

Turma X, Grupo XX

Relatório do trabalho de XX:

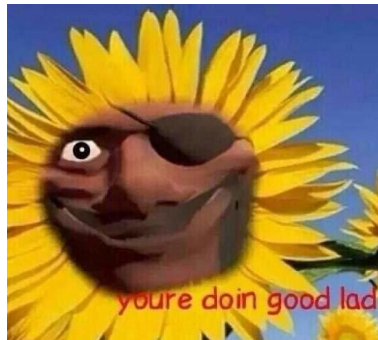
Nome do trabalho

Autor 1 ^{*} Autor 2 [†] Autor 3 [‡]

Universidade de Brasília, XX de YY de 20ZZ

Resumo

O \LaTeX é uma poderosa ferramenta de formatação de texto voltada para artigos acadêmicos. Embora demore a ser dominada, a excelente apresentação providenciada pela ferramenta é uma vantagem que deve ser considerada. Este repositório procura suprir a necessidade do usuário por um *template* bem feito de relatório em \LaTeX sem, entretanto, exigir conhecimentos profundos da ferramenta. Ao longo deste texto a estrutura do repositório, assim como as funcionalidades específicas deste projeto, são explicadas. Abaixo mostra-se uma bela imagem opcional de capa.



Sumário

1	Introdução	2
2	Básico do \LaTeX	2
2.1	Citações automáticas	2
2.2	Inserção de equações	3
2.3	Negrito, itálico e sublinhado	4
2.4	Atalhos de espaçamento vertical e quebra de linha	4
2.5	Inserção de figuras	4
3	Conhecimentos mais avançados	6
3.1	Inserção de código Python e MATLAB	6
3.2	Hyperlinks e URLs	7
3.3	Criação de tabela	8

^{*}Matrícula 1 – email 1

[†]Matrícula 2 – email 2

[‡]Matrícula 3 – email 3

1 Introdução

Para organizar o seu relatório em \LaTeX , recomenda-se usar a organização utilizada neste repositório:

- uma pasta `figures` com todas as figuras, gráficos e etc. do trabalho;
- uma pasta `listings` com todos os códigos a serem incluídos no relatório;
- uma pasta `sections` com todas as seções do relatório (arquivos `.tex`);
- uma pasta `src` com os arquivos `.tex` que preparam e estruturam o relatório.

Dependendo do tamanho do documento, pode vir a ser confortável criar uma 5ª pasta que contenha subseções, a `subsections`. Por outro lado, caso o documento seja livro ao invés de artigo, o diretório `sections` pode ser substituído por `chapters`, ou `parts`.

A pasta `src` confina a `main.tex`, `preâmbulo.tex` e o `setup_listings.tex`. É na `main` que o tamanho da fonte, título, autores e data são definidos, assim como as seções são incluídas; este é o único arquivo que deve ser compilado, gerando o `.pdf` do relatório. A compilação deve ocorrer via `pdfLaTeX`. Caso esteja no Overleaf, o compilador do projeto é exposto no Menu → Settings → Compiler.

No `preâmbulo.tex`, importa-se as pacotes do \LaTeX necessárias para formatar o `.pdf` em si, como a fonte do texto, cor tema do relatório, estilização das legendas, e moldura da página (essa que mostra o nome do professor, número da página, código da disciplina e etc.). Certifique-se de possuir esses pacotes instalados antes de compilar a `main`.

O `setup_listings.tex` formata a exibição de códigos em Python e MATLAB de maneira minuciosa.

Ao todo, não deve ser do interesse do leitor alterar os setups e o preâmbulo além do necessário para mudar a cor tema, moldura ou formatação dos códigos, mas sim escrever nos `.tex` da `sections`. Caso encontre alguma dúvida ou problema, contate o autor do template pelo próprio Github em X ou mande mensagem para thiagotomasdepaula@gmail.com.

2 Básico do \LaTeX

Apresentada a estrutura do projeto, passa-se a apresentar maneiras de escrever o `.pdf` em si. Não serão detalhados os documentos de preâmbulo e setups visto o texto ficaria longo demais e os códigos estão bem comentados. No que segue, espera-se que o leitor veja o `.pdf` e o código `.tex` da seção correspondente em paralelo.

2.1 Citações automáticas

O \LaTeX tem a extremamente útil capacidade de tornar automática a citação a figuras, equações matemáticas, tabelas, seções (e afins), e, é claro, referências bibliográficas, entre outros. Para tanto, simplesmente coloque o comando `\label{}` dentro do ambiente (`figure`, `table`, `equation`, etc.) ou logo abaixo do comando (`\section`, etc.) cuja referência é necessária, e faça a citação através do comando `\ref{}` ou `\autoref{}`. Em particular, equações também podem ser referidas por `\eqref{}`.

Por exemplo, foi colocado `\label{sec:citações}` nesta subseção, de forma que escrever `\ref{sec:citações}` retorna o número 2.1, enquanto que escrever `\autoref{sec:citações}` retorna Subseção 2.1. Confira alguns exemplos a seguir.

- “A seção `\ref{sec:básico}`” → “A seção 2”.
- “A `\autoref{sec:básico}`” → “A Seção 2”.
- “A equação `\ref{eq:seno}`” → “A equação 1”.
- “A `\autoref{eq:seno}`” → “A Equação 1”.
- “A equação `\eqref{eq:seno}`” → “A equação (1)”.
- “A figura `\ref{fig:exemplo}`” → “A figura 1”.
- “A `\autoref{fig:exemplo}`” → “A Figura 1”.
- “A figura `\ref{subfig:exemplo}`” → “A figura 2a”.

- “A \autoref{subfig:exemplo}” → “A **Figura 2a**”.

Por outro lado, referências bibliográficas são definidas e listadas na seção Referências, que não aparece no sumário. Para criar uma ficha de referências, escreva \bibitem{} dentro do ambiente thebibliography. O argumento do \bibitem{} é um apelido da referência respectiva, e a referência em si (título, tipo, autores, etc.) deve ser explicitada abaixo desse comando.

O código abaixo gera a bibliografia deste modelo. Note que este é um método manual, que pode vir a ser inconveniente para projetos com várias referências, mas em geral relatórios da graduação tendem a ter poucas fontes. O 5 é o número máximo de referências esperado, e é usado pelo \LaTeX para garantir que a lista fique alinhada. Confira mais informações de como fazer uma bibliografia [aqui](#).

```
\begin{thebibliography}{5}
  \bibitem{tipler}
    Paul A. Tipler e Gene Mosca.
    \textit{Física Volume 2, 5\textordfeminine Edição}.
    LTC--Livros Técnicos e Científicos Editora S.A., Rio de Janeiro, 2006.

  \bibitem{taylor}
    John R. Taylor.
    \textit{An Introduction to Error Analysis, Second Edition}.
    University Science Books, Sausalito (CA), 1997.

  \bibitem{britannica}
    Britannica, The Editors of Encyclopaedia.
    "servomechanism".
    Encyclopedia Britannica, 14 May. 2013,
    \url{https://www.britannica.com/technology/servomechanism}.
    Acessado 15 de janeiro de 2023.
\end{thebibliography}
```

Note o comando \url{} na última referência: é uma forma de inserir url's clicáveis no \LaTeX . Esse tipo de referência, assim como hyperlinks, serão vistos mais à frente.

2.2 Inserção de equações

Uma das grandes vantagens do \LaTeX é a facilidade em se inserir equações assim como a qualidade da formatação do texto matemático. Para colocar uma equação numerada, use o ambiente equation. Para uma não numerada, use equation* ou \$\$.

Por exemplo, o código

```
\begin{equation}
  \sin(z) =
  z
  - \frac{z^3}{3!}
  + \frac{z^5}{5!}
  - \dots
  =
  \sum_{n=0}^{\infty} \frac{z^{2n+1}}{(2n+1)!}
  \label{eq:seno}
\end{equation}
```

gera a **Equação 1**,

$$\sin(z) = z - \frac{z^3}{3!} + \frac{z^5}{5!} - \dots = \sum_{n=0}^{\infty} \frac{z^{2n+1}}{(2n+1)!} \quad (1)$$

enquanto que o código

```

$$
\Gamma(z) =
\int_0^\infty t^{z-1} e^{-t} \, dt
=
(z-1)!
$$

```

gera a equação abaixo.

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt = (z-1)!$$

Para colocar expressões matemáticas na linha do texto, use `$`. Mais detalhes [aqui](#).

2.3 Negrito, itálico e sublinhado

Para deixar em itálico, negritar, ou sublinhar um texto, use os comandos `\textit{}`, `\textbf{}` e `\underline{}`, respectivamente. O preâmbulo `.tex` fornece ainda o comando `\highlightbox{}` para destacar blocos de texto como, por exemplo, citações longas. Confira um exemplo de uso de cada comando a seguir.

“...l’amour est cent fois meilleur que la haine. L’espoir est meilleur que la peur. L’optimisme est meilleur que le désespoir.”

Para mais detalhes, procure [aqui](#). Por fim, repare que as aspas da citação foram feitas com acento agudo e apóstrofe, que é a forma padrão do \TeX para criá-las.

2.4 Atalhos de espaçamento vertical e quebra de linha

Um parágrafo novo pode ser iniciado deixando uma linha em branco entre dois blocos de texto. Para iniciar nova linha sem criar um parágrafo, use `\\` ao final da linha. Por exemplo, colocar `\\` aqui

gera linha aqui. Usar o `\\` além da linha em branco entre os blocos de texto gera tanto um parágrafo quanto um pequeno espaçamento vertical.

Por exemplo, este é um primeiro parágrafo,

E este é um segundo parágrafo com `\\` e linha em branco.

2.5 Inserção de figuras

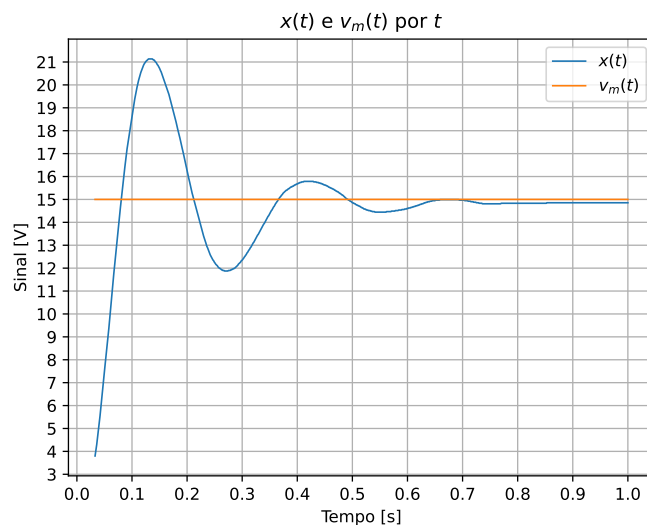
Para inserir uma figura no relatório use o comando `\includegraphics{}` dentro do ambiente `figure`. Tenha em mente que o arquivo da imagem deve estar em `figures`. Por exemplo, o código

```

\begin{figure}[H]\centering
\includegraphics[width=10cm]{gráfico do teste.pdf}
\caption{Figura de teste.}
\label{fig:exemplo}
\end{figure}

```

cria a imagem abaixo. O `H` serve para posicionar a figura no lugar mais próximo possível do esperado.

**Figura 1:** *Figura de teste.*

Para incluir múltiplas figuras, use o ambiente `subfigure` dentro do `figure`. Por exemplo, o código

```
\begin{figure}[H]\centering
  \begin{subfigure}{.45\textwidth}\centering
    \includegraphics[width=3cm]{unb_logo.png}
    \caption{Figura 1.}
    \label{subfig:exemplo}
  \end{subfigure}
  \hfill
  \begin{subfigure}{.45\textwidth}\centering
    \includegraphics[width=3cm]{unb_logo.png}
    \caption{Figura 2.}
  \end{subfigure}
  \caption{Figura dupla.}
\end{figure}
```

gera a figura dupla abaixo.

**(a)** *Figura 1.***(b)** *Figura 2.***Figura 2:** *Figura dupla.*

Note que o `.45\textwidth` garante que as figuras podem ficar lado a lado com alguma folga. No caso de 3 figuras lado a lado, a largura de cada uma não deve passar de `.33\textwidth`. Por fim, veja mais detalhes sobre inserção de figuras [aqui](#).

3 Conhecimentos mais avançados

3.1 Inserção de código Python e MATLAB

É comum nas engenharias o ensino de Python, e para aquelas que lidam com sinais e sistemas, como elétrica e mecatrônica, o uso do Matlab. Um código pode ser inserido no texto através do ambiente `lstlisting` fornecido pelo pacote `listings`; por exemplo, o código

```
\begin{lstlisting}[language=Python]
print('Hello World!')
\end{lstlisting}
```

gera

```
1 print('Hello World!')
2
```

ao passo que

```
\lstset{mystyle}
\begin{lstlisting}[language=Matlab]
clc; hold on; grid on;
for a = 0:10
    G = tf(1, [1 a]);
    bode(G);
end
'done!'
\end{lstlisting}
```

gera

```
1 clc; hold on; grid on;
2 for a = 0:10
3     G = tf(1, [1 a]); % malha aberta com pólo em -a
4     bode(G);
5 end
6 'done!'
7
```

Para trechos maiores de código, recomenda-se usar o comando `\lstinputlisting{}`, que inclui um código no texto de maneira automática. Neste modelo, o comando espera que os programas estejam na pasta `listings`. Por exemplo, o código

```
\lstinputlisting[
    language=Python, label={lst:código longo},
    caption={
        Código Python de maior tamanho.
    }]{exemplo.py}
```

resulta no [Listing 1](#).

Listing 1: *Código Python de maior tamanho.*

```
1 import os
2 import sys
3 from numpy import asarray
4 from PIL import Image
5
6
7 def img2data(image_file_name):
8     image = Image.open(image_file_name)
9     contents = asarray(image)
```

```

10
11     image_height, image_length = contents.shape[:2]
12
13     RGBs = [RGB for line in contents for RGB in line] # noqa
14
15     label, path = get_path_and_label(image_file_name)
16     data_name = label + '.data'
17
18     with open(os.path.join(path, data_name), 'w') as data_file:
19         sys.stdout = data_file
20         print(label, end=':\t')
21         print('.word %d, %d\n.byte' % (image_length, image_height))
22
23         column_cont = 0
24         for RGB in RGBs:
25             red_value, green_value, blue_value = RGB[:3]
26             color_byte = \
27                 (round(7 * red_value / 255) << 0) + \
28                 (round(7 * green_value / 255) << 3) + \
29                 (round(3 * blue_value / 255) << 6)
30             print(color_byte, end='')
31
32             num_spaces = 4 - len(str(color_byte)) # variable spacing
33             print(num_spaces * ' ', end='')
34
35             column_cont += 1
36             if column_cont == image_length:
37                 column_cont = 0
38                 print()
39
40     sys.stdout = sys.__stdout__
41     print("%s %dx%d created in '%s'" % (data_name, image_length,
42                                         image_height, path))
43
44 def get_path_and_label(image_file_name):
45     extension_removed = os.path.splitext(image_file_name)[0]
46     path, label = os.path.split(extension_removed)
47     return (label or os.path.basename(path)), path
48
49
50 if __name__ == '__main__':
51     img2data(sys.argv[1])

```

Note que as palavras chave de cada linguagem ficam destacadas, facilitando a leitura. Caso queira mudar as cores, modifique o `setup_listings.tex` na main. Para se aprofundar no uso do pacote listings, veja [este site](#) ou [a própria documentação](#).

3.2 Hyperlinks e URLs

URLs clicáveis e hyperlinks podem ser inseridos no relatório através dos comandos `\url{}` e `\href{}` disponibilizados no pacote `hyperref`. Recomenda-se que esse pacote seja o último a ser incluído para evitar conflitos com outros pacotes que mexem com citações, e também permite a customização das cores dos links, entre outros.

Confira um exemplo de uso de cada comando abaixo. Para mais detalhes, vide [este site](#) ou [a própria documentação](#).

- “O site `\url{https://latexcolor.com}` é ótimo” → “O site <https://latexcolor.com> é ótimo”
- “Acesse documentações dos pacotes `\href{https://ctan.org}{aqui}`.” → “Acesse documentações dos pacotes [aqui](https://ctan.org).”

3.3 Criação de tabela

Inserir uma tabela em \LaTeX pode vir a ser relativamente complicado. Se a tabela não possui células que abrangem mais de uma coluna ou linha por vez, o código fica relativamente simples: basta usar o ambiente `tabular` dentro do ambiente `table`, e usar os separadores de coluna e linha `&` e `\\` para preencher a tabela em si. Por exemplo, o código

```
\begin{table}[H]\centering
  \begin{tabular}{c | c c}
    \toprule
     $t_{\text{exec}}$  ( $\mu\text{s}$ ) &  $C$  &  $f$  (MHz) \\
    \midrule
    \midrule
    0 & 0 & 50 \\
    50 & 2500 & 50 \\
    100 & 5000 & 50 \\
    \bottomrule
  \end{tabular}
  \caption{Tabela simples.}
  \label{tab:simples}
\end{table}
```

gera a [Tabela 1](#).

t_{exec} (μs)	C	f (MHz)
0	0	50
50	2500	50
100	5000	50

Tabela 1: *Tabela simples.*

Como no caso de inserir uma figura, o `H` serve para posicionar a tabela precisamente. Os argumentos `c` e `|` do `tabular` centraliza uma coluna e cria uma linha vertical entre colunas, respectivamente. A quantidade de `c`'s estabelece a quantidade de colunas da tabela. Os comandos `\toprule`, `\midrule` e `\bottomrule` vêm do pacote `booktabs` e dão bom aspecto à tabela.

Criar células que abrangem múltiplas colunas ou linhas geram uma tabela mais complicada. Nesse caso, pode-se lançar mão dos comandos `\multirow{...}{...}` e `\multicolumn{...}{...}`, que recebem a quantidade de colunas/linhas a serem mescladas, o tipo de centralização da célula resultante, e o valor da célula resultante. Por exemplo, o código

```
\begin{table}[H]\centering
  \begin{tabular}{c c c}
    \toprule
    \multirow{2}{*}{ $n$ } & \multicolumn{2}{c}{ $I_1$ } \\
    \cmidrule{2-3}
    & teórico & real \\
    40 & 14256 & 14256 \\
    50 & 22316 & 22316 \\
    60 & 32176 & 32176 \\
    70 & 43836 & 43836 \\
    80 & 57296 & 57296 \\
    90 & 72556 & 72556 \\
    100 & 89616 & 89616 \\
  \end{tabular}
\end{table}
```


gera a tabela

n	I_1	
	teórico	real
40	14256	14256
50	22316	22316
60	32176	32176
70	43836	43836
80	57296	57296
90	72556	72556
100	89616	89616

Note ainda o uso do comando `\cmidrule{}`, uma versão mais abrangente do `\midrule`; recebe como argumento o números das colunas pelas quais a linha horizontal deve passar. No exemplo, ela atravessa a segunda e terceira coluna, evitando o n . Para entender melhor a criação de tabelas e/ou os comandos do booktab, visite [o Overleaf](#) ou a [documentação do pacote](#).

Referências

- [1] Paul A. Tipler e Gene Mosca. *Física Volume 2, 5ª Edição*. LTC–Livros Técnicos e Científicos Editora S.A., Rio de Janeiro, 2006.
- [2] John R. Taylor. *An Introduction to Error Analysis, Second Edition*. University Science Books, Sausalito (CA), 1997.
- [3] Britannica, The Editors of Encyclopaedia. "servomechanism". Encyclopedia Britannica, 14 May. 2013, <https://www.britannica.com/technology/servomechanism>. Acessado 15 de janeiro de 2023.