

# Relatório PAH

André Santos, Helena Alves, Pedro Carneiro

12/04/2013

# Contents

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Descrição do trabalho</b>	<b>4</b>
<b>3</b>	<b>Estrutura de dados</b>	<b>5</b>
<b>4</b>	<b>Implementação</b>	<b>5</b>
<b>5</b>	<b>Ferramenta PAH</b>	<b>7</b>
5.1	Composição e Instalação . . . . .	7
5.2	Invocação . . . . .	7
5.2.1	Outras opções . . . . .	8
<b>6</b>	<b>Funcionalidades da ferramenta</b>	<b>9</b>
6.1	Cabeçalhos do documento . . . . .	9
6.2	Índices . . . . .	9
6.3	Comandos em LaTeX e HTML . . . . .	9
6.4	Títulos do documento . . . . .	10
6.5	Texto Não Formatado . . . . .	10
6.6	Texto Formatado . . . . .	10
6.7	Listas . . . . .	11
6.7.1	Listas Numeradas . . . . .	11
6.7.2	Listas Não Numeradas . . . . .	11
6.7.3	Entradas de um dicionário . . . . .	11
6.8	Imagens . . . . .	12
6.9	Links . . . . .	12
6.10	Comentários . . . . .	12
6.11	Tabelas . . . . .	13
<b>7</b>	<b>Conclusão</b>	<b>14</b>

# 1 Introdução

A resolução do trabalho prático descrita no presente relatório refere-se à especificação de um pré-processador para gerar ficheiros em LaTeX e Html. Pretende-se neste trabalho que seja implementada uma nova linguagem, de maneira a simplificar a escrita das diversas funcionalidades que essa nova linguagem irá permitir.

Assim, neste trabalho prático pretende-se aplicar os conceitos adquiridos na Unidade Curricular de Processamento de Linguagens associados à utilização de expressões regulares para descrever padrões de textos, para futuramente gerar ficheiros nos formatos referidos, através da ferramenta Flex.

## 2 Descrição do trabalho

Linguagens como o LaTeX e o Html trazem grandes benefícios para a sociedade actual, contudo também possuem alguns pontos menos atraentes. Estes pontos dizem respeito a certas desvantagens das mesmas, nomeadamente, a complexidade de certos comandos. Além de outros aspectos, este foi um dos aspectos que levou à criação da ferramenta *PAH*. Uma maior facilidade na escrita de certos comandos leva a um maior desempenho por parte do programador, aumentando o seu tempo de trabalho. Deste modo, a simplificação de certos comandos foi um dos aspectos principais a serem melhorados. Esta nova linguagem não acarreta, de longe, todas as potencialidades e funcionalidades das duas linguagens em questão, no entanto, foram trabalhadas grande parte das funcionalidades mais usuais, tais como:

- Introdução de capítulos;
- Texto formatado;
- Texto não formatado;
- Formatação de tabelas;
- Formatação de listas não numeradas;
- Formatação de listas numeradas;
- Formatação de listas descritivas;
- Inclusão de imagens;
- Inclusão de links;
- Introdução de comentários no texto;
- Geração de índices;
- Geração de capa de documento;
- Protecção dos seguintes caracteres especiais em latex:  
%, \$, -, &, #,  
{, , ~  
, ^ (excepto nas tabelas)
- Entre outros.

### 3 Estrutura de dados

Para a criação dos ficheiros em LaTeX e Html, é necessário armazenar determinados dados, de maneira que, após o processamento do ficheiro de entrada, seja possível imprimir esses dados para os ficheiros finais respectivos. Desta forma, é necessário uma estrutura de dados capaz de suportar estes dados e permitir o seu acesso rápido e eficiente. Portanto, foi considerada uma implementação de listas ligadas para guardar os dados referentes às tags de fecho e aos cabeçalhos do documento.

### 4 Implementação

Como o objectivo do trabalho consiste em gerar ficheiros em LaTeX e Html, foi decidido elaborar dois ficheiros flex. Sendo assim, de seguida vai ser apresentada uma breve explicação sobre a implementação deste ficheiro, no caso do latex, embora grande parte das situações são equivalentes para ambos os ficheiros.

O ficheiro flex foi desenvolvido recorrendo a um autómato, sendo este constituído pelos estados: HEADER, LATEX, TABLE, VERB, VERB\_LINE, COMMENT\_LINE, COMMENT, IMG, LINK, LIST, DICTIONARY.

Na parte inicial são efectuados os includes dos ficheiros necessários, assim como a declaração de variáveis. Seguidamente inicia-se o estado INITIAL, de onde se pode saltar para a maioria dos outros estados. Quando é encontrada a tag `%%header%`, ele automaticamente inicia o estado HEADER. Neste estado, podem-se encontrar as tags `%title`, `%author` e `%date`, que dizem respeito ao título, autor e data do documento, respectivamente. Ainda no estado INITIAL, quando são encontradas as tags `%t[1-3]`, `%index%`, `%cover%`, são adicionados o título, o índice e a capa do documento ao ficheiro final.

Quando nos estados INITIAL, LIST, DICTIONARY é encontrada a tag `%%o1%` e `%%u1%`, inicia-se o estado LIST, onde se inicia o início de listas ordenadas e listas não ordenadas, respectivamente. Entretanto, dentro do estado LIST, quando encontrado a tag `%o1%` e `%u1%`, dá-se o fim das duas listas, e volta-se para o estado anterior. As listas descritivas caracterizadas pela tag `%%d1%`, funcionam de forma análoga, contudo, neste caso, inicia-se o estado DICTIONARY. Neste último estado, juntamente com o estado LIST, quando encontradas as tags `%it`, possibilitam a inserção de itens dentro das respectivas listas.

As tags `%b`, `%i` e `%u` representam os diversos tipos de formatação do texto, nomeadamente, texto escrito em negrito, texto escrito em itálico e texto sublinhado. Estas são inicializadas quando encontrados os estados INITIAL, LIST, DICTIONARY, querendo isto dizer que é possível inserir texto formatado em qualquer parte do texto, dentro de listas e dentro de entradas de dicionário.

Nos estados INITIAL, LIST, DICTIONARY são encontradas as tags `%img` e `%link%`, é iniciado o estado IMG e LINK, respectivamente. Ainda nestes últimos três estados podem

encontrar-se as tags `%%` e `%%comment%` , que dizem respeito aos comentários de linha e multi-linha, iniciando-se os estados `COMMENT_LINE` e `COMMENT`, respectivamente. Dentro do estado `COMMENT_LINE`, quando é encontrado a tag `\n` , significa que o comentário de linha termina ali, e então volta-se para o estado anterior. Em relação ao estado `COMMENT`, este funciona de forma análoga ao anterior, contudo volta ao estado anterior quando encontra a tag `%%comment%` . Relativamente aos estados `VERB` e `VERB LINE`, que dizem respeito ao verbatim multi-linha e verbatim de linha, funcionam de forma idêntica aos estados anteriores.

Quando no estado `INITIAL`, são encontradas as tags `%%table%` e `%%latex%` , são iniciados os estados `TABLE` e `LATEX`, correspondendo à inserção de tabelas e à inserção de comandos nativos LaTeX, respectivamente.

Relativamente à implementação do conversor para html desenvolvido em flex, esta segue as linhas gerais da implementação do conversor para latex, como seria de esperar.

Em ambos os ficheiros flex, é implementada uma stack de estados, disponível pela ferramenta flex, onde são utilizados as funções `yy_push_state` e `yy_pop_state`. Esta stack de estados, permite a implementação de estados dentro de outros estados, como por exemplo, imagens dentro de listas, listas dentro de listas, links dentro de listas, etc.

## 5 Ferramenta PAH

### 5.1 Composição e Instalação

A ferramenta PAH está desenvolvida em flex, sendo esta constituída pelos seguintes ficheiros:

**latex.fl:** código fonte da ferramenta para gerar latex;

**html.fl:** código fonte da ferramenta para gerar html;

**pah.c:** main do programa;

**functionsLatex.c:** funções em C usadas pelo ficheiro latex.fl;

**functionsHtml.c:** funções em C usadas pelo ficheiro html.fl;

**linkedlist.c:** módulo de listas ligadas;

**utilities.c:** módulo auxiliar em C que serve de suporte à implementação de ambos os ficheiros flex;

**makefile:** responsável pela configuração da ferramenta no sistema.

Para proceder à instalação da ferramenta, basta apenas copiar os ficheiros referidos anteriormente para uma directoria à escolha e executar a makefile. Desta forma, será criado um executável na mesma directoria, com o nome de *pah*, bem como será criada uma cópia do manual da ferramenta na directoria UNIX destinada a tal.

### 5.2 Invocação

Como um dos objectivos deste pré-processador é facilitar a conversão em latex e html, a invocação para gerar ficheiros para cada uma destas linguagens consiste em:

**Gerar LaTeX:** `./pah -latex ficheiroInput output.tex`

**Gerar HTML:** `./pah -html ficheiroInput output.html`

### 5.2.1 Outras opções

- **Gera `ficheirotex.tex` e `ficheirohtml.html`:** `./pah ficheiroInput ficheirotex.tex ficheirohtml.html`
- **Gera `ficheiro.tex` e `ficheiro.html`** `./pah ficheiroInput ficheiro`
- **Gera `Latex.tex` e `Html.html`:** `./pah ficheiro`
- **Erro por falta de ficheiro de input:** `./pah -latex ./pah -html ./pah`



## 6 Funcionalidades da ferramenta

Nas secções seguintes serão apresentadas e exemplificadas todas as funcionalidades oferecidas por parte desta nova ferramenta.

### 6.1 Cabeçalhos do documento

Num cabeçalho de um documento é possível encontrar título, autores e data de um documento. Como seria de esperar, a ferramenta *PAH* também proporciona esta funcionalidade, baseando-se nos seguintes comandos:

- **Título do documento:** `%title` Título
- **Autor do documento:** `%author` Autor
- **Data do documento** `%date` 02/04/2013

Além disso, para que seja possível que o documento possua uma capa, basta usar o comando `%cover%`, que em LaTeX diz respeito ao comando `\maketitle`, e em Html diz respeito a uma zona do documento, mais propriamente no início do documento, onde terá toda esta informação.

### 6.2 Índices

Para um índice de um documento ser gerado, basta introduzir o comando:

- `%index%`

Este comando deve ser colocado após a capa do documento e antes de qualquer conteúdo.

### 6.3 Comandos em LaTeX e HTML

Em qualquer momento, a introdução de comandos nativos de LaTeX/HTML pode ser necessária, de forma a cobrir as funcionalidades que a ferramenta *PAH* não proporciona. Desta forma, para ambas as linguagens, a introdução destes comandos é feita de forma muito simples, bastando para isso seguir os seguintes comandos:

- **Para LaTeX:** `%%latex%` Comandos nativos em latex `%latex%`

- **Para HTML:** `%%html%` Comandos nativos em html `%%html%%`

A introdução de comandos nativos pode ser efectuada em qualquer parte do documento.

## 6.4 Títulos do documento

Existem três tipos de títulos, sendo estes apresentados pela ordem de maior importância.

- `%t1{Título da secção}`
- `%t2{Título da subsecção}`
- `%t3{Título de subsubsecção}`

## 6.5 Texto Não Formatado

Uma das funcionalidades da ferramenta é permitir incluir texto que não seja interpretado pelo *PAH*. Desta forma, todo o texto compreendido entre os comandos `%%verbatim%` e `%%verbatim%%`, não será interpretado aquando da conversão entre os formatos, sendo este apresentado no documento sem quaisquer modificações. Este último caso é utilizado aquando da introdução de várias linhas. Caso o pretendido seja apenas a introdução de uma linha, basta introduzir o conteúdo dessa linha entre os comandos

`%%verb%` e `%%verb%%`

Sabendo que o texto presente entre os comandos acima referidos, não vai ser interpretado, isso também acontece com os comandos da própria linguagem, como por exemplo, se dentro das tags `%%verbatim%` e `%%verbatim%%` se encontrar a tag `%b{Latex}`, a palavra "Latex" não ficará a negrito, pois dentro destas tags não existe interpretação de comandos.

## 6.6 Texto Formatado

Ao longo do documento, partes do texto podem ser apresentadas em itálico, em negrito, ou até mesmo sublinhadas. Para isso, basta seguir os próximos comandos:

**Itálico:** `%i{Texto em itálico}`

**Negrito:** `%b{Texto em negrito}`

**Sublinhado:** `%u{Textto sublinhado}`

Estas operações de formatação de texto são muito usuais na escrita de documentos, e tendo isso em conta, esta nova ferramenta permite a formatação do texto em várias situações, nomeadamente, em listas numeradas, listas não numeradas, listas descritivas, links e em situações normais ao longo do documento. A composição destas é também suportada pela ferramenta, sendo assim possível, por exemplo, que o mesmo texto seja destacado a negrito e a sublinhado.

## 6.7 Listas

De seguida, são apresentadas os três tipos de listas suportadas pela ferramenta *PAH*.

### 6.7.1 Listas Numeradas

- Iniciada por `%%o1%` e finalizada por `%o1%%` ;
- Cada item da lista é precedido por `%it` .

### 6.7.2 Listas Não Numeradas

- Iniciada por `%%u1%` e finalizada por `%u1%%` ;
- Cada item da lista é precedido por `%it` .

### 6.7.3 Entradas de um dicionário

- Iniciada por `%%d1%` e finalizada por `%d1%%` ;
- Cada item é precedido por `%it{}` e o texto que ficará a negrito deverá ser colocado entre os parêntesis;
- A descrição do item deve ser colocada após o fim dos parêntesis.

**Exemplo:**

```
%%d1%  
  %it{Item} descrição do item  
  %it{Item} descrição do item  
%d1%%
```

Em relação aos três tipos de lista é possível implementar outros comandos dentro destas, como por exemplo, colocar o texto a negrito, a itálico e ainda, existe a possibilidade de poder colocar listas dentro de listas, sendo esta uma funcionalidade muito utilizada.

## 6.8 Imagens

Como qualquer outra ferramenta, esta também permite a inclusão de imagens no documento. Para isso, basta usar o comando `%img{path,caption}`, sendo "path" o caminho para a respectiva imagem e "caption" a legenda da mesma.

**Exemplo:** Este comando que a seguir se apresenta é o responsável pela a inclusão da imagem, que depois deste se encontra.

```
%img{exemplo.png, Logotipo Escolha de Engenharia}
```



Figure 1: Logotipo Escolha de Engenharia

## 6.9 Links

A presença de links num documento é permitida usando o comando `%link{url,texto}`. No campo "texto" é inserido o texto que fica associado ao link colocado no campo "url", como mostra o exemplo.

**Exemplo:**

```
%link{https://elearning.uminho.pt/, Se clicar aqui será redireccionado para a  
página blackboard da uminho.}
```

Se clicar aqui será redireccionado para a página blackboard da uminho.

## 6.10 Comentários

O uso de comentários é muito usual ao longo do texto, uma vez que facilitam a leitura do mesmo e não aparecem no documento final. No caso de o comentário abranger várias linhas, estas devem ser colocadas entre os comandos `%%coment%` e `%coment%%`. Por outro lado, se o pretendido for apenas um comentário de linha, coloca-se o texto a comentar após o comando `%%%`.

- `%%% Isto é um comentário de linha`

## 6.11 Tabelas

As tabelas têm a seguinte estrutura:

Nome	Idade
André	17
Maria	10
Rui	15

```
%%table%  
    %head Célula % Célula  
    %row  Célula % Célula  
%%table%%
```

- A tabela é iniciada com `%%table%` e finalizada com `%%table%%` ;
- Cada célula da tabela deve ser colocada após `%head` ou `%row` . A única diferença entre estes, consiste no formato do texto. Enquanto que o comando `%head` coloca o texto centrado e a negrito, o comando `%row` coloca o texto num formato normal.
- O comando `%` permite a separação das células da tabela, correspondendo, no entanto, ao número total de colunas.

Numa tabela, e em latex, caso seja inserido uma nova linha com um número de colunas inferior ao número de colunas da linha com mais colunas, automaticamente são desenhadas as restantes células em branco. Uma vez que em html é perfeitamente possível que cada linha tenha um número de colunas distintas, esse problema não se coloca.

Na introdução de tabelas, um dos aspectos principais tidos em conta consistiu em reduzir grande parte da complexidade dos comandos necessários para a sua implementação, tornando esta muito mais facilitada. Desta forma, apenas é necessário a introdução dos comandos acima referidos, tornando mais prática a sua introdução.

## 7 Conclusão

Como forma de conclusão pode-se referir que o trabalho prático foi, no entender do grupo, realizado com sucesso. As funcionalidades propostas pelo enunciado foram implementadas com sucesso, juntamente com a adição de algumas funcionalidades extras.

De uma forma geral, os resultados produzidos foram satisfatórios, de modo a obter uma solução simples e eficiente.