

**Mestrado Integrado em Engenharia Informática**  
**Processamento de Linguagens**  
2º Trabalho Prático

Francisco Oliveira (a78416)  
Raul Vilas Boas (a79617)  
Vitor Peixoto (a79175)

15 de Maio de 2018

# Conteúdo

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introdução</b>                      | <b>2</b> |
| <b>2</b> | <b>Requisitos</b>                      | <b>3</b> |
| 2.1      | Principais . . . . .                   | 3        |
| 2.2      | Extras . . . . .                       | 3        |
| <b>3</b> | <b>Demonstração</b>                    | <b>4</b> |
| 3.1      | Níveis de títulos . . . . .            | 4        |
| 3.1.1    | Exemplo de título SubSection . . . . . | 4        |
| 3.2      | Formatação de texto . . . . .          | 5        |
| 3.2.1    | Texto exemplo . . . . .                | 5        |
| 3.3      | Listas . . . . .                       | 6        |
| 3.3.1    | Listas não numeradas . . . . .         | 6        |
| 3.3.2    | Listas numeradas . . . . .             | 6        |
| 3.3.3    | Entradas tipo dicionário . . . . .     | 6        |
| 3.3.4    | Listas aninhadas . . . . .             | 7        |
| 3.4      | Extras . . . . .                       | 8        |
| <b>4</b> | <b>Conclusão</b>                       | <b>9</b> |

# Capítulo 1

## Introdução

No âmbito da unidade curricular de Processamento de Linguagens, foi-nos pedido que, baseado em Markdown, criássemos uma linguagem semelhante e uma ferramenta em Flex capaz de transformar esta notação mais leve em L<sup>A</sup>T<sub>E</sub>X.

Neste relatório será explicado o processo de desenvolvimento e as decisões tomadas ao longo da realização do trabalho.

## Capítulo 2

# Requisitos

Neste trabalho foram pedidos vários requisitos principais, os quais falaremos em breve, contudo, tomamos iniciativa de adicionar algumas funcionalidades extra, com intuito de aprofundarmos o nosso conhecimento e aprendizagem neste tópico.

Passamos agora a mostrar os requisitos em causa:

### 2.1 Principais

Foi-nos pedido que a nossa ferramenta fosse capaz de:

- Criar vários níveis de títulos
- Formatar o texto para:
  - Negrito
  - Itálico
  - Sublinhado
- Criar listas de tópicos:
  - Não-numerados
  - Numerados
  - Tipo entrada de dicionário

### 2.2 Extras

Ainda como extras, a nossa ferramenta é capaz de:

- Texto rasurado
- Aceitar alguns comandos especiais:
  - Footnote <sup>1</sup>
  - Comentários
  - Inserir novo parágrafo e quebra de página
  - Inserir imagens

---

<sup>1</sup>footnote

# Capítulo 3

## Demonstração

### 3.1 Níveis de títulos

A nossa linguagem permite a utilização de todos os níveis de títulos existentes no L<sup>A</sup>T<sub>E</sub>X.

Neste caso "Demonstração" seria um exemplo de título do tipo *Chapter* e "Níveis de títulos" um tipo *Section*. Neste caso, o título deste capítulo "Demonstração" seria um exemplo de título do tipo *Chapter* e "Níveis de títulos" um tipo *Section*. O título de cada nível deve ser sempre indicado no início de uma nova linha, seguindo de "#", sendo que o número de ocorrências de "#" indicam o nível do título (i.e. "#" indica *Chapter*, "##" indica *Section* e por aí adiante).

Para além disto, há a opção de adicionar ou não numeração ao título, bem como o seu aparecimento no índice. Os títulos que não devem ser numerados nem indexados devem seguir-se de um "\*" após todos os "#".

Na imagem abaixo, pode-se compreender melhor o funcionamento do processo explicado acima:

#### 3.1.1 Exemplo de título SubSection

Temos ainda *SubSubSection*, *Paragraph*, *SubParagraph*, bem como todos os anteriormente referidos títulos na sua versão com \* (asterisco).

#### Exemplo de título SubSection\*

Exemplo.

```
35 ^#####\*.+      fprintf(out, "\\subparagraph*{%s}", yytext+=7);
36 ^#####\*.+      fprintf(out, "\\paragraph*{%s}", yytext+=6);
37 ^####\*.+        fprintf(out, "\\subsubsection*{%s}", yytext+=5);
38 ^###\*.+         fprintf(out, "\\subsection*{%s}", yytext+=4);
39 ^##\*.+          fprintf(out, "\\section*{%s}", yytext+=3);
40 ^#\*.+           fprintf(out, "\\chapter*{%s}", yytext+=2);
41 ^#####.+        fprintf(out, "\\subparagraph{%s}", yytext+=6);
42 ^#####.+        fprintf(out, "\\paragraph{%s}", yytext+=5);
43 ^####.+         fprintf(out, "\\subsubsection{%s}", yytext+=4);
44 ^###.+          fprintf(out, "\\subsection{%s}", yytext+=3);
45 ^##.+           fprintf(out, "\\section{%s}", yytext+=2);
46 ^#.+            fprintf(out, "\\chapter{%s}", yytext+=1);
```

Figura 3.1: Código Flex - títulos

```
###*Exemplo de título SubSection*
Exemplo.
```

Figura 3.2: Exemplos de títulos

## 3.2 Formatação de texto

Para os diferentes tipos de formatação são usados os seguintes estilos:

- "\*" para texto bold;
- "/" para texto itálico;
- "\_" para texto sublinhado;
- "~" para texto rasurado.

Estes são apenas marcadores de início de formatação.

Para além disso, o estado *formatting*, utilizado para sabermos que estamos dentro de uma formatação, não é exclusivo o que permite que qualquer texto possa ser introduzido dentro da formatação bem como formatações adicionais, como será demonstrado.

```
\*           {fprintf(out, "\\textbf");yy_push_state(formatting);}
\/          {fprintf(out, "\\textit");yy_push_state(formatting);}
_           {fprintf(out, "\\underline");yy_push_state(formatting);}
~           {fprintf(out, "\\sout");yy_push_state(formatting);}
<formatting>\} {fprintf(out, ""); yy_pop_state();}
```

Figura 3.3: Código Flex - Formatação

### 3.2.1 Texto exemplo

Portanto como pedido, a nossa linguagem é capaz de acomodar texto em **bold**, em *itálico*, sublinhado, ~~rasurado~~ bem como qualquer combinação entre eles.

Por exemplo ***MAIS***<sup>1</sup>

```
Portanto como pedido, a nossa linguagem é capaz de acomodar texto em *{bold},
em /{itálico}, _{sublinhado}, ~{rasurado} bem como qualquer combinação entre eles.
%newline
Por exemplo *{/_{~{MAIS}}}}
%note{ texto em bold+itálico+sublinhado+rasurado}
```

Figura 3.4: Exemplos de formatação de texto

<sup>1</sup> texto em bold+itálico+sublinhado+rasurado

## 3.3 Listas

Outro dos requisitos era a capacidade de criar listas. Permitimos a criação de todos os tipos de listas pedidos, bem como listas aninhadas podendo estas ser ou não do mesmo tipo. Para criar uma qualquer lista deve-se utilizar as seguintes expressões:

### 3.3.1 Listas não numeradas

- ">>\*" para iniciar uma nova Lista não numerada
  - ">>\*" para iniciar uma sub Lista não numerada
  - ">\*" para continuar uma Lista não numerada

### 3.3.2 Listas numeradas

1. ">>dígito" para iniciar uma nova Lista numerada
  - (a) ">>dígito" para iniciar uma sub Lista numerada
  - (b) ">dígito" para continuar uma Lista numerada

### 3.3.3 Entradas tipo dicionário

**Exemplo** ">>[palavra]" para iniciar uma nova Entrada tipo dicionário

**NovoExemplo** ">>[palavra]" para iniciar uma nova Entrada tipo dicionário

**SubExemplo** >[palavra]" para continuar uma Entrada tipo dicionário

Aqui temos alguns exemplos.

```
\>\>{*          {fprintf(out, "\\begin{itemize}\\n\\item ");yy_push_state(bulletlist);}
\>\>[0-9]       {fprintf(out, "\\begin{enumerate}\\n\\item ");yy_push_state(numberlist);}
\>\>\[^\n]*\     {fprintf(out, "\\begin{description}\\n\\item%s ", yytext+2);yy_push_state(diclist);}
<bulletlist>\>\>{*          fprintf(out, "\\item ");
<bulletlist>\<    {fprintf(out, "\\end{itemize}"); yy_pop_state();}
<numberlist>\>\>[0-9]       fprintf(out, "\\item ");
<numberlist>\<    {fprintf(out, "\\end{enumerate}"); yy_pop_state();}
<diclist>\>\>\[^\n]*\       fprintf(out, "\\item%s ", ++yytext);
<diclist>\<        {fprintf(out, "\\end{description}"); yy_pop_state();}
```

Figura 3.5: Código Flex - Listas

---

<sup>2</sup>um qualquer dígito

### 3.3.4 Listas aninhadas

Apresentamos aqui um exemplo com várias listas aninhadas, e algumas delas formatadas.

1. Alínea 1
  - Ponto 1
    - Palavra 1** *Significado 1 itálico*
  - **Ponto 2 bold**
2. Alínea 2 sublinhado
  - Ponto 2
    - (a) Alínea 2.1 rasurado
      - ***Ponto 2.1 bold+itálico+sublinhado+rasurado***
3. Alínea 3
  - Palavra 1** Significado 1
    - (a) ~~Alínea 3.1 rasurado~~
  - Palavra 2** Significado 2
    - Alínea 3.2

```
###Listas aninhadas
Apresentamos aqui um exemplo com várias listas aninhadas,
e algumas delas formatadas.
>>1 Alínea 1
  >>* Ponto 1
    >>[Palavra 1] /{Significado 1 itálico}
    <
  >* *{Ponto 2 bold}
  <
>2 _{Alínea 2 sublinhado}
  >>* Ponto 2
    >>2 Alínea 2.1 rasurado
      >>* */{_{~{Ponto 2.1 bold+itálico+sublinhado+rasurado}}}}
      <
    <
  <
>1 Alínea 3
  >>[Palavra 1] Significado 1
    >>1 ~{Alínea 3.1 rasurado}
    <
  >[Palavra 2] Significado 2
    >>* Alínea 3.2
    <
  <
<
```

Figura 3.6: Exemplos da formatação de listas



### 3.4 Extras

Para além do que era pedido, fizemos ainda alguns extras. Primeiramente temos a *footnote*<sup>3</sup> que permite adicionar pequenas anotações ao texto, utilizando um simples comando `%note{texto da nota}`.

Temos também comentários, comandos para inserir novas linhas, novo parágrafo e novas páginas.

Podemos também adicionar imagens e efetuar transformações, tais como alterar o tamanho da imagem, bem como adicionar uma legenda.

Para adicionar uma imagem é necessário começar com `[dígito]` que será a escala a que a imagem será representada (escala entre 0.0 e 9.9). Depois da escala deve introduzir o caminho até à imagem, usando para tal o comando `[caminho]`. Finalmente há a hipótese de adicionar uma legenda usando `(legenda)` após o caminho.

```
\[([0-9]\.)?[0-9]\ { yytext[yytextlen-1]='\0';yytext++;
                      scale=yytext;
                      fprintf(out,"\\begin{figure}[h]\n");
                      fprintf(out,"\\centering\n");
                      yy_push_state(image);
}
<image>\[.*\ { yytext[yytextlen-1]='\0';yytext++;
               fprintf(out,"\\includegraphics[scale=%s]{%s}\n",scale,yytext);
}
<image>\\(.*\ { yytext[yytextlen-1]='\0';yytext++;
               fprintf(out,"\\caption{%s}\n",yytext);
               fprintf(out,"\\end{figure}\n");
               yy_pop_state();
}
<image>[.\\n] {fprintf(out,"\\end{figure}\n");yy_pop_state();}
```

Figura 3.7: Código Flex - Imagem

```
[0.9][imagem_flex.png](Código Flex - Imagem)
```

Figura 3.8: Exemplo de formatação de imagens

---

<sup>3</sup>texto rasurado também era um extra mas já foi falado

## Capítulo 4

# Conclusão

Este trabalho permitiu consolidar a matéria lecionada nas aulas da unidade curricular, mas também obter um maior conhecimento acerca do funcionamento do Flex, para além de permitir praticar o uso deste tipo de ferramentas para resolver problemas de conversão de linguagem, entre outros.  
Em suma, avaliamos a nossa prestação na resolução deste trabalho, como positiva.