos mesmos. Observando a primeira coluna, que diz respeito à numeração efetuada dos registos, podemos concluir que existem 29 registos de processos que contêm um código numérico. Na transformação do ficheiro processado.csv para o ficheiro registos.csv, foram mantidas apenas as colunas Código, Título, Descrição e Notas de Aplicação, e foi acrescentada uma coluna #Registo que numera cada um dos registos, sendo que na última linha da tabela poderemos ver através deste número a quantidade de processos existentes.

Estado	Código	Título
NIL	750	PRESTAÇÃO DE SERVIÇOS DE ENSINO E FORMAÇÃO
NIL	750.10	Gestão do aluno/formando
NIL	750.10.001	Seleção e seriação para ingresso no ensino ou formação

Figura 8: Excerto do ficheiro processado.csv

#Registo	Código	Título	Descrição	Notas de aplicação
1	750	PRESTAÇÃO DE SERVIÇOS DE ENSINO E FORMAÇÃO	Processo de prestação de serviços no domínio da educação/formação/qualificação da população, independentemente da idade ou do contexto (social, académico, profissional ou outro) - inclui a educação pré-escolar, o ensino qualificação profissional e avaliação permanente, em qualquer área de conhecimento.	Notas de aplicação Aplicar tanto às entidades que prestam os serviços referidos como às que solicitam a prestação dos mesmos e
2	750.10	Gestão do aluno/formando	Compreende as atividades relacionadas com apresentação de candidaturas, seleção de candidatos, matrícula, inscrições, transferências de alunos/formandos entre cursos, turnos ou estabelecimentos de ensino, e outras.	Quanto aplicável, estas atividades incluem o seguinte no âmbito da:
3	750.10.001	Seleção e seriação para ingresso no ensino ou formação	Avaliação da capacidade para a frequência de curso ou de ação de formação, bem como a redação das listas de admissão, designadamente as provas específicas, e a satisfação de pré-requisitos. Inicia com a candidatura a curso ou inscrição/inscrimento para formação e termina com a publicação da lista de produção final ou lista de admissão à formação. Inclui redação de provas de ingresso, aplicação de fórmulas e classificações mínimas, seleção dos candidatos, elaboração de listas finais e respetiva homologação.	Candidatos selecionados ou encaminhados para a formação profissional? Candidatos ao ingresso no âmbito da escolaridade obrigatória e do ensino regular profissional? Candidatos especiais (maiores de 25 anos)? Seleção dos candidatos ao curso de Estudos Avançados em Gestão Pública (CEAGP)? Bolsas das Palavras Africanas de Língua Oficial Portuguesa (PALOP) e Comunidade das Palavras de Língua Portuguesa (CPUP)? Concurso institucional? Programas multianuais de curso e transferência? Pré-graduação? Curso de Defesa Nacional? Formação ministrada pelo Centro de Formação de Funcionários de Justiça, para acesso e progressão no carreira de oficial de justiça? Formação para acesso e progressão no carreira de guarda-praça? Seleção para ingresso de alunos no ensino básico e secundário e ensino superior? Seleção para ingresso de jovens internados em centros educativos do ensino básico e secundário? Concurso de Ingresso na Formação Inicial - Técnico Administrativo e Fiscal e Técnico Judicial? Candidatos identificados para efeitos de encaminhamento na área da Práxis Judicial (PJ) segundo pré-requisitos? Edição para ingresso no Centro de Investigação Criminal da Polícia Judiciária? Formação ministrada no âmbito da medicina legal e cirúrgica forense? Formação ministrada pelo Centro de Formação do BSM? Formação para acesso ao carreira de conservador, notário e escrivão? Edição para ingresso no carreira de conservador e notário?

Figura 9: Excerto do ficheiro registos.csv

4.3 Alínea c)

O resultado desta alínea é um ficheiro, tipos.csv, contendo uma tabela que relaciona cada tipo de processo com o respetivo número de registos.

Tipo de processo	Número de Registos
PC	12
PE	5

Figura 10: Ficheiro tipos.csv

4.4 Alínea d)

O código que se segue representa o conteúdo do ficheiro grafo.dot, output da questão d) do enunciado. De seguida apresenta-se também um excerto daquele que seria o resultado deste grafo em forma de imagem.

```
1 graph {
2   "750.10.001" -- "Lei 45/2013";
3   "750.10.001" -- "Lei 45/2012";
4   "750.10.001" -- "DL 92/2011";
5   "750.10.001" -- "Lei 60/2011";
6   "750.10.001" -- "Portaria 181-D/2015";
7   "750.10.001" -- "Lei 9/2009";
8   "750.10.001" -- "Lei 2/2008";
9   "750.10.001" -- "Portaria 1500/2007";
10  "750.10.001" -- "Portaria 832/2007";
11  "750.10.001" -- "Portaria 401/2007";
12  "750.10.001" -- "DL 24/2006";
13  "750.10.001" -- "DL 174/2001";
14  "750.10.001" -- "Lei 166/99";
15  "750.10.001" -- "DL 70-A/2000";
16  "750.10.001" -- "DL 174/2001";
17  "750.10.001" -- "DL 4/98";
18  "750.10.001" -- "DL 48/86#";
19  "750.10.001" -- "DL 50/98 alterado pelos DL 70-A/2000 e DL 174/2001#";
20  "750.10.002" -- "Portaria 1141/2005";
21  "750.10.002" -- "DL 174/2001";
22  "750.10.002" -- "DL 29/2001";
23  "750.10.002" -- "DL 70-A/2000";
24  "750.10.002" -- "DL 174/2001";
25  "750.10.002" -- "DL 48/86#";
26  "750.10.002" -- "DL 50/98 alterado pelos DL 70-A/2000 e DL 174/2001#";
27  "750.10.300" -- "Lei 51/2012";
28  "750.10.300" -- "Lei 49/2005";
29  "750.10.300" -- "DL 50/98";
30  "750.10.300" -- "Lei 47/86";
31  "750.10.300" -- "Lei 21/85";
32  "750.10.300" -- "DL 174/2001";
33  "750.10.300" -- "DL 29/2001";
34  "750.10.300" -- "DL 48/86";
35  "750.10.300" -- "Portaria 213/2009";
36  "750.10.300" -- "Portaria 400/2007";
37  "750.10.600" -- "DL 174/2001";
38  "750.10.600" -- "Lei 2/2008";
39  "750.10.601" -- "DL 174/2001";
```

40 "750.10.602" -- "DL 29/2001";
 41 "750.10.602" -- "DL 3/2008#";
 42 "750.20.001" -- "DL 91/2013";
 43 "750.20.001" -- "DL 139/2012";
 44 "750.20.001" -- "DL 6/2001";
 45 "750.20.001" -- "DL 48/86";
 46 "750.20.001" -- "Lei 5/97#";
 47 "750.20.002" -- "Despacho Normativo 13/2014";
 48 "750.20.002" -- "DL 79/2014";
 49 "750.20.002" -- "DL 91/2013";
 50 "750.20.002" -- "DL 48/86";
 51 "750.20.002" -- "Portaria 243/2012";
 52 "750.20.002" -- "DL 174/2001";
 53 "750.20.002" -- "DL 286/89#";
 54 "750.20.300" -- "DL 5/2014";
 55 "750.20.300" -- "Portaria 81/2014";
 56 "750.20.300" -- "Lei 47/2006#";
 57 "750.20.301" -- "DL 75/2008#";
 58 "750.20.600" -- "DL 92/2014";
 59 "750.20.600" -- "Lei 45/2013";
 60 "750.20.600" -- "Lei 45/2012";
 61 "750.20.600" -- "Lei 9/2009";
 62 "750.20.600" -- "Lei 2/2008";
 63 "750.20.600" -- "Lei 39/2010";
 64 "750.20.600" -- "Portaria 1500/2007";
 65 "750.20.600" -- "Portaria 832/2007";
 66 "750.20.600" -- "Portaria 401/2007";
 67 "750.20.600" -- "DL 24/2006";
 68 "750.20.600" -- "DL 174/2001";
 69 "750.20.600" -- "Lei 166/99";
 70 "750.20.600" -- "DL 4/98";
 71 "750.20.601" -- "DL 205/2012";
 72 "750.20.601" -- "DL 215/2012";
 73 "750.20.601" -- "Lei 63/2007";
 74 "750.20.601" -- "Despacho Conjunto 275/2006";
 75 "750.20.601" -- "DL 44599/62#";
 76 "750.30.001" -- "DL 17/2016";
 77 "750.30.001" -- "Despacho Normativo 1-F/2016";
 78 "750.30.001" -- "DL 139/2012#";
 79 "750.30.300" -- "DL 17/2016";
 80 "750.30.300" -- "Despacho Normativo 1-F/2016";
 81 "750.30.300" -- "DL 139/2012#";
 82 "750.30.600" -- "DL 17/2016";
 83 "750.30.600" -- "Despacho Normativo 1-F/2016";
 84 "750.30.600" -- "DL 139/2012#";

```

85 "750.30.601" -- "DL 17/2016";
86 "750.30.601" -- "Despacho Normativo 1-F/2016";
87 "750.30.601" -- "DL 139/2012#";
88 "750.30.602" -- "DL 17/2016";
89 "750.30.602" -- "Despacho Normativo 1-F/2016";
90 "750.30.602" -- "DL 139/2012#";
91 }

```



Figura 11: Excerto do grafo resultante das linhas 71-90 do ficheiro grafo.dot

4.5 Alínea extra

Desta alínea resulta o ficheiro donos.csv apresentado na figura seguinte. Este contém uma tabela que indica para cada instituição o número de processos que lhes pertencem.

Instituições	Número de Processos
CMil	1
Grupo IP	8
INA	1
AP	12
DGES	1
IEFP	3
EPAE	1
FA	1
EC	1
FSS	1
ES	13
AL	1

Figura 12: Ficheiro donos.csv

5 Guia de Utilização

Para compilar todos os programas basta correr o comando **make**, graças à makefile que se apresenta de seguida. Para além de compilar o projeto, esta makefile trata também da limpeza dos ficheiros criados aquando da execução dos mesmos. Para usar esta funcionalidade basta correr o comando **make clean**.

```
1  run:
2      gawk -f a.gawk formacao.csv
3      gawk -f b.gawk processado.csv
4      gawk -f c.gawk processado.csv
5      gawk -f d.gawk processado.csv
6      gawk -f e.gawk processado.csv
7
8  a:
9      gawk -f a.gawk formacao.csv
10
11 b:
12     gawk -f b.gawk processado.csv
13
14 c:
15     gawk -f c.gawk processado.csv
16
17 d:
18     gawk -f d.gawk processado.csv
19
20 e:
21     gawk -f e.gawk processado.csv
22
23 clean:
24     rm -f processado.csv
25     rm -f registos.csv
26     rm -f tipos.csv
27     rm -f grafo.dot
28     rm -f donos.csv
29
30
```

De forma a obter uma imagem a partir do ficheiro `.dot` gerado, é preciso correr o seguinte comando, após ter instalado o software **graphviz**:

```
dot -Tpng grafo.dot > output.png
```

6 Conclusão

Após ter sido descrito todo o processo de desenvolvimento deste trabalho, desde a descrição do problema em causa até à implementação da solução, resta agora apresentar uma breve conclusão sobre todo o processo.

A realização deste projeto trouxe a consolidação de todos os conhecimentos adquiridos ao longo das aulas da unidade curricular acerca da ferramenta **Gawk** e do uso de expressões regulares. Assim, a partir destes dois, foi-nos possível construir diferentes processadores de textos que nos permitiram alcançar os objetivos estabelecidos no enunciado do problema.

De uma forma geral, terminamos este trabalho prático confiantes de que o seu objetivo foi cumprido e que todas as alíneas do enunciado que nos foi atribuído foram corretamente respondidas. Para além disso, consideramos que este trabalho permitiu também explorar o lado criativo do grupo, sendo que, a partir do enunciado base fornecido, o grupo esforçou-se em efetuar tarefas extra com o objetivo de demonstrar a grande variedade de informações que seriam possíveis de retirar do ficheiro fornecido.