

Processamento de Linguagens (3º ano de Curso)

**Processador de Inglês**

Relatório de Desenvolvimento

Alexandre Lopes Mandim da Silva  
A73674

Luís Miguel da Cunha Lima  
A74260

Hugo Alves Carvalho  
A74219

23 de Abril de 2017

### **Resumo**

Este relatório descreve todo o processo de desenvolvimento e decisões tomadas para a realização do segundo trabalho prático da Unidade Curricular de Processamento de Linguagens.

O problema a resolver consiste no desenvolvimento de um filtro de texto, utilizando o Flex, que seja capaz de ler textos em inglês expandindo todas as suas contrações que encontrar e gerar em HTML uma lista de todos os verbos deste mesmo texto.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
1.1	Estrutura do relatório . . . . .	2
<b>2</b>	<b>Caso de Estudo: Processador de Inglês</b>	<b>3</b>
2.1	Expansão de contrações . . . . .	3
2.2	HTML com os verbos no infinitivo . . . . .	3
2.2.1	Expressões Regulares . . . . .	3
2.2.2	Estruturas de Dados . . . . .	3
<b>3</b>	<b>Exemplos dos resultados gerados</b>	<b>5</b>
3.1	Expansão de Contrações . . . . .	5
3.2	Lista de verbos no infinitivo . . . . .	6
<b>4</b>	<b>Conclusão</b>	<b>8</b>
<b>5</b>	<b>Anexos</b>	<b>9</b>
5.1	Código Inglês Corrente . . . . .	9
5.1.1	Expansão de Contrações . . . . .	9
5.1.2	Lista de Verbos no Infinitivo . . . . .	9

# Capítulo 1

## Introdução

A resolução deste trabalho, passa pelo desenvolvimento de um filtro de texto em Flex que será utilizada para a resolução de um dos enunciados propostos. Para isso iremos utilizar técnicas leccionadas nas aulas desta Unidade Curricular.

Pretendemos com este trabalho aprimorar as nossas capacidades de escrever Expressões Regulares (ER), também a nossa experiência na utilização da linguagem de programação C.

O grupo decidiu escolher o enunciado "Processador de Inglês Corrente", como problema a resolver. Para a resolução deste enunciado é então importante utilizar a noção de Expressão Regular para reconhecer no texto quais as contrações presentes e substituí-las pelas palavras relacionadas e ainda encontrar os verbos que se encontrem no infinitivo.

Neste relatório apresentamos todos os passos e a justificação das decisões tomadas durante todo o processo, descrevemos as estruturas criadas para guardar o texto extraído pelo filtro e também uma apresentação do produto final (em HTML) obtido com a utilização do filtro criado por nós.

### 1.1 Estrutura do relatório

A elaboração deste documento tem por base a estrutura do relatório fornecida pelo docente.

O relatório encontra-se dividido em quatro capítulos. O primeiro capítulo faz uma rápida contextualização ao tema tratado neste relatório. Posteriormente são desenvolvidas e justificadas todas as tarefas envolvidas para desenvolver o trabalho. Não menos importante, segue-se o capítulo de apresentação dos testes efectuados à aplicação com os indispensáveis resultados. Para finalizar, faz-se uma análise crítica referente ao desenvolvimento do projeto e ao seu estado final.

## Capítulo 2

# Caso de Estudo: Processador de Inglês

### 2.1 Expansão de contrações

Nesta parte do trabalho, foi pedido que, perante um texto em inglês com contrações do tipo (*i'm, you're, etc.*), substituíssemos essas contrações para a forma extensível. Deste modo, recorrendo à ferramenta flex, e expressões regulares resolvemos o problema da seguinte forma: em primeiro lugar, usando expressões regulares, procuramos as contrações que pretendemos alterar e realizamos a respetiva alteração.

Para isto, focamo-nos nas principais contrações usadas na língua inglesa, como por exemplo, (*i'm, you're, you've, you'll, he's*).

De salientar que formas contraídas como, por exemplo, *why's* pode significar *why has, why is* e *why does*, deste modo consideramos o 's apenas para o verbo to be.

### 2.2 HTML com os verbos no infinitivo

#### 2.2.1 Expressões Regulares

Para encontrar os verbos, procuramos os que são antecidos por *to, can, could, shall, should, will, would, may* e *might*. Contudo estas palavras podem estar no início da frase ou a meio, assim sendo, usamos duas expressões regulares para fazer match a estas expressões. Na expressão regular usada para encontrar quando a palavra está no início da frase (AntecedenteMinuscula), procuramos pela palavra (*to, can, ...*) seguida por um espaço ou um \n seguida de outra palavra (verbo) que tem no mínimo dois caracteres. Quando está no meio da frase (AntecedenteMaiuscula) procuramos pela mesma expressão regular que no início, apenas com a exceção que tem que existir um espaço antes da expressão.

Para encontrar os verbos em frases interrogativas, seguimos o mesmo raciocínio anterior (início e meio da frase). Procuramos palavras como *do, does* e *did* seguidas de um sujeito (*i, you, he, she, it, we* e *they*) ou *not* e o verbo vem em seguida.

É importante salientar que nem sempre isto acontece, ou seja, nem sempre em seguida da palavra *to* vem um verbo, assim sendo, fazemos match a expressões que a segunda palavra não é um verbo.

Sempre que o flex encontra um match no texto é chamada a função `guardaVerbo(yytext)`. Esta função é explicada em seguida.

#### 2.2.2 Estruturas de Dados

De modo a organizar os verbos encontrados no texto da forma mais correta, decidimos criar uma matriz com 26 linhas em que cada uma das posições corresponde a uma letra do alfabeto. Assim, sempre que encontramos um novo verbo, o procedimento é o seguinte:

1. Encontrar a última palavra da expressão enviada com `yytext` para a função `guardaVerbo`

2. Com esta última palavra, correspondente ao verbo, encontrar o número da linha correspondente com a função auxiliar `getId`, que utiliza o primeiro carácter da string.
3. Verificar se o verbo já existe na matriz
4. Se não existe, adiciona o verbo no fim linha encontrada em (2)

Por último, para imprimir a lista de verbos no ficheiro HTML basta percorrer toda a matriz e imprimir cada uma das posições.

## Capítulo 3

# Exemplos dos resultados gerados

Para o enunciado escolhido, Processador Inglês, demonstramos neste capítulo o output gerado após a execução dos filtros de texto explicados nas secções anteriores.

### 3.1 Expansão de Contrações

Para demonstrar o funcionamento do filtro de texto desenvolvido para a alinea a, usamos um pequeno texto com varias contrações aplicadas no exercício da lingua inglesa. Ao apresentar os resultados decidimos mostrar ambos os textos para se poder verificar as diferenças entre eles, tornando assim mais facil a analise do resultado do filtro de texto desenvolvido.

```
1 Now then Mardy Bum
2 I see your frown
3 And it's like looking down the barrel of a gun
4 And it goes off
5 And out come all these words
6 Oh there's a very pleasant side to you
7 A side I much prefer
8 It's one that laughs and jokes around
9 Remember cuddles in the kitchen
10 Yeah, to get things off the ground
11 And it was up, up and away
12 Oh, but it's right hard to remember
13 That on a day like today when you're all argumentative
14 And you've got the face on
15
16 Well now then Mardy Bum
17 Oh I'm in trouble again, aren't I
18 I thought as much
19 Cause you turned over there
20 Pulling that silent disappointment face
21 The one that I can't bare
22
23 Can't we laugh and joke around
24 Remember cuddles in the kitchen
25 Yeah, to get things off the ground
26 And it was up, up and away
27 Oh, but it's right hard to remember
28 That on a day like today when you're all argumentative
29 And you've got the face on
30
31 Yeah I'm sorry I was late
32 Well I missed the train
33 And then the traffic was a state
34 And I can't be arsed to carry on in this debate
35 That reoccurs ,oh when you say I don't care
36 Well of course I do, yeah I clearly do!
37
38 So laugh and joke around
39 Remember cuddles in the kitchen
40 Yeah, to get things off the ground
41 And it was up, up and away
42 Still it's right hard to remember
43 That on a day like today when you're all argumentative
44 And you've got the face on
45
--
```

Figura 3.1: Letra da Música Mardu Bum - Artic Monkeys

```

hugocarvalho@hugocarvalho ~/Dropbox/Trabalhos PL/TP2 $ ./alineaA < testel.txt
Now then Mardy Bum
I see your frown
And it is like looking down the barrel of a gun
And it goes off
And out come all these words
Oh there's a very pleasant side to you
A side I much prefer
It is one that laughs and jokes around
Remember cuddles in the kitchen
Yeah, to get things off the ground
And it was up, up and away
Oh, but it is right hard to remember
That on a day like today when you are all argumentative
And you have got the face on

Well now then Mardy Bum
Oh I am in trouble again, aren't I
I thought as much
Cause you turned over there
Pulling that silent disappointment face
The one that I can not bare

Can not we laugh and joke around
Remember cuddles in the kitchen
Yeah, to get things off the ground
And it was up, up and away
Oh, but it is right hard to remember
That on a day like today when you are all argumentative
And you have got the face on

Yeah I am sorry I was late
Well I missed the train
And then the traffic was a state
And I can not be arsed to carry on in this debate
That reoccurs, oh when you say I don't care
Well of course I do, yeah I clearly do!

So laugh and joke around
Remember cuddles in the kitchen
Yeah, to get things off the ground
And it was up, up and away
Still it is right hard to remember
That on a day like today when you are all argumentative
And you have got the face on

```

Figura 3.2: Letra da Música Mardu Bum - Artic Monkeys aplicado o filtro de texto

## 3.2 Lista de verbos no infinitivo

Para exemplificar os resultados gerados pelo filtro de texto desenvolvido para esta segunda alínea, utilizamos como texto de entrada uma letra de uma música *Numb*.

Mostramos assim a lista de verbos encontrados, apresentada ao abrir a página html gerada, ordenados alfabeticamente. As imagens seguintes mostram os resultados obtidos:



```

1 I'm tired of being what you want me to be
2 Feeling so faithless, lost under the surface
3 I don't know what you're expecting of me
4 Put under the pressure of walking in your shoes
5 Caught in the undertow, just caught in the undertow
6 Every step that I take is another mistake to you
7 Caught in the undertow, just caught in the undertow
8
9 I've become so numb, I can't feel you there
10 Become so tired, so much more aware
11 By becoming this all I want to do
12 Is be more like me and be less like you
13
14 Can't you see that you're smothering me?
15 Holding too tightly, afraid to lose control
16 'Cause everything that you thought I would be
17 Has fallen apart right in front of you
18 Caught in the undertow, just caught in the undertow
19 Every step that I take is another mistake to you
20 Caught in the undertow, just caught in the undertow
21 And every second I waste is more than I can take!
22
23 I've become so numb, I can't feel you there
24 Become so tired, so much more aware
25 By becoming this all I want to do
26 Is be more like me and be less like you
27
28 And I know I may end up failing too
29 But I know you were just like me with someone disappointed in you
30
31 I've become so numb, I can't feel you there
32 Become so tired, so much more aware
33 By becoming this all I want to do
34 Is be more like me and be less like you
35
36 I've become so numb, I can't feel you there
37 I'm tired of being what you want me to be
38 I've become so numb, I can't feel you there
39 I'm tired of being what you want me to be
40

```

Figura 3.3: Letra da Música Numb - Linkin Park



Figura 3.4: Letra da Música Numb - Link Park aplicado o filtro de texto

## Capítulo 4

# Conclusão

A realização deste projeto foi fundamental para se consolidar a matéria leccionada quer nas aulas práticas quer nas teóricas, uma vez que as técnicas de utilização de expressões regulares aí aprendidas facilitou a implementação do problema. Estas técnicas permitiram recolher de forma clara e eficiente as informações pretendidas.

Foi ainda possível ao grupo aprofundar os conhecimentos da linguagem de HTML de pouco tinha sido utilizada por qualquer dos elementos do nosso grupo. O projeto "Processador de Inglês" foi muito enriquecedor, dado que obrigou a dedicar uma porção de tempo e a refinar as qualidades do grupo na utilização do FLEX e na construção das expressões regulares.

## Capítulo 5

# Anexos

### 5.1 Código Inglês Corrente

#### 5.1.1 Expansão de Contrações

```
%{  
%}  
%%  
  
'm      { printf(" am"); }  
're     { printf(" are"); }  
n't     { printf(" not"); }  
't      { printf(" not"); }  
've     { printf(" have"); }  
'll     { printf(" will"); }  
It's    { printf("It is"); }  
it's    { printf("it is"); }  
He's    { printf("He is"); }  
he's    { printf("he is"); }  
She's   { printf("She is"); }  
she's   { printf("she is"); }  
  
%%  
  
int yywrap(){  
    return 1;  
}  
  
int main(){  
    yylex();  
    return 0;  
}
```

#### 5.1.2 Lista de Verbos no Infinitivo

```
%option noyywrap
%option yylineno

%{

#include <stdio.h>
#include <ctype.h>
#include <string.h>

void guardaVerbo(char* verbo);
void inicia();
int getId(char c);
int existe(char* c, int id);
void adiciona(char* c, int id);
void gerarPagina();
void imprime();
char** lista[26];

}%

AntecedenteMaiuscula (Can|Could|Shall|Should|Will|Would|May|Might|Do|Did|Does|To)
AntecedenteMinuscula (can|could|shall|should|will|would|may|might|do|did|does|to)
```

```
Sujeito ([Ii][Yy]ou|[Hh]e|[Ss]he|[Ii]t|[Ww]e|[Tt]hey|[Nn]ot)

Declarativa1 ({AntecedenteMaiuscula}[\\ \\n][a-zA-Z]{2,})
Declarativa2 ((\\ ){AntecedenteMinuscula}[\\ \\n][a-zA-Z]{2,})
Interrogativa1 ({AntecedenteMaiuscula}(\\ ){Sujeito}(\\ ){a-zA-Z}{2,})
Interrogativa2 ((\\ ){AntecedenteMinuscula}(\\ ){Sujeito}(\\ ){a-zA-Z}{2,})

%%
{Declarativa1} { guardaVerbo(yytext); }
{Declarativa2} { guardaVerbo(yytext); }
{Interrogativa1} { guardaVerbo(yytext); }
{Interrogativa2} { guardaVerbo(yytext); }
<*.\\n { }
%%

void gerarPagina() {
FILE* f = fopen("alinea8.html", "w");
char* inicio = "<html> <head> <h1> PL-TP1b </h1> </head> </body>";
char* fim = "</body> </html>";

fprintf(f, "\\s", inicio);
```

```
int i, j;

for (i=0; i<26; i++) {
for (j=0; lista[i][j]; j++) {
fprintf(f, "<li> %s </li>", lista[i][j]);
}
}

fprintf(f, "\\s", fim);
fclose(f);
}

void inicia() {
int i;
for (i=0; i<26; i++) {
lista[i] = (char**)malloc(1024*sizeof(char**));
}
}

int existe (char* verbo, int id) {
int i;

for (i=0; lista[id][i]!=NULL; i++) {
if (strcmp(lista[id][i], verbo) == 0)
return 1;
}
}
```

```

void adiciona (char* verbo, int id) {
    int i;

    for(i=0; lista[id][i]!=NULL; i++);
    lista[id][i] = strdup(verbo);
}

int getId(char c) {
    char aux = tolower(c);
    int r = aux-97;
    return r;
}

void guardaVerbo(char* arg){
    char* pal = strtok(arg, " \n");
    char ultima[50];

    while(pal!=NULL) {
        strcpy(ultima,pal);
        pal = strtok(NULL, " \n");
    }

    int id = getId(ultima[0]);
    int r = existe(ultima, id);
    if (r == 0) {
        adiciona(ultima, id);
    }
}

```

```

void imprime() {
    int i, j;
    for (i=0; i<26; i++) {
        for (j=0; lista[i][j]!=NULL; j++) {
            printf("%s\n", lista[i][j]);
        }
    }
}

int main (int argc, char* argv[]){
    inicia();

    if (argc==2) {
        yyin = fopen(argv[1], "r");
    }

    yylex();

    //imprime();
    gerarPagina();

    return 0;
}

```