Universidade do Minho

Processamento de Linguagens

Trabalho Prático 1

Março 2019

André Guilherme Nunes Viveiros, A80524

César Augusto da Costa Borges, A81644

Luís José Rodrigues da Silva Macedo, A80494

Resumo

Este documento consiste no relatório correspondente ao primeiro trabalho prático realizado no âmbito da Unidade Curricular de Processamento de Linguagens, do curso de Mestrado Integrado em Engenharia Informática da Universidade do Minho, no ano letivo de 2018/2019. Este primeiro trabalho consiste em desenvolver uma solução a um desafio proposto pelos docentes da Unidade Curricular.

Índice

Índice de Figuras

1. Introdução

Este relatório apresenta e documenta a totalidade do trabalho desenvolvido no âmbito do primeiro trabalho prático da Unidade Curricular de Processos de Linguagem, do curso de Mestrado Integrado em Engenharia Informática da Universidade do Minho, no ano letivo de 2018/2019.

Este capítulo indica os objetivos, contextualiza e apresenta o caso de estudo, descrevendo também as motivações e objetivos do projeto.

1.1. Objetivos

Este trabalho tem como objetivos:

- A resolução de um desafio usando *ER* e *Flex*;
- A familiarização dum ambiente *Unix*;
- A familiarização das *ER*;
- O desenvolvimento de *Processadores de Linguagens Regulares*;
- A utilização do *Flex*

1.2. Contextualização e caso de estudo

Dos 8 enunciados propostos pelos docentes da *UC*, este grupo escolheu o enunciado 8, de acordo com a formula proposta. Considera-se o caso de

estudo a limpeza e normalização de um ficheiro com milhares de noticias de um jornal angolano. Através dessa normalização, espera-se a criação de um ficheiro *HTML* para cada noticia, a criação de um ficheiro *HTML* com todas as *tags*, em que cada *tag* é um *link* para outro ficheiro com *links* para todas as noticias com essa *tag*. Por ultimo, a criação de uma lista com todas as *tags* encontradas e numero de ocorrências.

1.3. Estrutura do relatório

Este documento apresenta a seguinte estrutura:

- No Capítulo 2 (fundamentação) refere-se as ideais gerais para o desenvolvimento da solução;
- O relatório é concluído no Capítulo 6 com observações relevantes.

2. Fundamentação

Tendo-se apresentado o caso de estudo e identificado as motivações e objetivos para o desenvolvimento da soluçao ao desafio em questão, este capítulo fundamenta a sua construção.

De forma a limpar as noticias usou-se *ER* com o objetivo de encontrar todas as características pertinentes de cada noticia, tais como:

- *Tags*;
- *ID*;
- Categoria;
- Data;
- Autor;
- Artigo.

Em conjunto com a limpeza, ocorre a normalização, ou seja, guarda-se os dados numa estrutura de dados. Após ter todas as noticias normalizadas, cria-se todos os ficheiros *html* necessários.

3. Normalização

Tendo-se a fundamentação do sistema, este capítulo explica o processo de normalização do artigo.

3.1. Limpeza dos Dados

Após a observação dos artigos do Jornal Angolano reparou-se que todas as notícias têm uma estrutura semelhante á descrita na Figura 1, ou seja, as informações de cada noticia encontram-se por esta ordem:

1. *tags*;
2. *ID*;
3. categoria;
4. titulo;
5. autor e a data;
6. texto do artigo.

Dito isto, decidiu-se criar uma estrutura onde fosse possível guardar a informação mais pertinente da mesma.

Figura 1: Exemplo de uma noticia.

3.2. Estrutura de Dados

De forma a guardar todos os artigos criou-se uma estrutura de dados capaz de guardar toda a informação de um artigo, isto é, uma estrutura chamada *Noticia*, que guarda toda a informação da noticia e uma estrutura chamada *Tag*, que guarda uma *tag*, o numero de repetições e os *IDs* das noticias onde aparece.

Noticia:

- *char *id*, o *ID*;
- *char *title*, o titulo
- *char *category*, a categoria;
- *char *date*, o autor e a data;
- *char **tags*, o conjunto das *tags*;
- *int lenght_tags*, o numero de *tags*;

- *char *text*, o texto.

Tag:

- *char *tag*, a *tag*;
- *int rept*, o numero de repetições;
- *char **id_noticia*, o conjunto dos *IDs* das noticias em que aparece.

Para que seja possível guardar todas as noticias lidas do ficheiro, decidiu-se usar estruturas auxiliares, da biblioteca *GLib*. Essas estruras são duas *GHashTable*, **tags** e **noticias**.

	Key	**Value**
tags	*tag*(*char **)	*Tag*(estrutura de dados)
noticias	*ID* da noticia(*char **)	*Noticia*(estrutura de dados)

De modo a inserir e percorrer estas estruturas externa usou-se as seguintes funções, da mesma biblioteca:

- *g_hash_table_insert*;
- *g_hash_table_lookup*;
- *g_hash_table_foreach*;

De forma a ter um conjunto de noticias por *tag*, o preenchimento da *GHashTable* tags é feito depois de ter percorrido todo o ficheiro e ter guardo todas as noticias, ou seja, percorre-se as noticias e por cada noticia percorre-se as *tags* dessa noticia, inserindo-as na *GHashTable* juntamente com o *ID* da noticia e aumentado o numero de repetiçoes.

4. Expressões Regulares

Após as estruturas estarem bem definidas, construiu-se um autómato para modelar o sistema de acordo com a nossa estrutura.

Este autómato contem 9 autómatos mais pequenos:

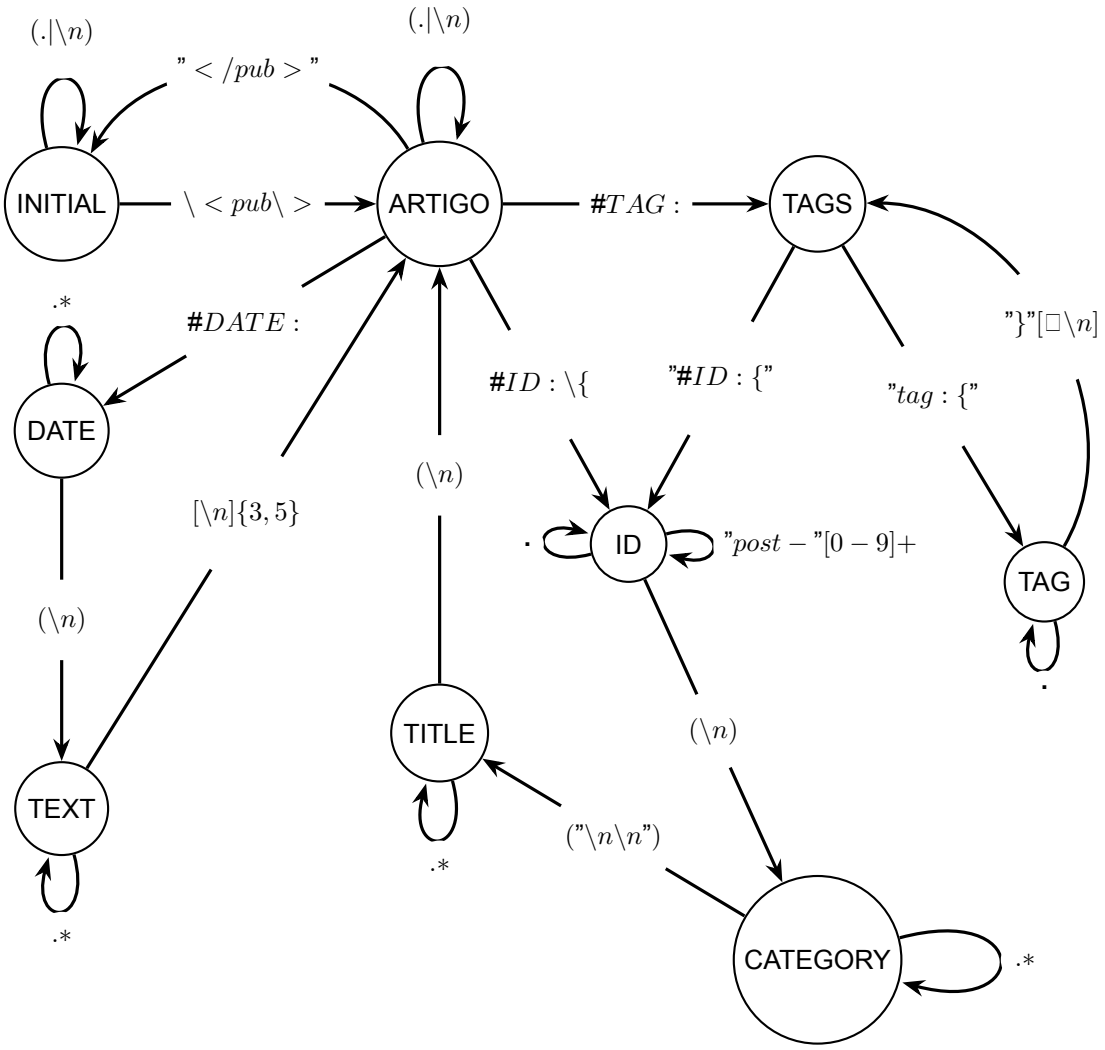
- INITIAL;
- ARTIGO;
- TAGS;
- TAG;
- ID;
- CATEGORY;
- TITLE;
- DATE;
- TEXT;

Quando o programa começa, tem à sua disposição as seguintes *ER*s:

1. $\backslash < pub \backslash >$
2. $< ARTIGO >$ " $< /pub >$ "
3. $< ARTIGO >$ #TAG :
4. $< ARTIGO >$ #$DATE$:
5. $< ARTIGO >$ #ID : $\backslash\{$
6. $< ARTIGO > (.|\backslash n)$
7. $< TAGS >$ "tag : {"
8. $< TAGS >$ "ID : {"
9. $< TAG >$ "}"$[\square\backslash n]$, onde $\square$ é igual a um espaço

10. $< TAG >$.
11. $< ID >$ "$post-$"$[0-9]+$
12. $< ID >$ $(\backslash n)$
13. $< ID >$.
14. $< CATEGORY >$ $.*$
15. $< CATEGORY >$ ("$\backslash n \backslash n$")
16. $< TITLE >$ $.*$
17. $< TITLE >$ $(\backslash n)$
18. $< DATE >$ $(\backslash n)$
19. $< DATE >$ $.*$
20. $< TEXT >$ $.*$
21. $< TEXT >$ $[\backslash n]\{3, 5\}$
22. $(.|\backslash n)$

Resumindo as listas anteriores, tem se o seguinte automato com todas as *ER*s e as respetivas ações:



Este autómato procura o inicio da noticia, entrando no estado ARTIGO e ignora tudo o resto. Dentro do ARTIGO ele procura o inicio das *tags*, entrando em TAGS, dos *ID*s, entrando em ID e da datae autor, entrando em DATE, ignorando o resto.

O autómato TAGS procura o inicio de uma *tag*, que se for encontrada entra em TAG, e o inicio do *ID*s. O inicio do *ID* é procurado duas vez pois, durante a analise do ficheiro com as noticias foram encontradas algumas exceções, em que uma delas não continha *tags*, e desta forma se não houver *tags* os *ID*s são encontrados no autómato ARTIGO.

Dentro do autómato TAG é procurado uma *tag* carácter a carácter devido ao problema intitula "*Black Hole*", ou seja, é encontrado o texto com mais caracteres e o pretendido é o de menor caracteres. Quando encontra o fim da *tag* volta a TAGS.

Se forem encontrados os *ID*s é procurado o *ID* que siga a forma "*post*-X", onde X é um numero positivo. Os restantes *ID*s são ignorados. Se for encontrado uma mudança de linha entra-se em CATEGORY.

A categoria entra-se numa só linha, logo guarda-se essa linha toda. Quando for encontrado duas mundaças de linha seguidas entra-se em TITLE.

O processo de procura pelo titulo é o mesmo que o da categoria com uma diferença, quando encontra apenas uma mudança de linha volta ao autómato ARTIGO.

Como a data e o autor está contida numa só linha, é apanhada a linha toda removendo os primeiro nove caracteres (não são pertinentes). Após ser encontrado uma mudança de linha, entra-se em TEXT para procurar o texto da noticia.

Para guardar o texto da noticia são apanhadas todas as linhas até encontrar três ou quatro ou cinco mudanças de linha seguidas, voltando ao autómato ARTIGO.

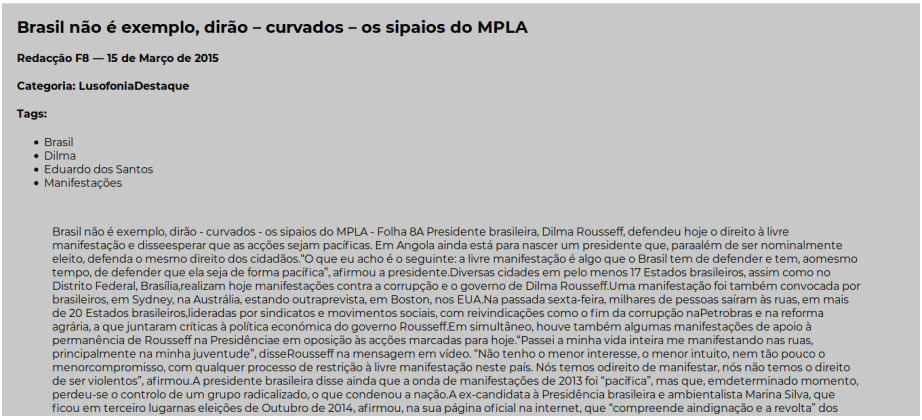
5. HTML

5.1. Noticias

Um dos casos de estudo era a criação de ficheiros *HTML* para apresentar as noticias e as *tags*. Para tal foram criados um pagina *HTML* com uma lista de links apresentados pelos títulos das noticias, para os ficheiros individuais *HTML* das noticias. Pode ser observado um exemplo na Figura 2.

Figura 2: Exemplo de um ficheiro HTML com as noticias.

Cada noticia foi estruturada com o titulo no topo da pagina, seguindo o autor e data, a categoria, a lista das *tags* e por ultimo o texto da noticia. Na Figura 3 é apresentado um exemplo de uma noticia.



Brasil não é exemplo, dirão – curvados – os sipaios do MPLA

Redacção F8 — 15 de Março de 2015

Categoria: LusofoniaDestaque

Tags:

- Brasil
- Dilma
- Eduardo dos Santos
- Manifestações

Brasil não é exemplo, dirão - curvados - os sipaios do MPLA - Folha 8A Presidente brasileira, Dilma Rousseff, defendeu hoje o direito à livre manifestação e disseesperar que as acções sejam pacíficas. Em Angola ainda está para nascer um presidente que, paraalém de ser nominalmente eleito, defenda o mesmo direito dos cidadãos."O que eu acho é o seguinte: a livre manifestação é algo que o Brasil tem de defender e tem, aomesmo tempo, de defender que ela seja de forma pacífica", afirmou a presidente.Diversas cidades em pelo menos 17 Estados brasileiros, assim como no Distrito Federal, Brasília,realizam hoje manifestações contra a corrupção e o governo de Dilma Rousseff.Uma manifestação foi também convocada por brasileiros, em Sydney, na Austrália, estando outraprevista, em Boston, nos EUA.Na passada sexta-feira, milhares de pessoas saíram às ruas, em mais de 20 Estados brasileiros,lideradas por sindicatos e movimentos sociais, com reivindicações como o fim da corrupção naPetrobras e na reforma agrária, a que juntaram críticas à política económica do governo Rousseff.Em simultâneo, houve também algumas manifestações de apoio à permanência de Rousseff na Presidênciae em oposição às acções marcadas para hoje."Passei a minha vida inteira me manifestando nas ruas, principalmente na minha juventude", disseRousseff na mensagem em vídeo. "Não tenho o menor interesse, o menor intuito, nem tão pouco o menorcompromisso, com qualquer processo de restrição à livre manifestação neste país. Nós temos odireito de manifestar, nós não temos o direito de ser violentos", afirmou.A presidente brasileira disse ainda que a onda de manifestações de 2013 foi "pacífica", mas que, emdeterminado momento, perdeu-se o controlo de um grupo radicalizado, o que condenou a nação.A ex-candidata à Presidência brasileira e ambientalista Marina Silva, que ficou em terceiro lugarnas eleições de Outubro de 2014, afirmou, na sua página oficial na internet, que "compreende aindignação e a revolta" dos

Figura 3: Exemplo de um ficheiro HTML com uma noticias.

5.2. Tags

A pagina com todas as *tags* contem uma lista com todas as *tags* e o numero de repetições, Figura 4. Para alem do nome da *tag* e do numero de repetições, o nome da *tag* está ancorado a uma pagina com uma lista das noticias onde essa *tag* aparece, Figura 5.

Tag	Number of repetitions
Processo de Paz	1
confiança	2
casa-ce. cabinda	1
gás	6
construtoras	1
slazar	1
um morto	1
genocida	1
puto	1
nativos	1
Chico Pobre	1
resultados	15
Mpindi André	1
duplicar	1
convocação	1
morro do moco	1
relógio	1
tony neves	1
Belas Shopping	1
José Saramago	1
Narciso Benedito	1
mobilidade	1
prejuízos	4
Matchedje	1

Figura 4: Exemplo de um ficheiro HTML com as tags e as suas repetições.

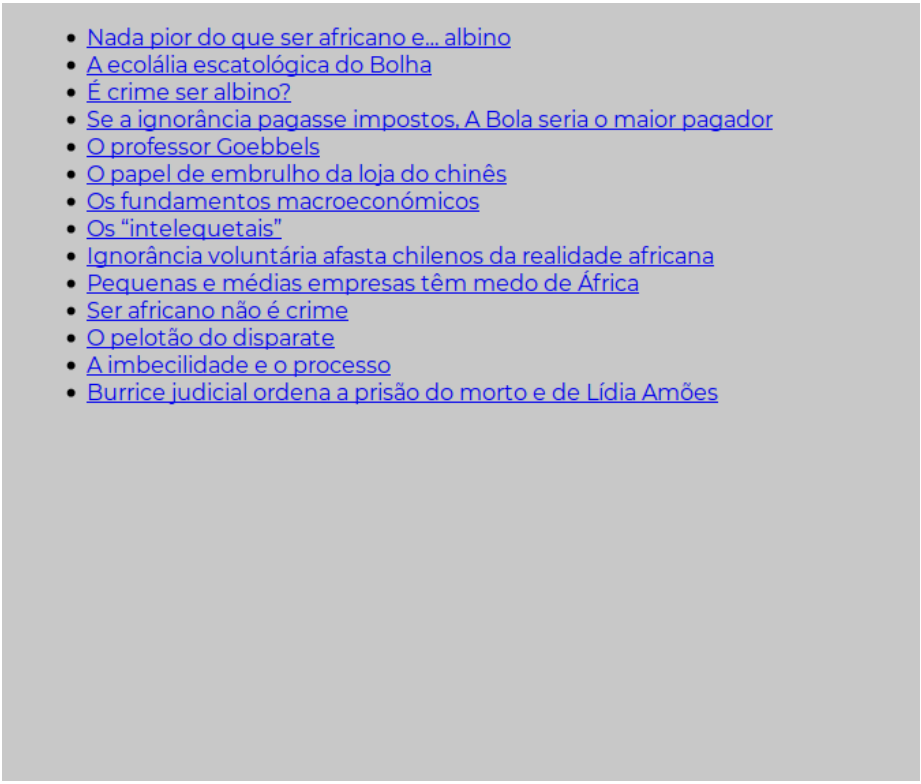


Figura 5: Exemplo de um ficheiro HTML com as noticias que contêm a tag ”ignorância.

6. Conclusão

Em conclusão, o programa desenvolvido é capaz de processar um ficheiro com noticias, que seguem uma estrutura especifica, guardando as informações pertinentes de cada noticia numa estrutura de dados, atraves de *ER*s de da ferramenta *Flex* e criando ficheiros *HTML* com:

- Uma lista com o titulo das noticias, ancorado aos ficheiros individuais das noticias;
- As informações guardadas de uma noticia, para todas as noticias;
- Uma lista com o nome das *tags*, ancorado a um ficheiro com a lista de noticias dessa *tag*;
- Uma lista com o titulo das noticias com uma certa *tag*, ancorado ao ficheiro da noticia.

Para alem disso, o grupo consolidou os conhecimentos aprendidos sobre *ER*s, uso da ferramenta *Flex*.

Lista de Siglas e Acrónimos

ER	Expressões Regulares (*Regular Expressions*)
UC	Unidade Curricular
ID	Identificador (*Identity*)
XML	*Extensible Markup Language*