Universidade do Minho

Mestrado Integrado em Engenharia Informática

Processamento de Linguagens

Filipe Monteiro (a80229)
Bruno Martins (a80410)

28 de Abril de 2019

Resumo

Este trabalho tem como foco a utilização da ferramenta *Flex* juntamente com expressões regulares para o processamento de artigos da *Wikipedia* portuguesa devolvendo um resumo em dois formatos, *HTML*, *LaTeX* ou *Markdown* no contexto da unidade curricular de Processamento de linguagens. Para isso desenvolveram-se um série de estruturas de dados capazes de armazenar os textos processados de forma a produzir os artigos no formato desejado.

Conteúdo

1 Introdução **3**

2 Conceitos Base **3**
2.1 Formato do XML 3

3 Estruturas de dados **5**

4 Análise dos artigos **5**

5 Representação dos Dados **7**

6 Conclusões **7**

7 Anexos **8**

1 Introdução

Este projeto consiste em, dado um ficheiro *XML* com artigos da *Wikipédia PT*, gerar resumos com certos critérios no formato *HTML*, *LaTeX* e também *Markdown* O resumo tem que conter o título de cada artigo bem como uma *Infobox* se existir, o primeiro parágrafo do desenvolvimento e as categorias associadas a ele. Para isso foi utilizado o analizador lexixo *Flex* que permite através de expressões regulares fazer uma filtragem e impressão de todo o conteúdo neceessário numa estrutura apropriada. Para a geração de estruturas de auxílio foi utilizada a linguagem de programação C.

2 Conceitos Base

Para a total compreensão deste relatório é necessário o conhecimento prévio de alguns conceitos. Numa primeira instância Expressão Regular será um termo bastante comum e que podemos definir como uma forma concisa e flexível de analizar conjuntos de caracteres. Flex, é o analizador léxico que será utilizado para em conjunto com as expressões regulares formar os textos prentendidos. Estruturas de dados, são formas de armazenar informação de forma a manipula-la como pretendemos. XML, é um formato de texto muito utilizado na internet, poderemos ver um exemplo no subcapítulo seguinte.

2.1 Formato do XML

Para este projeto, o *XML* tem um formato específico atribuído pela *Wikipedia* no entanto, as marcas importantes são as de *title*, *Info/*, *”’* e *[[Categoria:]]*. Podemos ver isso no excerto seguinte:

```
<page>
    <title>Albino Forjaz de Sampaio</title>
    <ns>0</ns>
    <id>224</id>
    <revision>
      <id>51866589</id>
      <parentid>51059268</parentid>
      <timestamp>2018-04-21T18:15:50Z</timestamp>
      <contributor>
        <username>Robertogilnei</username>
        <id>135704</id>
      </contributor>
      <model>wikitext</model>
      <format>text/x-wiki</format>
      <text xml:space="preserve">{{Info/Biografia
 |nome                 = Albino Forjaz de Sampaio
 |nascimento_data      = {{dni|19|1|1884|sem idade}}
 |nascimento_local     = [[Lisboa]]
 |morte_data           = {{morte|13|3|1949|19|1|1884}}
 |morte_local          = Lisboa
 |nacionalidade        = [[Portugal|Português]]
 |ocupação             = [[Escritor]] e bibliógrafo
 |magnum_opus          = ''Porque me orgulho de ser português''
 |prémios              =
}}

'''Albino Maria Pereira Forjaz de Sampaio''' ([[Lisboa]], [[19

== Início de carreira ==
Albino Forjaz de Sampaio começou a sua carreira literária aos 1


{{Portal3|Mitologia greco-romana|Mitologia}}
[[Categoria:Heróis da mitologia grega]]
[[Categoria:Pessoas da Guerra de Troia]]
[[Categoria:Personagens da Ilíada]]
[[Categoria:Semideuses da mitologia greco-romana]]</text>
<sha1>4ykq1q2yefugpe3dny8cgm3zdsxklnd</sha1>
</revision>
</page>
```

Figura 1: Excerto de XML

3 Estruturas de dados

Para o armazenamento dos dados processados foram criadas duas estruturas de dados:

- *Article*, que consiste guradar um memória os textos;
- *Vector*, que consiste num array de *Articles*.

A estrutura *Article* tem como atributos um array de caracteres para guardar o titulo dos artigos, um array de arrays de caracteres para guardar a *Infobox*, um array de array de caracteres para guardar o resumo do artigo, um inteiro para guardar o numero de categorias desse artigo, um inteiro para guardar o numero de palavras num resumo e um array de caracteres para guardar o *URL* desse artigo na *WikipediaPT*. Podemos ver essa estrutura na figura seguinte: A estrutura vector é caracterizada por um array de artigos, um

```
typedef struct article{
  char*  title;
  char* info;
  char**  abstract;
  char** category;
  int n_category;
  int n_words;
  char*  url;
}* Article;
```

Figura 2: Estrutura *Article*

inteiro que define o número de artigos e um inteiro para facilitar a alocação de memória, alocando assim vários para artigos de uma só vez.

```
typedef struct vector{
  Article* vector;
  int size;
  int used;
}* Vector;
```

Figura 3: Estrutura *Vector*

4 Análise dos artigos

A utilização das expressões regulares neste trabalho teve em conta a particularidade da ferramenta utilizada, o *Flex*, que por vezes não permite

certas expressões como outras linguagens. Para filtrar os artigos propostos foram definidos estados, estes estados começam todos no *INITIAL* e a partir daí deserola-se o processamento.
Para encontrar o título de um artigo o texto é processado linha a linha e, sempre que é encontrada a *tag* < *title* > é começado a ler-se o conteúdo seguinte. Logo que o conteúdo lido for <> tudo o que está entre estas duas *tags* é escrito na estrutura como título.
No caso de existência de *Infobox* entramos no estado *INFO* e é começada a captação de caracteres logo a seguir ao aparecimento do padrão {*Info/*, captam-se todos os caracteres execepto links, referências. Quando o analisador lexico deteta {} voltamos ao estado inicial.
Para guardar o resumo de cada artigo é iniciado o estado *ABSTRACT* quando o analisador deteta a primeira ocurrência de três pelicas seguidas. Ao entrar neste estado são ignoradas todas os aprecimentos de *({ conteúdo })* bem como {{ *conteúdo* }}. Alguns caracteres também são ignorados na leitura, nomeadamente, *'* — { e o *newline*. Este estado termina logo que seja encontrado um *newline*.
Para guardar as categorias de cada um dos artigos, sempre que é encontrado um início de linha *[[Categoria:* entra-se no estado *CATEGORY* e é lido todo o conteúdo dessa linha até à occurrência de *]]* que marcam o final da categoria. Quando esta marca é encontrada volta-se ao estado inicial para que, caso existam mais categorias processo se repita.

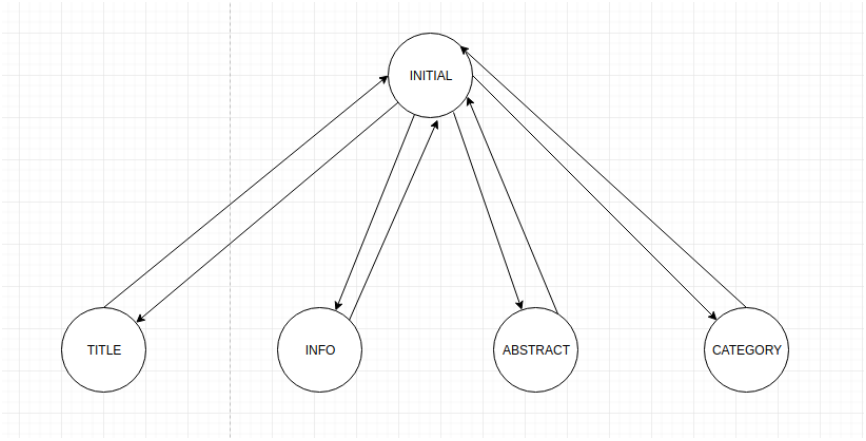


Figura 4: Autómato de estados

Para o auxílio na construção das expressões regulares também foram criadas

5 Representação dos Dados

Depois da filtragem dos artigos através do FLEX, é necessário representar os dados obtidos de uma forma clara e de fácil gestão para o utilizador. Para isso criamos três possíveis visualizações: **HTML, LaTeX e Markdown**.

Começando na exportação em HTML. Este gera sempre um ficheiro (index.html) e depois n outros ficheiros também .html que contém os artigos. Cada artigo tem a sua página, contendo o título, as categorias a que pertence, imagens/informação extra do artigo e por fim um resumo deste. Para além disto, possui ligação para o index e vice-versa por forma a facilitar a leitura e procura de todos os artigos da desejada categoria.
Para isto, o programa á medida que encontra artigos da categoria desejada, imprime o seu conteúdo, formatado, para um ficheiro, enquanto preenche o index.html, que é finalizado no final do programa. A *template* é fixa, mas de fácil ajuste caso seja necessário.

Os formatos latex e Markdown foram criados como extras, caso o utilizador pretendesse um documento de fácil leitura, e simples viusalização dos artigos. Este são formatos que o programa simplesmente filtra os artigos desejados e imprime para um ficheiro, conforme uma *template* já definida.

Para tornar a visualização dos dados mais agradável, no caso do HTML e do Markdown, caso apareçam imagens nas informações extras estas aparecem na página resultante. Por fim, as palavras que vêm entre [[ ]] são realçadas no texto para uma melhor representação do texto real.

6 Conclusões

O grupo conclui que as inconistências muitas vezes apresentadas pelos artigos *XML* contribuiram para uma dificuldade acrescida a este trabalho. Concluímos também que o flex não parece ser a ferramenta mais apropriada devido à falta de certas capacidades tais como o *lookahead ou lookbehind*. Contudo, apesar de nem todos os objetivos propostos terem sido atingidos consideramos que os que foram alcançados foram bem conseguidos por parte do grupo.

7 Anexos

1 Albino Forjaz de Sampaio

Artigo Original
Categorias:

- Heris da mitologia grega;
- Pessoas da Guerra de Troia;
- Personagens da Ilada;
- Semideuses da mitologia greco-romana;

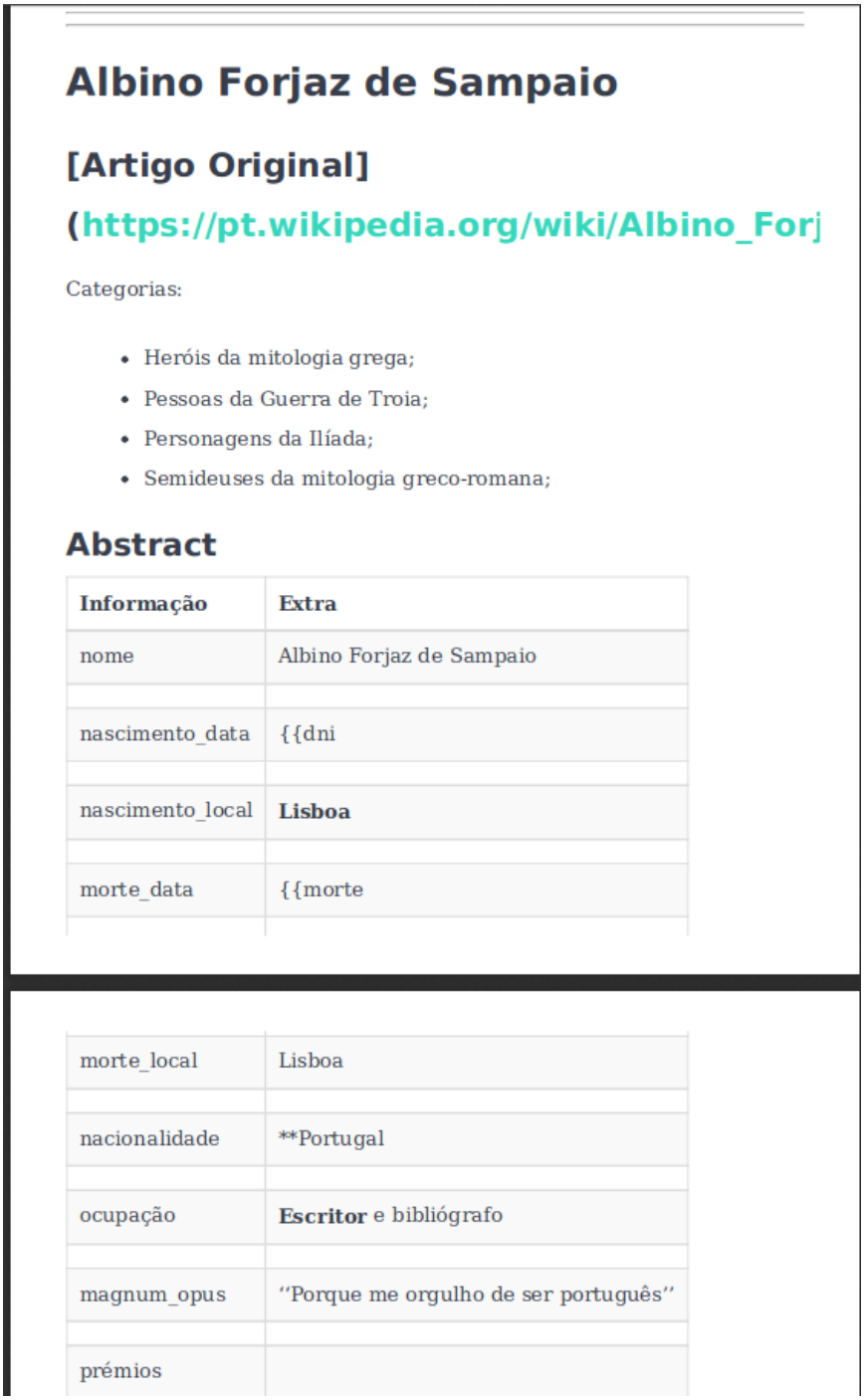
1.1 Abstract

Info	
nome	Albino Forjaz de Sampaio
nascimento$_{d}ata$	dni—19—1—1884—sem idade
nascimento$_{l}ocal$	Lisboa
morte$_{d}ata$	morte—13—3—1949—19—1—1884
morte$_{l}ocal$	Lisboa
nacionalidade	Portugal—Portugus
ocupao	Escritor e bibligrafo
magnum$_{o}pus$	"Porque me orgulho de ser portugus"
prmios	

Albino Maria Pereira Forjaz de Sampaio **Lisboa**, **19 de Janeiro de 1884** Lisboa, **13 de Maro** de **1949**) foi um **escritor** e...

Figura 5: Exmeplo de um artigo representado em Latex

Figura 6: Exmeplo de um artigo representado em HTML



Albino Forjaz de Sampaio

[Artigo Original]

(https://pt.wikipedia.org/wiki/Albino_Forj

Categorias:

- Heróis da mitologia grega;
- Pessoas da Guerra de Troia;
- Personagens da Ilíada;
- Semideuses da mitologia greco-romana;

Abstract

Informação	**Extra**
nome	Albino Forjaz de Sampaio
nascimento_data	{{dni
nascimento_local	**Lisboa**
morte_data	{{morte
morte_local	Lisboa
nacionalidade	**Portugal
ocupação	**Escritor** e bibliógrafo
magnum_opus	''Porque me orgulho de ser português''
prémios	

Figura 7: Exmeplo de um artigo representado em Markdown