Bolsa Instrutor Dinf/PRAE

Universidade Federal do Paraná
Deparamento de Informática
Pró-Reitoria de Assuntos Estudantis

Curso LaTeX

Professor Responsável :
Eduardo J. Spinosa.

CURITIBA - PR
2 de Setembro de 2014

Conteúdo

1 Introdução:

O que é o LaTeX , links importantes, estrutura básica de um documento em LaTeX , classes de documentos, pacotes e codificação.

1.1 O que é o LaTeX:

O LaTeX é um conjunto de macros para TeX (linguagem de programação para edição de textos) que facilita a edição de textos complexos, arquivos LaTeX tem a extensão .tex, e precisam ser compilados para um formato legível como .pdf, .dvi, .ps... em um arquivo .tex nem tudo que você escrever será visto pelo leitor, a vantagem disso é que o arquivo a ser visualizado não precisa guardar informações sobre edição e o leitor não pode editar o texto(a menos que tenha o arquivo .tex).

Para escrever em LaTeX você precisa de um editor de texto de sua preferência, como bloco de notas, emacs, gedit... e um compilador de LaTeX; para o ambiente Windows você pode usar o MIKTeX, um editor e compilador de arquivos com extensão .tex.

No ambiente Linux, use o comando **latex arquivo.tex**, para compilar e gerar **arquivo.dvi**, ou **pdflatex** para gerar o **arquivo.pdf**(precisa compilar antes).

1.2 Links:

Mais informações sobre o LaTeX na internet:

- Obtendo MIKTeX(Windows):
  http://miktex.org/.
- Wikibooks LaTeX :
  http://pt.wikibooks.org/wiki/Latex.
- The (not) so short Introduction to LaTeX :
  http://www.ctan.org/tex-archive/info/lshort/.
- ABNTeX :
  http://abntex.codigolivre.org.br.

1.3 Estrutura básica:

Vamos ver um exemplo de um arquivo fonte para o LaTeX :

```
\documentclass[11pt, ar4paper]{article}
% Comandos globais
\usepackage[portuguese]{babel}
```

```
\usepackage[latin1]{inputenc}
\begin{document}
% Texto e comandos locais
Hello World!
\end{document}
```

O texto foi obtido com o ambiente verbatim:
`\begin{verbatim}`
O texto nesse ambiente é impresso exatamente como digitado.
`\end{verbatim}`

O ambiente verbatim cria um paragrafo para o texto, para o mesmo efeito porém dentro de uma linha use o comando `\verb!texto!` , se você usar o carácter ! no texto, pode substitui-lo por outro não usado no texto, como # .

Um arquivo em LaTeX contém o texto e comandos de como o texto deve ser processado, palavras são separadas por um ou mais espaços, parágrafos são separados por uma ou mais linhas em branco; comandos são iniciados com \, uma \ sozinha produz um espaço; lembre-se de não digitar nenhum dos caracteres especiais `& $ # % _ { } ^ ~ \` exceto como um comando, para imprimir esses caracteres use os comandos:
`\& \$ \# \% \_ \{ \} \^{} \~{} $\backslash$`;
para "aspas" use ``` ``aspas'' ```, veja também os comandos para fechar/criar páginas e pulo de linha em 3.1 e comandos para modificar o layout da página em 4.

Um ambiente é uma região do texto que tem um tratamento especial definido pelo autor, um ambiente começa com `\begin{nome do ambiente}` e termina com `\end{nome do ambiente}`, como por exemplo o ambiente `\begin{document} e \end{document}` que define onde começa e termina o arquivo a ser processado pelo LaTeX

Normalmente todo arquivo .tex tem um preâmbulo e um corpo, tudo entre `\documentclass{}` e `\begin{document}` é o preâmbulo do arquivo, as definições aqui afetam todo o documento, como tipo do documento, formato do papel, altura e largura do texto, se nenhuma definição for escolhida o LaTeX define valores standart; `\documentclass{}` define a classe do documento, `\begin{document}` e `\end{document}` definem o começo e o final do arquivo, o corpo do arquivo, com o texto e seus comandos locais; oquê estiver escrito após % é um comentário e será ignorado pelo LaTeX .

1.4 Classes de documentos e pacotes

1.4.1 Classes de documentos

No comando `\documentclass[opções]{estilo}`, o **estilo** pode ser: **article**, **report**, **book** ou **letter**, as **opções** são: **10pt**, **11pt** ou **12pt** para o tamanho base das letras do texto; **a4paper** se o papel for A4 (para o estilo

letter não é necessário nenhuma indicação), **landscape** para a impressão no modo paisagem, **twocolumn** para a impressão em duas colunas, **twoside** para a impressão nos dois lados do papel, **titlepage** para que no estilo **article** seja gerada uma página separa com o título (para os outros estilos não é necessária esta opção).

1.4.2 Pacotes

`\usepackage[portuguese]{babel}` indica para usar o pacote **babel**, com a opção de língua **portuguese**, gera datas e nomes como Capítulo, Bibliografia em português com estilo brasileiro.

`\usepackage{graphicx,color}` indica para usar os pacotes **graficx** e **color**, que permitem incluir figuras e colorir o texto.

`\graphicspath{{../figuras/}}` o sistema vai procurar as figuras na pasta **figuras** que fica na mesmo pasta com os arquivos .tex.

`\usepackage[latin1]{inputenc}` indica para usar o pacote **inputenc** com a opção **latin1**, que define uma codificação para os caracteres em que os acentos são digitados diretamente pelo teclado.

`\usepackage{amsthm,amsfonts}` indica para usar os pacotes da American Mathematical So-ciety **amsthm** e **amsfonts**. O primeiro, entre outras coisas, define um estilo para a escrita dos teoremas e o segundo adiciona alguns estilos de letras, por exemplo $\mathbb{R}$, $\mathbb{C}$ e $\mathbb{N}$ foram geradas com `$\mathbb{R}$`, `$\mathbb{C}$` e `$\mathbb{N}$` por causa da adição deste pacote.

`\setlength{\textwidth}{16 cm}` indica que a largura do texto é de 16 cm.

`\setlength{\textheight}{20 cm}` indica que a altura do texto é de 20 cm.

`\evensidemargin 0 cm` indica que a margem esquerda das páginas pares é zero (tamanho real da margem é a soma da variável `\hoffset`).

`\oddsidemargin 0 cm` indica que a margem esquerda das páginas é zero.

`\topmargin 0 cm` indica que a margem superior é zero.

`\baselineskip 65 mm` define a distância entre as linhas como sendo de 65 mm.

`\sloppy` reduz o número de divisões nas palavras que são impressas nos finais das linhas.

Você pode também "criar" novos comandos com o `\newcommand`, como definir nomes mais simples para comandos com nomes grandes, por exemplo:

`\newcommand{\binv}{\backslash}` novo nome para o comando que imprime \ .

`\newcommand{\til{\~{}}` o comando `\til` agora imprime ~.

1.5 Codificação:

Para usar uma linguagem diferente da inglesa é preciso fazer alguns "ajustes" para que o LaTeX codifique corretamente o texto, para isso precisamos usar o pacote `\usepackage[língua]{babel}`, o pacote **babel** irá ajustar automaticamente as características da língua que você escolheu, se você não tiver o pacote da língua instalado o documento irá compilar mas você não terá as traduções corretas na impressão do documento; você pode escolher mais de uma língua no pacote como: `\usepackage[linguaA,linguaB]{babel}`.

Usamos também o pacote para a codificação do texto:

`\usepackage[codificação]{inputenc}`, **inputenc** diz ao LaTeX qual a codificação do arquivo .tex, assim você pode usar acentuação diretamente com as teclas do teclado; a codificação do arquivo depende do sistema operacional(UNIX,Windows...), é comum o uso da ISO-8859-1, para isso adicione o pacote:

`\usepackage[latin1]{inputenc}`.

A maioria dos sistemas operacionais usam a decodificação Unicod(UTF-8) como padrão, para esses sistemas (como o Ubuntu) use o pacote:

`\usepackage[utf8]{inputenc}`.

Para a língua portuguesa usamos os pacotes:

```
\usepackage[portuguese]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

você pode substituir **portuguese** por **brazilian**, lembre-se de usar a codificação correta, se você definiu o arquivo como:

UTF-8 use o pacote `\usepackage[utf8]{inputenc}`,

ISO-8859-1 use o pacote `\usepackage[latin1]{inputenc}`.

2 Seções:

Título e abstract, seções, apêndice e geração de índices, mensagens de erro, include, input e subfile.

2.1 Include, input e subfiles:

Quando o texto se torna muito complexo podemos facilitar o trabalho quebrando o arquivo .tex em partes, usar múltiplos arquivos é uma tática fácil no LaTeX com os comandos:

`\input{}, \include{}, \includeonly{} e \subfile{}`.

2.1.1 Input:

O comando `\input{nome do arquivo.tex}` é usado para unir pedaços de arquivos .tex, todo o conteúdo do **nome do arquivo.tex** será considerado, exemplo:

arquivos.tex:

```
\begin{document}
\input{./arquivo1.tex}
\input{./arquivo2.tex}
\input{./arquivo3.tex}
\end{document}
```

quando o **arquivos.tex** for compilado, o LaTeX irá procurar na pasta corrente o **arquivo1.tex**, **arquivo2.tex** e **arquivo3.tex** e irá unir **todo** o conteúdo deles em ordem, se algum arquivo não estiver na pasta corrente você precisa detalhar o destino dele, exemplo:

`\input{./meus arquivos/arquivo1.tex}`

(cuidado com os nomes dos arquivos, muitas vezes espaços em branco podem gerar erros, use _ no lugar).

2.1.2 Include:

Podemos usar também o comando **include** no preâmbulo quando existirem muitos arquivos: `\includeonly{nome do arquivo1, nome do arquivo2...}` e os comandos `\include{nome do arquivo1}`, no corpo do arquivo.

Existe também o comando `\includepdf{}` que insere uma ou mais páginas de um arquivo PDF, para isso você precisa do pacote

`\usepackage[final]{pdfiles}`, você pode também usar o pdfpages packages; agora use o comando

`\includepdf[pages=1-2]{nome do arquivo.pdf}`, para imprimir as páginas 1 e 2

`\includepdf[pages=-]{nome do arquivo.pdf}`, para imprimir todas as páginas; compile a saída com o pdflatex para funcionar perfeitamente.

2.1.3 Subfile:

Os comandos `input{} e include{}` não são compilados individualmente, mas apenas quando o arquivo que os contém é compilado, se você precisa que os arquivos menores também sejam compilados previamente, você pode usar o pacote subfiles; primeiro no arquivo superior você precisa adicionar o pacote `usepackage{subfiles}`, e agora em vez de usar os comandos

`input{nome do arquivo} e include{nome do arquivo}` use o

`\subfile{nome do arquivo}`, a última parte é adicionar no início dos arquivos menores o comando:

`\documentclass[arquivo superior.tex]{subfiles}`.

2.2 Título e abstract:

Para gerar um título do documento usamos os comandos no corpo do arquivo

```
\title{título}

\author{autor1 \\endereço1 \and autor2 \\endereço2}

\date{data}

\maketitle
```

Você pode omitir o comando `\date{data}`, assim o LaTeX imprime a data atual; existe também o comando `\thanks{rodapé}` para texto de rodapé que pode ser usado no título, autor e data, como por exemplo:

`\author{João \thanks{ O cara}}`;

se o tipo do documento for **book** os comandos geram uma página separada para o título do documento.

Você pode ainda criar um título mais complexo com o ambiente titlepage: `\begin{titlepage} \end{titlepage}`;

usando um arquivo separado para a capa, adicionando ela no documento com o comando `\input{}`, podemos criar modelos mais complexos como por exemplo:

capa.tex:

```
\begin{titlepage}

  \begin{center}

    \textmd{ \LARGE Grupo}\\[1cm]
    % por uma logo bacana aqui %
    \textsc{ \Large Universidade Federal do Paraná}\\
    \textsc{ \Large Deparamento de}\\[1.5cm]

    \HRule\\[0.3cm]
    { \huge Título }
    \HRule\\[1.5cm]

      \emph{Professor Responsável :} \\
      Nome.\\[1cm]

      \emph{Bolsistas : } \\
      Nomes.\\

    \vfill
```

```
    { \large \uppercase{Curitiba - PR}} \\
    { \large \today }

  \end{center}

\end{titlepage}
```

Em publicações cientificas é comum começar o documento com um resumo sobre o assunto que será tratado, para isso usamos o ambiente **abstract**, disponível apenas para os tipos de documento **article** e **report**, no corpo do arquivo use:

```
\begin{abstract}
Resumo do documento.
\end{abstract}
```

2.3 Seções, apêndices e índices:

2.3.1 Seções:

Você pode usar os seguintes comandos para produzir um seccionamento automático e sequencial:

```
\part{título},
\chapter{título},
\section{título},
\subsection{título}
\subsubsection{título}
\paragraph{título}
\subparagraph{título}
```

O comando `\part{}` é apenas para o tipo de documento **book**, o comando `\chapter{}` **apenas** para os tipos **book** e **report**, os outros comandos podem ser usados em qualquer tipo de documento, exceto **letter**; esses comandos formam uma hierarquia, os capítulos (`\chapter{}`) são divididos em seções (`\section{}`), que são divididas em subseções e assim por diante.

Se o tipo de documento for **book**, então o comando `\chapter{}` sempre começa o capítulo em uma nova página e normalmente, nas páginas de números ímpares, gerando uma página em branco se necessário; para evitar que ele gere uma página em branco, no tipo de documento defina:

`\documentclass[....,ondeside]{book}` imprime apenas em um lado do papel ou

`\documentclass[....,openany]{book}` os capítulos podem começar em páginas pares também.

A numeração das seções é automática, para `\part{}` a numeração é romana (Part I,Part II...), capítulos e seções são números decimais e apêndices (um caso particular de `\chapter`) são letras.

Você pode tirar a numeração das subseções e subsubseções com o comando `\setcounter{secnumdepth}{1}`, o valor padrão é 2, ou com o comando `\section*{}` em cada seção; o comando `\setcounter{section}{3}` define o início da numeração, nesse caso a próxima seção será 4.

Para os capítulos você pode usar os comandos `\frontmatter e \mainmatter`, as páginas entres esses comandos teram a numeração com algarismos romanos, e os capítulos após `\mainmatter` serão numeradas em arábicos e recomeçam da página 1; existe também o comando `\backmatter`, os capítulos depois dele não serão numerados.

2.3.2 Apêndices:

Para criar um apêndice no tipo de documento **book** ou **report** use o comando:

```
\appendix
\chapter{Primeiro Apêndice}
```

no tipo de documento **article** use:

```
\appendix
\section{Primeiro Apêndice}
```

2.3.3 Índices:

O índice ou sumário (table of contents) é gerado automaticamente com o comando `\tableofcontents`, se você usar o pacote `\usepackage[portuguese]{babel}`, o título será Sumário, para mudar o nome você precisa redefinir o comando:

`\renewcommand{\contentsname}{novo nome}`, antes de usar `\tableofcontents`; as vezes é preciso compilar três vezes ou mais para que o índice seja impresso corretamente. Exemplo:

```
\documentclass[11pt, a4paper]{article}
\begin{document}
\input(capa.tex)
\tableofcontents
\input{./capitulo01.tex}
\input{./capitulo02.tex}
\end{document}
```

Você pode usar também o pacote **makeindex**

2.4 Mensagens de erro:

Quando o LaTeX encontra um erro ele apresenta uma mensagem e para a execução, a descrição do erro começa após o ! e a linha onde o erro foi encontrado aparece logo abaixo:

```
!Missing $ inserted
¡inserted text¿
l.181 ...
```

para continuar a execução você precisa digitar uma das opções:

Tecla:	**Ação**:
x	Para imediatamente e sai do programa(e**x**it).
q	Ignora e continua o programa sem mostrar outros erros(**q**uietly)
e	Para o programa e abre o editor de texto na linha do erro(**e**dito)
h	Mostra uma possível solução para o erro(**h**elp)
i	Insere uma solução e continua o programa, a solução é temporária você precisa mudar o arquivo para que o erro não aconteça novamente(**i**nput)
r	Continua o programa ignorando erros, limite de 100 erros(**r**un)

2.4.1 Erros conhecidos:

Alguns erros comuns e suas soluções:

-
```
!Too many }'s.
l.6 \date Abril 2011}
```

Muitos {'s, sempre que você abrir { não esqueça de fechar }.

-
```
!Undefined control sequence.
l.6 \dtae Abril 2011
```

Comando indefinido, o erro mais comum, erro de digitação; a menos que você defina um nome diferente para algum comando, \dtae não é `\date`.

-
```
!Missing $ inserted
```

Não está no modo matemático, um carácter que pode ser usado apenas no modo matemático foi inserido num texto normal, você pode mudar para o modo matemático com o comando `\begin{math} \end{math}`; esse erro pode também aparecer se você usar uma codificação errada, como por exemplo definir o arquivo como UTF-8 sem adicionar o pacote `\usepackage[utf8]{inputenc}` ou definir o arquivo como ISO8859-1 sem usar o pacote `\usepackage[latin1]{inputenc}`, veja codificação 1.5.

- !LaTeX Error: File paralisy.sty not found.

Falta de pacote, quando você usa o comando `\usepackage` para usar um pacote específico do LaTeX ele irá procurar o nome do pacote.sty, você pode ter errado no nome do pacote ou o pacote não está instalado na sua máquina, você pode baixar oas arquivos .sty e colocar na pasta do documento a ser compilado.

- Package Babel Warning: No hyphenation patterns were loaded for the language 'Latin'
  I will use the patterns loaded for \language=0 instead.

Um erro comum ocorre do pacote **Babel** e não do LaTeX , esse erro ocorre quando queremos usar o pacote `\usepackage[latin1]{babel}`, a solução é instalar o pacote da língua usada, veja codificação 1.5.

3 Formatação:

Quebra de linhas e páginas, tipos de texto(cores, tamanhos e estilos), alinhamento e footnote.

3.1 Quebra de linhas e páginas:

Algumas vezes o LaTeX pode fechar/criar páginas e pular linhas de um modo indesejado, para evitar isso usamos os comandos(veja também os comandos para modificar o layout da página em 4):

Comando:	**Ação**:
\newline ou (\\)	Pula uma linha.
*	Pula uma linha e evita a criação de uma nova página.
\linebreak[número]	Pula uma linha dependendo do **número**, que define uma prioridade (0,Provavelmente ignorado; 4,com certeza um pulo de linha).
\newpage	Fecha a página atual e começa uma nova página.
\pagebreak[número]	Começa uma nova página, **número** define a prioridade [0,4].
\nopagebreak[número]	Evita que uma página seja fechada, **número** define a prioridade [0,4].
\clearpage	Fecha a página atual.

3.2 Alinhamento:

Por padrão os paragrafos em LaTeX são sempre completos, para criar um alinhamento particular você deve usar os ambiente, ou os comandos para alinhar figura ou apenas uma linha dsejada:

`\begin{flushleft} \end{flushleft}` ou o comando `\raggedright`,

alinhamento para esquerda;
`\begin{center} \end{center}` ou o comando `\centering`,
centralizado;
`\begin{flushright} \end{flushright}` ou o comando `\raggedleft`,
alinhamento para a direita.

Para modificar a distância entre dois paragrafos você pode usar o comando:

`\vspace{xcm}`, onde x é a distância; veja a seção de layout em 4.

3.3 Tipos de texto:

3.3.1 Estilos e tamanhos:

1. Para alterar o tamanho e estilo de **todo o texto**:

   `\renewcommand*\rmdefault{estilo}\normalfont\upshape`,

   escolhendo o **estilo** que desejar (é preciso também adicionar o pacote do **estilo**), isso só irá funcionar para as fontes definidas no LaTeX se você quiser outros tipos de fontes você terá que baixar e instalar, um bom tutorial para instalação de fontes

2. Para alterar **apenas uma parte do texto** usamos os comandos:

- **Estilos**:

  `\textit{itálico}` produz *itálico*,

  `\textbf{negrito}` produz **negrito**,

  `\textrm{romano}` produz romano,

  `\textsf{sans serif}` produz sans serif,

  `\texttt{máquina de escrever}` produz `máquina de escrever`,

  `\textsc{caixa alta}` produz CAIXA ALTA.

- **Tamanhos**:

  `{\tiny o menor}` produz o menor,

  `{\scriptsize muito pequeno}` produz muito pequeno,

  `{\footnotesize menor}` produz menor,

  `{\small pequeno}` produz pequeno,

  `{\large grande}` produz grande,

  `{\Large maior}` produz maior,

  `{\LARGE maior ainda}` produz maior ainda,

{\huge ainda maior} produz ainda maior,

{\Huge o maior de todos} produz o maior de todos.

3.3.2 Cores:

Para escrever em cores é preciso adiciona o pacote:\usepackage{graphicx,color}; e depois usar os comandos:

\textcolor{blue}{texto em azul} produz texto em azul,
\textcolor{red}{texto em vermelho} produz texto em vermelho,
ou:
{\color{yellow}texto em azul} produz texto em amarelo,
{\color{green}texto em vermelho} produz texto em verde.

3.4 Footnote:

Criar um texto de rodape é simples, você apenas precisa usar o comando \footnote{texto}; exemplo:

Footnote em 3,2,1.\footnote{!FOOTNOTE!}[1]

Para mudar o tipo de marcação você precisa redefinir o comando:

\renewcommand{\thefootnote}{\estilo{footnote}{}, onde **estilo** pode ser **arabic**(1,2,3...), **roman**(i,ii,iii...), **Roman**(I,II,III...), **alph**(a,b,c...) ou **Alph**(A,B,C...).

4 Layout da página:

Ambiente minipage, orientação da página com o ambiente landscape, páginas em multicolunas com multicols, hspace, vspace, vfill, hrulefill.

4.1 Minipage:

Quando queremos alinhar textos ao lado de figuras/tabelas na página podemos usar o ambiente **minipage**, que cria uma página pequena(**minipage**) com o conteúdo do ambiente para depois alinha-la na página conforme desejado:

```
\begin{minipage}[posição minipage][altura][posição conteúdo]{largura}
Texto,figura,tabela,etc
\end{minipage}
```

[1] !FOOTNOTE!

a **posição minipage** define a posição da **minipage** na página, valores: **c**(centro),**t**(topo) e **b**(fundo), valor padrão **h**(onde foi declarada); a **posição conteúdo** é a posição do conteúdo dentro da minipage(**t**, **c** e **b**); **altura** e **largura** definem as dimensões da **minipage**.

Exemplo:

```
\begin{minipage}{7cm}
Texto do lado esquerdo.
\end{minipage}\hfill
\begin{minipage}{7cm}
Figura/tabela do lado direito.
\end{minipage}
```

produz:

Texto do lado esquerdo. Figura/tabela do lado direito.

4.2 Orientação da página:

Por padrão um documento em LaTeX tem a orientação da página como **portrait**, para mudar usamos o ambiente **landscape**, há dois modos de mudar a orientação, local e global(todo o documento);

para modificar todo o documento usamos o pacote:

`\usepackage[landscape]{geometry}`, porém se você também for definir o tamanho do papel com o **geometry** use:

`\usepackage[a4paper,landscape]{geomertry}`.

Para mudar a orientação local use o ambiente **landscape**, por exemplo numa tabela:

```
\begin{landscape}
\begin{table}
\caption{Exemplo}
\centering % Opcional
\begin{tabular}{|l|c|}
\hline
A & B \\
\hline
C & D \\
\hline
\end{tabular}
\end{table}
\end{landscape}
```

produz:

Tabela 1: Exemplo

A	B
C	D

4.3 space e fill:

Muitas vezes o LaTeX pode quebrar páginas e linhas de um modo indesejado, podemos usar os comandos de formatação de texto 3.1, mas isso pode não funcionar com figuras,tabelas,etc para resolver esses problemas usamos os comandos que modificam o layout da página:

`\hspace{xcm}`, cria um espaço horizontal vazio de tamanho **x**;
`\vspace{xcm}`, cria um espaço vertical vazio de tamanho **x**;
`\hfill`, abreviação de `\hspace{\fill}`, preenche o espaçamento horizontal padrão(tamanho da linha);
`\vfill`, abreviação de `\vspace{\fill}`, preenche o espaçamento vertical padrão(tamanho da página);
você pode usar o comando `\hrulefill{xcm}`, para modificar o comando `\hfill`, para não preencher toda a linha.

4.4 Multicolunas:

Para criar um documento com duas colunas por página você pode usar no tipo de documento o argumento **twocolumn**:
`\documentclass[twocolumn]{article}`, porém esse modo nem sempre é satisfatório, você pode ter problemas com a posição de figuras,tabelas,etc.
Por isso usamos o ambiente **multicols**, você precisa do pacote `\usepackage{multicol}` para usar o ambiente; um exemplo de layout de duas colunas:

```
\begin{multicols}{2}
1 coluna \vfill
2 coluna
\end{multicols}
```

produz:

1 coluna 2 coluna

Você pode usar o comando `\columnseprule{}` antes do ambiente **multicols** para definir a separação entre as colunas(valor padrão 0pt), exemplo:

```
\setlength{\columnseprule}{1pt}
\begin{multicols}{2}
1 coluna \vfill
2 coluna
\end{multicols}
```

produz:

1 coluna | 2 coluna

para mudar o espaço horizontal entre as colunas(valor padrão 10pt) usamos o comando antes do ambiente **multicols**: `\columnsep{}`, exemplo:

```
\setlength{\columnsep}{20pt}
\begin{multicols}{2}
```

5 Tabelas e listas:

Tabelas(table e tabular), orientação de tabelas, multi linhas e colunas; listas(itemizes, enumeration e description), teoremas.

5.1 Tabelas

5.1.1 Tabular:

Para criar tabelas usamos o ambiente **tabular**:

```
\begin{tabular}{formato das colunas}
linhas
\end{tabular}
```

O **formato das colunas** define o número de colunas, suas características e o tipo de separação entre as colunas, os símbolos para o **formato das colunas** são:

l, a coluna é alinhada à esquerda;

c, a coluna é centralizada;

r, a coluna é alinhada à direita;

no lugar dos símbolos **l**, **c** e **r** você pode usar o símbolo **p{xcm}**, que define o tamanho das linhas da coluna, onde **x** é o tamanho desejado; para definir o tipo de separação entre as colunas usamos os símbolos:

`|` desenha uma linha vertical;

`||` desenha duas linhas verticais;

@{texto} insere o **texto** em todas as linhas, entre as duas colunas onde ele aparece.

As linhas devem conter as entradas de cada linha da tabelas separando colunas com o símbolo **&** e terminadas com \\; você pode usar o comando \hline entre as **linhas** para criar uma linha horizontal de comprimento igual ao da tabela, dois \hline juntos criam duas linhas horizontais; para concatenar colunas usamos o comando: `\multicolumn{número}{formato}{texto}`, que concatena **número** colunas, o **formato** deve conter um dos símbolos para definir colunas(l, r ou c) e o tipo de separação(`|`,`||`).

Vamos construir uma tabela simples:

```
\begin{center}
\begin{tabular}{|l||c|c|c|c|c|}
\hline
\multicolumn{6}{|c}}{\textbf{1 semestre 2011}}\\
\hline
Horário &Seg &Ter &Qua &Qui &Sex\\
\hline\hline
13:30-14:30 & & & & & \\
\hline
14:30-15:30 & & & & & \\
\hline
15:30-17:30 & & & & & \\
\hline
17:30-19:00 & & & & & \\
\hline
19:00-21:00 & & & & & \\
\hline
21:00-22:00 & & & & & \\
\hline
\end{tabular}
\end{center}
```

Produz:

1 semestre 2011					
Horário	Seg	Ter	Qua	Qui	Sex
13:30-14:30					
14:30-15:30					
15:30-17:30					
17:30-19:00					
19:00-21:00					
21:00-22:00					

Para que uma coluna tenha mais de uma linha de texto, no formato das colunas use o símbolo **p{xcm}** com um tamanho pequeno o suficiente para forçar o LaTeX a mudar de linha.

Para aumentar a distância entra as linhas use, entre o `\begin{center}` e o `\begin{tabular}`, o comando `renewcommand{\arraystretch}{fator}`, onde **fator** é o número de vezes que a distância entre as linhas deve ser aumentada em relação ao padrão, por exemplo **fator** igual a 1.5, aumenta em 50% a distância entre as linhas.

Para colorir uma célula de uma tabela você precisa adicionar o pacote: `\usepackage[table]{xcolor}`, e use o comando:

`\cellcolor[número]{cor}`, onde **número** é a intensidade da **cor** que você escolheu, você pode omitir o **número** e usar a intensidade padrão.

5.1.2 Table:

O ambiente **tabular** nem sempre satisfaz a sua necessidade quando do você precisa construir uma tabela, por isso usamos o ambiente **tabular** dentro do ambiente **table**:

```
\begin{table}[h!]
\caption{Exemplo}
\begin{center}
\begin{tabular}{|l|c|}
\hline
A & B \\
\hline
C & D \\
\hline
\end{tabular}
\end{center}
\end{table}
```

produz:

Tabela 2: Exemplo

A	B
C	D

Com o ambiente **table**, você pode escolher onde a tabela será impressa na página, com `\begin{table}[x]`, **x** define onde a tabela será impressa:

h, onde foi declarada(**h**ere);

t, no topo da página(**t**op);

b, abaixo da página(**b**ottom);

!, força a tabela a ser posicionada onde você escolheu.

Você pode com o ambiente **table** criar uma referência para a tabela com o comando `\label{}`, e depois usar `~\ref{}` para se referir a ela:

```
\begin{table}
\begin{tabular}
.
.
\end{tabular}
\caption{Um exemplo de table}
\label{minha_tabela}
\end{table}
```

5.2 Listas:

Uma lista é um ambiente no LaTeX , não esqueça que você pode usar um ambiente dentro de outro para criar uma lista personalizada:

5.2.1 itemize:

O ambiente itemize produz uma lista onde os itens são procedidos de •;

os itens são separados por um espaço adicional; exemplo:

```
\begin{itemize}
\item item1
\item item2
\end{itemize}
```

produz:

- item1
- item2

5.2.2 enumerate:

O ambiente enumerate produz uma lista onde os itens são numerado em arábicos, no primeiro nível, letras no segundo e algaritmos romanos no terceiro nível; exemplo:

```
\begin{enumerate}
\item item de primeiro nível
\begin{enumerate}
\item item de segundo nível
\begin{enumerate}
\item item de terceiro nível
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

produz:

1. item de primeiro nível
    (a) item de segundo nível
        i. item de terceiro nível

5.2.3 description:

O ambiente description produz uma lista diferente das outras, pois você pode nomear cada item; exemplo:

```
\begin{description}
\item[primeiro] O primeiro item
\item[segundo] O segundo item
\end{description}
```

produz:

primeiro O primeiro item

segundo O segundo item

você pode usar o comando `\hfill` e o pulo de linha para ter um efeito diferente:

```
\begin{description}
\item[primeiro] \hfill \\
O primeiro item
\item[segundo] \hfill \\
O segundo item
\end{description}
```

produz:

primeiro
: O primeiro item

segundo
: O segundo item

5.3 Teoremas:

Para escrever teoremas,lemas, corolários... você precisa do pacote: `\package{amsthm}`,
agora é só definir o ambiente no corpo do documento com o comando:
`\newtheorem{meu_ambiente}{definição}`,
usando o ambiente:

```
\begin{meu_ambiente}
Texto
\end{meu_ambiente}
```

produz:

definição 1. *Texto*

os teoremas definidos são numerados automaticamente e podem ser usados em qualquer parte do documento, para criar teoremas sem numeração usamos: `\newtheorem*{}`;

vamos ver alguns teoremas mais elaborados:

`\newtheorem{teo}{Teorema}[section]`, [**section**] faz com que a seção seja impressa junto com o número do teorema;

`\newtheorem{lema}[teo]{Lema}`,

`\newthorem{cor}[teo]{Corolário}`

`\newtheorem{prop}[teo]{Proposição}`

esses ambientes definidos com [**teo**] seguem a numeração do ambiente definido anteriormente **teo**, vamos ver o resultado:

```
\begin{teo}[Pitágoras]
Em todo triângulo retângulo o quadrado do comprimento da hipotenusa
é igual a soma dos quadrados dos comprimentos dos catetos.
\end{teo}
```

produz:

Teorema 5.1 (Pitágoras)**.** *Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.*

Para provas de teoremas existe o ambiente especial **proof**:

```
\begin{proof}[Prova de um teorema importante]
Prova do teorema
\end{proof}
```

produz:

Prova de um teorema importante. Prova do teorema □

você pode também modificar o estilo da impressão dos teoremas com comando: `\theoremstyle{estilo}`, onde **estilo** pode ser **plain**(padrão) usado para teoremas, lemas, preposições; **definition** usado para definições e exemplos; **remark** usado para notas; exemplo:

```
\theoremstyle{definition}
\newtheorem{defi}{Definição}
```

usando o ambiente criado:

```
\begin{defi}
Definimos A como...
\end{defi}
```

produz:

Definição 1. Definimos A como...

6 Inserindo figuras

Essa sessão mostra como são colocados elementos gráficos no LaTeX.

6.1 Formatos suportados

O LaTeXnão gerência as imagens diretamente, o único processamento que é feito é desenhar uma caixa aonde ela será inserida no documento. O pacote *graphicx* é quem irá fazer o trabalho gerênciá-las.

O único formato de imagem que pode ser usado *ao compilar com o* LaTeXé o **eps**. Se estiver gerando um documento com o pdfLaTeXpode ser usado os seguinte formatos de imagem : **jpg, png, imagens vetorizadas e eps**.

6.2 Incluindo uma imagem

Iremos usar o comando *includegraphcis* do pacote *graphicx* para inserir as imagens ao documento, o comand possui as seguintes opções :

width especifica a largura da imagem.

height especifica a altura da imagem.

keepaspectratio essa opção é setada por true ou false. Se for setado como true, não distorce a imagem ao aumentar ou diminuir o seu tamanho.

scale altera o tamanho da imagem em porcentagem, ou seja, 1 é o tamanho original, 2 é o dobro, 0.5 é a metade e assim por diante.

angle rotaciona o imagem em graus no sentido anti-horário.

Abaixo iremos mostar uma série de exemplos do uso do comando *includegraphics*.

O comando abaixo mostra a forma mais simples de colocar uma imagem:

```
\includegraphics{minha_foto.eps}
```

Do que jeito que foi usado o comando a imagem será inserida sem modificações. Vale lembrar que a busca pela imagem passada ao includegraphics é procurada a partir do ditório corrente do documento que é compilado pelo LaTeX.

Um problema que acontece é quando o tamanho original da imagem é muito grande. Isso faz com que o LaTeXprocure um lugar aonde ela possa ser inserida resultando em efeitos indesejávies. Uma forma de contorna esse problema é por seta os valores de altura e largura da imagem, como abaixo:

```
\includegraphics[width=xx, height=yy]{minha_foto.eps}
```

Os valores xx e yy podem ser especificados em centimetros ou milimetros.

Setar os valores de altura e largura pode distorcer a imagem e iss pode ser resolvido de duas formas. Uma delas é por usar a opção *scale* :

```
\includegraphics[scale=kk]{minha_foto.eps}
```

Dessa forma podemo setar 1 se quisermos o tamanho original. Analogamente podemos setar 0.5 para metade ou 0.25 para um quarto do tamanho original da imagem.

A segunda forma seria por usar a opção *keepaspectratio* :

```
\includegraphics[keepaspectratio=true, width=xx]{minha_foto.eps}
```

6.3 Ambiente figure

Usar o ambiente *figure* é interessante por uma séria de motivos que vão desde alinhar liguras a colocar descrição ou fazer referências a ela, abaixo temos um exemplo de como usá-lo :

```
\begin{figure}[htb]
  \includegraphics{minha_foto.eps}
\end{figure}
```

As opções do ambiente *figure* são :

h indica que a imagem dever ser colocado exatamente nesse local do cumento.

t indica que a imagem deve ser colocada na parte de cima da página.

b indica que a imagem deve ser colocada no fim da página.

Dentro do ambiente também podemos combinar uma série de comandos como abaixo:

```
\begin{figure}[h]
  \caption{Aqui vai uma descrição da imagem}
  \label{fig:ref_minha_foto}
  \centering
  \includegraphics{minha_foto.eps}
\end{figure}
```

Um aspecto importante é a ordem em que os comando são dados. Nesse caso será colocado a descrição primeiro e depois a imagem. Se o comando label tivesse ficado depois do includegraphics teriamos a descrição embaixo da figura. E por fim comando label cria uma referência para o local aonde a imagem foi inserida, caso queria fazer uma refência a ela basta fazer :

```
\ref{fig:ref_minha_foto}
```

6.4 Ambiente SCfigure

Além do ambiente figure temos o SCfigure que serve para colocar a descrição ao lado de uma imagem, que é feito da seguinte forma :

```
\begin{SCfigure}
  \centering
  \includegraphics{minha_foto.eps}
  \caption{descrição ao lado da foto.}
\end{SCfigure}
```

6.5 Ambiente wrapfigure

Outro efeito que podemos controlar via ambiente é o posicionamento de um texto junto a uma imagem. Para usá-lo devemos incluir o pacote *wrapfigure* no preâmbulo. O comando *wrapfigure* tem a seguinte sintaxe:

```
\begin{wrapfigure}{alinhamento}{comprimento}
```

O campo alinhamento é referente a posição do texto, que pode ser r para direita e l para esquerda. O campo comprimento é refente ao tamanho do quadro a ser montado pelo wrapfigure para por a imagem dentro.

No exemplo abaixo usamos *textwidth* que é um tamanho relativo ao texto para facilitar disposição dos elementos.

```
\begin{wrapfigure}{r}{0.5\textwidth}
  \begin{center}
    \includegraphics[width=0.48\textwidth]{imagem.eps}
  \end{center}
  \caption{Uma imagem}
\end{wrapfigure}
```

6.6 Usando subfigure com imagens

O ambiente *subfigure* pode ser usado tanto com imagens como tabelas. Mas nesse caso estaremos interessados em trabalhar com imagens. Usar subfigure é interessante pois podemos agrupar figuras e serem tratadas dentro de um único ambiente, e para usá-lo devemos incluir o pacote *subfigure*.

```
\begin{figure}
  \centering
  \subfloat[figura 1]{ \includegraphics{minha_figura1.eps} }
  \subfloat[figura 2]{ \includegraphics{minha_figura2.eps} }
  \subfloat[figura 3]{ \includegraphics{minha_figura3.eps} }
  \caption{Coleção de figuras}
  \label{fig:colecao}
\end{figure}
```

7 Referências

Iremos mostrar como fazer referências no LaTeXusando *labels, hiperlinks* e BibTeX.

7.1 Usando labels

Podemos usar as *labels*, colocadas em figuras, tabelas, texto e outros, juntamente com o comando *ref* para referenciá-las. Abaixo temos uma descrição desses comandos :

label deixa uma marca no texto para ser usada depois. Por exemplo, : `\label{nome_marca}` o nome nome_marca será usado por comandos como *ref*.

ref usa um nome definido por uma *label* e imprime o número do objeto, que pode ser várias coisas. Um exemplo do uso do comando *ref* seria : `\ref{nome_marca}` em que nome_marca é um nome de uma *label*.

E por fim temos o comando *pageref* que é equivalente ao comando *ref* só que imprime o número da página da *label*.

7.1.1 Organizando labels

É normal que num texto tenha bastante *labels* e como ela pode ser usada em diversos locais. A seguinte organização é sugerida para ficar mais fácil saber o que está sendo referenciado :

fig: figura.

sec: sessão.

tab: tabela.

eq: equação.

lst: lista.

Um exemplo dessa organização pode ser visto abaixo :
`Veja a figura \ref{fig:teste} na página \pageref{fig:teste}.`

7.2 Usando hiperlinks

Agora iremos mostrar como criar hiperlinks no LaTeXusando os comandos *hypertarget* e *hyperlink* do *pacote* ***hyperref*** . Segue abaixo a descrição e funcionamentos desses comandos :

hypertarget cria uma ponto de referência no texto da seguinte forma : `\hypertarget{nome_ref}{texto que será impresso}`

hyperlink cria um link para uma *hypertarget* da seguinte forma : `\hyperlink{nome_ref}{fazendo`

Ao usarmos o *hyperlink* ele destaca a palavra no texto, por colocar um cor diferente ou uma caixa nela[2].

Podemos mudar esse comportamento por setar opções no pacote, como abaixo :

colorlinks=true deixa a palavra colorida ao invés de criar uma quando em torno dela.

citecolor=green a palavra usada para fazer uma citação a um livro fica de cor verde.

Existem diversas opções que podem ser usadas, para maiores informações acesse o seguinte link http://en.wikibooks.org/wiki/LaTeX/Hyperlinks.

7.3 Gerando lista de tabelas e figuras

Em trabalhos acadêmicos é normal fazer uma lista de figuras e tabelas junto com o índice. Iremos mostrar como fazer isso com o LaTeXutilizando os *captions* colocados tanto em figuras quanto em tabelas.

Assim como o índice é feito pelo comando `\tableofcontents` a lista de figuras e tabelas são geradas pelos respectivos comandos : `\listoffigures` e `\listoftables` . Lembrando que estas listas são feitas por coletar os *captions* inseridos nas figuras ou tabelas.

7.4 Fazendo referências com o BibTeX

Uma maneira de fazer referências a livros, periódicos ou outras fontes de informação é por usar o BibTeX. Ele as organiza uma arquivo .bib de acordo com o tipo de documento.

Como exemplo iremos mostrar como colocar artigo e um livro num arquivo BibTeX. Ao criarmos uma referências temos campo obrigatórios e opcionais, os campos opcionais são indicados por um % na frente.

```
@article{nome_citacao_artigo,
  author    = ",
  title     = ",
  journal   = ",
  %volume   = ",
  %number   = ",
  %pages    = ",
  year      = "",
  %month    = ",
  %note     = ",
}
```

[2]Você pode ajustar essas opções no pacote ***hyperref***

```
@book{nome_citacao_livro,
  author    = ",
  title     = ",
  publisher = ",
  %volume   = ",
  %number   = ",
  %series   = ",
  %address  = ",
  %edition  = ",
  year      = "",
  %month    = ",
  %note     = ",
}
```

Depois de ter criado o arquivo .bib precisamos especificar no arquivo .tex aonde ele está e o tipo de formatação das referências. Isso é feito no fim do arquivo .tex com os seguintes comandos :

```
\bibliographystyle{plain}
\bibliography{arquivo_bib}
```

O comando *bibliographystyle* é quem define o tipo de formatação, existem diversos tipos além do plain. O comando *bibliography* diz aonde está o arquivo .bib, um observação importante é que não é necessário colocar o .bib no final do nome do arquivo. Feito isso estamos quase prontos parar gerar o documento com as citações.

O LaTeXsó vai inserir apenas as referências que você pedir pelos comandos `\cite{nome_citacao_artigo}` e `\nocite{nome_citacao_livro}`. Caso queria gerar o documento com todas as citações na sessão de referências basta fazer um `\nocite{*}`.

Com as citações feitas no arquivo .tex vamos compilá-lo para o LaTeXdescobrir que existe citações sem referências. Isso será anotado nos arquivo .log e .aux . Para podermos fazer com que o LaTeXsaiba aonde está as referências, iremos usar o comando *bibtex* no arquivo .aux para fazer os devidos ajustes. E por fim vamos compilar o arquivo .tex novamente para usar as informações no .aux para construir as referências de forma certa.

8 Modo Matemático 1

O modo matemático do LaTeXé muito útil para escrever formulas, equações, provas e teoremas matemáticos. Para entrar no ambiente matemático basta fazer `$ x^2 +2x -1$` ou `\[ x^2 +2x -1\]`.

Na verdade os símbolos `$ ... $` são a forma abreviada de `\begin{math} ... \end{math}` assim como `\[ ... \]` é a abreviação de `\begin{equation} ... \end{equation}`. Existe uma diferença sutil entre o *math* e *equation* que será explorado mais adiante.

8.1 Letras gregas e símbolos

Como dito anteriormente o modo matemático é usado para facilitar a vida de quem tem de escrever equações. Um exemplo disso pode ser visto abaixo com a tabela das letras gregas.

Tabela 3: Exemplo de letras gregas

Simbolo	Como escrever em LaTeX
α	`\alpha`
β	`\beta`
γ	`\gamma`
δ	`\dela`

Além disso ainda temos as seguinte formatações abaixo para letras :

Tabela 4: Exemplos de formatação de símbolos

Comando em LaTeX	Resultado esperado	Descrição
`\mathnormal{l}`	l	para notações matemáticas.
`\mathfrak{A}`	$\mathfrak{A}$	usado em notações algébricas.
`\mathcal{A}`	$\mathcal{A}$	usando em notações de conjunto.
`\mathbb{Z}`	$\mathbb{Z}$	usado para conjuntos especiais.

Ainda existe mais tipos de formatação como `a'` a' e `\vec{a}` $\vec{a}$.

Ainda com relação aos símbolos temos os relacionais, binários e outros. Existe uma grande quantidade de símbolos em LaTeXe é mais interessante termos uma tabela com deles e ir aprendendo de acordo com a necessidade. No link a seguir temos uma tabela bem completa dos símbolos matemáticos : http://amath.colorado.edu/documentation/LaTeX/Symbols.pdf

8.2 Potências e índices

Para escrever potências e índices o LaTeXse utiliza de uma ideia simples que pode ser expandida para outros contextos.

Ao escrever um índice podemos pensar que ele está subscrito e para isso iremos fazer _{escrever o índice}. Então teremos o seguinte resultado :

Exemplo	Código LaTeX
a_0	`$a_0$`
$\mathcal{C}_{l-1}$	`$\mathcal{C}_{l-1}$`

Com relação as potências podemos pensar que elas estão sobrescritas e usaremos ̂para escrever a potência entre chaves, da seguinte forma :

Exemplo	Código LaTeX
2^{n-1}	`$2^{n-1}$`
$n^5 + 4n^2 + 2$	`$n^5 + 4n^2 + 2$`

E por fim ainda podemos misturar numa mesma equação índices e potências da seguinte forma `k_{n+1} = n^2 + k^2_n - k_{n-1}` resultando no exemplo abaixo :

$$k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

8.3 Frações

A divisão pode ser escrita da forma tradicional usado símbolo que é o operador de divisão. Ou escrevê-la usando o comando *frac* que deve se informado o numero e denominador com a seguinte sintaxe `\frac{numero}{denominador}` . Um exemplo disso seria o seguinte `S_n = \frac{a_1(q^n -1)}{q-1}`

$$S_n = \frac{a_1(q^n - 1)}{q - 1}$$

E por fim ainda podemos colocar uma fração dentro da outro como no exemplo `\frac{\frac{1}{x}+\frac{1}{y}}{y-z}`

$$\frac{\frac{1}{x} + \frac{1}{y}}{y - z}$$

8.4 Somatórios e produtório

A escrita de somatórios e produtórios em LaTeXé bastante simples e basta fazer o seguinte para escrevê-los :

Tabela 5: Como escrever somatório e produtório

Expressão	Como escrever em LaTeX
$\sum_{i=0}^{n} a_i$	`$\sum_{i = 0}^{n} a_i$`
$\prod_{i=0}^{n} a_i$	`$\prod_{i = 0}^{n} a_i$`

Repare que usando $ para escrever uma equação ela ficou com uma aparência não muito boa. Isso pode ser resolvido por usar o comando *displaystlye* dentro do modo matemático. Como no exemplo abaixo :

Sem *displaystyle*	código em LaTeX
$\sum_{i=0}^{n} a_i$	`$\sum_{i = 0}^{n} a_i$`

Com *displaystyle*	código em LaTeX
$\displaystyle \sum_{i=0}^{n} a_i$	`$\displaystyle \sum_{i = 0}^{n} a_i$`

O comando *displaystyle* pode ser usado não só apenas em somatórios. Mas também em quanto expressão que for usada no modo matemático com *math* ou $.

9 Modo Matemático 2

9.1 Matrizes e Arrays

Para escrevermos matrizes iremos usar o ambiente *matrix* do modo matemático. Abaixo temos um exemplo simples de matriz :

```
\begin{matrix}
a & b & c \\
d & e & f \\
g & h & i
\end{matrix}
```

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

Assim como nas tabelas as colunas da matriz são separadas por &. E como não temos de especificar quantas colunas queremos devemos colocar um nova linha par dizer que a coluna acabou. E as linhas terminam somente com um *end* do ambiente *matrix*. Lembrando que é necessário adicionar o pacote *amsmath* para compilar corretamente o arquivo TeX com a matriz.

As matrizes normalmente são delimitadas por parênteses. Para isso iremos usar ambientes pré-definidos que já incluem um delimitador, alguns deles seriam :

Ambiente	Delimitador
pmatrix	( )
bmatrix	[ ]
vmatrix	||

Existem outros mas por uma questão didática iremos mostrar um exemplo usando o ambiente *pmatrix*. Abaixo segue o uso do ambiente *pmatrix* para construir uma matriz.

```
A_{m,n} =
\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\
a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\
\vdots  & \vdots  & \ddots & \vdots  \\
a_{m,1} & a_{m,2} & \cdots & a_{m,n}
\end{pmatrix}
```

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

9.2 Texto nas equações

Às vezes é necessário colocar um pouco de texto juntamente com equações. É importante observar que modo matemático o texto será tratado como se fosse um monte de letras ou símbolos, o que não é o efeito desejado. Veja o que acontece quando fazemos `$x^2 < 2^x somente se x \geq 5$` no modo matemático :

$$x^2 < 2^x somentesex \geq 5$$

Se queremos que o LaTeXtrate como texto as palavras dentro do matemático devemos usar o comando *text*. Caso queira adicionar formatação ao texto temos os seguintes comandos :

text apenas texto sem formatação.

textit texto em *itálico*.

textbf texto em **negrito**.

textrm texto na fonte Times New Roman.

Reescrevendo o exemplo apresentado acima `x^2 < 2^x \textrm{ somente se } x \geq 5` com teremos o seguinte resultado :

$$x^2 < 2^x \text{ somente se } x \geq 5$$

Observe que teve ser colocado espaços antes e depois do texto entre as expressões. Isso acontece pois os espaços são desconsiderados dentro do modo matemático.

9.3 Enumeração de Equações

Ao usar o ambiente *equation* o LaTeXenumera automaticamente as suas equações. Além disso podemos colocar uma label dentro desse ambiente podendo criar uma lista de equações ou fazer referência cruzada. Abaixo temos um exemplo simples do uso do *equation* :

```
\begin{equation}
f(x) = (x+a)(x+b)
\end{equation}
```

$$f(x) = (x+a)(x+b) \tag{1}$$

Assim como fazemos em outros ambientes podemos colocar uma label dentro dele. Normalmente usamos o comando *ref* para fazer referência uma label. Com relação ao ambiente *equation* ele também pode ser usado mas é mais recomendado usar o *eqref* nesse caso. Abaixo temos um exemplo de como fazer uma referência a uma equação.

```
\begin{equation} \label{eq:eq1}
5^2 -5 = 20
\end{equation}
Referencia a equação \eqref{eq:eq1}
```

$$5^2 - 5 = 20 \tag{2}$$

Referencia a equação (2)

A enumeração das equações é relativa a seção ou a subseção. Esse comportamento é setado no preambulo do documento da seguinte forma :

Numeração relativo a seção `\numberwithin{equation}{section}`

Numeração relativo a subseção `\numberwithin{equation}{subsection}`

9.4 Comando align

Quando queremos colocar mais de uma equação ou ela ocupa mais de uma linha o comando *align* nos permite fazer a formatação necessária para isso. Um exemplo seria colocar estas equações $a_n = a_1 + (n-1)r$ e $a_n = a_{n-1} + r$ dentro de um ambiente *subequations*, para isso basta fazer o seguinte :

```
\begin{subequations}
\begin{align}!
a_n = a_1 +(n-1)r \\
a_n = a_{n-1} + r
\end{align}
\end{subequations}
```

$$a_n = a_1 + (n-1)r \tag{3a}$$

$$a_n = a_{n-1} + r \tag{3b}$$

Agora se temos uma equação ou expressão que é muito grande para caber numa linha devemos fazer o seguinte :

```
\begin{align}
f(x) &= x^4 + 7x^3 + 2x^2 \nonumber \\
&\qquad {} + 10x + 12 \end{align}
```

$$f(x) = x^4 + 7x^3 + 2x^2 + 10x + 12 \tag{4}$$

É importante observar que a escrita tem a mesma dinâmica do que escrever uma matriz. Ou seja o & separa cada coluna. Nesse caso temos f(x) na primeira coluna e na segunda temos a equação. Como usual o \\ indica a quebra de linha. O comando *nonumber* é utilizado pois a enumeração no

ambiente *align* é diferente do *equation*. E por isso precisamos dizer para o *align* não enumerar a equação na linha. Nesse caso apenas a última linha não possui o *nonumber* porque quero enumerar apenas a última linha da equação.

9.5 Ambientes cases

Para escrevermos funções que são escritas usando { iremos usar o ambiente *dcases*. Assim como outro ambientes já mostrados o ambiente *cases* é escrito no mesmo estilo de uma tabela. Abaixo temos um exemplo de uso do *cases* :

```
\[
u(x) =
\begin{cases}
\exp{x} & \text{if } x \geq 0 \\
1       & \text{if } x < 0
\end{cases}
\]
```

$$u(x) = \begin{cases} \exp x & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

10 Beamer 1

10.1 Estrutura básica

O LaTeXtem uma classe destinada a fazer apresentação de slides que é o *beamer*[3]. A estrutura mais básica de uma apresentação em *beamer* é a seguinte :

```
\documentclass{beamer}
\begin{document}
    \begin{frame}
    \frametitle{Título do slide vai aqui}
    Texto do slide vai aqui.
    \end{frame}
\end{document}
```

Assim como qualquer tipo de documento em LaTeXtemos de especificar o tipo de documento, que nesse caso é o *beamer*. Dentro do início e fim do documento é que temos os frames. Nos frames é aonde iremos colocar o texto para aparecer no slide. Isto significa que cada ambiente frame corresponde a um slide. E por fim temos o comando *frametitle* que como o nome já diz coloca o título para aquele slide em específico. Isso nos possibilita criarmos uma seção e dentro dela termos slides com diferentes títulos, isso será explorado mais a frente.

É importante lembrar que dentro de um frame são válidos ambientes como itemize, enumerate, minipage, equation, figure e outros. A criação do

[3] É importante dizer que o *beamer* não vem no pacote mais básico do LaTeX

título da apresentação continua sendo colocada no preambulo do documento só que nessa caso ela pode ser mais elaborada, como no exemplo abaixo :

```
\title{O nome do título}
\subtitle{Aqui vai o subtítulo}
\author{Eu \inst{1} \and Você \inst{2}}
\institute{
     \inst{1}
     Instituição A
     \and
     \inst{2}
     Instituição B
}
\date
\subject{Curso de LaTeX}
```

10.2 Criando título

E para criar o título podemos fazer de duas maneiras. Uma usando o *maketitle* e a outra por `\frame{\titlepage}`, lembrando que ambas devem estar dentro de um frame. É recomendado que seja utilizado apenas um frase em separado para o título da apresentação.

Para criarmos o índice da apresentação é bem simples, basta usar o já conhecido comando *tableofcontents*. Nesse momento é importante dizer que ao criar as seções você deve fazê-las fora dos frames, a fim de deixar o seu documento melhor estruturado. O exemplo abaixo ilustra esse fato :

```
\section{Assunto A}
\begin{frame}
     \frametitle{título do frame}
      texto ...
\end{frame}
\section{Assunto B}
\begin{frame}
     \frametitle{título do frame}
      texto ...
\end{frame}
```

10.3 Temas para o beamer

No *beamer* temos a possibilidade de alterar o layout do documento de forma bastante simples. No preambulo podemos setar o tema que será usado na apresentação. Existem diversos deles e eles podem ser encontrados em : http://www.pletscher.org/writings/latex/beamerthemes.php No exemplo abaixo mostrar como setar o tema *warsaw* :

```
\usetheme{Warsaw}
```

10.4 Blocos

O *beamer* tem caixas de texto que normalmente são usadas para destacá-lo. Temos três tipos delas são :

block caixa de texto normal.

alertblock caixa de texto destinada a mensagens de aviso ou para chamar atenção sobre alguns aspecto.

exampleblock como o próprio nome já diz essa caixa de texto serve para colocar exemplos.

Abaixo temos um exemplo de como as caixas de texto são usadas :

```
\begin{frame}
     \begin{block}{Essa é uma caixa}
         Aqui vai o texto destina destinada a ela.
     \end{block}
     \begin{block}{Essa é uma caixa de alerta}
         Aqui vai o texto destina destinada a ela.
     \end{block}
     \begin{block}{Essa é uma caixa de exemplo}
         Aqui vai o texto destina destinada a ela.
     \end{block}
\end{frame}
```

11 Beamer 2

11.1 Ambiente columns

Vimos os ambientes que criam caixas de texto como *block* organizando o texto horizontalmente. Além disso temos o ambiente *columns* que estrutura o texto verticalmente. Ou seja cria colunas numa apresentação colocando o texto lado a lado. O código abaixo faz esse tipo de divisão do texto :

```
\begin{columns}[c]
\column{.5\textwidth}
Conteúdo da primeira coluna. \\

     \column{.5\textwidth}
Segundo coluna \\
Mais uma linha

\end{columns}
```

É importante observar que a cada *column* cria uma coluna. Com isso não é necessário dizer quantas colunas queremos, basta apenas um *column* para criar uma coluna. Além disso dentro de um comando *column* nada impede termos outro ambiente *columns*. Isso significa que podemos criar uma coluna dentro de uma coluna. Outro aspecto do comando *column* é a largura da coluna que é especificada pelas medidas de comprimento do LaTeX, como centímetros, `\textwidth` e outros.

A disposição dos elementos dentro do ambiente *columns* é definida pela letra entre colchetes logo após o nome do ambiente. Existe basicamente duas formas de alinhar o texto sendo ele centralizado `\begin{columns}[c]` , ou com o topo da coluna `\begin{columns}[t]`.

11.2 Comando pause

Em algumas apresentações vemos o efeito de fazer aparecer item a item de forma incremental. Isso pode ser feito no beamer por usado o comando *pause*. Essa animação pode ser feita pelo código abaixo :

```
\begin{frame}
\frametitle{Tíutlo do frame}
Começa aqui a animação
\pause
item um
\pause
item dois
\end{frame}
```

O resultado disso é uma apresentação que no primeiro slide temos apenas a primeira frase. E uma nova página para cada *pause*. Numa apresentação isso faz com que as demais frases apareçam uma por vez.

Podemos produzir esse mesmo efeito usando o ambiente *itemize* adicionando parâmetros aos comandos *item*. Os argumentos são inseridos da seguinte forma :

item$< n- >$ aparece a partir da n-ésima vez em diante.

item$< n-m >$ fica visível a partir da n-ésima até a m-ésima.

item$< n >$ aparece apenas na n-ésima vez.

Abaixo temos um exemplo de como usar o *itemize* para criar itens que aparecem apenas em determinados momentos :

```
\begin{itemize}
\item sempre será mostrado
\item<2-> Aparece a partir da segunda
```

```
\item<2-4> É mostrado da segunda até a quarta
\item<4> Mostrado apenas na quarta vez
\end{itemize}
```

11.3 Customizando o tema

O beamer possui uma série de temas padrões e a partir deles podemos mudar alguns aspectos. Dentre eles iremos trabalhar com as cores que compõe um tema. Parar isso iremos usar dois tipos de comandos sendo um para uma solução pronta e outra que você especifica os atributos. Para solução pronta iremos usar o comando *usecolortheme* descrito abaixo :

usecolortheme altera as cores do tema a partir de uma lista pré-definida.

O link a seguir tem uma matriz que mostras as combinações das cores que podem ser feitas usando o comando *usecolortheme* e o temas do LaTeX: http://www.hartwork.org/beamer-theme-matrix/

Caso seja necessário uma maior customização do tema, ela é feita usando os comandos abaixo :

useinnertheme especifica o layout interno da apresentação. Ele tem as seguintes opções :

- rectangles
- circles
- inmargin
- rounded

useoutertheme especifica as cores as cores do cabeçalho e dos roda pé de cada slide. Para ele podemos setar os seguintes valores :

- infolines
- miniframes
- shadow
- sidebar
- smoothbars
- smoothtree
- split
- tree

Para maiores informações sobre o uso do *beamer* para fazer apresentações sugiro olhar o seguinte link : www.math.utah.edu/~smith/AmberSmith_GSAC_Beamer.pdf