Armazenamento e Consulta Relacional de Texto: o caso de "Os Lusíadas"

David Paiva Fernandes
david.paiva.fernandes@gmail.com

Resumo

Este trabalho apresenta um sistema de armazenamento e consulta relacional de texto fazendo uso de um SGBD (Sistema de Gestão de Base de Dados). Tendo por base o poema épico "Os Lusíadas", descreve detalhadamente a metodologia seguida para identificação da gramática do texto ASCII da obra original, bem como apresenta o desenvolvimento dos autómatos finitos de reconhecimento da linguagem da obra. Propõe ainda um modelo de dados para a estruturação de textos e apresenta alguns exemplos de consultas SQL (Structured Query Language) que ilustram algumas das suas potencialidades. Por fim indicam-se algumas pistas de desenvolvimento, nomeadamente no sentido de uma maior generalização do modelo tornando-o eventualmente aplicável a outro tipo de obras.

Palavras-chave: análise de conteúdo, SGDB, LISP, texto, análise lexical

Title: Storage and relational querying of text: "The Lusíads" case

Abstract

This paper presents a system of storage and relational query of text by use of a DBMS (Database Management System). Based on the Portuguese epic poem "Os Lusíadas", we describe in detail the methodology used to identify the grammar of the ASCII text and present the development of finite automata for token recognition. We also propose a data model and present some examples of SQL (Structured Query Language) queries that illustrate some of the capabilities of the system. Finally we present some indications for future development towards a more generalized model in order to make it suitable to other textual works.

Keywords: content analysis, DBMS, LISP, text, lexical analysis

1- Introdução

Segundo Holsti [1969, p.24] toda a comunicação é composta por seis elementos básicos: emissor, processo de codificação, mensagem, canal de transmissão, recetor e processo de descodificação sendo que a *análise de conteúdo* incide sobre o estudo da mensagem qualquer que seja o seu formato: um romance, um diário, um discurso. Embora a metodologia da *análise de conteúdo* vise apenas a mensagem, dos seus resultados podem produzir-se inferências sobre qualquer um dos outros cinco elementos, com três propósitos principais:

inferir sobre características do texto, sobre causas ou antecedentes da mensagem e efeitos da comunicação.

Incluído na *análise de conteúdo* encontramos o estudo quantitativo de textos literários, que ascende aos finais do séc. XIX com períodos de atividade intensa nas décadas de 30 e 80 do séc. XX, em particular com a utilização de meios computacionais para a realização de estudos literários de estilo e atribuição de autoria. Estes sistemas procuram representar numericamente aspectos ou características dos textos através da aplicação de métodos matemáticos precisos e centram a sua atenção, maioritariamente, nas questões de estilo e atribuição de autoria mas podem também ser utilizados para investigar questões interpretativas mais vastas como enredo, género, tema e período. [Hoover, 2008].

Este trabalho apresenta o desenvolvimento de um sistema de armazenamento e consulta relacional de texto fazendo uso de um SGBD. Para tal descreve-se detalhadamente a metodologia seguida desde a identificação da estrutura do texto original, a sua formalização através de expressões regulares e a criação do autómato finito que as reconhece. Propõe-se ainda um modelo de dados que suporta a obra utilizada como teste, ”Os Lusíadas”, bem como se apresentam alguns exemplos de consultas SQL que ilustram as potencialidades do sistema. Por fim, indicam-se algumas pistas de desenvolvimento no sentido de uma maior generalização do modelo, o que o tornaria aplicável a outras obras com outro tipo de organização interna, em poesia ou em prosa e não apenas literárias.

Com vista a criar um ambiente propício ao desenvolvimento de análise de conteúdo de “Os Lusíadas”, na implementação realizada distinguem-se três fases:

- Identificação da estrutura do texto original e Modelação ER (Entity-relationship) e relacional;
- Transformação do texto base (análise lexical) e produção das instruções DML (Data Manipulation Language) para inserção da informação na base de dados;
- Análise de conteúdo com a criação de Scripts DML de exemplo de consulta aos dados.

Este artigo organiza-se da seguinte forma: na secção 2 são descritos os fundamentos teóricos dos autómatos finitos; na secção 3 é modelada a base de dados relacional; na secção 4 é implementada a análise lexical; na secção 5 é realizada a análise de conteúdo e por fim são apresentadas as conclusões na secção 6.

2 - Autómatos Finitos

Começaremos por apresentar alguns dos fundamentos teóricos subjacentes ao trabalho desenvolvido.

Um *alfabeto* é um conjunto não vazio de símbolos distintos e costuma designar-se por Σ. Alguns exemplos de alfabetos são:

- $\Sigma_a = \{0,1\}$ - alfabeto binário
- $\Sigma_b = \{0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f\}$ - alfabeto hexadecimal
- $\Sigma_c = \{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$ - alfabeto inglês

Uma *palavra*, em sentido lato, é uma sequência finita de símbolos de um alfabeto, e recorrendo aos alfabetos referidos acima podemos dar os seguintes exemplos: $\{011010\}, \{0d0a\}, \{was\}, \{saw\}$.

Comprimento de uma palavra e palavra nula: Considera-se que o comprimento de uma palavra é o número de símbolos que a compõem e denota-se por

$$|\,palavra\,| \tag{1}$$

Uma palavra nula (ou vazia) é uma palavra composta por nenhum símbolo, denota-se por ε e considera-se que é uma palavra passível de ser gerada a partir de qualquer alfabeto.

Naturalmente verifica-se que $|\varepsilon| = 0$.

Potências de um alfabeto: Chama-se potência k de um alfabeto, Σ^k, ao conjunto de todas as palavras de comprimento k que podem ser formadas a partir de símbolos do alfabeto, por exemplo:

$$\Sigma_a^2 = \{00,01,10,11\} \tag{2}$$

O conjunto de todas as palavras que se podem formar com os símbolos de um determinado alfabeto denota-se por Σ^*

$$\Sigma^* = \bigcup_{i=0}^{\infty} \Sigma^i \tag{3}$$

Note-se que Σ^0 representa a palavra nula.

A um subconjunto de palavras de um Σ^* chama-se *linguagem* e denota-se por L.

$$L \subseteq \Sigma^* \tag{4}$$

É importante reconhecer que o conjunto L não é formado de uma maneira completamente arbitrária; existe frequentemente uma regra, ou conjunto de regras, que regula a sua formação e de que são exemplo, num certo sentido, as gramáticas de línguas naturais.

Um autómato finito é uma máquina que, como qualquer outra, transforma algo, que inicia a sua laboração e, sob determinadas condições, pára. Um autómato finito processa pois entradas e em cada momento pode verificar-se que o autómato se encontra em um ou mais estados bem determinados. Um ou mais destes estados são chamados de estados finais e uma vez atingido um destes estados, o autómato pára. Neste caso diz-se que o autómato *reconheceu* as entradas processadas.

Um autómato pode parar sem que tenha atingido um estado final e neste caso diz-se que o autómato não reconheceu as entradas processadas.

Formalmente um autómato é um tuplo com 5 elementos

$$A = (Q,\Sigma,\delta,q_0,F) \tag{5}$$

em que $Q = \{p_0, p_1,\ldots,p_n\}$ é o conjunto de estados que o autómato pode assumir, $\Sigma = \{a_0, a_1,\ldots,a_n\}$ é o conjunto de entradas, δ é uma função de transição, $q_0 \in Q$ é o estado inicial do autómato antes de processar qualquer entrada e $F \subseteq Q$ um conjunto não vazio de estados finais, estados em que o autómato pára de processar entradas.

Função de transição faz corresponder um par $\{p_i, a_k\}$ a um estado p_f com $p_i, p_f \in Q$ e $a_k \in \Sigma$.

$$p_f = \delta(p_i, a_k) \tag{6}$$

Diz-se que no estado p_i e em face de uma entrada a_k, o autómato transita para o estado p_f.

Se $\delta(p,a) = \varnothing$ diz-se que o autómato *morre*, significando isso que não reconhece as entradas processadas.

Transição-ε: Existe um tipo especial de transição que não obriga à ocorrência de uma entrada. Numa transição deste tipo, a única condição para que o autómato passe para o estado p_f é encontrar-se no estado p_i.

$$\delta(p_i, \varepsilon) = p_f \tag{7}$$

Note-se que a passagem para o estado p_f é automática e em face de uma transição ε, o autómato encontra-se simultaneamente em dois estados.

Determinismo: Diz-se que um autómato é determinista (DFA - deterministic finit automata) se a cada par $\{p_i, a_k\}$ faz corresponder um e um só p_f:

$$p_a = \delta(p_i, a) \wedge p_b = \delta(p_i, a) \Rightarrow p_a = p_b : p_i, p_a, p_b \in Q, a \in \Sigma \tag{8}$$

Diz-se não-determinista (NFA - nondeterministic finit automata) no caso contrário e incluem-se nesta categoria todos os autómatos com transições-ε.

3 – Implementação da Base de Dados

A ideia de base passa por utilizar as potencialidades de um SGBD no que toca a armazenamento e consulta de informação. A primeira tarefa consiste em passar o texto original de um formato já digital mas relacionalmente desestruturado (ficheiro ASCII) para um formato estruturado que permita o seu armazenamento numa base de dados relacional. Segue-se depois a criação de instruções DML (Data Manipulation Language) para inserção da informação na base de dados.

O problema foi dividido em três fases:

- Identificação da gramática do texto original e Modelação ER e relacional;
- Transformação do texto base (análise lexical) e produção das instruções DML para inserção da informação na base de dados;
- Criação de Scripts DML de exemplo de consulta aos dados.

A estrutura do poema é conhecida e em certo sentido similar a um grande número de obras não apenas literárias, nomeadamente em prosa.

Tabela 1: Estrutura da obra

Estrutura de ''Os Lusíadas''	Estrutura generalizada
Cantos	Capítulos
Estrofes	Parágrafos
Versos	Frases
Palavras	Palavras

Sabemos que o poema é composto por 10 cantos, cada um dos cantos por um número variável de estrofes e cada uma das estrofes por 8 versos. Analisado o texto de base [OSL, 2011] verifica-se que cada canto se inicia pela palavra CANTO seguida de um ESPAÇO e de um número de ordem. Cada estrofe é precedida de um número identificador da sua ordem dentro de cada canto como exemplificado na Tabela 2 (¶ indica mudança de linha).

Tabela 2: Extrato representativo

CANTO 1¶
¶
1¶
As armas e os barões assinalados,¶
Que da ocidental praia Lusitana,¶
(...)
E entre gente remota edificaram¶
Novo Reino, que tanto sublimaram;¶
¶
2¶
E também as memórias gloriosas¶
(...)

Para permitir uma maior generalização optou-se por não limitar quantitativamente a ocorrência de alguns dos elementos. Assim, assumiu-se que o poema poderia ser composto por nenhum, um ou mais cantos e de forma similar no caso das estrofes e chegou-se ao esboço das produções de uma gramática que reconhece o texto escolhido:

Tabela 3: Produções da gramática

```
LUSÍADAS  →  CANTO*
CANTO     →  'CANTO' NÚMERO¶¶+ESTROFE*
ESTROFE   →  NÚMERO¶+VERSO*¶+
VERSO     →  CARACTER+¶
```

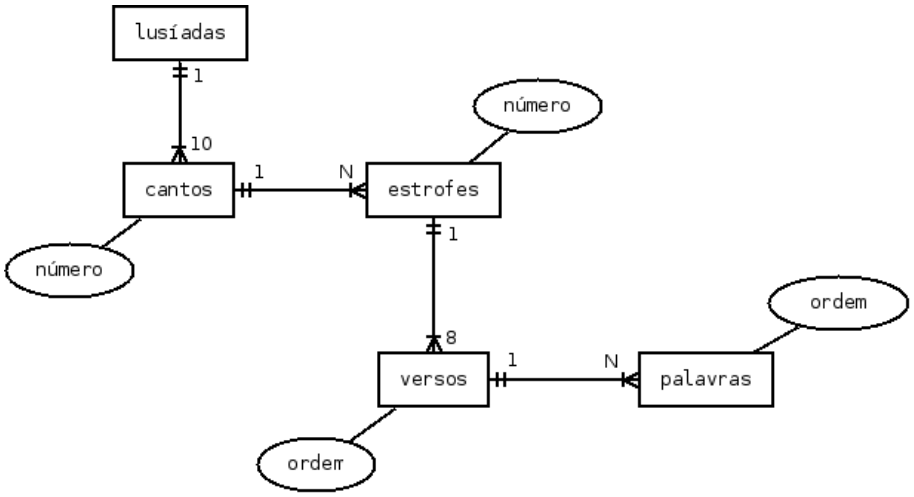

Figura 1: Modelo ER

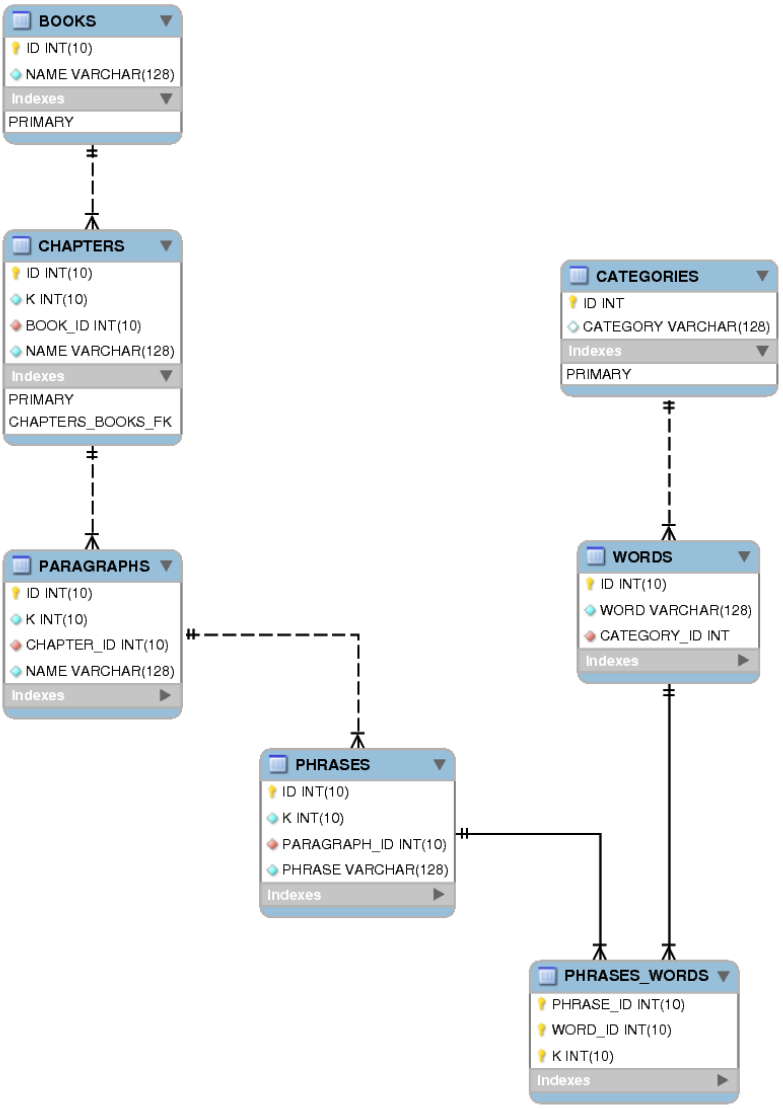

Figura 2: Modelo relacional

Com o objectivo de simplificar algumas das produções, substituíram-se manualmente no texto original as expressões CANTO n por um único carácter, o $, e contemplou-se a possibilidade de ocorrerem mais do que uma mudança de linha consecutiva mantendo-se apenas a obrigatoriedade de os versos serem separados por apenas uma e as estrofes e cantos por duas ou mais.

Obtém-se assim uma expressão regular que incorpora todas as produções referidas na Tabela 3 já com os ajustes referidos.

$$\text{LUSÍADAS} \rightarrow (\$¶¶+(\text{NÚMERO}¶+(\text{CARACTER}+¶)*¶+)*)* \quad (9)$$

A obra apresenta uma estrutura hierárquica regular, o que facilita a visualização imediata dos modelos conceptuais ER e relacional.

4 – Análise Lexical

Uma linguagem aceite por uma expressão regular é também aceite por um autómato não determinista com transições ε (ε-NFA) e estes podem ser transformados em autómatos deterministas (DFA) [Hopcroft et al., 2001, p.101,60]. Assim, com o objectivo de encontrar um ε-NFA que aceitasse a expressão regular (9) começamos por definir autómatos ε-NFA independentes para cada uma das expressões regulares referidas na Tabela 3, utilizando para tal o método de Thomson [Xing, 2004]. Os autómatos resultantes podem ser vistos nas figuras 3, 4, 5 e 6.

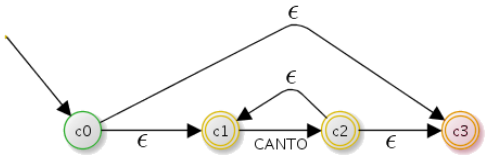

Figura 3 : LUSÍADAS=CANTO*

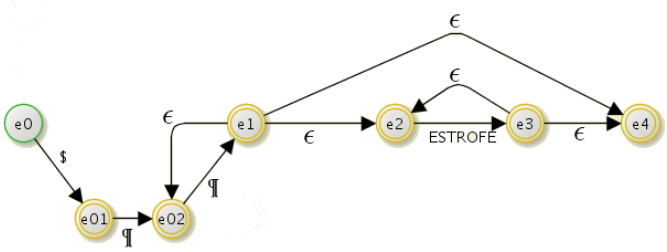

Figura 4 : CANTO=$¶¶+ESTROFE*

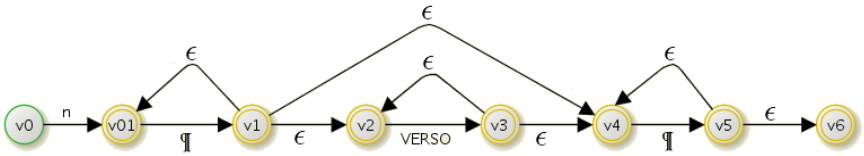

Figura 5 : ESTROFE=n¶+VERSO*¶+

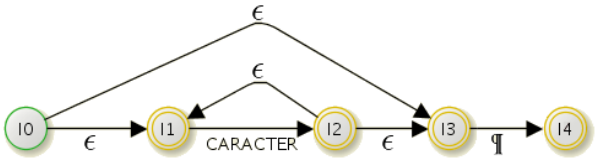

Figura 6 : VERSO=CARÁCTER+¶

Procedeu-se à união dos vários autómatos, substituindo sucessivamente os elementos não terminais de cada autómato por um autómato mais específico.

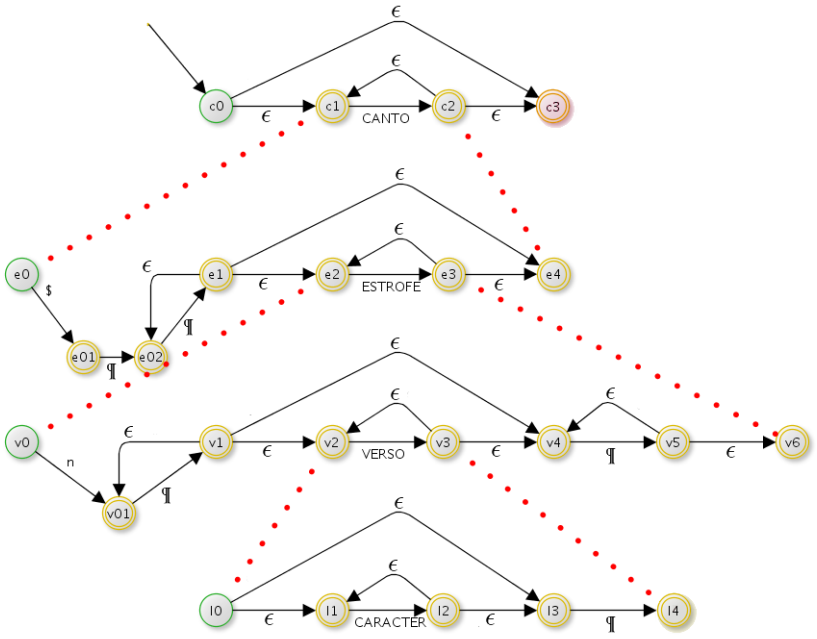

Figura 7 : União

Identificando os estados redundantes

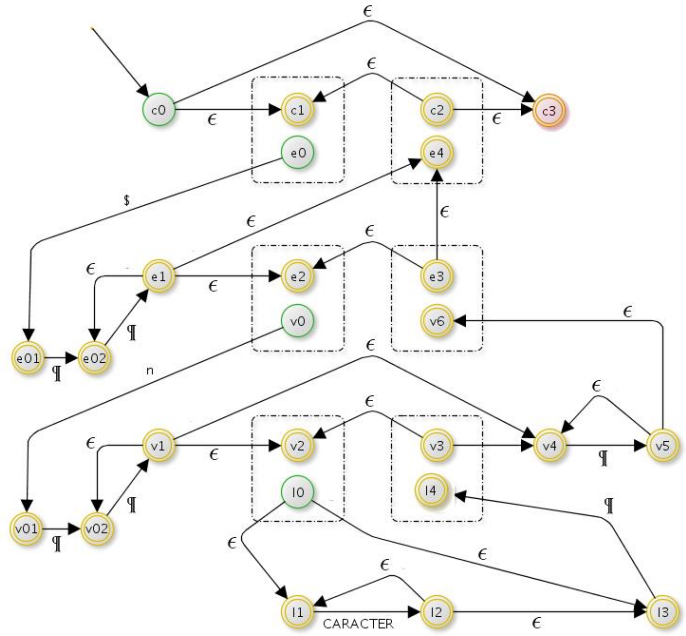

Figura 8 : Estados redundantes

Procedendo-se à sua simplificação, conseguimos determinar um autómato finito sem elementos não terminais.

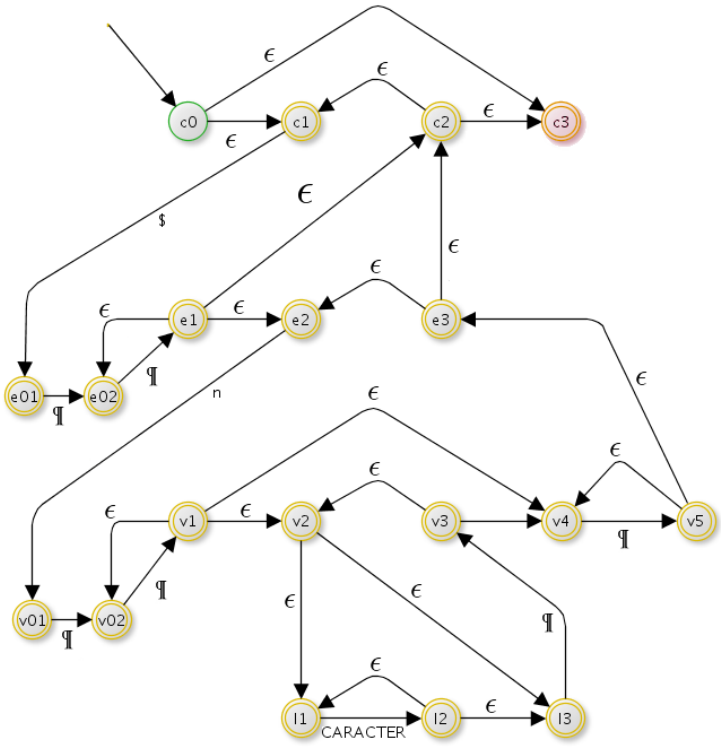

Figura 9 : ε -NFA final

De seguida, procede-se à eliminação das transições ε [Hopcroft et al., 2001, p.77] obtendo-se daí um DFA que permite a implementação direta em código, trabalho esse efetuado em *Common Lisp*. Note-se que a parte da análise lexical que trata de identificar cada uma das palavras (*tokenize*) foi implementada num DFA distinto não descrito neste documento.

A ideia de base para a implementação do DFA foi a de conseguir representar o texto original numa linguagem de tipo "MARKUP" (como por exemplo XML - eXtensible Markup Language) e a escolha da *Common Lisp* como linguagem de implementação, não tendo sido determinada por este objetivo, acabou por orientar a escolha da linguagem de "MARKUP" já que apenas foi necessário dar devido uso às listas ligadas e às funcionalidades nativas de processamento de listas que a LISP possui e definiu-se uma linguagem baseada em *Property Lists* [CLP, 2007] que agrupa a informação numa estrutura de tipo hierárquico semelhante ao XML.

Exemplo da primeira estrofe do primeiro canto:

```
(:ROOT
 (:B
  (:C 1
   (:E 1
    (:V 1 (:VALL "As armas e os barões assinalados,")
          1 2 3 4 5 6)
    (:V 2 (:VALL "Que da ocidental praia Lusitana,")
          7 8 9 10 11)
    ...
    (:V 7 (:VALL "E entre gente remota edificaram")
          3 32 33 34 35)
    (:V 8 (:VALL "Novo Reino, que tanto sublimaram;")
          36 37 7 38 39)))))
```

em que **:C** corresponde ao CANTO, **:E** à estrofe e **:V** ao verso.

Cada verso é composto por uma STRING identificada com :VALL que contém o verso completo tal como encontrado no texto original, bem como uma lista de índices que apontam para uma lista de palavras únicas.

A razão porque se mantêm as duas representações prende-se com o facto de não termos efetuado tratamento dos sinais de pontuação, pelo que a forma mais simples de podermos reproduzir a obra a partir do modelo seria através do armazenamento do verso no seu formato original; foi o que se fez e é essa a sua única utilidade; todas as outras consultas são efetuadas sobre a tabela de palavras únicas.

O algoritmo que implementa o DFA de análise lexical baseia-se no facto de as funções em LISP serem *First Class Objects* e como tal poderem ser argumentos de outras funções.

Um ciclo principal serve como *scanner* e itera sobre cada um dos caracteres (sem recuo) e invoca a função correspondente ao estado corrente do DFA passando o carácter como argumento. Cada função de estado implementa as suas transições, alterando para tal a função de estado corrente e passando de novo o controle à função principal.

Após a transformação da obra do formato ASCII original para o formato intermédio estruturado acima referido, geramos as instruções SQL de inserção na base de dados.

5 - Análise de Conteúdo

O objetivo deste trabalho é o de permitir efectuar operações relacionadas com análise de conteúdo sobre “Os Lusíadas", pelo que se apresentam de seguida algumas consultas efetuadas e alguns resultados obtidos. Os exemplos que apresentamos pretendem demonstrar não apenas resultados quantitativos como sejam contagens, mas também apresentar resultados que impliquem ou demonstrem análises relacionais.

Começamos com alguns resultados simples a que se seguem resultados relacionais, para os quais incluímos as instruções SQL que lhes dão origem bem como representações gráficas que os ilustram.

Alguns factos numéricos

- O poema estende-se por 10 cantos, num total de 1102 estrofes de 8 versos cada uma. O canto mais curto é o 7º com 87 estrofes e o mais longo o 10º com 156 sendo 110.2 o número médio de estrofes por canto.
- São utilizadas 9018 palavras distintas num total de 55400 palavras e 247434 caracteres.
- 4930 palavras figuram uma única vez em toda a obra.
- A palavra mais vezes repetida numa única estrofe é a palavra ’que’ havendo 1 estrofe com 10 repetições, 2 com 9, 3 com 8, 11 com 7 e 16 com 6.
- São duas as palavras mais longas constituídas por 16 letras, aparecem ambas em versos do Canto 9º e são elas: determinadamente e impossibilidades.
- A palavra ’não’ surge por 585 vezes enquanto ’sim’ apenas uma.
- Os versos mais longos são compostos por 11 palavras enquanto os mais curtos, em número de 39, possuem apenas 3 palavras.

Palavras distintas por comprimento

Nesta consulta procuramos conhecer a distribuição de frequência de palavras tendo em conta o seu comprimento, isto é, o número de caracteres.
O universo da análise consiste no conjunto de palavras distintas e não na ocorrência total, pelo que parece natural que a distribuição seja normal.

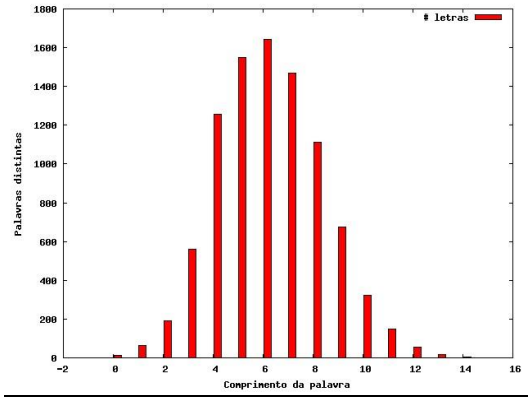

Figura 10 : Distribuição de palavras distintas por número de caracteres

```
SELECT LENGTH(WORD) TAMANHO,
       COUNT(*) FREQUENCIA_ABSOLUTA
FROM WORDS
GROUP BY LENGTH(WORD)
```

Letras em início e fim de palavra

Nesta consulta procuramos conhecer a distribuição de palavras tendo em conta as letras de início e fim. Nota para a elevada frequência de palavras terminadas nas vogais ***a*** e ***o*** e na consoante ***s***.

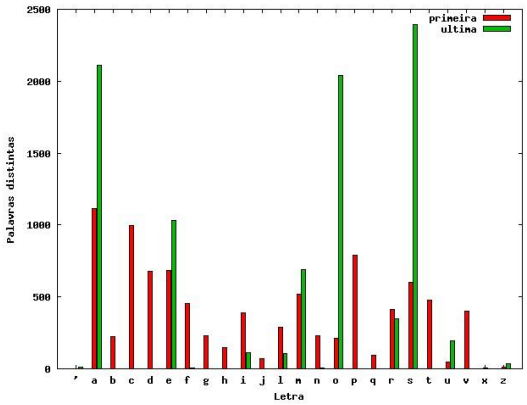

Figura 11 : Distribuição de palavras distintas por caracteres de início e fim

```
; letras em início de palavra
SELECT SUBSTRING(WORD,1,1) PRIMEIRA_LETRA,COUNT(*)
FREQUENCIA_ABSOLUTA
FROM WORDS
GROUP BY SUBSTRING(WORD,1,1)

; letras em fim de palavra
SELECT SUBSTRING(WORD,LENGTH(WORD),1) ULTIMA_LETRA,COUNT(*)
FREQUENCIA_ABSOLUTA
FROM WORDS
GROUP BY SUBSTRING(WORD,LENGTH(WORD),1)
```

Palavras mais frequentes com mais de 3 letras

Neste caso procuramos conhecer as palavras mais utilizadas e limitamos o universo de pesquisa às palavras com 4 ou mais letras.

Tabela 4 : Palavras mais frequentes

Palavra	Frequência
mais	283
quem	232
gente	230
terra	222
para	205
como	195
onde	177
grande	138
tanto	132
seus	107
quando	107
assim	102
porque	102
mundo	101
este	86
reino	86

```
SELECT W.WORD PALAVRA, COUNT(*) K
FROM PHRASES_WORDS PW
  INNER JOIN WORDS W
    ON PW.WORD_ID=W.ID
WHERE LENGTH(W.WORD)>3
GROUP BY PW.WORD_ID, W.WORD
ORDER BY COUNT(*) DESC
```

Pesquisa de antónimos

Nestas consultas, que evidenciam já características não apenas quantitativas mas também de ordem semântica, procuramos comparar a frequência de palavras com conotação positiva por oposição a palavras com conotação negativa: belo/feio, alto/baixo, grande/pequeno. Verifica-se que as palavras ditas *positivas* são em muito maior frequência o que talvez se explique pelo carácter de exaltação da obra em causa.

Tabela 5 : Frequência de "antónimos"

Palavra	Frequência
bela	24
belo	7
feia	12
feio	7

Palavra	Frequência
alta	33
alto	37
baixa	2
baixo	5

Palavra	Frequência
grande	138
grandemente	4
grandes	49
grandeza	4
grandíloqua	1
grandíloquo	1
grandíssima	6
grandíssimo	3
grandíssimos	1
grandura	1
pequena	10
pequenas	1
pequeno	10
pequenos	7

Palavra	Frequência
certa	10
certo	36
errada	1
errado	1

```
SELECT W.WORD PALAVRA, COUNT(*) K
FROM PHRASES_WORDS PW
  INNER JOIN WORDS W
    ON PW.WORD_ID=W.ID
WHERE W.WORD LIKE 'bel_' OR W.WORD LIKE 'fei_'
GROUP BY PW.WORD_ID, W.WORD
ORDER BY W.WORD
```

Rácio palavras distintas/palavras totais por canto

Nesta consulta procuramos um indicador de "criatividade" calculando o rácio entre o número de palavras distintas versus número total de palavras por canto. Verifica-se que o rácio oscila entre os 32% e os 38% sem que nenhum CANTO se destaque de forma acentuada.

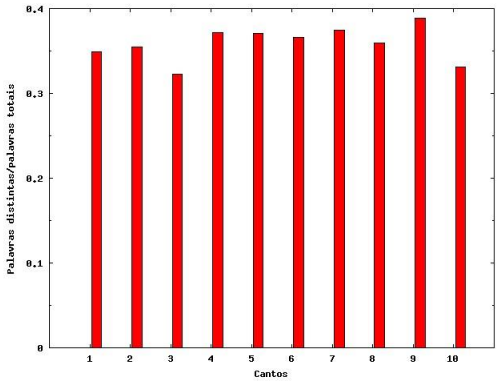

Figura 12 : Rácio de palavras únicas por canto

```
SELECT A.CANTO, A.K/B.K
FROM
  (SELECT SQ.CANTO, COUNT(*) K
  FROM (select DISTINCT C.ID CANTO, PW.WORD_ID
    FROM CHAPTERS C
      INNER JOIN PARAGRAPHS P ON C.ID=P.CHAPTER_ID
      INNER JOIN PHRASES PH ON P.ID=PH.PARAGRAPH_ID
      INNER JOIN PHRASES_WORDS PW ON PH.ID=PW.PHRASE_ID) SQ
  GROUP BY SQ.CANTO) A,
  (SELECT SQ.CANTO, COUNT(*) K
  FROM (select C.ID CANTO, PW.WORD_ID
    FROM CHAPTERS C
      INNER JOIN PARAGRAPHS P ON C.ID=P.CHAPTER_ID
      INNER JOIN PHRASES PH ON P.ID=PH.PARAGRAPH_ID
      INNER JOIN PHRASES_WORDS PW ON PH.ID=PW.PHRASE_ID) SQ
  GROUP BY SQ.CANTO) B
WHERE A.CANTO=B.CANTO
```

Repetição de palavras num mesmo verso

Com esta consulta procuramos conhecer palavras (e o verso onde ocorrem) que surjam mais do que duas vezes num mesmo verso.

PHRASE	WORD	COUNT(*)
Que costumes, que Lei, que Rei teriam?	que	3
De Argos, da Hidra a luz, da Lebre e da Ara.	da	3
Que costumes, que lei, que terra tinha.	que	3
De desesperação, de fome, de ira?	de	3
Que triunfos, que palmas, que vitórias?	que	3
De fazendas, de reinos e de impérios:	de	3
Que mortes, que perigos, que tormentas,	que	3
Os velhos e os meninos os seguiam,	os	3
Sem marido, sem Reino, e sem ventura.	sem	3
De ouro, e de seda e de algodão tecida.	de	3
De Quíloa, de Mombaça e de Sofala;	de	3
Que, por que não passassem, sem que nela	que	3

```
SELECT PW.PHRASE_ID, P.PHRASE, W.WORD, COUNT(*)
  FROM WORDS W
    INNER JOIN PHRASES_WORDS PW
      ON W.ID=PW.WORD_ID
    INNER JOIN PHRASES P
      ON PW.PHRASE_ID=P.ID
    INNER JOIN PARAGRAPHS PG
      ON P.PARAGRAPH_ID=PG.ID
    INNER JOIN CHAPTERS C
      ON PG.CHAPTER_ID=C.ID
  WHERE C.BOOK_ID=1 /* os Lusiadas */
    AND LENGTH(W.WORD)>1
GROUP BY PW.PHRASE_ID, P.PHRASE, W.WORD
HAVING COUNT(*) >2
ORDER BY COUNT(*) DESC
```

Identificação indirecta de categorias morfológicas

Os dois exemplos seguintes, sem recorrer a outros meios como sejam dicionários, procuram identificar categorias morfológicas, no caso *verbos*, *substantivos* e *adjetivos*, através de dois métodos diferentes, relacionados com co-ocorrência de palavras. Neste primeiro caso procuramos identificar verbos, pesquisando palavras terminadas na letra '*r*' (verbo no infinitivo) mas cujo radical surja também em palavras terminadas no sufixo '*mos*', forma verbal do presente do indicativo ou pretérito perfeito na primeira pessoa do plural.

abrir	achar	alevantar	andar	apartar	buscar	chamar	chegar
deixar	der	descobrir	desembarcar	entender	entrar	enxergar	esperar
estar	fazer	ferir	fizer	for	houver	ir	levar
lograr	navegar	ouvir	partir	passar	poder	por	puder
querer	receber	saber	servir	ter	tomar	tornar	vencer
ver	vier	vir					

```
SELECT DISTINCT W.WORD
  FROM WORDS W
    INNER JOIN PHRASES_WORDS PW
      ON W.ID=PW.WORD_ID
    INNER JOIN PHRASES P
      ON PW.PHRASE_ID=P.ID
    INNER JOIN PARAGRAPHS PG
      ON P.PARAGRAPH_ID=PG.ID
    INNER JOIN CHAPTERS C
      ON PG.CHAPTER_ID=C.ID
  WHERE C.BOOK_ID=1 /* os Lusíadas */
     AND W.WORD LIKE '%R'
     AND CONCAT(SUBSTR(W.WORD,1,LENGTH(W.WORD)-1),'MOS') IN
       (SELECT W.WORD PALAVRA
         FROM WORDS W
           INNER JOIN PHRASES_WORDS PW
             ON W.ID=PW.WORD_ID
           INNER JOIN PHRASES P
             ON PW.PHRASE_ID=P.ID
           INNER JOIN PARAGRAPHS PG
             ON P.PARAGRAPH_ID=PG.ID
           INNER JOIN CHAPTERS C
             ON PG.CHAPTER_ID=C.ID
         WHERE C.BOOK_ID=1 /* os Lusíadas */
            AND W.WORD LIKE '%MOS')
ORDER BY W.WORD
```

No segundo caso, utilizando associações sintagmáticas [Rapp, 2002], procuramos identificar substantivos e adjetivos pesquisando palavras precedidas de artigo '*o*', '*a*', '*um*' ou '*uma*', e que ocorram também na forma plural, isto é, com sufixo '*s*'.

As primeiras 100 palavras surgidas:

tanto	força	fama	musa	marte	vitória	cítara	seu
do	coroa	vária	áurea	vista	maneira	roupa	roda
meu	muita	usada	ferro	fogo	vontade	doutra	praia
bandeira	quanto	fúria	escura	outra	rica	fortaleza	toda
fonte	fresca	neve	vira	via	nossa	pouca	divina
bruta	estranheza	âncora	outro	cova	vela	linda	presa
grande	morte	terra	cujo	lei	forte	formosa	adarga
clara	teu	memória	cabeça	mãe	língua	seta	leva
montanha	frota	roxa	famosa	ilha	má	pedra	ela
serra	suspeita	gente	costa	pequena	forma	leda	vida
armada	cidade	nova	causa	companhia	tempo	deusa	porta
sua	pintura	lusitana	parte	noite	guerra	certa	água
guerreira	nau	glória	verdade				

```
SELECT DISTINCT TALVEZ_SUBSTANTIVOS.PALAVRA FROM
  (SELECT PW.PHRASE_ID VERSO_ID, W.WORD PALAVRA, PW.K ORDEM
  FROM WORDS W
    INNER JOIN PHRASES_WORDS PW
      ON W.ID=PW.WORD_ID
    INNER JOIN PHRASES P
      ON PW.PHRASE_ID=P.ID
    INNER JOIN PARAGRAPHS PG
      ON P.PARAGRAPH_ID=PG.ID
    INNER JOIN CHAPTERS C
      ON PG.CHAPTER_ID=C.ID
  WHERE C.BOOK_ID=1 /* os Lusiadas */) TALVEZ_SUBSTANTIVOS
INNER JOIN
  (SELECT PW.PHRASE_ID VERSO_ID, W.WORD PALAVRA, PW.K ORDEM
  FROM WORDS W
    INNER JOIN PHRASES_WORDS PW
      ON W.ID=PW.WORD_ID
    INNER JOIN PHRASES P
      ON PW.PHRASE_ID=P.ID
    INNER JOIN PARAGRAPHS PG
      ON P.PARAGRAPH_ID=PG.ID
    INNER JOIN CHAPTERS C
      ON PG.CHAPTER_ID=C.ID
  WHERE C.BOOK_ID=1 /* os Lusiadas */
  AND W.WORD IN ('A','O','UM','UMA')) ARTIGOS
ON   TALVEZ_SUBSTANTIVOS.VERSO_ID=ARTIGOS.VERSO_ID
 AND TALVEZ_SUBSTANTIVOS.ORDEM=ARTIGOS.ORDEM+1
 AND CONCAT(TALVEZ_SUBSTANTIVOS.PALAVRA,'S') IN
  (SELECT W.WORD PALAVRA
  FROM WORDS W
    INNER JOIN PHRASES_WORDS PW
      ON W.ID=PW.WORD_ID
    INNER JOIN PHRASES P
      ON PW.PHRASE_ID=P.ID
    INNER JOIN PARAGRAPHS PG
      ON P.PARAGRAPH_ID=PG.ID
    INNER JOIN CHAPTERS C
      ON PG.CHAPTER_ID=C.ID
  WHERE C.BOOK_ID=1 /* os Lusiadas */
      AND W.WORD LIKE '%S')
```

6 – Conclusão

Apresentámos neste artigo o desenvolvimento de uma ferramenta para efectuar *análise de conteúdo* sobre texto literário, no caso, "Os Lusíadas". De alguma maneira, código e modelo de dados desenvolvidos estão limitados pela obra escolhida como exemplo, mas pensamos que apresentámos ideias que sugerem a validade do modelo a outros textos.

Fazendo uso de autómatos finitos e das potencialidades de um SGBD, propomos um método de representar o texto literário através de um modelo de dados relacional, o que permite o seu armazenamento estruturado e assim a consulta relacional do seu conteúdo.

Seria de grande utilidade permitir a execução de inquéritos de ordem qualitativa tendo em conta, por exemplo, a categoria morfológica das palavras, categorização essa já contemplada no modelo de dados na tabela CATEGORIES mas não tratada nem utilizada.

A estrutura em três níveis da obra utilizada (cantos/capítulos, estrofes/parágrafos e versos/frases) particulariza o modelo, pelo que seria também vantajoso desenvolver um modelo mais generalizado, sem limite de níveis de forma a suportar estruturas mais complexas: partes, secções, subsecções, parágrafos, frases, etc.

O texto base escolhido contém informação sobre partes do texto em discurso direto através da sua sinalização com aspas (“), bem como apartes entre parênteses. Também neste aspecto, nada foi feito para tratar essa informação e permitir, por exemplo, obter todo um bloco de texto em discurso direto.

Em alguns casos, os autómatos implementados resultam específicos para a ”versão” d”Os Lusíadas” utilizada e seria de grande interesse providenciar uma forma de ultrapassar esta limitação.

Quanto à exploração dos dados, afinal o objectivo do sistema, desenvolvemos uma aplicação WEB que permite a execução de QUERIES à base de dados. A sua utilização exige conhecimentos de SQL.

Conhecidas as potencialidades de consulta relacional, resta portanto apelar à imaginação e encontrar predicados de pesquisa com interesse e a partir de factos quantitativos lançar questões de ordem qualitativa cuja resposta ajude a uma melhor compreensão do texto, como por exemplo, que razão haverá para que a palavra **’sim’** figure apenas uma única vez em toda a obra.

Agradecimentos
O autor gostaria de mencionar que para este artigo muito contribuíram os conhecimentos adquiridos nas unidades curriculares do Prof. Luís Cavique e do Prof. Jorge Morais do curso da Licenciatura em Informática.

Bibliografia

CLP, (2007). The property lists. http://www.cs.cmu.edu/Groups/AI/html/cltl/clm/node108.html.

Holsti, O. (1969). Content analysis for the social sciences and humanities. Addison-Wesley Pub. Co., [online], http://www.questia.com/PM.qst?a=o&d=54363997, 21 de Setembro de 2011

Hoover, D.L. (2008). Quantitative analysis and literary studies. http://www.digitalhumanities.org/companionDLS/

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to Automata Theory, Languages, and Computation. Addison Wesley, 2 edition.

OSL, (2011). Os Lusíadas. http://www.fullbooks.com/Os-Lus-adas-Os-Lusiadas1.html.

Rapp, R. (2002). The computation of word associations: comparing syntagmatic and paradigmatic approaches. In Proceedings of the 19th international conference on Computational linguistics - Volume 1, COLING'02, pag 1-7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Xing, G. (2004). A simple way to construct nfa with fewer states and transitions. In Proceedings of the 42nd annual Southeast regional conference, ACM-SE 42, pages 214–218, New York, NY, USA. ACM.

Programador de computadores, particularmente interessado em linguagens de programação, da RPG-II à C#, e em pesquisa, transformação e organização de informação, nomeadamente não estruturada. Uma breve passagem pela formação profissional revelou a paixão pelo ensino e o desejo de desenvolver trabalho nessa área. Frequenta o 3º ano da Licenciatura em Informática da Universidade Aberta.