

Processamento de Linguagens e Compiladores(3º ano de LCC)

**Trabalho Prático 4**

Relatório de Desenvolvimento

Bruno Jardim  
(A91680)

Eduardo Freitas  
(A91643)

Pedro Fernandes  
(A91699)

10 de novembro de 2021

## Resumo do Trabalho Prático

Para este trabalho prático foi-nos proposto a filtração de um documento BibTeX, que é uma ferramenta de formatação de citações bibliográficas em documento L<sup>A</sup>T<sub>E</sub>X. O formato usado separa as informações conforme pedido no enunciado ou seja, filtra toda a informação não necessária e apresenta a tag e a chave que representam o texto filtrado. Assim sendo, também fornece ao utilizador o autor e o título de todas as obras filtradas. Para além desta informação, é também apresentado um índice de autores com as chaves que representam todas as suas obras.

## Um Processador de BibTeX

**Área:** Processamento de Linguagens

**Enquadramento:** Processamento de texto

**Problema 1:** Pretendemos filtrar as categorias e fazer a contagem das mesmas.

**Problema 2:** Complementar o procedor anteriormente criado de forma a apresentar os autores e título das obras.

**Problema 3:** Criar um índice com os autores e as *tags* únicas dos textos escritos pelo mesmo.

**Problema 4:** Construir um grafo com as relações dos autores

**Objetivo:** O objetivo do relatório é demonstrar a nossa abordagem e as decisões tomadas para a resolução do problema em questão, explicando e justificando cada passo dado.

# Capítulo 1

## Informação a tratar

Neste capítulo iremos explorar um exemplo para demonstrar o tipo de informação que o nosso processador consegue tratar. Num documento BibTeX temos um *bloco* que começa por uma *tag* que começa por "@" seguida de uma palavra, depois terá uma *chave* única que nos permite facilmente encontrar uma obra específica, de seguida terá informação da obra. Com esta informação iremos preceder a uma contagem de ocorrências de cada *tag*, algo que de seguida é colocado num documento HTML.

## Capítulo 2

# Filtragem e tratamento da informação

Neste capítulo analisaremos o desenvolvimento do nosso programa bem como abordaremos as decisões que tivemos de tomar na realização do mesmo. Sendo assim, vamos explicar como é que a informação é filtrada e ainda como é realizado o armazenamento da mesma. O primeiro processo de filtramento é detetar um *bloco*, após a deteção teremos que filtrar as diversas informações que necessitamos.

### 2.1 Alínea a) : Análise e contagem de tags.

Para este efeito usamos uma ER para detetar quando encontramos uma tag e usamos a função *re.split()* da biblioteca *re*, para obtermos a tag não formatada. Formatamos as tags a partir da função *.title()* do Python para não haver variações no formato e efetuamos uma contagem das ocorrências.

### 2.2 Alínea b) : Filtragem de chaves, autores e títulos.

A partir da mesma ER referida em 2.1 usamos a segunda parte do *re.split()* para obtermos a chave única de cada obra.

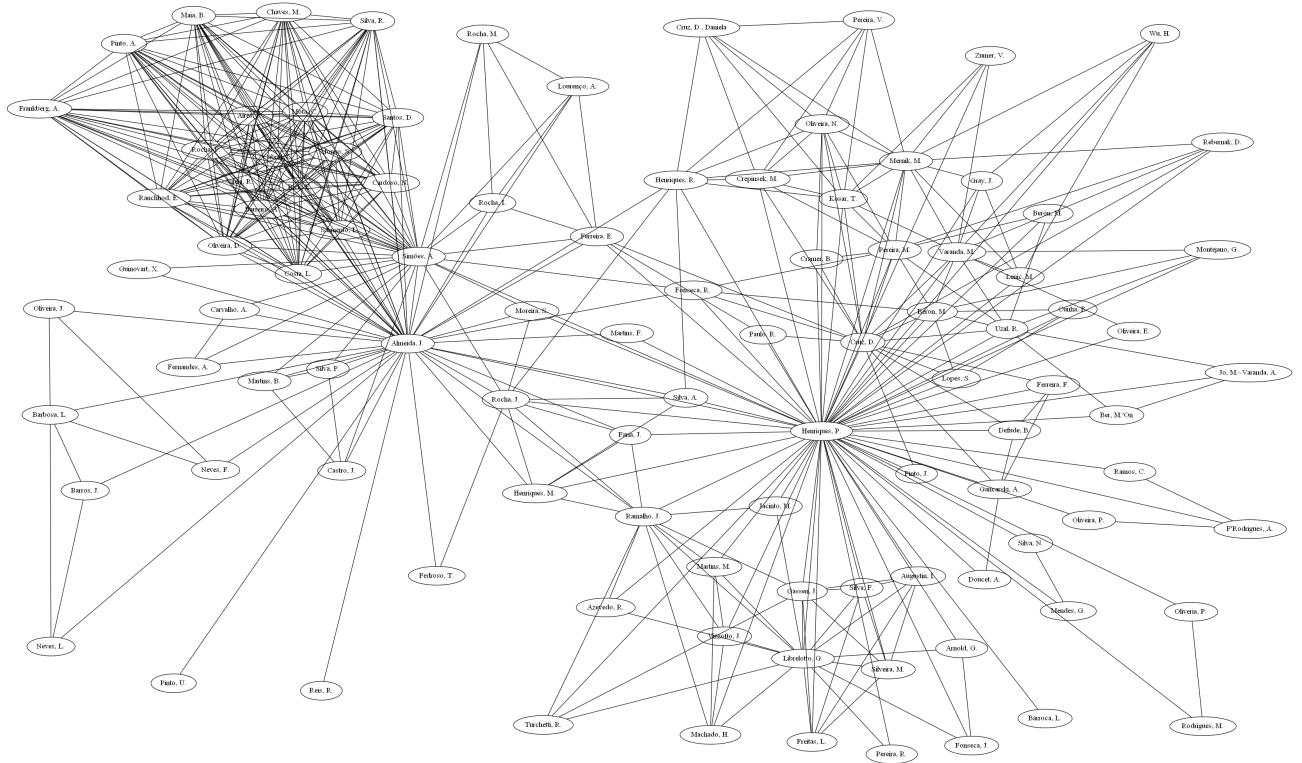
De seguida procedemos a filtrar a informação necessária (autor/autores e título), e.g., *author = {Daniela da Cruz and Pedro Rangel Henriques}*, seguidamente adicionamos a informação a um dicionário estruturado da seguinte forma `{ 'tag' : { 'chave': { { 'author': 'nome do autor' }, { 'title': 'titulo texto' } } }`

### 2.3 Alínea c) : Índice de autores.

Para a resolução deste exercício optamos pela criação de um dicionário que associa os autores às respetivas chaves, através da reutilização do dicionário anteriormente criado na alínea b). Neste caso, este novo dicionário possui os nomes ainda não formatados da maneira que pretendíamos, ou seja, para além de ainda não estarem individualizados também ainda se encontram no formato original. Desta forma, para realizarmos a formatação dos nomes foi feito um *re.split()* nos "And", obtendo assim uma lista com todos os nomes desagregados. De seguida, os nomes foram filtrados de modo a apresentarem a formatação, e.g., *"Henriques, P."*.

Para concluir, criamos um dicionário onde as *keys* são os autores com os nomes formatados e os *values* uma lista com todas as chaves dos textos em que o autor esteve envolvido.

## 2.4 Alínea d) : Grafo de relacionamento entre autores.



Para ser possível criar este grafo usamos o dicionário utilizado em 2.3.

O processo para estabelecer as relações é o seguinte:

- Percorrer os autores todos.
- Para cada autor, percorrer todas as chaves.
- Percorrer todos os autores e verificar se eles têm a mesma chave.
- Caso sim criamos um *tuplo* com o nome dos dois autores e armazenamo-lo numa lista.
- Finalmente passamos os *tuplos* ao *PyGraphViz* e o grafo é construído.

## Capítulo 3

# Apresentação da Informação Filtrada

Neste último capítulo, vamos demonstrar com a ajuda de um exemplo a utilização do nosso programa. Assim sendo, utilizaremos um bloco aleatório para comprovar as formatações realizadas pelo nosso processador.

```
@Book{RH02,
  author=   {José Carlos Ramalho and Pedro Rangel Henriques},
  title =   {XML \& XSL: da teoria à prática},
  publisher = {Editora FCA},
  series =   {Série Tecnologias de Informação ISBN-972-722-347-8},
  edition=   {1st Ed.},
  month =    {Oct},
  year  =    {2002}
}
```

Através dos processos descritos no Capítulo 2 quando tratamos a informação obtemos o seguinte output no código HTML.

```
@Book:1
      RH02
              Author: José Carlos Ramalho And Pedro Rangel Henriques
              Title: Xml \& Xsl: Da Teoria À Prática
```

Através dos processos descritos na alínea 2.3 obtemos o seguinte output, como podemos observar a chave *RH02* está presente no índice do autor *Ramalho, J.* e no do autor *Henriques, P.*

```
Ramalho, J. ....
['Ramalho95', 'Ramalho96', ..., 'LARH09', 'RH02']
```

```
Henriques, P. ....
['graminteractivas1990', 'Ramalho95', ..., 'RRH02', 'ORH06', 'RH02', 'cruz09']
```

## Capítulo 4

# Conclusão

Para concluir, ao realizar este trabalho desenvolvemos as nossas capacidades de interpretação e tratamento de informação. A utilização de todas as ferramentas necessárias para a realização deste projecto ajudou-nos a desenvolver a nossa capacidade de filtração de informação, sendo esta competência algo muito benéfico para qualquer programador.