

Processamento de Linguagens (3º ano de Curso)

**Trabalho Prático 2**

Relatório de Desenvolvimento

Grupo 43

Gonçalo Esteves  
(A85731)

João Araújo  
(A84306)

Rui Oliveira  
(A83610)

28 de Junho de 2020

## **Resumo**

Este relatório visa a documentar o nosso trabalho, que se baseia na implementação de um programa que guarda informação no domínio das *Humanidades Digitais* e gera a mesma em HTML, num formato de cadernos de anotações. Paralelamente, tem-se como objetivo demonstrar a capacidade de escrever gramáticas independentes de contexto para criar Linguagens de Domínio Específico e desenvolver processadores de linguagens segundo o método da tradução dirigida pela sintaxe, suportado numa gramática tradutora, utilizando geradores de compiladores como o par flex/yacc no contexto da Unidade Curricular.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Análise e Especificação</b>	<b>3</b>
2.1	Descrição informal do problema . . . . .	3
2.2	Especificação dos Requisitos . . . . .	3
<b>3</b>	<b>Concepção/desenho da Resolução</b>	<b>4</b>
3.1	Estruturas de dados . . . . .	4
3.2	Flex . . . . .	4
3.3	Yacc . . . . .	5
<b>4</b>	<b>Codificação e Testes</b>	<b>6</b>
4.1	Alternativas, Decisões e Problemas de Implementação . . . . .	6
4.2	Modo de Utilização . . . . .	6
4.3	Testes realizados e Resultados . . . . .	6
<b>5</b>	<b>Conclusão</b>	<b>9</b>

# Capítulo 1

## Introdução

Integrado na cadeira de *Processamento de Linguagens*, foi desenvolvido o segundo trabalho prático. Segundo a fórmula dada, o enunciado atribuído foi o segundo, que pedia o desenvolvimento de Linguagens de Desenvolvimento Específico para um caderno de anotações em *Humanidades Digitais*, sendo que o caderno é composto por diversos pares (documento, triplos). Para a realização deste projeto, exploramos a utilização das ferramentas do Flex e do Yacc e, além disso, trabalhamos com geração de páginas HTML.

## Capítulo 2

# Análise e Especificação

### 2.1 Descrição informal do problema

Tal como já fora referido, o enunciado deste projeto foi o segundo da lista de enunciados concedida, que se encontra dividido na elaboração de duas componentes:

1. Escrever uma gramática para definir a DSL de um caderno de anotações;
2. Construir um processador que reconheça e valide cadernos de anotação, escritos na DSL definida acima, gerando um site HTML de acordo com as seguintes regras:
  - 2.1. Criar uma página por cada elemento (conceito) - tópico, sujeito, objeto - que agrupe a informação recolhida, e que hiperlink com tudo o que for possível;
  - 2.2. Incluir na página HTML de um sujeito `:x` uma imagem `"la.jpg"` sempre que encontrar no caderno um triplo da forma `:x :img "la.jpg"`.

### 2.2 Especificação dos Requisitos

Os requisitos para a realização deste trabalho prático são os seguintes:

1. Especificar a gramática concreta da linguagem de entrada;
2. Desenvolver um reconhecedor léxico e sintático para essa linguagem com recurso ao par de ferramentas geradoras Flex/Yacc;
3. Construir o gerador de código que produza a resposta solicitada. Esse gerador de código é construído associando ações semânticas de tradução às produções da gramática, recorrendo uma vez mais ao gerador Yacc.

## Capítulo 3

# Concepção/desenho da Resolução

### 3.1 Estruturas de dados

Por forma a guardar a informação recolhida sobre cada página HTML existente (mais precisamente, a informação relativa às imagens a serem inseridas e ao conteúdo dos triplos), bem como quais as páginas que foram criadas, foram construídos três arrays dinâmicos. Num deles (*htmls*) foi guardado o nome de todas as páginas HTML criadas; noutra (*imagens*), a informação relativa a quais as imagens que deverão ser inseridas em cada página; e por fim, o terceiro (*refers*) que foi utilizado para guardar as informações dos triplos, tratadas como sendo referências.

A restante informação, suportada nos documentos associados aos triplos, é inserida automaticamente no ficheiro HTML correspondente, após a leitura total do documento (de referir que por "documento" entende-se o primeiro elemento de um par (documento, triplos)), por oposição às imagens e às referências, que apenas são inseridas nos HTML respetivos no final da leitura de todo o ficheiro com os diversos pares (documento, triplos).

### 3.2 Flex

Desenvolvida a gramática passamos ao desenvolvimento do processador flex. As decisões tomadas na escolha dos filtros a utilizar tiveram um papel fundamental, uma vez que o analisador léxico da aplicação é o responsável por identificar os tokens principais da linguagem de entrada. Para isso, foram definidos vários contextos e expressões regulares para capturar todos os conceitos, expressões, títulos, subtítulos e informações, enviando-os posteriormente para o *Yacc* para que o reconhecimento sintático pudesse ser feito.

Por forma a capturar o título de um documento, por exemplo, o raciocínio aplicado foi o seguinte: quando encontramos um elemento que corresponda à expressão regular

$$@tit:[ ]*[^\\n]+\\n$$

guarda-se o conteúdo deste (após ser processado) em *yylval.svalue*, variável definida no yacc, e é retornado o símbolo variável *TITULO*. Para filtrar e capturar os restantes elementos, o raciocínio foi semelhante, tendo em conta as regras que cada um possui.

### 3.3 Yacc

No que diz respeito à gramática do caderno de anotações foram utilizadas 23 produções, 14 símbolos não terminais e 11 símbolos terminais, dos quais 8 são símbolos variáveis e 3 são sinais.

Depois de capturados os elementos da linguagem através do *Flex*, estes são enviados ao analisador sintático. Aqui, recorrendo à gramática definida, é feito o seu reconhecimento, dando-se então a instrução para inserção da informação resultante dessa análise na estrutura de dados. Assim, para esse reconhecimento, foram caracterizadas as seguintes produções:

```
Caderno      : ListaPares
              ;
ListaPares    : ListaPares Par
              | Par
              ;
Par           : Documento Triplos
              ;
Documento     : TRESIGUAIS CONCEITO TITULO Texto
              ;
Texto         : Texto Excerto
              | Excerto
              ;
Excerto       : SUBTITULO
              | INFO
              | PARAGRAFO
              ;
Triplos       : INITRIPLoS ListaTriplos
              ;
ListaTriplos  : ListaTriplos Triplo
              | Triplo
              ;
Triplo        : CONCEITO Relacoes '.'
              ;
Relacoes     : Relacoes ';' Relacao
              | Relacao
              ;
Relacao       : TipoRelacao Objetos
              ;
TipoRelacao   : CONCEITO
              | EXPRESSAO
              ;
Objetos       : Objetos ',' TipoObjeto
              | TipoObjeto
              ;
TipoObjeto    : CONCEITO
              | EXPRESSAO
              ;
```

## Capítulo 4

# Codificação e Testes

### 4.1 Alternativas, Decisões e Problemas de Implementação

Relativamente à geração dos ficheiros HTML, optamos por desenhar um esquema simples, de forma a obter uma visualização rápida e intuitiva de toda a informação fornecida. Além disso, decidimos criar uma página referente ao índice, onde se encontram os *hyperlinks* de todas as páginas HTML criadas, com o objetivo de facilitar o acesso a todas elas. Por fim, relativamente às informações nos triplos, optamos por inserir as mesmas como sendo referências e, tanto estas como as imagens, optamos por inseri-las no fim dos respetivos ficheiros.

### 4.2 Modo de Utilização

Tirando proveito do uso de uma makefile, o nosso programa torna-se de fácil criação e execução. Primeiramente, e dentro da diretoria onde estão guardados os ficheiros .l e .y, deverão ser também guardados os ficheiros que pretendemos ler (o ficheiro onde estão os pares (documento, triplo) bem como as imagens a inserir no HTML). Em seguida, é necessário correr o comando *make* e, só depois, correr o comando *./processador <nome do ficheiro>*, ou seja, correr o executável passando como argumento o nome do ficheiro de texto que queremos que seja analisado por forma a criar uma pasta com os ficheiros HTML necessários.

### 4.3 Testes realizados e Resultados

Para se poder construir as páginas HTML de forma eficiente, rápida e modular, implementamos no Yacc algoritmos capazes de gerar ficheiros HTML através da informação recebida após processamento do Flex. Com o intuito de demonstrar os pontos abordados ao longo do relatório, apresentamos a seguir um dos HTML obtidos bem como o seu aspeto visual, tendo por base o ficheiro exemplo fornecido pela equipa docente. Neste caso, apresentamos o ficheiro de índice, a partir do qual se poderá aceder a todas as outras páginas HTML criadas:



```

<!DOCTYPE html>
<html>
  <head>
    <meta charset='utf-8' />
    <title>indice</title>
  </head>
  <body>
    <h1>Índice</h1>
    <h2>B</h2>
    <ul>
      <li><a href='Brasil.html'>Brasil</a></li>
    </ul>
    <h2>C</h2>
    <ul>
      <li><a href='CarolinaDA.html'>CarolinaDA</a></li>
    </ul>
    <h2>E</h2>
    <ul>
      <li><a href='Etnólogo.html'>Etnólogo</a></li>
    </ul>
    <h2>F</h2>
    <ul>
      <li><a href='FranciscoSM.html'>FranciscoSM</a></li>
    </ul>
    <h2>G</h2>
    <ul>
      <li><a href='Gerês.html'>Gerês</a></li>
    </ul>
    <h2>J</h2>
    <ul>
      <li><a href='JorgeDomingos.html'>JorgeDomingos</a></li>
    </ul>
    <h2>M</h2>
    <ul>
      <li><a href='Mecenas_cultural.html'>Mecenas cultural</a></li>
    </ul>
    <h2>O</h2>
    <ul>
      <li><a href='OndinaL.html'>OndinaL</a></li>
      <li><a href='OnorioL.html'>OnorioL</a></li>
      <li><a href='OntinaL.html'>OntinaL</a></li>
    </ul>
    <h2>P</h2>
    <ul>
      <li><a href='Person.html'>Person</a></li>
      <li><a href='Porto.html'>Porto</a></li>
    </ul>
  </body></html>

```

# Índice

## B

- [Brasil](#)

## C

- [CarolinaDA](#)

## E

- [Etnólogo](#)

## F

- [FranciscoSM](#)

## G

- [Gerês](#)

## J

- [JorgeDomingos](#)

## M

- [Mecenas cultural](#)

## O

- [OndinaL](#)
- [OnorioL](#)
- [OntinaL](#)

## P

- [Person](#)
- [Porto](#)

Figura 4.1: index.html

## Capítulo 5

# Conclusão

Terminado o desenvolvimento deste trabalho prático, acreditamos ter conseguido cumprir com os objetivos estabelecidos, referindo como exemplo uma maior compreensão das potencialidades das ferramentas estudadas e utilizadas no desenvolvimento deste projeto, bem como o desenvolvimento de conhecimentos relativos ao *parsing* de ficheiros e posterior reorganização dos mesmos. Além disso, podemos afirmar que o uso destas ferramentas simplificou a análise de um caderno de anotações e respetiva geração automática de ficheiros HTML.

De realçar que fomos capazes de implementar também o "extra" requisitado no enunciado, dispondo assim o nosso programa da capacidade de interpretar relações inversas.