



UNIVERSIDADE FEDERAL DE SANTA MARIA  
CAMPUS CACHOEIRA DO SUL

---

**L<sup>A</sup>T<sub>E</sub>X 101**

---

*Autores :*

Prof<sup>a</sup>. Dr<sup>a</sup> Ana Luisa Soubhia  
Elias Teixeira da Costa  
Flavio Luan Müller Freitas  
Laís Brum Menezes  
Marcos Alves dos Santos  
Prof. Dr. Vinícius Maran

Esta apostila tem como objetivo apresentar um compilado de informações abordadas nos minicursos sobre *LaTeX* ministrados pelos alunos da Universidade Federal de Santa Maria (UFSM), campus Cachoeira do Sul.

Este material é resultante da primeira etapa do projeto de ensino *Cadernos Digitais para Engenharias - Aplicações em LaTeX, Python e Jupyter*, financiado pela Universidade Federal de Santa Maria (UFSM) através do edital FIEN 2019.

Coordenador do Projeto: Prof. Dr. Vinícius Maran. Contato:  
[vinicius.maran@ufsm.br](mailto:vinicius.maran@ufsm.br)

Sugestões sobre o material podem ser informadas [nest link](#).

Este trabalho está licenciado sob uma licença [Creative Commons](#) “[Attribution-NonCommercial-ShareAlike 3.0 Unported](#)”.



## Conteúdo

<b>1</b>	<b>Introdução ao L<sup>A</sup>T<sub>E</sub>X</b>	<b>4</b>
<b>2</b>	<b>Estrutura de um Documento</b>	<b>5</b>
2.1	Utilização de Pacotes . . . . .	5
2.1.1	Pacote Obrigatório . . . . .	5
2.1.2	Pacotes Opcionais . . . . .	7
<b>3</b>	<b>Formatação</b>	<b>8</b>
3.1	Título do documento . . . . .	8
3.2	Quebra de Linha e Página . . . . .	9
3.3	Alinhamento . . . . .	9
3.4	Secções . . . . .	9
3.5	Espaçamento . . . . .	10
3.6	Cores . . . . .	10
3.7	Estilo e Tamanho . . . . .	13
3.8	Nota e Rodapé . . . . .	14
3.9	Cabeçalho e rodapé da página . . . . .	16
3.10	Espaços e medidas . . . . .	17
<b>4</b>	<b>Tabelas, Listas, Imagens e Minipage</b>	<b>18</b>
4.1	Tabelas . . . . .	18
4.2	Listas . . . . .	21
<b>5</b>	<b>Figuras</b>	<b>23</b>
5.1	Minipage . . . . .	27
<b>6</b>	<b>Novos comandos e ambientes</b>	<b>29</b>
<b>7</b>	<b>Símbolos</b>	<b>30</b>
7.1	Alfabeto Grego . . . . .	30
7.2	Símbolos Matemáticos . . . . .	30
<b>8</b>	<b>Matemática</b>	<b>31</b>
8.1	Construindo equações . . . . .	31
8.1.1	Potência e Índices . . . . .	31
8.2	Frações e Raízes . . . . .	32
8.3	Matriz e delimitadores . . . . .	33
8.4	Funções . . . . .	37
8.4.1	Somatório . . . . .	37
8.4.2	Limite . . . . .	38

8.4.3	Integral	38
8.4.4	Tipos especiais de letras	38
8.5	Fórmulas em várias linhas	39
8.6	Linhas em equações	40
8.6.1	Empilhando	42
8.7	Comando <code>phantom</code>	42
8.8	Teoremas	42

# 1 Introdução ao L<sup>A</sup>T<sub>E</sub>X

Antes de saber o que é o L<sup>A</sup>T<sub>E</sub>X[1] é preciso conhecer o TEX. TEX é um programa criado por Donald Knuth na década de 70 que tem como finalidade aumentar a qualidade de impressão com base nas impressoras da época e é utilizado para processar textos e fórmulas matemáticas.

L<sup>A</sup>T<sub>E</sub>X é um programa que reúne comandos que utilizam o TEX como base de processamento e foi criado por Leslie Lamport na década de 80 com o objetivo de facilitar o uso do TEX através de comandos para diferentes funções.

É um editor de textos especialmente voltado para a área matemática contendo comandos para montar as mais diversas fórmulas. Gera textos de alta qualidade tipográfica (espaçamento entre palavras, combinação de letras etc...). Além de ser muito bom para fazer textos grandes como livros. O LaTeX é um processador baseado no estilo lógico. Os programas de processamento de texto podem ser divididos em duas categorias:

**Estilo visual:** Nestes processadores de texto, existe um menu na tela apresentando os recursos que podem ser usados, sendo selecionados através do mouse. E o texto que você digita aparece na tela da mesma forma que vai ser impresso. Isso é conhecido como WHAT-YOU-SEE-IS-WHAT-YOU-GET (WYSIWYG). Ex: Word.

**Estilo lógico:** Nesta categoria o processamento é feito em duas etapas distintas.

- O texto a ser impresso e os comandos de formatação são escritos em um arquivo fonte com o uso de um editor, (isso não impede que também haja um menu na tela onde os comandos podem ser selecionados, isto é apenas um adicional oferecidos por fabricantes para facilitar a digitação).
- Em seguida este arquivo é compilado e gera um arquivo de saída que pode ser visualizado. Ex: HTML

Os comandos LaTeX foram criados com base em diversos estudos sobre diagramação. Isto foi importante para fazer com que o LaTeX entenda o que o autor deseja fazer, por exemplo, como diferenciar um texto matemático de uma citação de fala. Segundo os estudos, existem formas que permitem tornar o texto muito mais claro. O tamanho deve ter um limite ideal para facilitar a leitura assim com o da fonte, o espaçamento das linhas, letras e palavras também têm uma medida ideal.

Geralmente, quando o autor está trabalhando com um processador visual, ele comete muitos erros por não conseguir conciliar uma boa estética com uma estrutura lógica e bem compreensível. Utilizando o LaTeX, que é um processador lógico, o autor se preocupará mais com o conteúdo. Dessa forma, seu texto não terá apenas uma boa estética e estrutura coerente, mas também com um bom conteúdo. Além disso, com a troca de apenas um comando, algo pode ser mudado futuramente com

facilidade, o que deixa o documento muito mais flexível.

Vamos ver o que acontece tecnicamente:

O autor escreve seu documento usando os comandos do LaTeX. O LaTeX entende o que o autor quis dizer e transforma os comandos digitados em uma linguagem inteligível pelo TEX, ou seja, o texto escrito em linguagem LaTeX é processado por um compilador seguindo as regras dessa linguagem. Isto é, transforma um arquivo .tex (que possui o código tex) em um arquivo .dvi (device independent). Este último é o documento pronto para ser visualizado.

Claro que não se pode esquecer que seu computador deve ter a biblioteca do LaTeX para fazer isso. Alguns dos sistemas TEX, atuais, disponíveis no mercado são: fptex, pctx, miktex (Windows) e tetex (Linux, embora provavelmente já o tenha instalado).

Esse arquivos .dvi podem ser lidos independentemente da versão do editor usado, logo pode ser lido em qualquer sistema, contanto que ele tenha um programa específico para lê-lo. Outra vantagem do LaTeX é sua estabilidade, ou seja, a probabilidade de se encontrar um bug nele é mínima e justamente por ser free software seu sistema é aberto, o que permite que qualquer um corrija possíveis bugs ou que possa adaptá-lo às suas necessidades. A cada momento surgem novos pacotes com funções criadas por usuários espalhados por todo o mundo <sup>1</sup>.

## 2 Estrutura de um Documento

Vamos aprender agora sobre a estrutura do L<sup>A</sup>T<sub>E</sub>X.

Todo documento escrito em L<sup>A</sup>T<sub>E</sub>X tem um preâmbulo, onde os comandos são listados e uma área onde o texto será digitado chamada corpo de texto.

### 2.1 Utilização de Pacotes

#### 2.1.1 Pacote Obrigatório

Vamos agora falar da utilização dos pacotes que definem o formato do texto que iremos trabalhar.

---

<sup>1</sup>Aqui cabe uma ressalva, escrevemos este minicurso no site <https://www.overleaf.com>, um editor de documentos do formato L<sup>A</sup>T<sub>E</sub>X online. Caso disponha de conexão com a internet no ambiente de trabalho, recomendamos fortemente a utilização dessa plataforma. Ou se preferir baixe e instale o pacote offline em seu computador.

Para começar a trabalhar com textos no L<sup>A</sup>T<sub>E</sub>X precisamos declarar o seguinte comando no preâmbulo:

### Código

```
\documentclass[tamanho da fonte]{tipo de documento}
```

O tamanho da fonte é escolhido pelo usuário. O tipo de documento pode ser:

- **article:** Para a digitação de artigos
- **report:** Para relatórios
- **book:** Para livros e apostilas
- **letters:** Para cartas

Esses são os formatos de digitação mais usados no L<sup>A</sup>T<sub>E</sub>X. Mais tarde aprenderemos a criar nosso próprio tipo de documento e utilizar outros criados por outros usuários.

Junto do tamanho da fonte, separados por vírgulas, também podemos definir o tamanho da folha e se a disposição do texto será em colunas ao invés de linhas. Exemplo:

### Código

```
\documentstyle[twocolumn,12pt,a4]{article}
```

Esse comando declara que o texto será apresentado em duas colunas na página, com fonte tamanho 12pt em uma folha de tamanho a4, no formato artigo. Alguns estilos disponíveis no L<sup>A</sup>T<sub>E</sub>X são:

- **Tamanho:** Padrão da letra: 11pt ou 12pt(pontos), o último é usado com mais frequência;
- **twoside:** Imprime em ambos os lados da página;
- **oneside:** Imprime em um só lado da página;
- **twocolumn:** Produz o texto disposto em duas colunas na página;
- **onecolumn:** Produz o texto disposto em uma coluna;
- **landscape:** Produz uma página na forma de paisagem;

- **leqno:** Isto faz com que a numeração das fórmulas sejam colocadas a esquerda em vez de a direita;
- **fleqn:** Faz com que a fórmula fique localizada na margem esquerda em vez de estar centralizada;
- **openright:** Faz com que os capítulos sejam iniciados apenas nas páginas ímpares;
- **openany:** Permite que os capítulos sejam iniciados nas páginas ímpar ou par.
- **para declarar o tamanho da folha, simplesmente digite o seu formato em letra minúscula dentro dos colchetes, utilizando virgula caso tenha declarado mais atributos.**

Obs: O comando `\twocolumn` inicia uma nova página produzindo duas colunas nela a partir do ponto onde foi colocado no texto. Para voltar a disposição de uma coluna basta usar o comando `\onecolumn`.

### 2.1.2 Pacotes Opcionais

Pacotes são um conjunto de arquivos que implementam uma determinada característica adicional para os documentos escritos em LaTeX. São muito utilizados para a digitação de documentos mais elaborados, aumentando e muito a capacidade de processamento do L<sup>A</sup>T<sub>E</sub>X. A forma de chamar esses documentos no preâmbulo é a seguinte:

#### Código

```
\usepackage[opcional]{pacote}
```

onde o campo 'pacote' é obrigatório e o opcional é necessário somente para certos pacotes ou ocasiões. Abaixo estão listados alguns pacotes que podem ser declarados no preâmbulo:

- **graphicx:** permite a inserção de gráficos no texto.
- **color:** permite a inserção de cor no texto.
- **babel:** Faz a tradução de nomes do Inglês para o Português na estrutura do documento. Requer que 'brazilian' seja digitado no campo opcional do pacote:  
`\usepackage[brazilian]{babel}`



- **inputenc:** permite a digitação direta de caracteres como til, cedilha, circunflexo, agudo, etc, sem precisar declarar o pacote fontenc. Necessita do opcional [utf8] para funcionar dessa maneira.
- **fontenc:** permite que o L<sup>A</sup>T<sub>E</sub>X entenda a acentuação feita diretamente pelo teclado. É usado com o opcional [T1].
- **amsfonts:** Define alguns estilos de letras para o ambiente matemático.
- **fancyhdr:** Para fazer cabeçalhos personalizados.

**Observação:** Alguns pacotes dependem de outros para funcionar. A lista completa de pacotes disponíveis no L<sup>A</sup>T<sub>E</sub>X pode ser encontrada [aqui](#).

## 3 Formatação

Vamos tratar agora dos aspectos da formatação do texto.

### 3.1 Título do documento

A declaração do título do documento, autor, data e outros atributos do documento é feita nas primeiras linhas do corpo do documento da seguinte forma:

#### Código

```
\usepackage{tipo do documento}
...
\begin{document}
\title{Título}
\author{autor ou autor1 \and autor2}
\date{data}
\maketitle
\begin{abstract}
  Aqui vai um resumo da obra
  caso o autor deseje.
\end{abstract}
```

Utilizando os estilos article e report é possível usar o ambiente abstract para fazer um breve resumo sobre o documento.

## 3.2 Quebra de Linha e Página

Para forçar uma quebra de linha use o comando `\linebreak`. A parte anterior ao comando ficará justificada.

### Código

```
Esta linha está \linebreak justificada.
```

### Exemplo

```
Esta                linha                está
justificada.
```

Para quebrar a página existem dois comandos: `\newpage` e `\pagebreak[num]`.

`\newpage` passa todo o texto digitado abaixo desse comando para a próxima página. `\pagebreak[num]` força a quebra da página, onde o argumento opcional `[num]` é um valor inteiro de 1 a 4 que definem a prioridade do funcionamento do comando, onde 4 é o maior valor.

`\nopagebreak[num]` impede a página de ser quebrada no local onde o comando foi colocado.

## 3.3 Alinhamento

É possível alinhar o texto de três formas no L<sup>A</sup>T<sub>E</sub>X:

<code>\begin{flushleft}</code>	<code>\begin{center}</code>	<code>\begin{flushright}</code>
texto alinhado a	texto alinhado ao	texto alinhado a
esquerda	centro	direita
<code>\end{flushleft}</code>	<code>\end{center}</code>	<code>\end{flushright}</code>

## 3.4 Secções

Quando o texto é longo, este pode ser dividido em várias partes. O L<sup>A</sup>T<sub>E</sub>X contém comandos para dividi-lo de forma organizada e coerente.

### Código

```

\part {parte - só pode ser usado em report ou book}
\chapter {capítulo - só pode ser usado em report ou book}
\section {seção}
\subsection {subseção}
\subsubsection {sub-sub-seção}
\paragraph {parágrafo}
\subparagraph*{subparágrafo não numerado}

```

Onde o conteúdo entre chaves representa o nome que o usuário dará a parte em questão. Todas elas são numeradas automaticamente pelo L<sup>A</sup>T<sub>E</sub>X, sendo essa uma grande vantagem sua comparada a outros editores de texto que não conseguem fazer isso. Caso o usuário não deseje numerar a parte basta colocar um asterisco entre o comando e a chave que carrega seu nome como indicado no último exemplo.

Quando adicionar ou remover partes do documento, compile-o duas vezes para que o L<sup>A</sup>T<sub>E</sub>X reconheça corretamente a nova numeração das partes.

### 3.5 Espaçamento

O L<sup>A</sup>T<sub>E</sub>X não reconhece espaços "crus" digitados no meio do texto, equações e após alguns símbolos e caracteres especiais. Quando isso acontecer os comandos abaixo podem ser usados para forçar o programa a colocar um espaço pré definido no lugar onde ele foi digitado:

<code>\,</code> - espaço pequeno	<code>\;</code> - espaço grande
<code>\:</code> - espaço médio	<code>\!</code> - espaço negativo (backspace)

### 3.6 Cores

Para usar cores é necessário declarar no preâmbulo o comando `\usepackage{color}`. Dessa forma o L<sup>A</sup>T<sub>E</sub>X carrega o pacote color e reconhece os comandos de cores. Exemplos:

- `\definecolor{nome}{modelo}{parâmetro}` Este define uma cor personalizada, onde nome é o nome da cor, modelo é o modelo da cor com o principal sendo: rgb (red, green, blue) e parâmetro é o código da cor segundo o modelo usado.
- `\textcolor{cor}{palavra}` Muda cor da palavra selecionada. A cor pode ser escrita diretamente em inglês ou usar o `\definecolor` definindo o nome da

cor em português.

Exemplo:

no preâmbulo:

#### Código

```
\definecolor{verdemar}{rgb}{0.59,0.78,0.65}
```

no corpo do texto:

#### Código

```
\textcolor{verdemar}{texto em cor personalizada}
```

Resultado:

#### Exemplo

```
texto em cor personalizada
```

- `\color{cor}` Muda a cor do texto inteiro. Para mudar apenas um trecho do texto use chaves neste trecho.

Ex:

#### Código

```
{\color{blue}{trecho do texto}}
```

#### Exemplo

```
trecho do texto
```

**Nota:** Use esse comando entre chaves para evitar que outras partes do texto mudem de cor.

- `\pagecolor{cor}` Muda a cor do fundo da página.
- `\colorbox{cor}{texto}` Gera uma caixa com o fundo da cor que foi escolhida.

#### Código

```
\colorbox{red}{texto em caixa com fundo vermelho}
```

#### Exemplo

```
texto em caixa com fundo vermelho
```

Também é possível gerar uma caixa com bordas usando o seguinte comando:

#### Código

```
\fcolorbox{black}{red}{texto em caixa com fundo vermelho e
borda preta.}
```

#### Exemplo

texto em caixa com fundo vermelho e borda preta.

onde a primeira chave declara a cor da borda e a segunda a cor do fundo da caixa.

E por último, usando o pacote tcolorbox é possível gerar caixas de textos personalizadas, com bordas arredondadas, fundo com uma cor, letra com outra, em duas partes distintas...

#### Código

```
\begin{tcolorbox}[width=\textwidth,colback={green},
title={caixa com bordas arredondadas},colbacktitle=yellow,
coltitle=blue]
  \blindtext[1]
\end{tcolorbox}
```

#### Exemplo

caixa com bordas arredondadas

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

### Código

```
\begin{tcolorbox}[width=\textwidth,colback={red},
title={Caixa quadrada},
outer arc=0mm,colupper=white]
  \blindtext[1]
\end{tcolorbox}
```

### Exemplo

Caixa quadrada

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

Importante: dependendo do visualizador de DVI as cores não serão vistas, mas se converter em PS ou PDF, aparecerá normalmente. Obs: O pacote `blindtext` apenas gera o texto lorem ipsum dentro das caixas.

## 3.7 Estilo e Tamanho

Os seguintes comandos alteram o estilo da letra no texto:

- O comando `\textbf{texto}` **transforma o texto em negrito.**
- O comando `\textsf{texto}` transforma o texto em Sans Serif.
- O comando `\textsl{texto}` *transforma o texto em slanted.*
- O comando `\textsc{texto}` TRANSFORMA O TEXTO EM SMALL CAPS.

O comando `\texttt{texto}` transforma o texto em letra de máquina.

O comando `\textrm{texto}` transforma o texto em romano.

Abaixo estão alguns comandos que alteram o tamanho das letras no texto:

O comando `\tiny{texto}` deixa o texto assim.

O comando `\scriptsize{texto}` deixa o texto assim.

O comando `\footnotesize{texto}` deixa o texto assim.

O comando `\small{texto}` deixa o texto assim.

O comando `\normalsize{texto}` deixa o texto assim.

O comando `\large{texto}` deixa o texto assim.

O comando `\LARGE{texto}` deixa o texto assim.

O comando `\huge{texto}` deixa o texto assim.

### 3.8 Nota e Rodapé

Notas de rodapé são produzidas no texto com o comando `\footnote{texto}` digitado no lugar que se quer referenciar algo. Ex: Este texto tem uma nota no final da

página `\footnote{descrição da referência}`.

Produz:

Este texto tem uma nota no final da página <sup>2</sup>.

Também é possível colocar referências na borda lateral da página. Assim, a nota ficará na altura da linha de texto em que foi colocada.

Ex: Este texto tem uma referência na lateral da

página `\marginpar{descrição da referência}`.

Gera:

Este texto tem uma referência na lateral da página .

referência  
da la-  
teral  
da  
página

---

<sup>2</sup>esta é a nota do pé da página

**OBS:** O comando `\marginpar{descrição da referência}` não numera as referências. O comando `\footnote{texto}` numera e linka automaticamente sua posição no texto com sua referencias no rodapé da página. Experimente clicar sobre o número da referência para ver o que acontece.



### 3.9 Cabeçalho e rodapé da página

Com o pacote `\usepackage{fancyhdr}`, é possível editar o estilo do cabeçalho e do rodapé das páginas do documento. Neste documento por exemplo foram utilizados os seguintes comandos no preâmbulo:

```
\title{Minicurso \LaTeX}
\author{versão 1.0 - 2019}
\date{01/08/2019}
\makeatletter
\let\thetitle\@title
\let\theauthor\@author
\let\thedata\@date
\makeatother
\pagestyle{fancy}
\fancyhf{}
\rhead{\theauthor}
\lhead{\thetitle}
\cfoot{\thepage}
```

Onde `\rhead{\theauthor}` significa que se trata da nota do lado superior direito da página, que será representada pelo autor do documento, previamente descrito. Mas Poderia ser outra coisa como por exemplo, `\rightmark` para representar seção ou `\leftmark` para representar o capítulo. Isso também se aplica ao comando `\lhead{\thetitle}` que mostra o título do arquivo na borda superior esquerda. E por ultimo, `\cfoot{\thepage}` mostra no centro da borda inferior da página o número da página do documento.

Abaixo estão listados alguns comandos que podem ser usados nos argumentos das notas de cabeçalho e rodapé das páginas.

- `\thepage` Mostra o número da página do documento que o usuário está lendo
- `\leftmark` Mostra o nome do capítulo que está sendo lido.
- `\rightmark` Mostra o nome da seção que está sendo lida.
- `\chaptername` Mostra o nome do capítulo que está sendo lido.
- `\thechapter` Mostra o número do capítulo que está sendo lido.
- `\thesection` Mostra o número do capítulo que está sendo lido.

### 3.10 Espaços e medidas

Através dos seguintes comandos pode ser dado espaçamento entre as palavras:

- `\hspace{medida}` Põe espaço entre as palavras, a unidade da medida pode ser em:
  - (cm) centímetros
  - (in) polegadas
  - (pt) pontos
  - (ex) medida equivalente a altura da letra x
  - (em) medida relativa a largura da letra M

Dica: Utilize as duas ultimas medidas pois são baseadas na fonte utilizada no documento.

#### Código

```
Linha com espaço de 6em \hspace{6em} entre as palavras.
```

#### Exemplo

```
Linha com espaço de 6em          entre as palavras.
```

- `\vspace{medida}` Põe espaço vertical entre as linhas.

#### Código

```
Uma linha  
\vspace{3em}  
separada por 3em de outra.
```

**Exemplo**

Uma linha
-----------

separada por 3em de outra.
----------------------------

Outros comandos úteis:

- `\hrulefill` Produz uma linha horizontal \_\_\_\_\_
- `\dotfill` Produz uma linha pontilhada horizontal .....

## 4 Tabelas, Listas, Imagens e Minipage

### 4.1 Tabelas

Tabelas são feitas da seguinte maneira:

**Código**

```
\begin{tabular}{| l | c | r |} \hline
coluna 1 linha 1 & coluna 2 linha1 & coluna 3 linha1 \\ \hline
coluna 1 linha 2 & coluna 2 linha2 & coluna 3 linha2 \\ \hline
coluna 1 linha 3 & coluna 2 linha3 & coluna 3 linha3 \\ \hline
\end{tabular}
```

As instruções `l`, `c`, `r` referem-se ao alinhamento do texto dentro da tabela, esquerda, centro e direita respectivamente. Cada elemento separado por `|` indica o estilo da coluna e o número de elementos separados pela pipeline é a quantidade de colunas que a tabela terá. A pipeline faz uma separação vertical dos elementos da tabela. Para gerar uma linha horizontal separando os elementos, use `\hline` ao final de cada quebra de linha. Para pular colunas use `&` (e comercial).

**Exemplo**

coluna 1 linha 1	coluna 2 linha1	coluna 3 linha1	
coluna 1 linha 2	coluna 2 linha2	coluna 3 linha2	
coluna 1 linha 3	coluna 2 linha3	coluna 3 linha3	

É possível criar tabelas com colunas de múltiplos alinhamentos, multicolumnas, sem linhas verticais ou horizontais. Colunas com múltiplos alinhamentos existem dentro de tabelas com multicolumnas. Para criar uma tabela nesse estilo, defina no segundo par de chaves em qual colunas se deseja o multi alinhamento e dentro da tabela, a partir de qual linha se deseja que a tabela passe a ser multicolumna. Esse tipo de tabela é gerado da maneira abaixo.

**Código**

```
\begin{tabular}{ | c | lr | } \hline
números & \multicolumn{2}{ c| }{letras e símbolos} \\ \hline
1 & a & % \\ \cline{2-3}
2 & b & * \\ \hline
\end{tabular}
```

**Exemplo**

números	letras e símbolos	
1	a	%
2	b	*

Repare que a tabela é declarada com duas colunas, porém apresenta três a partir do ponto em que o comando `\multicolumn` foi declarado. Esse comando apresenta três argumentos, `\multicolumn{n}{pos}{item}` onde `n` é o numero de colunas a ser mesclada, `pos` é o posicionamento do conteúdo e `item` é o nome do conteúdo em si. O comando `\cline{2-3}` põe linha horizontal apenas entre as linhas indicadas (dois e três).

Também é possível alterar a largura das colunas utilizando o comando `p{medida}` no lugar das letras `c,r,l` na declaração da tabela. Desta maneira é possível digitar textos dentro da tabela, onde serão automaticamente ajustados a parte da tabela que recebeu tamanho personalizado.

Colocando tabelas dentro do comando `table` é possível alterar o local da página onde as tabelas aparecerão.

Ex:

**Código**

```

\begin{table}[h]
  \centering
  \begin{tabular}{p{4cm}|c}
    & \\\ \hline
    &
  \end{tabular}
\end{table}

```

### Exemplo

Frases serão automaticamente formatadas	Mas não são nessa coluna sem formatação
para caber nessa coluna de 4cm.	fim da tabela

Para posicionar a tabela em lugares diferentes da página use as letras no argumento do table para:

- h - Mantém a tabela aonde ele foi colocada
- b - Coloca a tabela na parte inferior da página
- t - Coloca a tabela na parte superior da página
- p - Coloca a tabela numa página separada

Entretanto, pode ser que o L<sup>A</sup>T<sub>E</sub>X não o aceite por motivos estéticos.

## 4.2 Listas

Para gerar listas, use os comandos mostrados abaixo.

### Código

O que precisamos para fazer café(lista padrão):

```
\begin{itemize}
  \item açúcar
  \item água quente
  \item café
  \item filtro
\item garrafa térmica
\end{itemize}
```

### Exemplo

O que precisamos para fazer café(lista padrão):

- açúcar
- água quente
- café
- filtro
- garrafa térmica

### Código

O que precisamos para fazer café(lista com símbolos):

```
\begin{itemize}
  \item[{$\clubsuit$}] açúcar
  \item[{$\ast$}] água quente
  \item[{$\star$}] café
  \item[{$\vee$}] filtro
\item[{$\cup$}] garrafa térmica
\end{itemize}
```

### Exemplo

O que precisamos para fazer café:

- ♣ açúcar
- \* água quente
- \* café
- ∨ filtro
- ∪ garrafa térmica

### Código

O que precisamos para fazer café(lista enumerada):

```
\begin{enumerate}
  \item açúcar
  \item água quente
  \item café
  \item filtro
\item garrafa térmica
\end{enumerate}
```

### Exemplo

O que precisamos para fazer café:

1. açúcar
2. água quente
3. café
4. filtro
5. garrafa térmica

### Código

O que precisamos para fazer café(lista com descrição):

```
\begin{description}
  \item[açúcar] para adoçar
    o café
  \item[água quente] para passar
    no pó do café
  \item[café] em pó, para passar a
    bebida
  \item[filtro] para não deixar o
    pó se misturar com o liquido
  \item[garrafa térmica] para guar-
    dar o café recém passado
\end{description}
```

### Exemplo

O que precisamos para fazer café:

**açúcar** para adoçar o café  
**água quente** para passar no pó do café  
**café** em pó, para passar a bebida  
**filtro** para não deixar o pó se misturar com o liquido  
**garrafa térmica** para guardar o café recém passado

## 5 Figuras

Para inserir arquivos de imagem é necessário o pacote `\usepackage{graphicx}`. Ela pode estar nos seguintes formatos: jpg, png, pdf ou eps.

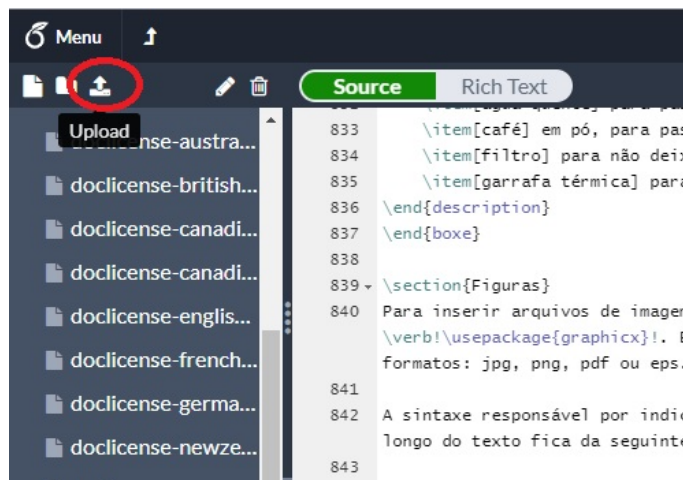
A sintaxe responsável por indicar a figura que será inserida ao longo do texto fica da seguinte forma:

```
\includegraphics[parametro=comprimento]{arquivo.jpg}
```

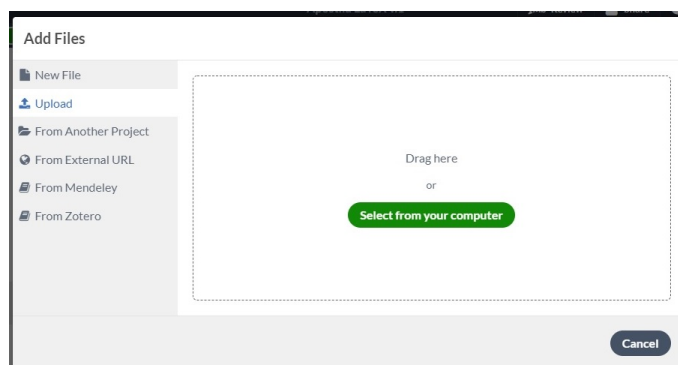
Importante: Para que a figura seja inserida no texto, ela deve estar na mesma pasta que o arquivo main.tex. Para fazer upload de imagens para o overleaf, siga as instruções abaixo.

### Código

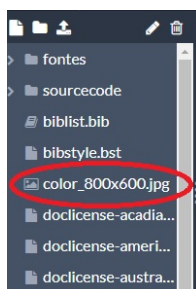




(a): Selecione a opção upload no canto superior esquerdo



(b): Escolha o(s) arquivos de seu computador(select from your computer).



(c): após o upload eles serão listados em ordem alfabética e podem ser usado no comando includegraphics

Também pode ser inserida dentro do comando `figure` para um ajuste mais fino, onde `\label{rótulo:nome}` indica um rótulo, pode ser `fig`, `tab`, para figura, tabela, ou qualquer outra coisa que se queira marcar. O nome do rótulo é usado nas referências do item no documento. Por exemplo, a figura 1(b) mostra o simbolo da Universidade Federal de Santa Maria. Recomenda-se usar rótulos diferentes para coisas diferentes a fim de se evitar confundir os itens marcados.

### Código

```
\begin{figure}[h!]  
  \centering  
  \includegraphics[width=0.5\textwidth]{color_800x600.jpg}  
  \caption{\UFSM}  
  \label{fig:UFSM}  
\end{figure}
```

### Exemplo



A letra no argumento opcional da declaração do comando da figura significa:

- h - aqui, mesmo lugar onde foi colocado
- t - acima, no topo da página
- b - abaixo, na parte inferior da página
- p - em uma página extra
- ! - declarado ao lado da letra, força a localização especificada pela letra

Os argumentos usados no comando `\includegraphics[]{}` são:

Comando	Descrição
<code>width</code>	Corresponde a largura da figura.
<code>height</code>	Corresponde a altura da figura.
<code>scale</code>	Corresponde a escala da figura.
<code>angle</code>	Corresponde a uma rotação no sentido horário.
<code>page</code>	Apenas para PDF's, indica a página a ser utilizada.

Também é possível colocar duas ou mais imagens lado a lado usando o comando `subfigure` e o pacote `subcaption`:

### Código

```
\begin{figure}[h!]
  \centering
  \begin{subfigure}[b]{0.4\linewidth}
    \includegraphics[width=\linewidth]{color_800x600.jpg}
    \caption{Símbolo 1: \UFSM}
  \end{subfigure} ~~
  \begin{subfigure}[b]{0.4\linewidth}
    \includegraphics[width=\linewidth]{color_800x600.jpg}
    \caption{Símbolo 2: \UFSM}
  \end{subfigure}
  \caption{O mesmo brasão da \UFSM \, duas vezes}
  \label{fig:\UFSM2}
\end{figure}
```

### Exemplo



(a): Universidade Federal de Santa Maria



(b): Universidade Federal de Santa Maria

O mesmo brasão da Universidade Federal de Santa Maria duas vezes

Os "tils" apenas colocam um espaço entre as imagens para que as legendas não fiquem grudadas. Quanto mais vezes o comando `subfigure` for declarado dentro do `figure`, mais imagens aparecerão dentro da mesma imagem. Tenha em mente que precisará alterar a escala das imagens de modo a fazê-las aparecer de forma coerente na página.

## 5.1 Minipage

Utilizada para colocar itens lado a lado, ou simplesmente para dar destaque a algo no meio do texto, esse ambiente cria uma área com formato de uma página, com largura definida através do comando:

### Código

```
\begin{minipage}[pm] [altura] [pt] {largura}
  texto
\end{minipage}
```

`pm` refere-se a posição da minipágina no texto. Pode ser `t`(top), `c`(center) ou `b`(bottom); topo, centro e inferior, respectivamente. Altura é a altura da minipágina, `pt` refere-se ao alinhamento do texto dentro da minipágina, sendo atribuído também as letras `t`, `c` e `b`. Largura refere-se a largura da minipágina. As notas de referências dos itens da `minipage` aparecem no final da `minipage`, não no final da página.

### Código

```
\begin{minipage}[t]{0.5\linewidth}
Esta é uma minipágina, alinhada à esquerda da página,
ocupando metade do tamanho da linha e com uma nota
de referência no final da
minipágina\footnote{nota do final da minipage}.
\end{minipage}
```

### Exemplo

Esta é uma minipágina, alinhada à esquerda da página, ocupando metade do tamanho da linha e com uma nota de referência no final da minipágina<sup>a</sup>.

---

<sup>a</sup>nota do final da minipage

### Código

```
\begin{minipage}{0.2\textwidth}
\begin{tabular}{|c|c|c|}
\hline
A & B & C \\
\hline
1 & 2 & 3 \\
\hline
4 & 5 & 6 \\
\hline
\end{tabular}
\end{minipage}

\begin{minipage}{0.2\textwidth}
\begin{tabular}{c|c|c}
A & B & C \\
\hline
1 & 2 & 3 \\
\hline
4 & 5 & 6 \\
\end{tabular}
\end{minipage}

\begin{minipage}{0.2\textwidth}
\begin{tabular}{|c|c|c|}
\hline
A & B & C \\
\hline
1 & 2 & 3 \\
\hline
4 & 5 & 6 \\
\hline
\end{tabular}
\end{minipage}
```

**Exemplo**

A	B	C	A	B	C	A	B	C
1	2	3	1	2	3	1	2	3
4	5	6	4	5	6	4	5	6

Declarar mini páginas uma imediatamente após a outra, sem linhas em branco entre os comandos, faz ambas se alinharem lado a lado, prestando-se atenção a sua escala.

## 6 Novos comandos e ambientes

Eventualmente algumas coisas serão repetidas tantas vezes durante o trabalho que sua digitação pode ser simplificada com a criação de comandos. É possível predefinir comandos para que palavras ou itens customizados apareçam no texto sem a necessidade de digitar a sua sintaxe completa. Por exemplo, se desejamos digitar Universidade Federal de Santa Maria várias vezes durante o texto, podemos usar o comando:

```
\newcommand{\UFSM}{Universidade Federal de Santa Maria}
```

no preâmbulo, depois é só usar `\UFSM` durante o texto para aparecer a frase Universidade Federal de Santa Maria. De modo geral, ao declarar o comando "new command", a primeira chave refere-se ao comando que usaremos no texto para fazer aparecer o que esta dentro da segunda chave. Se o que se quer abreviar com a criação de um comando é um símbolo matemático, por exemplo, basta usar o comando `\newcommand{\ag}{\ensuremath{\alpha}}`. Dessa forma, para digitar a letra grega alpha no texto basta usar o comando `\ag` ao invés de chamar o ambiente matemático e digitar o nome do símbolo. O comando `\ensuremath{}` apenas força o L<sup>A</sup>T<sub>E</sub>X a iniciar o ambiente matemático e colocar nele o que está dentro das chaves.

## 7 Símbolos

Abaixo estão listados alguns dos símbolos mais usados no cotidiano, junto de seus comandos.

### 7.1 Alfabeto Grego

<code>\alpha</code> $\alpha$	<code>\kappa</code> $\kappa$	<code>\tau</code> $\tau$	<code>\Xi</code> $\Xi$
<code>\beta</code> $\beta$	<code>\lambda</code> $\lambda$	<code>\upsilon</code> $\upsilon$	<code>\Pi</code> $\Pi$
<code>\gamma</code> $\gamma$	<code>\mu</code> $\mu$	<code>\phi</code> $\phi$	<code>\Sigma</code> $\Sigma$
<code>\delta</code> $\delta$	<code>\nu</code> $\nu$	<code>\varphi</code> $\varphi$	<code>\Upsilon</code> $\Upsilon$
<code>\epsilon</code> $\epsilon$	<code>\xi</code> $\xi$	<code>\chi</code> $\chi$	<code>\Phi</code> $\Phi$
<code>\varepsilon</code> $\varepsilon$	<code>\pi</code> $\pi$	<code>\psi</code> $\psi$	<code>\Psi</code> $\Psi$
<code>\zeta</code> $\zeta$	<code>\varpi</code> $\varpi$	<code>\omega</code> $\omega$	<code>\Omega</code> $\Omega$
<code>\eta</code> $\eta$	<code>\rho</code> $\rho$	<code>\Gamma</code> $\Gamma$	
<code>\theta</code> $\theta$	<code>\varrho</code> $\varrho$	<code>\Delta</code> $\Delta$	
<code>\vartheta</code> $\vartheta$	<code>\sigma</code> $\sigma$	<code>\Theta</code> $\Theta$	
<code>\iota</code> $\iota$	<code>\varsigma</code> $\varsigma$	<code>\Lambda</code> $\Lambda$	

### 7.2 Símbolos Matemáticos

Para usar alguns dos símbolos utilize o pacote: `\usepackage{amssymb}`

<code>\pm</code> $\pm$	<code>\doteq</code> $\doteq$	<code>\oint</code> $\oint$	<code>\supseteq</code> $\supseteq$
<code>\mp</code> $\mp$	<code>\leq</code> $\leq$	<code>\propto</code> $\propto$	<code>\nsupseteq</code> $\not\supseteq$
<code>\times</code> $\times$	<code>\geq</code> $\geq$	<code>\therefore</code> $\therefore$	<code>\nsubseteq</code> $\not\subseteq$
<code>\ast</code> $\ast$	<code>\geqslant</code> $\geqslant$	<code>\because</code> $\because$	<code>\subseteq</code> $\subseteq$
<code>\div</code> $\div$	<code>\nless</code> $\nless$	<code>\forall</code> $\forall$	<code>\ni</code> $\ni$
<code>\setminus</code> $\setminus$	<code>\ngtr</code> $\ngtr$	<code>\angle</code> $\angle$	<code>\in</code> $\in$
<code>\sim</code> $\sim$	<code>\nleqslant</code> $\nleqslant$	<code>\measuredangle</code> $\measuredangle$	<code>\notin</code> $\notin$
<code>\approx</code> $\approx$	<code>\ngeqslant</code> $\ngeqslant$	<code>\exists</code> $\exists$	<code>\sqrt{1}</code> $\sqrt{1}$
<code>\simeq</code> $\simeq$	<code>\infty</code> $\infty$	<code>\cap</code> $\cap$	<code>\frac{1}{2}</code> $\frac{1}{2}$
<code>\cong</code> $\cong$	<code>\ll</code> $\ll$	<code>\cup</code> $\cup$	
<code>\neq</code> $\neq$	<code>\gg</code> $\gg$	<code>\subset</code> $\subset$	
<code>\equiv</code> $\equiv$	<code>\sum</code> $\sum$	<code>\supset</code> $\supset$	
<code>\not\equiv</code> $\not\equiv$	<code>\int</code> $\int$	<code>\subseteq</code> $\subseteq$	

## 8 Matemática

Vamos agora falar do ambiente matemático. As equações são apresentadas nesse modo, podendo aparecer no meio do texto ou em destaque. Para digitar equações no meio do texto, coloque-as entre  $\$ $$  e, coloque-as entre  $\$\$ \$\$$  para que apareçam em destaque. Para digitar equações em bloco usa-se o comando:

```
\begin{equation}
  equação
\end{equation}
```

ou

```
\[
  equação
/]
```

Equações em destaque são numeradas automaticamente de acordo com sua seção. Para que isso não aconteça, use o comando `\nonumber` ao final de cada linha da equação dentro do comando. Isso é muito útil para evitar que por exemplo a resolução de uma equação seja reconhecida como um conjunto de equações pelo L<sup>A</sup>T<sub>E</sub>X.

### 8.1 Construindo equações

#### 8.1.1 Potência e Índices

Potências são feitas usando o comando:  $\$base^{\text{expoente}}\$$

E índices são feitos usando:  $\$base_{\text{índice}}\$$

#### Código

```
\log_{a}b
```

#### Exemplo

```
\log_a b
```



**Código**

```
ax^{2}+bx+c=0
```

**Exemplo**

```
ax^2 + bx + c = 0
```

## 8.2 Frações e Raízes

Frações são feitas com o comando:

```
\frac{numerador}{denominador}
```

Exemplo de soma de frações:

**Código**

```
$$\frac{1}{2} + \frac{1}{2} = 1$
```

**Exemplo**

```
 $\frac{1}{2} + \frac{1}{2} = 1$ 
```

Para usar raízes, use `$$\sqrt{[]}{}$` onde o que está dentro dos colchetes `[]` significam o grau da raiz e o que está dentro das chaves `{}` é o que vai dentro da raiz. Se os colchetes forem omitidos, a raiz é quadrada!

**Código**

```
$$\sqrt{[3]{9}}=3$
```

**Exemplo**

$$\sqrt[3]{9} = 3$$

**8.3 Matriz e delimitadores**

Matrizes e seus delimitadores fazem parte de um ambiente chamado `array` que separa os itens em linhas e colunas. Esse ambiente é declarado do mesmo jeito que as tabelas.

**Código**

```
\begin{equation} \nonumber
  \begin{array}{lr}
    x & y \\
    z & w \\
  \end{array}
\end{equation}
```

**Exemplo**

$$\begin{array}{lr} x & y \\ z & w \end{array}$$

Delimitadores são usados juntos de matrizes para representar sistemas, conjuntos, etc. Podem ser usados parênteses, colchetes, chaves e módulo como delimitadores. Veja exemplos de seus usos abaixo.

Parenteses:

### Código

```
\[
  \left(
  \begin{array}{lr}
    x & y \\
    z & w \\
  \end{array}
  \right)
\]
```

### Exemplo

$$\left( \begin{array}{lr} x & y \\ z & w \end{array} \right)$$

Colchetes:

### Código

```
\[
  \left[
  \begin{array}{lr}
    x & y \\
    z & w \\
  \end{array}
  \right]
\]
```

**Exemplo**

$$\begin{bmatrix} x & y \\ z & w \end{bmatrix}$$

Chaves(repare na peculiaridade de seu uso):

**Código**

```
\[
  \left\{
  \begin{array}{lr}
    x & y \\
    z & w
  \end{array}
  \right.
\]
```

**Exemplo**

$$\left\{ \begin{array}{lr} x & y \\ z & w \end{array} \right.$$

Módulo:

### Código

```
\[
  \left|
  \begin{array}{lr}
    x & y \\
    z & w
  \end{array}
  \right|
\]
```

### Exemplo

$$\left| \begin{array}{lr} x & y \\ z & w \end{array} \right|$$

Colocar um ponto após o `\right`. faz o delimitador do lado direito desaparecer.

Chaves(repare na peculiaridade de seu uso):

### Código

```
\[
  \left\{
  \begin{array}{lr}
    x & y \\
    z & w
  \end{array}
  \right\}
\]
```

**Exemplo**

$$\left\{ \begin{array}{cc} x & y \\ z & w \end{array} \right\}$$

Repare também no uso do comando `\[ \]` em bloco para representar o conjunto de informações em destaque no texto e no fato de que ele não numera equações.

## 8.4 Funções

Vamos tratar agora das funções básicas do cálculo e como elas podem ser expressas aqui. Todos os comandos estão listados abaixo, ao lado de seus resultados na tela após a compilação do documento.

### 8.4.1 Somatório

**Código**

```
\[  
\sum_{n=0}^{100} n + 1  
\]
```

**Exemplo**

$$\sum_{n=0}^{100} n + 1$$

### 8.4.2 Limite

#### Código

```
\[
\lim_{x\rightarrow\infty} \frac{x^3}{e^x}
\]
```

#### Exemplo

$$\lim_{x \rightarrow \infty} \frac{x^3}{e^x}$$

### 8.4.3 Integral

#### Código

```
\[
f(t)= \int_0^{\infty} x dx
\]
```

#### Exemplo

$$f(t) = \int_0^{\infty} x dx$$

### 8.4.4 Tipos especiais de letras

Dentro do ambiente matemático também há a possibilidade de se mudar os tipos de letras da seguinte forma:

$\mathrm{X Y Z}$  → XYZ

$\mathcal{X Y Z}$  →  $\mathcal{X} \mathcal{Y} \mathcal{Z}$

$\mathbf{X Y Z}$  → **XYZ**

$\mathsf{X Y Z}$  → XYZ

$\mathtt{X Y Z}$  → XYZ

$\mathit{X Y Z}$  → XYZ

$\mathbb{N I Z Q} \rightarrow \text{NIZQ}$

**Nota:** Não se esqueça de usar `\usepackage{amsfonts}` no preâmbulo. Recomenda-se a sua utilização apenas para representar conjuntos. Seu uso para digitar equações pode fazê-las aparecer com letras substituídas por símbolos.

## 8.5 Fórmulas em várias linhas

É uma combinação do ambiente `array` com o ambiente matemático de equação, chamado de `eqnarray`. Também são usados `&` e `\\` com a opção de usar `\nonumber` para não numerar as fórmulas, podendo também ser usado um asterisco na declaração e no fechamento do comando matemático, e de deixar de usar os argumentos `c,l,r`.

### Código

```
\begin{eqnarray*}
  \int_{-L}^L f(x)\,dx \quad \& \quad a_{0} \quad \int_{-L}^L dx + \\
  \sum_{m=1}^{\infty} a_{m} \\
  \int_{-L}^L \cos \frac{m \pi x}{2}\, dx \quad \\
\end{eqnarray*}
```



**Exemplo**

$$\int_{-L}^L f(x) dx = a_0 \int_{-L}^L dx + \sum_{m=1}^{\infty} a_m \int_{-L}^L \cos \frac{m\pi x}{2} dx$$

Para usar a mesma referência para todas as equações, basta declarar no preâmbulo o pacote:

```
\usepackage{subeqnarray}
```

**Código**

```
\begin{subeqnarray}\label{eq:sub}
\slabel{sub1} a^2=&b^2 + c^2\\
\slabel{sub2} a &=&b - 5
\end{subeqnarray}
```

**Exemplo**

$$a^2 = b^2 + c^2 \quad (1a)$$

$$a = b - 5 \quad (1b)$$

Dessa maneira, para se referir ao conjunto de equações basta usar (`\ref{eq:sub}`).

**8.6 Linhas em equações**

Os comandos a seguir mostram algumas maneiras de se destacar equações com linhas.

**Código**

```
 $\overline{xyz}$ 
```

**Exemplo**

$$\overline{xyz}$$

**Código**

```
\underline{xyz}
```

**Exemplo**

$$\underline{xyz}$$
**Código**

```
\overbrace{xyz}
```

**Exemplo**

$$\overbrace{xyz}$$
**Código**

```
\underbrace{xyz}
```

**Exemplo**

$$\underbrace{xyz}$$
**Código**

```
\overleftarrow{xyz}
```

**Exemplo**

$$\overleftarrow{xyz}$$
**Código**

```
\overrightarrow{xyz}
```

**Exemplo**

$$\overrightarrow{xyz}$$

Há a possibilidade de se colocar sob e sobrescritos:

**Código**

```
\overbrace{xyz}^{\{123\}}
```

**Exemplo**

$$\overbrace{xyz}^{123}$$

### 8.6.1 Empilhando

Usando `\stackrel{acima}{abaixo}` é possível colocar um símbolo ou texto acima de outro.

Exemplo:

#### Código

```
\stackrel{a}{b}C$ $\rightarrow$ $\stackrel{1}{2}D$
```

#### Exemplo

$$\begin{matrix} a \\ b \end{matrix} C \rightarrow \begin{matrix} 1 \\ 2 \end{matrix} D$$

## 8.7 Comando phantom

Muito útil quando se quer escrever algo alinhado em sob e sobrescritos.

#### Código

```
$U_{ij}^{\phantom{i}n}$
```

#### Exemplo

$$U_{ij}^r$$

## 8.8 Teoremas

O L<sup>A</sup>T<sub>E</sub>X possui um comando especial que define um ambiente com estrutura para digitar teoremas, proposições, etc. Comece declarando no preâmbulo o comando:

```
\newtheorem{ambiente}[título]{numeração}
```

onde ambiente é um nome qualquer que será dado para iniciar o ambiente no texto, título é a denominação que aparecerá antes de qualquer texto dentro desse ambiente, como teorema, lei, axioma, etc e numeração refere-se ao número de vezes que o ambiente foi chamado dentro da seção (parte, capítulo, etc). Este comando pode ser colocado em qualquer parte do seu documento. Em seguida deve ser usado o ambiente com o nome escolhido para escrever o texto.

### Código

```
\newtheorem{mat}{teorema}

\begin{mat}[Pitágoras]
  Em qualquer triângulo retângulo, o quadrado do
  comprimento da hipotenusa é igual à soma dos quadrados
  dos comprimentos dos catetos.
\end{mat}
```

### Exemplo

**teorema 1 (Pitágoras)** *Em qualquer triângulo retângulo, o quadrado do comprimento da hipotenusa é igual à soma dos quadrados dos comprimentos dos catetos.*

### Código

```
\newtheorem{mat}{teorema}

\begin{mat}[comentário]
  Esse é apenas um comentário para demonstrar
  a funcionalidade do comando.
\end{mat}
```

### Exemplo

**teorema 2 (comentário)** *Esse é apenas um comentário para demonstrar a funcionalidade do comando.*

## Referências

- [1] H Kopka and PW Daly. A guide to  $\{\backslash\text{LaTeX}\}$ -document. 1995.