

LATEX

Noções Intermediárias



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Caxias do Sul

ISBN 978-65-86734-01-0

Luís Henrique Ribeiro da Silva
Greice da Silva Lorenzetti Andreis

L^AT_EX

Noções Intermediárias

Autores

Luís Henrique Ribeiro da Silva
Greice da Silva Lorenzetti Andreis

Edição

Luís Henrique Ribeiro da Silva
Greice da Silva Lorenzetti Andreis

Edição Visual

Luís Henrique Ribeiro da Silva



Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul,
Campus Caxias do Sul

51 S586l	Silva, Luís Henrique Ribeiro da LaTeX : noções intermediárias / Luís Henrique Ribeiro da Silva, Greice da Silva Lorenzetti Andreis. -- Caxias do Sul, RS, 2019. 113 p.: il. (colors.) Inclui referências ISBN 978-65-86734-01-0 1. Licenciatura em matemática. 2. LaTeX - noções intermediárias. 3. Tipografia matemática. I. Andreis, Greice da Silva Lorenzetti. II. Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul. III. Título.
-------------	--

CDU 51

Ficha catalográfica elaborado pela bibliotecária Jaçanã Egges Pando - CRB 10/1936

PREFÁCIO

Meu primeiro contato com o LaTeX foi com a escrita de minha dissertação de mestrado, em 2006. O aprendizado desta Linguagem foi realizado de forma totalmente autônoma, buscando tutoriais disponíveis na internet. Já nos primeiros textos me encantei com as maravilhas proporcionadas pelo LaTeX na área da Matemática. Facilidade na escrita de fórmulas longas e complexas, formatação padronizada e elegância nos documentos. Hoje como docente, continuo utilizando o LaTeX e incentivando colegas e estudantes a adotar esta Linguagem.

Em 2017 o professor Érick Scopel submeteu o projeto de ensino “LaTeX: Noções básicas”, vindo a trabalhar com dois estudantes da Licenciatura em Matemática, Munique dos Santos Lima (bolsista) e Luís Henrique Ribeiro da Silva (voluntário), e com colaboração do professor Lucas Pinto Dutra. O projeto foi um sucesso.

Como o professor Érick se ausentou em 2018 para cursar seu doutorado, o Luís Henrique me procurou para dar continuidade ao projeto, com cooperação do professor Lucas Pinto Dutra, o qual saiu para doutorado logo na sequência. Aceitei com muito entusiasmo. Submetemos então o projeto de ensino “LaTeX: Noções intermediárias”, que tinha como objetivo oportunizar ao estudante do curso de Licenciatura em Matemática o aprimoramento na linguagem LaTeX, bem como incentivar servidores e demais estudantes do *Campus* a adotarem o LaTeX como uma possível solução para problemas de formatação e padronização de documentos. O Luís Henrique atuou como bolsista e na sequência como voluntário, criando o texto base deste volume. Ao longo de 2018 foram ofertados minicursos em dois níveis: noções básicas e noções intermediárias, os quais foram executados pelo Luís Henrique e pela Munique.

Devido à boa adesão por parte dos estudantes do *Campus* Caxias do Sul aos minicursos, em 2019 submeti o projeto de ensino “Disseminando a linguagem LaTeX”, tendo o Luís Henrique e a Munique como voluntários, que trabalharam na edição final desta coleção e na aplicação de minicursos no *Campus* Caxias do Sul e no *Campus* Farroupilha.

A proposta de disseminar a Linguagem LaTeX vem se concretizando ao longo destes três anos. Muitos alunos já o utilizam para produzir planos de aula, listas de exercícios, apresentações e trabalhos de conclusão de curso.

Este livro apresenta de forma original os comandos do LaTeX por meio de exemplos e de uma escrita didática sobre o tema.

Convido você a se aventurar pelas diversas possibilidades que o LaTeX oferece. Com certeza esta obra o ajudará nesta aventura. Boa leitura e/ou consulta!

Caxias do Sul, 20 de julho de 2019.
Greice da Silva Lorenzetti Andreis.

Sumário

1	A classe BEAMER	9
1.1	O primeiro documento	9
1.2	O ambiente <code>frame</code>	10
1.3	Adicionando título e subtítulo	12
1.4	Alterando as cores de fundo e do texto	13
1.5	Alterando as dimensões da apresentação	14
1.6	Temas pré-definidos	14
1.7	Blocos	15
1.7.1	Tipos de Blocos	16
1.8	Alinhamento do texto	18
1.9	O ambiente <code>columns</code>	18
1.9.1	Alinhamento vertical no ambiente <code>columns</code>	21
1.9.2	Alinhamento horizontal no ambiente <code>columns</code>	21
1.10	Listas	22
1.10.1	Lista com marcadores - <code>itemize</code>	22
1.10.2	Lista enumerada - <code>enumerate</code>	23
1.10.3	Lista descritiva - <code>description</code>	24
1.11	Criando sobreposições	25
1.11.1	Os comandos <code>uncover</code> e <code>visible</code>	25
1.11.2	O comando <code>only</code>	26
1.11.3	O comando <code>item</code>	27
1.12	Inserindo Imagens	29
1.13	Dividindo a apresentação em seções e subseções	29
1.14	Criando o sumário	32
1.14.1	Sumário no início de uma seção	33
1.15	O parâmetro <code>fragile</code>	33
1.16	Nota de rodapé	34
1.17	Botões e <i>hyperlinks</i>	34
1.18	Inserindo áudio e vídeo	36
1.18.1	O comando <code>sound</code>	36

1.18.2	O comando <code>movie</code>	36
2	Desenhando com o <i>TikZ</i>	37
2.1	Considerações Iniciais	37
2.2	Estabelecendo coordenadas	39
2.3	O comando <code>draw</code>	39
2.3.1	Espessura da linha	40
2.3.2	Cor da linha	40
2.3.3	Tipo da linha	42
2.3.4	Formas geométricas pré-definidas	42
2.3.5	Extremidades da linha	44
2.3.6	O comando <code>grid</code>	44
2.4	O comando <code>fill</code>	45
2.5	O comando <code>path</code>	46
2.6	O comando <code>coordinate</code>	46
2.7	O comando <code>node</code>	48
2.8	O parâmetro <code>scale</code>	49
2.9	O comando <code>tikzset</code>	49
2.10	O comando <code>clip</code>	50
2.11	O ambiente <code>scope</code>	51
3	Representação gráfica de figuras bidimensionais e tridimensionais	52
3.1	O comando <code>plot</code>	52
3.2	O pacote <code>pgfplots</code>	54
3.2.1	O ambiente <code>axis</code>	55
3.2.2	Gráficos bidimensionais com o comando <code>addplot</code>	56
3.2.3	Funções definidas por partes	58
3.2.4	Superfícies com o comando <code>addplot3</code>	59
4	Mais representações	64
4.1	Polígonos Regulares	64
4.1.1	Pentágono regular	64
4.1.2	Hexágono regular	65
4.1.3	Polígono regular de n lados	65
4.1.4	Ângulos e vértices	67
4.1.5	A biblioteca <code>angles</code>	68
4.2	Retas paralelas interceptadas por uma transversal	71
4.2.1	A biblioteca <code>intersections</code>	71
4.3	Relação entre conjuntos	75
4.4	Diagramas de Venn	76

4.5	Ciclo Trigonométrico	78
4.6	Intervalos na Reta	80
5	Pôster com a classe tikzposter	82
5.1	Os parâmetros opcionais da classe <code>tikzposter</code>	83
5.2	Inserção de dados de identificação	85
5.3	O comando <code>block</code>	85
5.4	Separando o texto em colunas	86
5.5	Temas	87
6	Modelo de pôster	88
7	Introdução aos Macros	91
7.1	O comando <code>newcommand</code>	91
7.2	O comando <code>newenvironment</code>	93
8	O pacote <code>tcolorbox</code>	94
8.1	Introdução	94
8.2	Personalizando as caixas	95
8.3	Espaçamento	97
8.4	O comando <code>newtcolorbox</code>	99
8.5	O ambiente <code>tcblisting</code>	100
8.6	Criando caixas enumeradas	101
9	Bibliografia com o pacote <code>abntex2cite</code>	102
9.1	O ambiente <code>thebibliography</code>	102
9.2	O pacote <code>abntex2cite</code>	103
9.3	O arquivo de referências - <code>.bib</code>	103
9.4	Criando a primeira lista de referências	105
9.5	O comando <code>citeoption</code>	110
9.6	Criando as citações	111
	Referências	113

Capítulo 1

A classe BEAMER

A finalidade deste capítulo é apresentar uma visão geral da classe BEAMER, partindo de sua estrutura básica. Nesse sentido, é necessário que o leitor tenha um conhecimento mínimo da linguagem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, visto que muitos comandos e ambientes utilizados aqui são dessa linguagem.

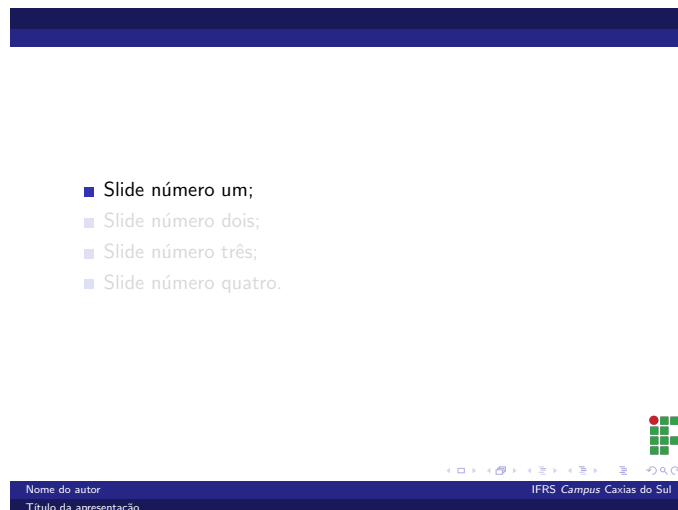
1.1 O primeiro documento

Uma apresentação em BEAMER consiste em, basicamente, um conjunto de quadros (*frames*, em inglês) e cada *frame* é composto por uma série de *slides*.

O termo *slide* (do inglês, que significa deslizar) será utilizado para fazer referência ao conteúdo de uma única página gerada no arquivo PDF. No entanto, no senso comum, esse termo é utilizado para se referir a cada quadro da apresentação. Por exemplo: “meu trabalho tem 21 slides”. Portanto, aqui a nomenclatura correta seria: “meu trabalho tem 21 quadros”. O exemplo a seguir elucida esse fato.

Exemplo 1 A Figura 1.1 mostra um quadro contendo quatro slides.

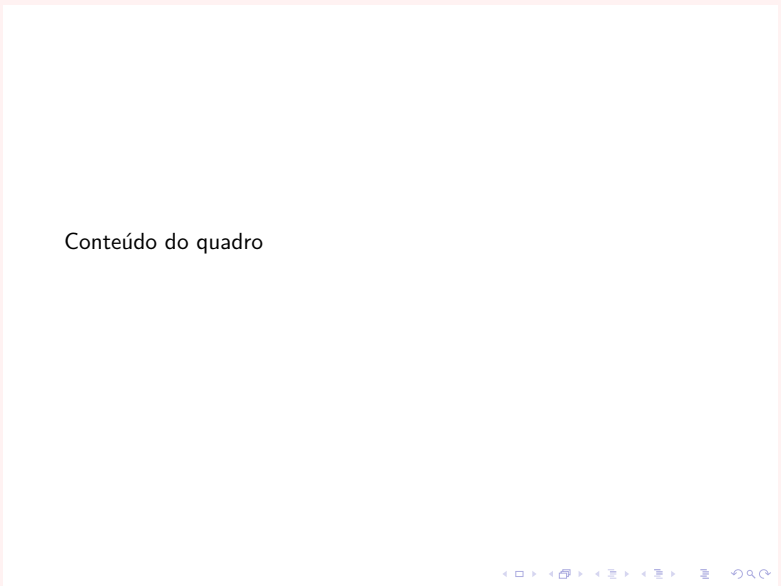
Figura 1.1: Exemplo de quadro contendo 4 slides.



Exemplo 2 Ao compilar os comandos básicos no BEAMER, obtém-se o resultado a seguir.

```
\documentclass{beamer}
\usepackage[utf8]{inputenc}

\begin{document}
  \begin{frame}
    Conteúdo do quadro
  \end{frame}
\end{document}
%%\end{corpo}
```



Conteúdo do quadro

1.2 O ambiente frame

Como foi visto no Exemplo 2, o documento criado foi uma página retangular contendo o texto “Conteúdo do quadro”. Portanto, tudo o que é colocado dentro de um ambiente `frame` é tratado como conteúdo de um quadro.

Exemplo 3 Código-fonte (sem preâmbulo) que gera um documento com dois quadros.

```
\begin{frame}
  Conteúdo do Quadro 1
\end{frame}
\begin{frame}
  Conteúdo do Quadro 2
\end{frame}
```

Para criar o primeiro quadro, contendo informações como título, subtítulo, autor e data, é necessário declarar algumas informações no preâmbulo.

Exemplo 4 *Código-fonte que identifica as informações da apresentação. Estes comandos são inseridos no preâmbulo do documento.*

```
\title{Título da apresentação}
\subtitle{Subtítulo da apresentação}
\author{Nome do autor}
\institute{IFRS \textit{Campus} Caxias do Sul}
\date{2018}
```

Para que essas informações tenham algum efeito no documento criado, é necessário utilizar o comando `titlepage` em um quadro.

Exemplo 5 *Código-fonte (sem preâmbulo) que gera o quadro contendo as informações da apresentação. As informações que foram declaradas no preâmbulo se encontram no Exemplo 4.*

```
\begin{frame}
  \titlepage
\end{frame}
```

Título da apresentação
Subtítulo da apresentação

Nome do autor

IFRS *Campus* Caxias do Sul

2018

O BEAMER permite inserir um logotipo em todos os quadros da apresentação com apenas um comando. Este logotipo pode ser um texto ou uma imagem. No caso de ser uma imagem, utiliza-se o comando `includegraphics` dentro do comando `logo`.

```
\logo{\includegraphics[scale=<escala>]{<nome da imagem>}}
```

Esse comando irá adicionar o logotipo em todos os quadros da apresentação. A Seção 1.12 traz mais detalhes sobre a sintaxe do comando `\includegraphics`.

Observação 1 *O posicionamento do logotipo na página é definido pelo tema utilizado. Caso não seja utilizado nenhum tema, a posição será no canto inferior direito. Os temas se encontram na Seção 1.6.*

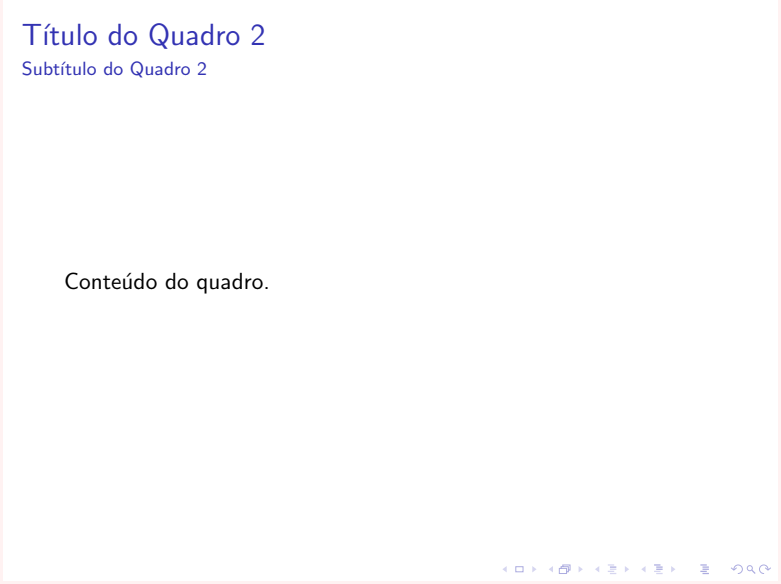
1.3 Adicionando título e subtítulo

Para criar um quadro com título e subtítulo, utilizam-se os comandos `frametitle` e `framesubtitle`, respectivamente. O conteúdo do quadro é digitado logo abaixo dos comandos de título e subtítulo.

Exemplo 6 *Quadro contendo título e subtítulo.*

```
\begin{frame}
  \frametitle{Título do Quadro 2}
  \framesubtitle{Subtítulo do Quadro 2}

  Conteúdo do quadro.
\end{frame}
```



Título do Quadro 2
Subtítulo do Quadro 2

Conteúdo do quadro.

Visualização de um slide de apresentação. O slide tem um fundo branco e está centralizado em um ambiente de apresentação com fundo rosa claro. No topo, há o título "Título do Quadro 2" em azul escuro e o subtítulo "Subtítulo do Quadro 2" em azul mais claro. Abaixo, o texto "Conteúdo do quadro." está centralizado. Na base do slide, há uma barra de navegação com ícones para voltar, avançar, etc.

Outra possível forma de identificar o título em um quadro é escrevê-lo em um parâmetro obrigatório ao lado do comando que inicia o quadro.

Exemplo 7 *Forma alternativa de identificar o título de um quadro.*

```
\begin{frame}{Título do Quadro 2}
```

1.4 Alterando as cores de fundo e do texto

É possível alterar as cores do plano de fundo e do texto em um documento. Para alterar a cor de fundo de todos os quadros do BEAMER, utiliza-se seguinte comando no preâmbulo.

```
\setbeamercolor{normal text}{bg=cor desejada!porcentagem da cor}
```

O parâmetro `normal text` se refere ao texto normal do documento e `bg` é uma abreviação de *background* (do inglês, plano de fundo). Ainda, é possível utilizar uma porcentagem da cor, ou seja, uma cor mais “clara”.

```
\setbeamercolor{normal text}{fg=cor desejada!porcentagem da cor}
```

Note que a única diferença em relação ao comando anterior é que aqui utiliza-se `fg`, uma abreviação de *foreground*, que significa primeiro plano. Em outras palavras, é todo o conteúdo que está na camada acima do *background*, com exceção dos títulos. Para trocar a cor dos títulos, utiliza-se um comando semelhante.

```
\setbeamercolor{titlelike}{fg=cor desejada!porcentagem da cor}
```

Exemplo 8 *Alterando a cor do plano de fundo para 10% amarelo e do texto para 70% vermelho.*

```
\setbeamercolor{normal text}{bg=yellow!10, fg=red!70}
```

Título do Quadro 2

Subtítulo do Quadro 2

Conteúdo do quadro.

1.5 Alterando as dimensões da apresentação

Todas as telas, seja de um computador, de uma televisão ou de um *smarthphone*, possuem uma característica chamada *aspect ratio* (proporção da tela, em português). Embora existam tamanhos de telas diferentes, a proporção da tela é semelhante em muitos desses dispositivos. Televisores e monitores antigos, em sua maioria, possuíam a proporção da tela 4 : 3. A medida da esquerda (4) se refere à proporção horizontal da tela, e a medida da direita (3) à proporção vertical. Hoje em dia, a maioria das televisões novas possuem proporção da tela 16 : 9, ou seja, 16 unidades na horizontal e 9 unidades na vertical.

Por padrão, o BEAMER trabalha com a proporção 4 : 3, mas é possível alterar essa propriedade no preâmbulo com o comando

```
\documentclass[aspectratio=<proporção desejada>]{beamer}
```

Exemplo 9 *Utilizando a proporção da tela 16 : 9.*

```
\documentclass[aspectratio=169]{beamer}
```

Observação 2 *A proporção 16 : 9 foi escrita como 169, removendo os dois pontos. Outras proporções disponíveis são 1610 (16 : 10), 149 (14 : 9), 54 (5 : 4) e 32 (3 : 2).*

1.6 Temas pré-definidos

Os temas tornam fácil a modificação da aparência de uma apresentação. O BEAMER traz incorporado um conjunto de 26 temas. São eles:

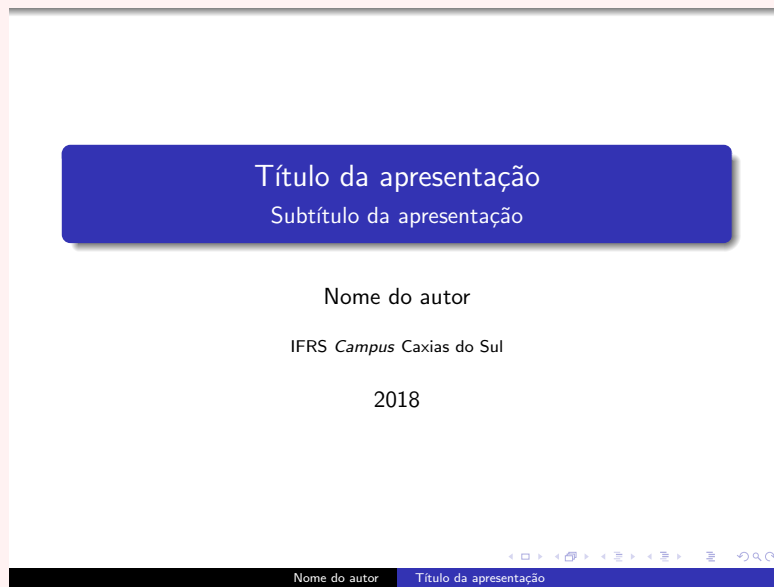
```
AnnArbor, Antibes, Bergen, Berkeley, Berlin, Boadilla, CambridgeUS, Copenhagen,  
Darmstadt, Dresden, Frankfurt, Goettingen, Hannover, Ilmenau, JuanLesPins,  
Luebeck, Madrid, Malmoe, Marburg, Montpellier, PaloAlto, Pittsburgh, Rochester,  
Singapore, Szeged, Warsaw
```

Para aplicar um tema, utiliza-se o comando `usetheme` no preâmbulo.

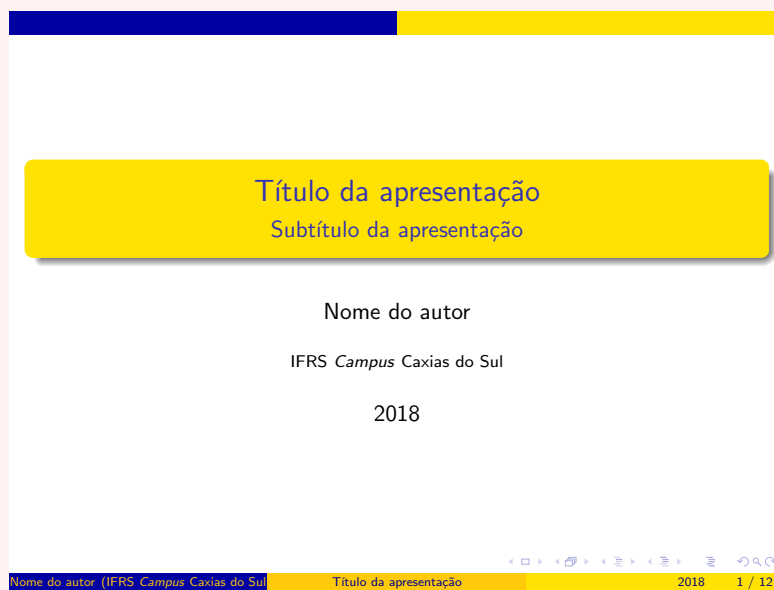
```
\usetheme{<tema escolhido>}
```

Exemplo 10 *Aplicações dos temas “Warsaw” e “AnnArbor”. As linhas de comando que foram apresentadas nos Exemplos 4 e 5 foram omitidas.*

`\usetheme{Warsaw}`



`\usetheme{AnnArbor}`



Observação 3 *A maioria dos temas leva nomes de cidades. Por esse motivo, é necessário escrevê-los com as iniciais maiúsculas.*

1.7 Blocos

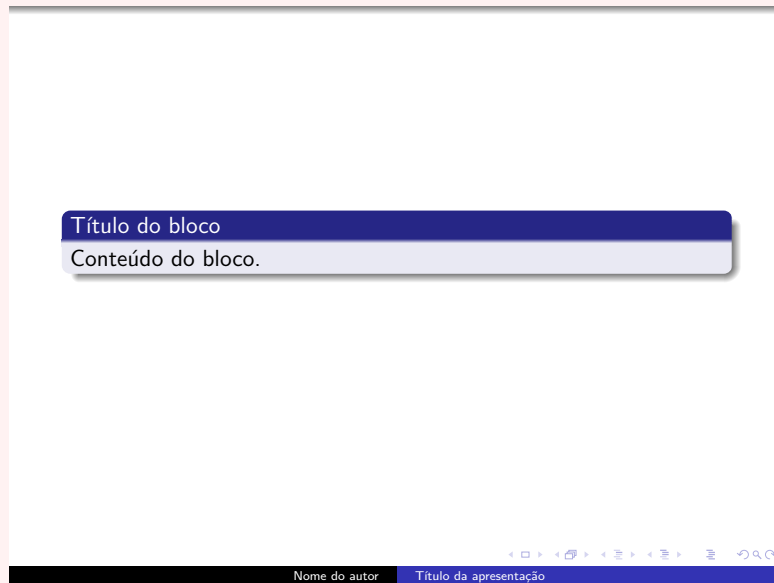
O BEAMER permite adicionar caixas predefinidas chamadas *blocks* (blocos, em português). Para inserir um bloco em um documento BEAMER, utiliza-se o ambiente `block`.

Exemplo 11 *Bloco inserido dentro de um quadro. O tema utilizado é o Warsaw.*

```

\begin{block}{Título do bloco}
  Conteúdo do bloco.
\end{block}

```



1.7.1 Tipos de Blocos

Existem modos de personalizar os blocos. O primeiro modo de fazer isso diz respeito à composição e ao formato do bloco. É possível criar um bloco com título, sem título ou com uma barra superior fina.

Exemplo 12 *Diferentes tipos de blocos.*

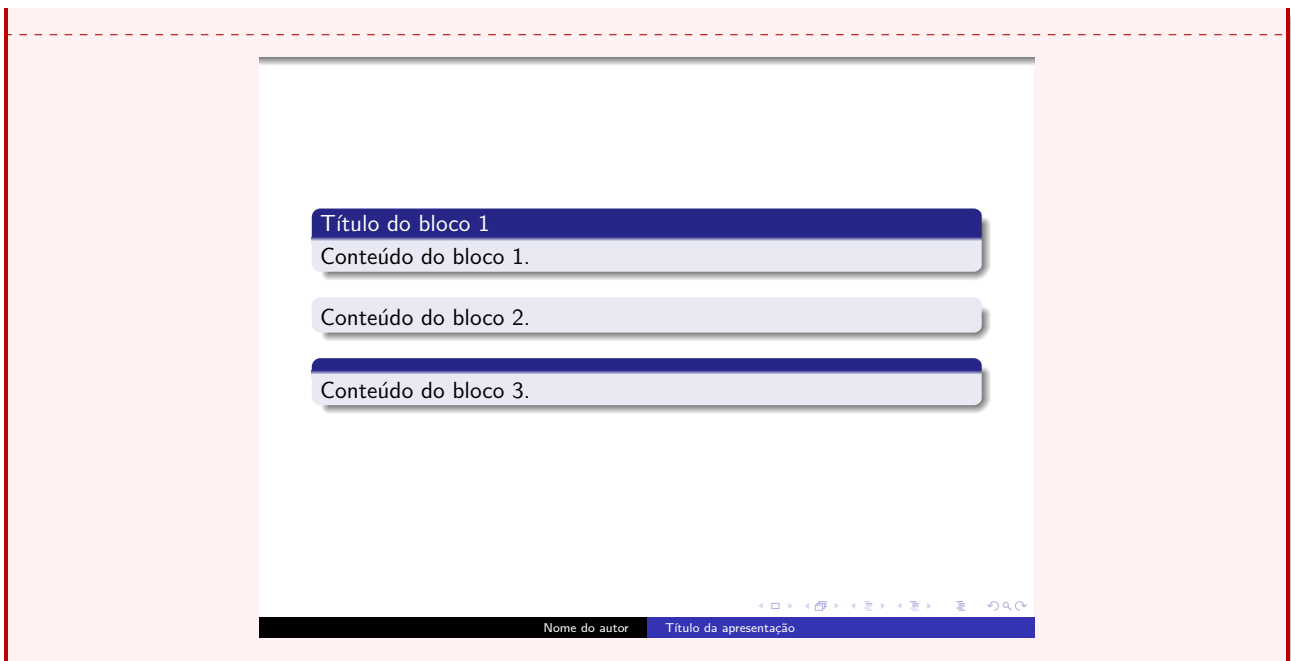
```

\begin{block}{Título do bloco 1}
  Conteúdo do bloco 1.
\end{block}

\begin{block}{}
  Conteúdo do bloco 2.
\end{block}

\begin{block}{\ }
  Conteúdo do bloco 3.
\end{block}

```

Observação 4 Quando um bloco não recebe título, o \LaTeX joga a primeira letra do conteúdo do bloco como sendo o título deste bloco. Por isso, é necessário utilizar uma das três formas apresentadas para incluir um bloco.

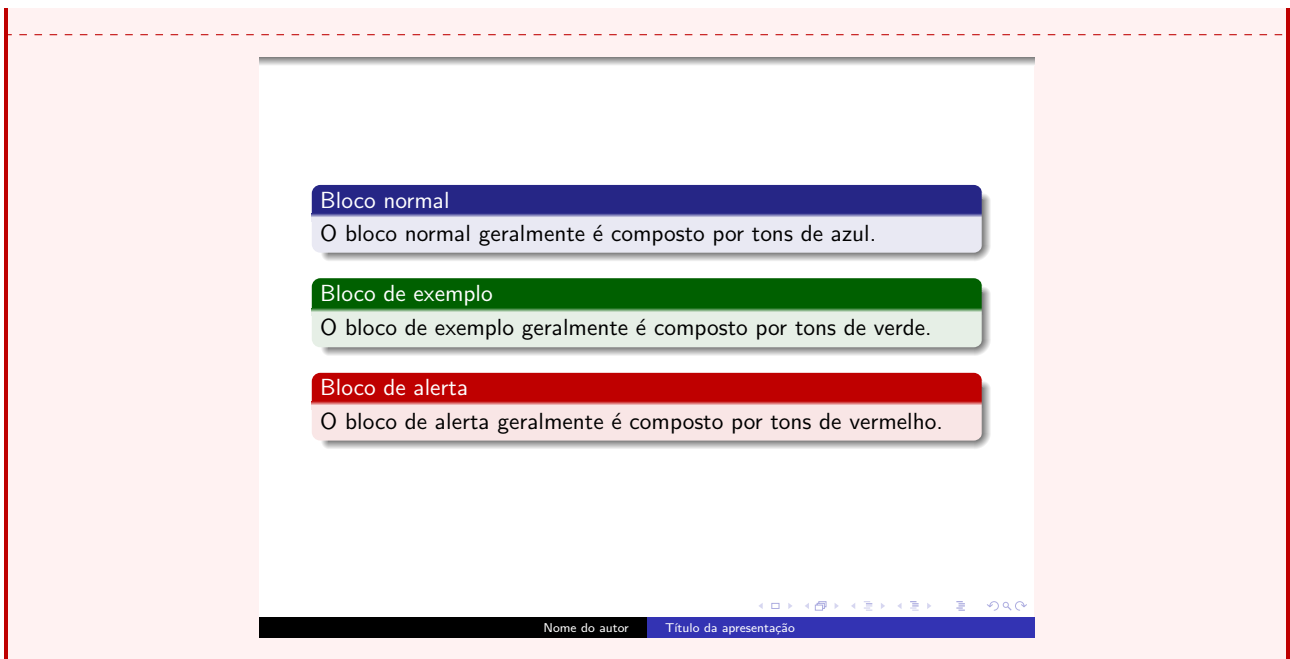
É possível trocar as cores dos blocos. Por padrão, o \LaTeX tem, de forma pré-definida, três tipos de blocos com cores diferentes. Essa funcionalidade pode não estar disponível em todos os temas.

Exemplo 13 *Diferentes tipos de blocos.*

```

\begin{block}{Bloco normal}
  O bloco normal geralmente é composto por tons de azul.
\end{block}
\begin{exampleblock}{Bloco de exemplo}
  O bloco de exemplo geralmente é composto por tons de verde.
\end{exampleblock}
\begin{alertblock}{Bloco de alerta}
  O bloco de alerta geralmente é composto por tons de vermelho.
\end{alertblock}

```



1.8 Alinhamento do texto

As opções de alinhamento de texto dentro de um quadro estão dispostas abaixo.

- alinhar à esquerda (padrão): utiliza-se o ambiente `flushleft`;
- alinhar à direita: utiliza-se o ambiente `flushright`;
- centralizar: utiliza-se o ambiente `center` ou o comando `centering`;
- justificar: utiliza-se o ambiente `justify` ou o comando `justify`.

Note que os comandos acima são os mesmos utilizados em um documento \LaTeX .

Observação 5 *Para utilizar os comandos referentes à justificação do texto é necessário adicionar um pacote no preâmbulo: `ragged2e`.*

1.9 O ambiente `columns`

O ambiente utilizado para exibir texto ou figuras em colunas dentro de um `frame` é o `columns`. É possível definir as colunas de duas formas dentro desse ambiente: utilizando o comando `column` ou criando um ambiente interno chamado `column`.

Utilizando o ambiente `column`, a estrutura é a que segue.

```

\begin{columns}[<opções>]
  \begin{column}{<largura da coluna>}
    Conteúdo da coluna da esquerda.
  \end{column}
  \begin{column}{<largura da coluna>}
    Conteúdo da coluna da direita.
  \end{column}
\end{columns}

```

Para o comando `column`, utiliza-se o código-fonte do quadro abaixo.

```

\begin{columns}[<opções>]
  \column{<largura da coluna>} Conteúdo da coluna da esquerda.
  \column{<largura da coluna>} Conteúdo da coluna da direita.
\end{columns}

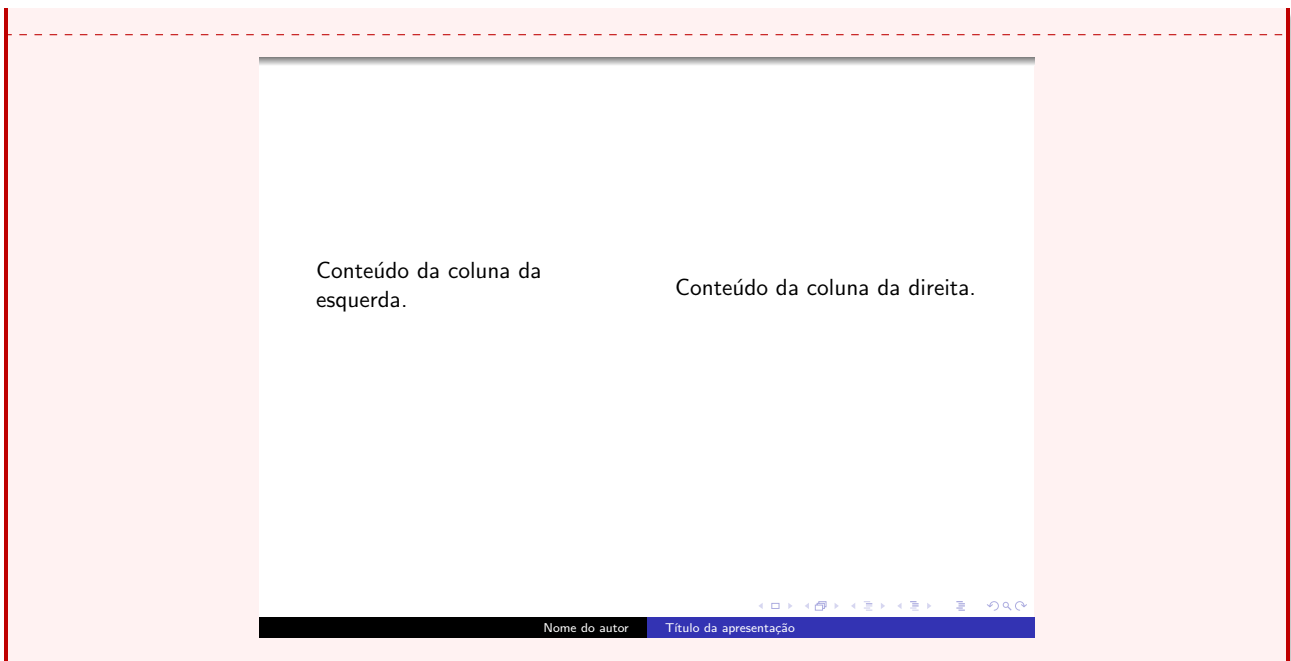
```

Exemplo 14 *Coluna com 5cm de largura. O resultado é o mesmo para os dois modos apresentados acima.*

```

\begin{columns}
  \begin{column}{5cm}
    Conteúdo da coluna da esquerda.
  \end{column}
  \begin{column}{5cm}
    Conteúdo da coluna da direita.
  \end{column}
\end{columns}

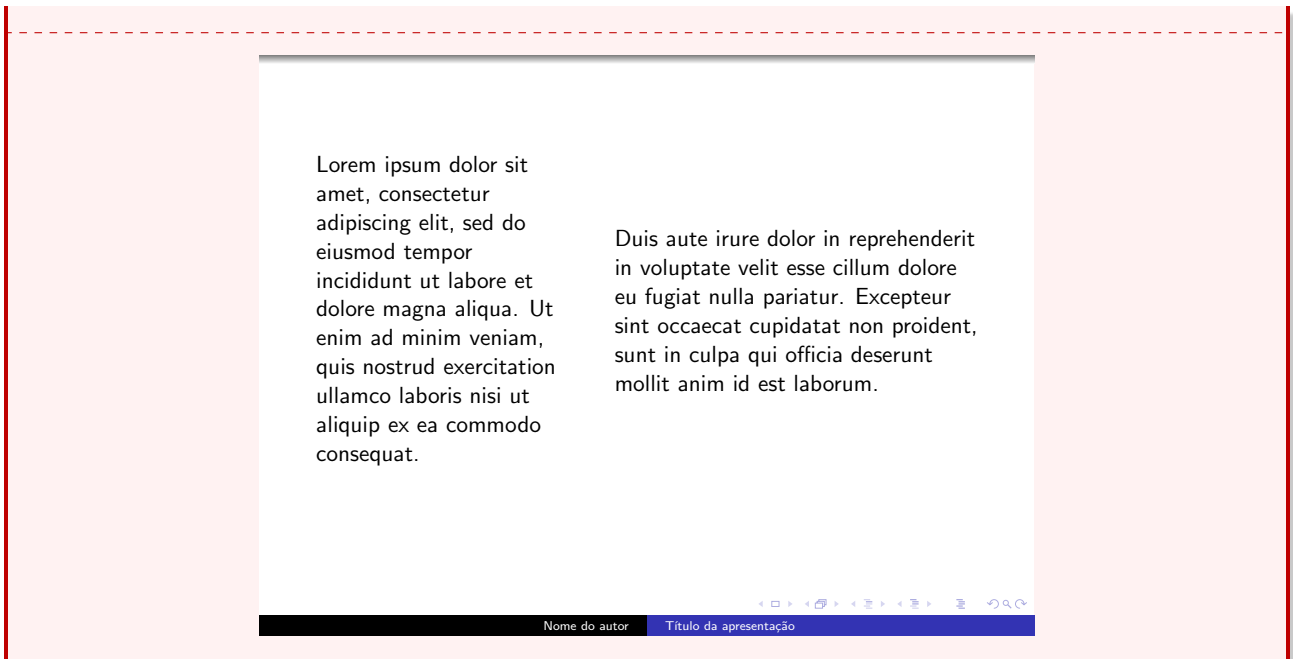
```



Exemplo 15 *Texto em duas colunas com larguras diferentes (esquerda: 4cm e direita: 6cm).*

```
\begin{columns}
  \begin{column}{4cm}
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do
    eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  \end{column}

  \begin{column}{6cm}
    Duis aute irure dolor in reprehenderit in voluptate velit esse
    cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat
    cupidatat non proident, sunt in culpa qui officia deserunt
    mollit anim id est laborum.
  \end{column}
\end{columns}
```



1.9.1 Alinhamento vertical no ambiente `columns`

Para alinhar o texto ou uma imagem de uma coluna verticalmente, utilizam-se as opções abaixo.

- `b`: alinha o conteúdo da coluna inferiormente (*bottom* - inferior);
- `t`: alinha o conteúdo da coluna superiormente (*top* - topo);
- `c`: alinha o conteúdo da coluna no centro do quadro (*center* - centro).

Exemplo 16 *Alinhando o conteúdo da coluna superiormente.*

```
\begin{columns}[t]  
<...>  
\end{columns}
```

Observação 6 *Em alguns casos, a utilização da opção “t” pode não funcionar. Nesse caso, recomenda-se a utilização da opção “T” (em maiúsculo).*

1.9.2 Alinhamento horizontal no ambiente `columns`

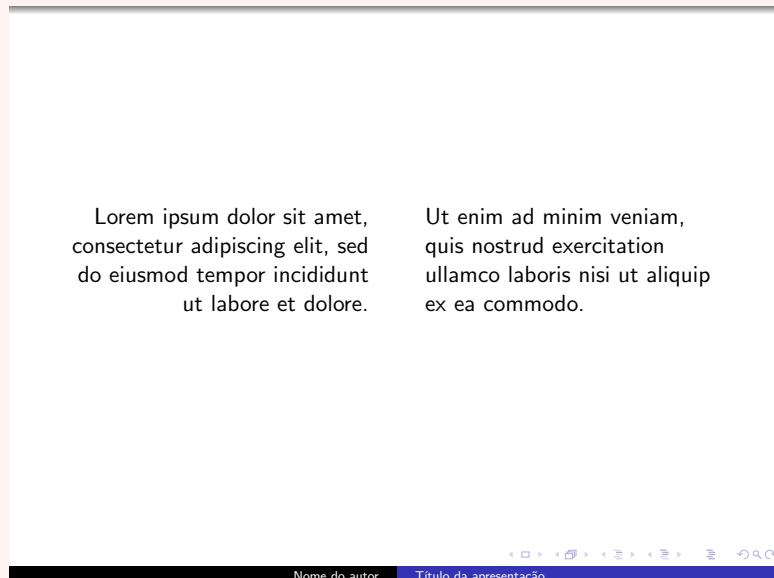
Para alinhar o conteúdo de uma coluna horizontalmente, utilizam-se os mesmos comandos da Seção 1.8.

Exemplo 17 *Alinhando o conteúdo da coluna da esquerda à direita e o conteúdo da coluna da direita à esquerda.*

```

\begin{columns}
  \begin{column}{5cm}
    \begin{flushright}
      Lorem ipsum dolor sit amet, consectetur adipiscing elit,
      sed do eiusmod tempor incididunt ut labore et dolore.
    \end{flushright}
  \end{column}
  \begin{column}{5cm}
    \begin{flushleft}
      Ut enim ad minim veniam, quis nostrud exercitation
      ullamco laboris nisi ut aliquip ex ea commodo.
    \end{flushleft}
  \end{column}
\end{columns}

```



1.10 Listas

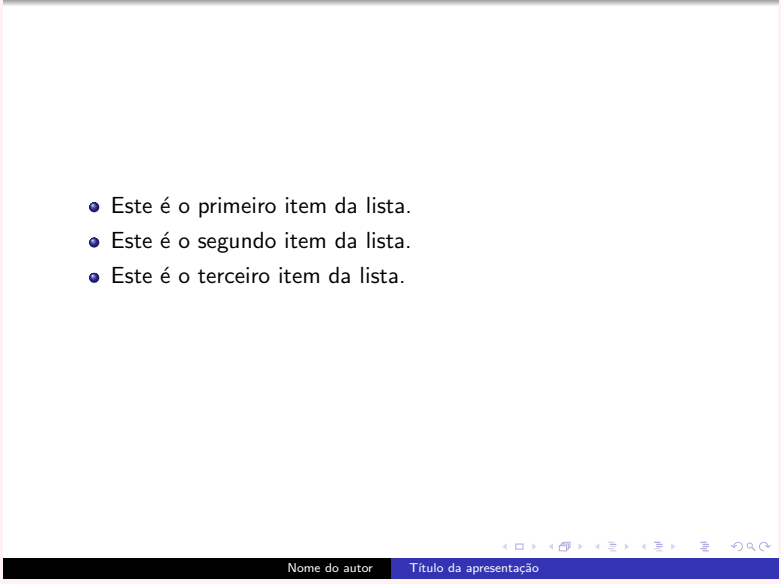
A estrutura básica para adicionar uma lista é a mesma utilizada na criação de documentos de texto em \LaTeX . É possível adicionar três tipos de listas no BEAMER.

1.10.1 Lista com marcadores - itemize

Para inserir uma lista com marcadores em um quadro utiliza-se os comandos apresentados no Exemplo 18.

Exemplo 18 *Lista com marcadores simples com 3 itens.*

```
\begin{itemize}
  \item Este é o primeiro item da lista.
  \item Este é o segundo item da lista.
  \item Este é o terceiro item da lista.
\end{itemize}
```

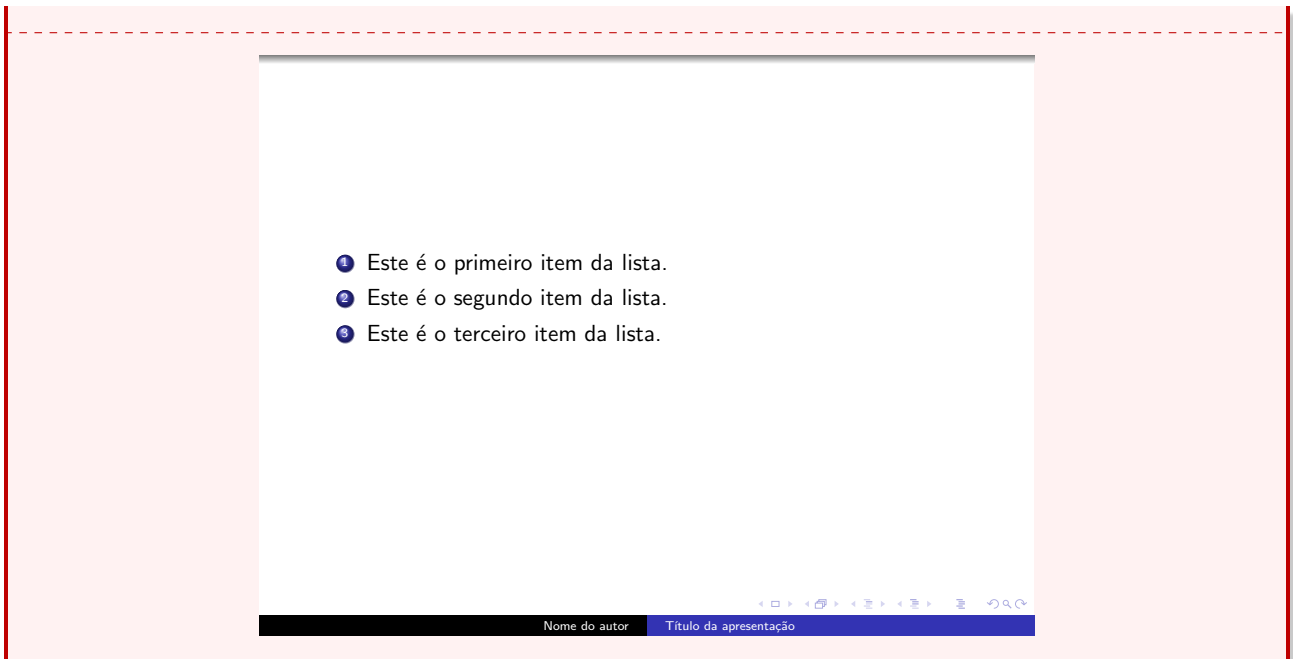
- 
- Este é o primeiro item da lista.
 - Este é o segundo item da lista.
 - Este é o terceiro item da lista.

1.10.2 Lista enumerada - enumerate

Para inserir uma lista enumerada no BEAMER são necessários os comandos apresentados no Exemplo 19.

Exemplo 19 *Lista enumerada simples com 3 itens.*

```
\begin{enumerate}
  \item Este é o primeiro item da lista.
  \item Este é o segundo item da lista.
  \item Este é o terceiro item da lista.
\end{enumerate}
```



1.10.3 Lista descritiva - description

O terceiro tipo de lista é a lista descritiva. A lista do tipo `description` permite ao usuário personalizar o marcador.

Exemplo 20 *Lista descritiva simples com 3 itens.*

```
\begin{description}
  \item[Item 1.] Este é o primeiro item da lista.
  \item[Item 2.] Este é o segundo item da lista.
  \item[Item 3.] Este é o terceiro item da lista.
\end{description}
```

A screenshot of a presentation slide showing the output of the LaTeX code above. The list items are displayed with blue text and blue markers: "Item 1. Este é o primeiro item da lista.", "Item 2. Este é o segundo item da lista.", and "Item 3. Este é o terceiro item da lista.". The slide includes the same navigation bar and footer as the previous slide.

1.11 Criando sobreposições

O BEAMER traz um comando simples mas não muito flexível para criar o efeito de sobreposição de *slides*. Nesse contexto, entende-se sobreposição como o aparecimento progressivo de linhas ou itens dentro de um quadro. O conceito *slide* apresentado aqui se refere aos itens de um quadro, e não ao número de quadros dentro de uma apresentação. O comando `pause` cria esse efeito de sobreposição.

Para criar esse efeito, insere-se o comando `pause` após o final uma palavra ou frase. Tudo o que vier após o `pause` não será exibido até avançar o `slide`.

Exemplo 21 *Para criar um simples questionário, com perguntas e respostas, utiliza-se o comando `pause`. A resposta não deve aparecer ao mesmo tempo que a pergunta.*

1. *Qual é a raiz quadrada do número 121? Resposta: 11.*
2. *Qual é a raiz quadrada do número 144? Resposta: 12.*

```
\begin{frame}
\begin{itemize}
  \item Qual é a raiz quadrada do número 121? \pause 11.
  \item Qual é a raiz quadrada do número 144? \pause 12.
\end{itemize}
\end{frame}
```

Nesse sentido, o comando `pause` é limitado. Caso seja necessário que um item apareça apenas no primeiro `slide`, o comando `pause` não dá conta. Por isso, existem outros comandos que permitem uma variedade maior de ações.

1.11.1 Os comandos `uncover` e `visible`

Os comandos `uncover` e `visible` exibem os *slides* em forma de cascata. É como se cada *slide* fosse “descoberto” de forma progressiva e posicionado abaixo do anterior. Portanto, o único *slide* visível é o atual.

```
\visible<especificações>{Texto do slide}\\
\uncover<especificações>{Texto do slide}\\
```

Ao utilizar o `uncover`, é possível adicionar um efeito de transparência nos slides anteriores e posteriores ao atual. Já o comando `visible` torna esses slides invisíveis. Para adicionar esse efeito de transparência, utiliza-se um novo comando no preâmbulo.

```
\setbeamercovered{transparent}
```

Destaca-se que o `uncover` significa, em português, descobrir ou revelar. Em outras palavras, é como se o `uncover` literalmente apenas “tapasse” e “destapasse” o conteúdo. Já o comando `visible` (visível, em português) torna o *slide* atual visível, enquanto os outros ficam invisíveis.

O parâmetro `especificações` permite personalizar o efeito visualizado no Exemplo 22. A Tabela 1.1 traz alguns exemplos de como isso pode ser feito.

Tabela 1.1: Tabela-exemplo de especificações aplicadas a um *slide*.

Comando	O que vai produzir
<code>\comando<1></code>	A informação aparece apenas no <i>slide</i> 1.
<code>\comando<2-></code>	A informação aparece a partir no <i>slide</i> 2.
<code>\comando<2-4></code>	A informação aparece entre os <i>slides</i> 2 e 4.
<code>\comando<1, 3, 4></code>	A informação aparece nos <i>slides</i> 1, 3 e 4.

O comando pode ser `uncover` ou `visible`.

Exemplo 22 *Exemplo de um quadro contendo 5 slides utilizando o comando `uncover`.*

```
\begin{frame}
\uncover<1> {O logaritmo natural é o logaritmo de base  $e$ ;} \\
\uncover<2> {O número irracional  $e=2,718\dots$  é chamado de número de Euler};\\
\uncover<3> {A função  $f(x)=\ln(x)$  corta o eixo  $x$  em  $x=1$ ;}\\
\uncover<4> {A derivada de  $f(x)=\ln(x)$  é  $f'(x)=\frac{1}{x}$ ;}\\
\uncover<5> {A integral de  $f(x)=\ln(x)$  é  $\int \ln(x)dx = x \cdot \ln(x) - x +$ 
 $\rightarrow C$ .}
\end{frame}
```

Observação 7 *É necessário pular a linha ao utilizar os comandos apresentados nessa subseção. Se isso não for feito, a informação aparecerá lado a lado.*

1.11.2 O comando `only`

O comando `only` (apenas, em português) funciona de modo semelhante. A diferença é que o `only` mantém todos os *slides* no centro da página. De certa forma, esse comando sobrepõe os *slides* anteriores. A sintaxe é a mesma, com exceção das duas barras no final do comando.

```
\only<especificações>{Texto do slide}
```

Exemplo 23 Criando um quadro contendo 5 slides com o comando `only`.

```
\begin{frame}
\only<1> {O logaritmo natural é o logaritmo de base  $e$ , onde  $e=2,718\dots$ .}
\only<2> {O número irracional  $e$  é chamado de número de Euler.}
\only<3> {A função  $f(x)=\ln(x)$  corta o eixo  $x$  em  $x=1$ .}
\only<4> {A derivada de  $f(x)=\ln(x)$  é  $f'(x)=\frac{1}{x}$ .}
\only<5> {A integral de  $f(x)=\ln(x)$  é  $\int \ln(x)dx = x \cdot \ln(x) - x + C$ .}
\end{frame}
```

1.11.3 O comando `item`

É possível aplicar os mesmos efeitos vistos anteriormente em uma lista.

Exemplo 24 Criando uma lista enumerada utilizando sobreposição, sendo que cada item apareça a partir do slide atual.

```
\begin{frame}
\begin{enumerate}

\item<1-> {O logaritmo natural é o logaritmo de base  $e$ , onde  $e=2,718\dots$ .}
\item<2-> {O número irracional  $e$  é chamado de número de Euler.}
\item<3-> {A função  $f(x)=\ln(x)$  corta o eixo  $x$  em  $x=1$ .}

\end{enumerate}
\end{frame}
```

Observação 8 Caso o comando de transparência apresentado na Subseção 1.11.1 esteja no preâmbulo, ele também aplicará esse efeito quando uma lista for utilizada.

Os três tipos de listas apresentados na Seção 1.10 podem ser utilizados. Ao criar um quadro utilizando listas, é possível generalizar o modo de como a sobreposição funciona. Ao inserir um parâmetro opcional ao lado do ambiente da lista, é possível aplicar o mesmo efeito sobre todos os slides.

Exemplo 25 O mesmo efeito do Exemplo 24 pode ser produzido com o código-fonte do quadro a seguir.

```

\begin{frame}
\begin{enumerate}[+>]

\item {O logaritmo natural é o logaritmo de base  $e$ , onde  $e=2,718\dots$ .}
\item {O número irracional  $e$  é chamado de número de Euler.}
\item {A função  $f(x)=\ln(x)$  corta o eixo  $x$  em  $x=1$ .}

\end{enumerate}
\end{frame}

```

É adicionado somente o parâmetro [$\langle + \rangle$], em que “+” é o número do slide atual (interpretado pelo BEAMER). Ou seja, no *slide* 1 o *software* entenderá que o símbolo + será igual a 1, e assim por diante.

Além disso, é possível adicionar os efeitos vistos anteriormente dentro de uma lista. Para isso, utiliza-se o seguinte parâmetro opcional ao lado de uma lista:

```
[<+-| comando@+>
```

de modo que o comando pode ser `only`, `visible`, `uncover` e `alert`. O comando `alert` destaca o texto do slide atual na cor vermelha.

Exemplo 26 *Criando uma lista em que o item atual apareça em destaque.*

```

\begin{frame}
\begin{enumerate}[<+-| alert@+>]

\item {O logaritmo natural é o logaritmo de base  $e$ , onde  $e=2,718\dots$ .}
\item {O número irracional  $e$  é chamado de número de Euler.}
\item {A função  $f(x)=\ln(x)$  corta o eixo  $x$  em  $x=1$ .}

\end{enumerate}
\end{frame}

```

Acredita-se que a utilização de listas seja a forma mais prática e efetiva de dispor os *slides* dentro de um quadro, visto que seu nível de personalização é mais aprimorado do que a utilização dos comandos `only`, `visible` e `uncover`.

Observação 9 *A especificação (ou seja, o que vai dentro de $\langle \rangle$) pode ser utilizada em blocos e em imagens. A ideia é a mesma utilizada até agora.*

1.12 Inserindo Imagens

O passo a passo para inserir uma imagem no BEAMER é o mesmo utilizado em um documento L^AT_EX. É necessário declarar no preâmbulo o pacote `graphicx`.

```
\usepackage{graphicx}
```

E no corpo do documento utiliza-se o comando `includegraphics` com dois parâmetros.

```
\includegraphics[scale=escala]{imagem.formato}
```

Observação 10 *Os formatos de imagens aceitos são vários. Recomenda-se a utilização dos arquivos PDF, PNG ou JPG.*

Para alinhar uma imagem em um quadro, utilizam-se os ambientes `flushleft`, `right` e `center`.

1.13 Dividindo a apresentação em seções e subseções

Os comandos utilizados para estruturar uma apresentação em seções e subseções estão dispostos no quadro abaixo.

```
\section{Nome da seção}
\subsection{Nome da subseção}
```

Exemplo 27 *Conjunto de quadros estruturados com seções. A figura a seguir mostra apenas o primeiro quadro da apresentação.*

```
\section{Introdução}
\begin{frame}
  Este é o quadro da introdução.
\end{frame}

\section{Metodologia}
\begin{frame}
  Este é o quadro da metodologia.
\end{frame}

\section{Desenvolvimento}
```

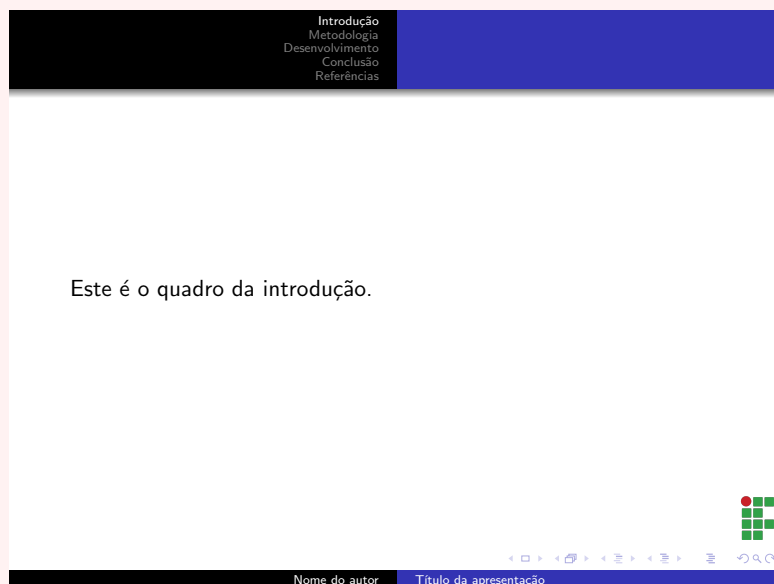
```

\begin{frame}
  Este é o quadro do desenvolvimento.
\end{frame}

\section{Conclusão}
\begin{frame}
  Este é o quadro da conclusão.
\end{frame}

\section{Referências}
\begin{frame}
  Este é o quadro das referências.
\end{frame}

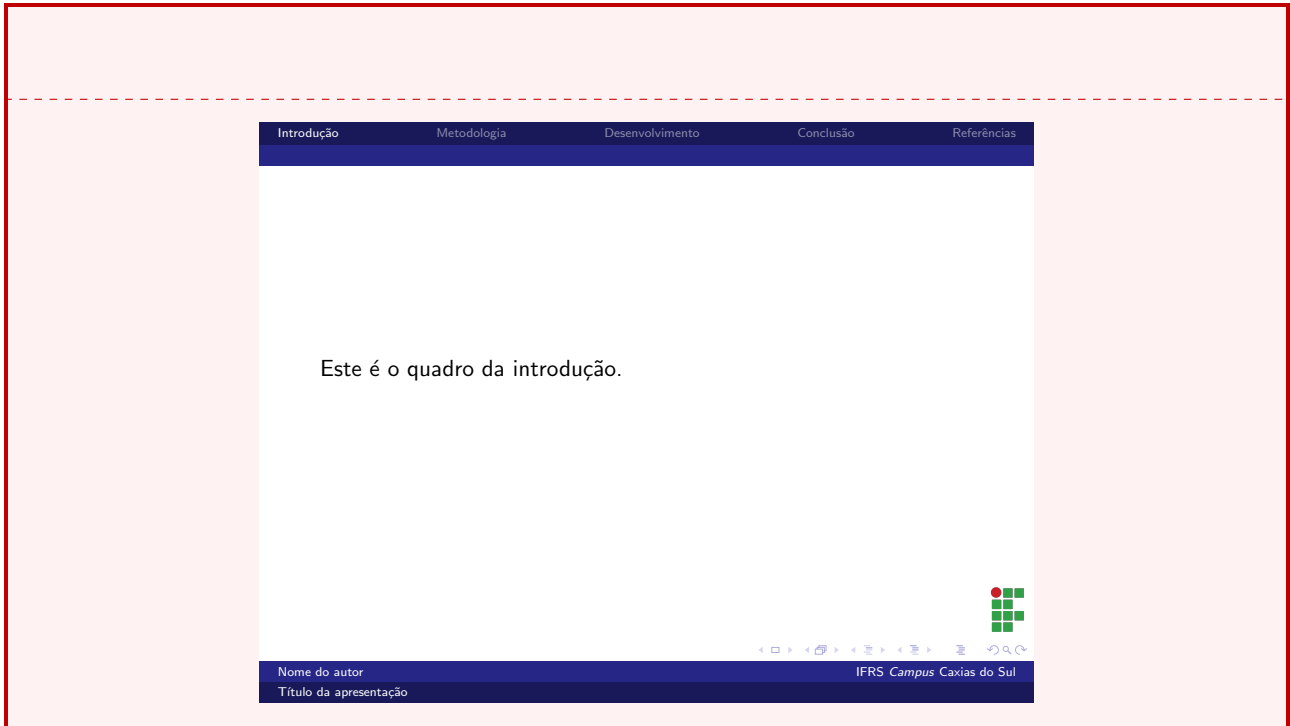
```



Note que no código-fonte os comandos de seção e subseção são inseridos logo acima do início do quadro o qual se deseja nomear. É possível navegar entre as seções ao clicar em uma seção na parte superior da imagem.

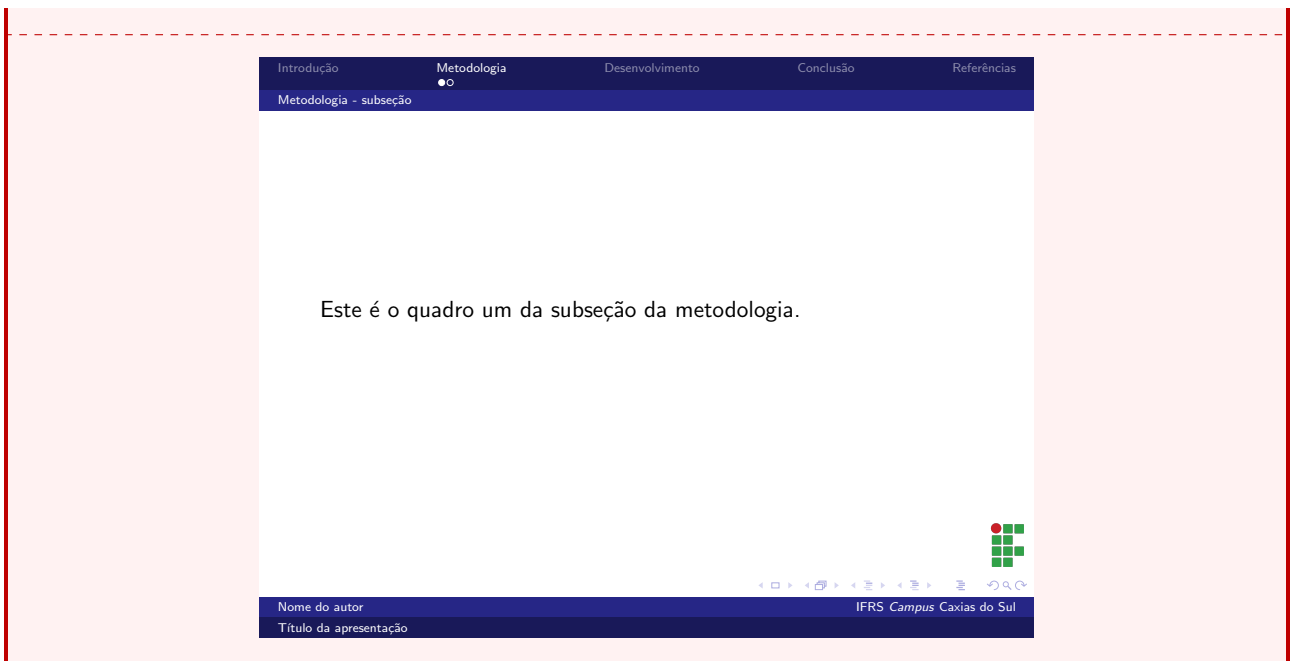
Observação 11 *O modo como as seções aparecem em um slide varia de acordo com o tema escolhido.*

Exemplo 28 *Código-fonte do Exemplo 27 com o tema Berlin.*



Exemplo 29 *Adicionando uma Subseção no quadro Metodologia. O tema utilizado é o Berlin.*

```
\subsection{Metodologia - subseção}
\begin{frame}
  Este é o quadro um da subseção da metodologia.
\end{frame}
\begin{frame}
  Este o quadro dois da subseção da metodologia.
\end{frame}
```



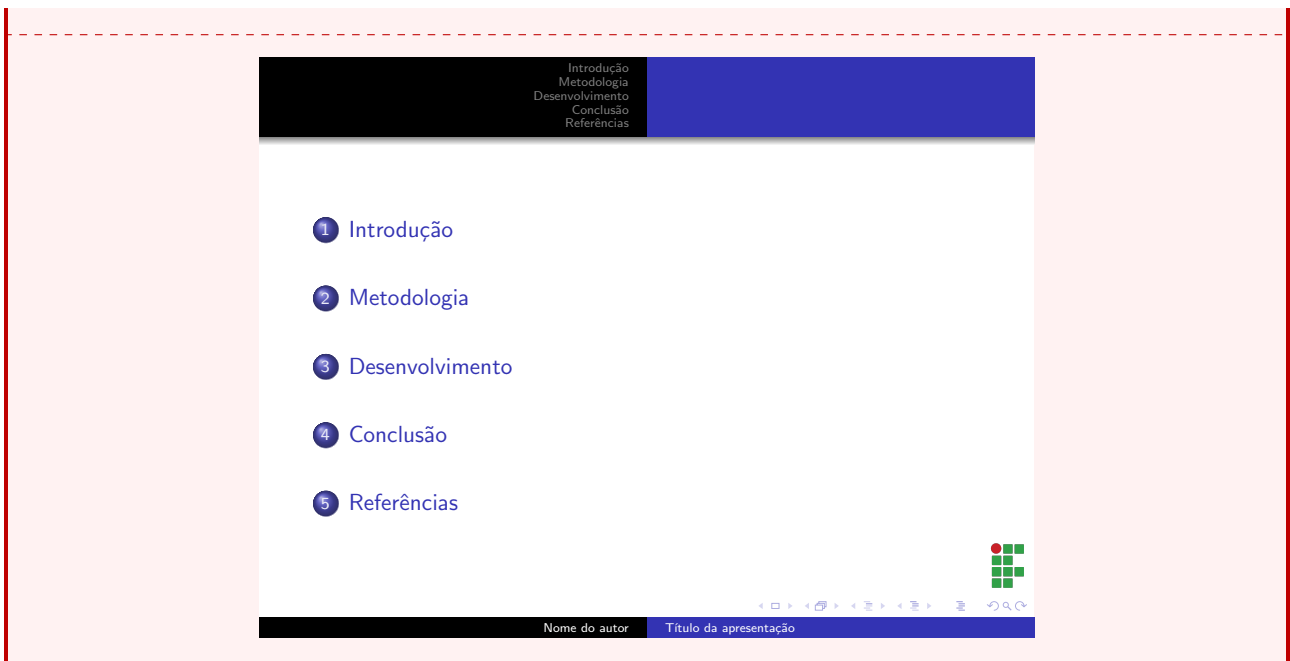
1.14 Criando o sumário

Para construir um sumário, utiliza-se o comando `tableofcontents` dentro de um quadro.

```
\begin{frame}
  \tableofcontents
\end{frame}
```

Exemplo 30 *Utilizando `tableofcontents` com os comandos do Exemplo 27.*

```
\begin{frame}
  \tableofcontents
\end{frame}
<código-fonte do Exemplo 25>
```

1.14.1 Sumário no início de uma seção

É possível inserir o sumário a cada início de seção. Para isso, é possível utilizar o comando `AtBeginSection` no preâmbulo do documento.

```
\AtBeginSection{
  \begin{frame}
    \tableofcontents
  \end{frame}
}
```

Para que a seção atual fique destacada, adiciona-se um parâmetro opcional no comando `tableofcontents`.

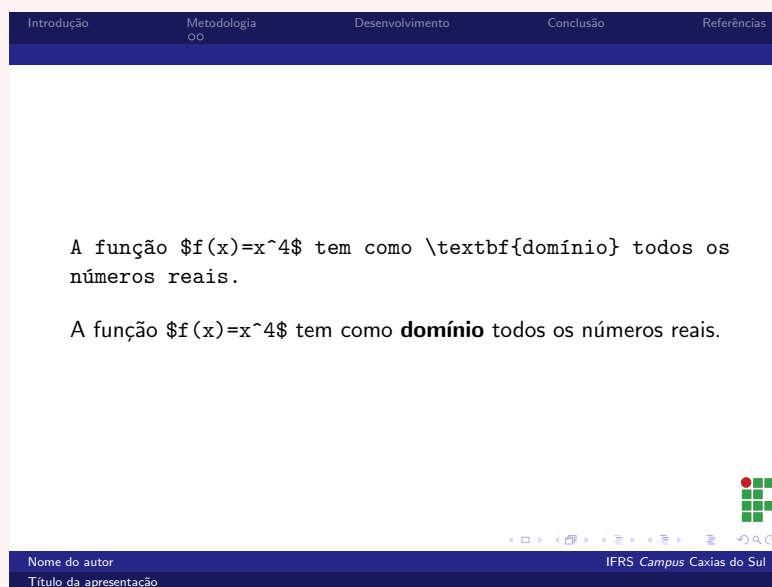
```
\tableofcontents[currentsection]
```

1.15 O parâmetro fragile

Em um documento redigido em $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ é possível inserir trechos de código-fonte dentro do documento. Para fazer isso no BEAMER, é necessário utilizar um parâmetro opcional no quadro desejado chamado `fragile`. Isso pode ser feito de duas formas. A primeira é utilizando o ambiente `verbatim`. Nesse caso, os comandos inseridos serão interpretados como texto e serão escritos sem formatação. A segunda forma é utilizando o comando `verb`. Esse comando permite um controle maior, podendo ser inserido individualmente, ou seja, apenas em trechos em que o código-fonte necessita aparecer.

Exemplo 31 Utilizando o parâmetro `fragile` juntamente com o comando `verb`.

```
\begin{frame}[fragile]
\begin{verbatim}
A função  $f(x)=x^4$  tem como \textbf{domínio} todos os
números reais.
\end{verbatim}
A função \verb|$f(x)=x^4$| tem como \textbf{domínio} todos os
números reais.
\end{frame}
```



1.16 Nota de rodapé

Para inserir uma nota de rodapé utiliza-se o comando `footnote`.

```
\footnote{texto da nota de rodapé}
```

1.17 Botões e *hyperlinks*

Para inserir um *hyperlink* em um quadro no BEAMER é necessário utilizar o comando `href`.

```
\href{<link>}{<descrição>}
```

É possível inserir *hyperlinks* internos. Desse modo, ao clicar em uma palavra específica de um quadro, o BEAMER redirecionará o usuário para outro quadro. Para isso, utiliza-se o comando `hyperlink`.

```
\hyperlink{<quadro destino>}{<descrição>}
```

Para utilizar esta funcionalidade é necessário identificar o quadro para o qual o redirecionamento será feito. Para identificar um quadro utiliza-se o comando `label`.

```
\label{<quadro destino>}
```

Exemplo 32 Para criar uma frase no quadro 2 (Metodologia) que volte para o quadro 1 (Introdução), utiliza-se o seguinte conjunto de comandos.

```
\section{Introdução}
  \begin{frame}\label{intro}
    Este é o quadro da introdução.
  \end{frame}

\section{Metodologia}
  \begin{frame}
    Este é o quadro da metodologia. \hyperlink{intro}{Clique aqui
    para voltar para a introdução}.
  \end{frame}
```

O BEAMER permite adicionar botões de redirecionamento. Nenhum pacote adicional é necessário para essa função.

```
\hyperlink{<quadro destino>}{\beamerbutton{<descrição>}}
```

Exemplo 33 Para inserir um botão simples, utiliza-se o comando `beamerbutton`.

```
\hyperlink{intro}{\beamerbutton{Voltar}}
```

Existem outras variações de botões. São elas:

```
\hyperlink{<quadro destino>}{\beamergetobutton{<descrição>}}
\hyperlink{<quadro destino>}{\beamerskipbutton{<descrição>}}
\hyperlink{<quadro destino>}{\beamerreturnbutton{<descrição>}}
```

Observação 12 As variações dos botões são apenas visuais. É necessário definir manualmente o destino desejado.

1.18 Inserindo áudio e vídeo

É possível inserir arquivos de áudio e vídeo dentro de uma apresentação em BEAMER. No entanto, é necessário salientar que nem todos os leitores de arquivo PDF são compatíveis com essa função. Para utilizar essa função é necessário um novo pacote.

```
\usepackage{multimedia}
```

1.18.1 O comando sound

Para inserir uma faixa de áudio, utiliza-se o comando `sound`.

```
\sound[<parâmetros opcionais>]{<descrição>}{<arquivo de áudio>}
```

O parâmetro `<descrição>` é o texto que aparecerá para iniciar a reprodução da faixa de áudio e o `<arquivo de áudio>` é o nome do arquivo a ser utilizado. Esse arquivo deve estar em uma pasta ou subpasta junto ao arquivo `.TEX`.

Observação 13 *Por padrão, o \LaTeX não incorpora arquivos de áudio junto ao documento PDF. Para incorporar um arquivo de áudio, adiciona-se o parâmetro opcional `inlinesound`.*

Para fazer com que o áudio inicie automaticamente, utiliza-se o parâmetro opcional `autostart`.

1.18.2 O comando movie

Para inserir um vídeo em uma apresentação, utiliza-se o comando `movie`.

```
\movie[<parâmetros opcionais>]{<descrição>}{<arquivo de vídeo>}
```

Os parâmetros `descrição` e `arquivo de vídeo` funcionam da mesma forma que no comando `sound`. Como parâmetros opcionais, os mais relevantes são:

- `showcontrols`: apresenta uma barra de comandos quando o vídeo é reproduzido;
- `autostart`: inicia o vídeo de forma automática;
- `width`: modifica a largura do vídeo;
- `height`: modifica a altura do vídeo;
- `externalviewer`: reproduz o vídeo externamente ao leitor PDF.

Observação 14 *O \LaTeX não incorpora arquivos de vídeo junto ao documento PDF. Desse modo, o arquivo de vídeo deve ser mantido em uma mesma pasta ou subpasta do arquivo \TeX .*

Capítulo 2

Desenhando com o TikZ

2.1 Considerações Iniciais

O TikZ é um pacote de desenho gráfico baseado na linguagem L^AT_EX. *Softwares* como o *Adobe Illustrator* e o *CorelDraw* permitem criar um desenho com ferramentas pré-definidas em tempo real, enquanto um desenho utilizando o TikZ é criado a partir de um conjunto de comandos, que são executados em sequência. O TikZ opera, basicamente, a partir de coordenadas. Para utilizar esta ferramenta é necessário um pacote adicional.

```
\usepackage{tikz}
```

É possível iniciar um desenho de duas formas: com um comando ou com um ambiente.

```
\tikz[<opções>]{<comandos TikZ>}

\begin{tikzpicture}[<opções>]
  <comandos TikZ>
\end{tikzpicture}
```

No entanto, acredita-se que a utilização de um ambiente torna a visualização do código-fonte mais clara. Em desenhos com muitos comandos, o ambiente pode melhorar a organização visual do código-fonte.

É possível enumerar quatro vantagens do pacote TikZ:

- Documento padronizado, de modo que a simbologia é a mesma no texto e nas imagens;
- Desenhos simples podem ser criados de forma rápida;
- O posicionamento do recurso gráfico é muito preciso;
- É possível utilizar macros.

Observação 15 Este material possui a seguinte estrutura: à esquerda localiza-se o código-fonte e à direita o desenho produzido. Em alguns casos, o código-fonte aparecerá na parte superior e o desenho na parte inferior do quadro.

Exemplo 34 Intervalo aberto em uma reta.

```

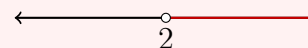
\begin{tikzpicture}
\tikzstyle{aberto}=[fill=white, draw=black, circle,
↪ inner sep=1.2pt]

\coordinate (A) at (-2,0);
\coordinate (B) at (0,0);
\coordinate (C) at (2,0);
\coordinate (C1) at (1.95,0);

\draw[thick, <->] (A)--(B) --(C);
\draw[red!80!black, thick] (B)--(C1);

\node at (B)[aberto]{};
\node at (B)[below]{$2$};
\end{tikzpicture}

```



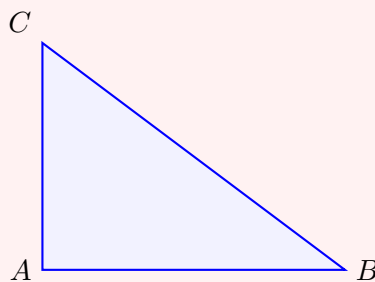
Exemplo 35 Triângulo retângulo.

```

\begin{tikzpicture}
\coordinate (A) at (0,0) node at (A)[left]{\small{$A$}};
\coordinate (B) at (4,0) node at (B)[right]{\small{$B$}};
\coordinate (C) at (0,3) node at (C)[above left]{\small{$C$}};

\fill[blue!5] (A) -- (B) -- (C) -- cycle;
\draw[blue, thick] (A) -- (B) -- (C) -- cycle;
\end{tikzpicture}

```



2.2 Estabelecendo coordenadas

No TikZ a orientação de um desenho se dá por coordenadas (retangulares ou polares). Para estabelecer uma coordenada retangular, utiliza-se a notação padrão $(0,0)$. Caso não seja colocada nenhuma unidade de medida, o \LaTeX entenderá como centímetro. Caso seja necessário utilizar outra unidade, como milímetros, escreve-se $(0\text{ mm},0\text{ mm})$.

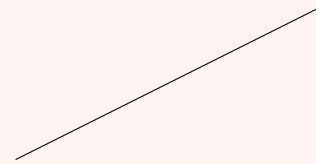
Para estabelecer uma coordenada polar utiliza-se a notação $(\theta:r)$, em que θ é a coordenada angular (em graus) e r é a coordenada radial (em centímetros). Nesse caso, recomenda-se a utilização da unidade de medida para a coordenada radial, mesmo ela não sendo necessária, por exemplo, $(30:2\text{ cm})$.

2.3 O comando draw

O comando `draw` cria linhas por meio de duas ou mais coordenadas. A palavra *draw* (do inglês) significa desenhar.

Exemplo 36 *Linha com início em $(0,0)$ e final em $(4,2)$.*

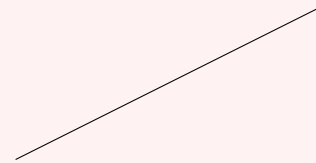
```
\begin{tikzpicture}
  \draw (0,0) -- (4,2);
\end{tikzpicture}
```



A mesma linha pode ser representada em coordenadas polares, conforme exemplo abaixo.

Exemplo 37 *Linha do Exemplo 36 em coordenadas polares.*

```
\begin{tikzpicture}
  \draw (0:0cm) -- (26.56:4.47cm);
\end{tikzpicture}
```



Observação 16 *O separador decimal utilizado no \LaTeX é o ponto. A vírgula é utilizada para coordenadas retangulares e o símbolo dois-pontos é utilizado para coordenadas polares.*

2.3.1 Espessura da linha

Para alterar a espessura da linha, utiliza-se um parâmetro opcional junto ao comando `draw`.

```
\begin{tikzpicture}
  \draw[<espessura>] <...>;
\end{tikzpicture}
```

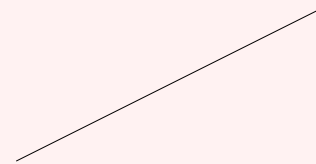
A tabela a seguir contém algumas opções de parâmetros que definem a espessura da linha.

Tabela 2.1: Parâmetros que definem a espessura da linha.

parâmetro	espessura
<code>ultra thin</code>	0.1pt
<code>very thin</code>	0.2pt
<code>thin</code>	0.4pt
<code>semithick</code>	0.6pt
<code>thick</code>	0.8pt
<code>very thick</code>	1.2pt
<code>ultra thick</code>	1.6pt
<code>line width</code>	definido pelo usuário

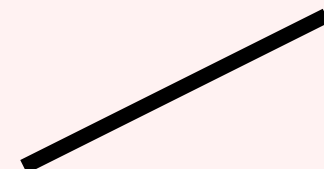
Exemplo 38 *Linha do Exemplo 36 com espessura 0.1pt (ultra thin).*

```
\begin{tikzpicture}
  \draw[ultra thin] (0,0) -- (4,2);
\end{tikzpicture}
```



Exemplo 39 *Linha do Exemplo 36 com espessura 0.2cm.*

```
\begin{tikzpicture}
  \draw[line width=0.2cm] (0,0) -- (4,2);
\end{tikzpicture}
```



2.3.2 Cor da linha

Para alterar a cor da linha, adiciona-se um parâmetro opcional juntamente ao comando `draw`. As formas mais usuais de selecionar uma cor são a partir das cores pré-definidas, da mistura de duas cores ou da criação de uma cor personalizada.


```

\begin{tikzpicture}
  \draw[<cor da linha>] <...>;
\end{tikzpicture}

```

A Tabela 2.2 apresenta as cores pré-definidas no \LaTeX .

Tabela 2.2: Cores pré-definidas no \LaTeX .

parâmetro	cor
red	vermelho
green	verde
blue	azul
cyan	ciano
magenta	magenta
yellow	amarelo
gray	cinza
darkgray	cinza escuro
lightgray	cinza claro
brown	marrom
lime	lima
olive	verde-oliva
orange	laranja
pink	rosa
purple	roxo
teal	verde-azulado
violet	violeta

Observação 17 *É possível criar uma cor utilizando o comando `definecolor` do pacote `xcolor`.*

Exemplo 40 *Linha do Exemplo 39 na cor laranja.*

```

\begin{tikzpicture}
  \draw[line width=0.2cm, orange] (0,0) -- (4,2);
\end{tikzpicture}

```



Observação 18 *Os parâmetros opcionais são separados por vírgula.*

2.3.3 Tipo da linha

Para alterar o tipo de linha utiliza-se um parâmetro opcional.

```
\begin{tikzpicture}
  \draw[<tipo da linha>] <...>;
\end{tikzpicture}
```

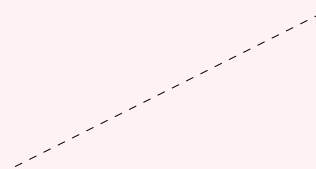
A Tabela 2.3 apresenta os tipos de linha que o TikZ possui.

Tabela 2.3: Tipos de linha.

parâmetro	tipo
solid	sólida
dashed	tracejada
dotted	pontilhada (padrão)
dashdotted	tracejada e pontilhada
densely dotted	pontilhada (densa)
loosely dotted	pontilhada (esparso)
double	dupla
dash dot dot	traço ponto ponto

Exemplo 41 *Linha do Exemplo 38 tracejada.*

```
\begin{tikzpicture}
  \draw[ultra thin, dashed] (0,0) -- (4,2);
\end{tikzpicture}
```



2.3.4 Formas geométricas pré-definidas

O comando `draw` também permite desenhar algumas formas geométricas por meio de comandos simples. As figuras geométricas que o TikZ possui são: retângulo, círculo e elipse. Além disso, é possível traçar um arco com um comando especial. Um retângulo é desenhado por meio do comando `rectangle`.

```
\begin{tikzpicture}
  \draw (<vértice inferior esquerdo>) rectangle (<vértice superior direito>);
\end{tikzpicture}
```

Exemplo 42 *Retângulo de dimensões 4cm × 2cm utilizando o comando rectangle.*

```
\begin{tikzpicture}
  \draw (0,0) rectangle (4,2);
\end{tikzpicture}
```



Exemplo 43 *Quadrado de lado 2cm utilizando o comando rectangle.*

```
\begin{tikzpicture}
  \draw (0,0) rectangle (2,2);
\end{tikzpicture}
```

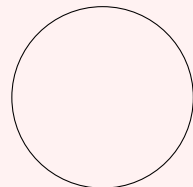


Para desenhar uma circunferência, utiliza-se o comando `circle` com um parâmetro `radius` (raio, em português).

```
\begin{tikzpicture}
  \draw (<centro>) circle[radius=<raio>];
\end{tikzpicture}
```

Exemplo 44 *Circunferência de raio 1.2cm.*

```
\begin{tikzpicture}
  \draw (0,0) circle[radius=1.2cm];
\end{tikzpicture}
```

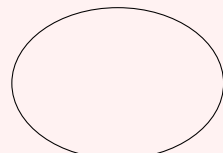


O comando utilizado para desenhar uma elipse é o mesmo, apenas com uma modificação. Indica-se `x radius` como o raio em x e `y radius` como o raio em y .

```
\begin{tikzpicture}
  \draw (<centro>) circle[x radius=<raio em x>, y radius=<raio em y>];
\end{tikzpicture}
```

Exemplo 45 *Elipse de raios $x = 1.4cm$ e $y = 1cm$.*

```
\begin{tikzpicture}
  \draw (0,0) circle[x radius=1.4cm, y radius=1cm];
\end{tikzpicture}
```



Observação 19 *Existem outras formas de desenhar círculos e elipses. No entanto, acredita-se que as formas utilizadas nos Exemplos 44 e 45 são as mais intuitivas.*

Para desenhar um arco, utiliza-se o comando `arc` junto ao comando `draw`.

```
\begin{tikzpicture}
  \draw (<centro>) arc (<ângulo inicial> : <ângulo final> : <raio>);
\end{tikzpicture}
```

Exemplo 46 *Arco de 150°.*

```
\begin{tikzpicture}
  \draw (0,0) arc (0:150:1cm);
\end{tikzpicture}
```



2.3.5 Extremidades da linha

Para inserir uma seta na extremidade de uma linha, utilizam-se os parâmetros opcionais da Tabela 2.4.

Tabela 2.4: Algumas setas do pacote TikZ.

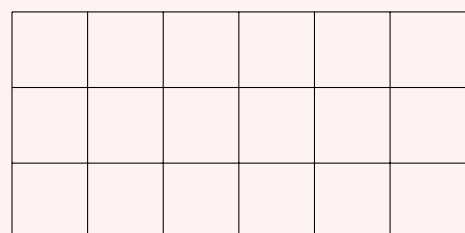
parâmetro	tipo de seta
<code>- ></code>	
<code>< - ></code>	
<code>- >></code>	
<code>- >>></code>	
<code> - ></code>	

2.3.6 O comando grid

O comando `grid` cria uma malha. Esse comando é amplamente utilizado em gráficos, a fim de tornar a visualização dos pontos mais clara em um plano cartesiano.

Exemplo 47 *Malha utilizando o comando grid.*

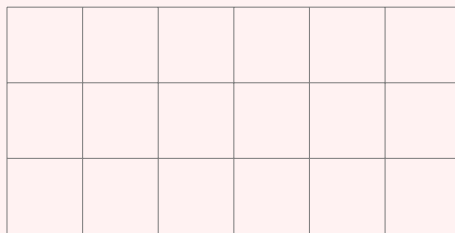
```
\begin{tikzpicture}
  \draw (-3,-1) grid (3,2);
\end{tikzpicture}
```



Para alterar a cor ou a espessura da linha, utiliza-se os comandos da Seção 2.3.

Exemplo 48 *Malha com espessura da linha thin e cor da linha cinza.*

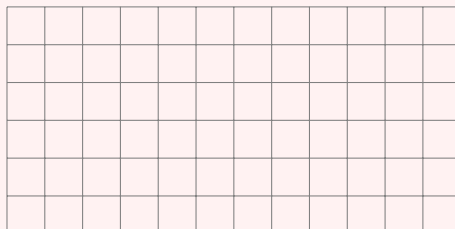
```
\begin{tikzpicture}
  \draw[thin, gray] (-3,-1) grid (3,2);
\end{tikzpicture}
```



Para alterar o passo da malha, ou seja, quanto ela varia em x e em y , utiliza-se o parâmetro `step`.

Exemplo 49 *Malha do Exemplo 48 com 5mm de passo.*

```
\begin{tikzpicture}
  \draw[thin, gray, step=5mm] (-3,-1) grid
  → (3,2);
\end{tikzpicture}
```

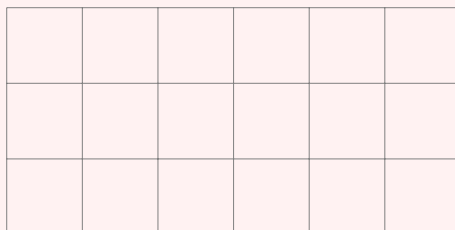


Observação 20 *Caso seja necessário ajustar este parâmetro para cada eixo de modo individual, utilizam-se os parâmetros `xstep` e `ystep`.*

O parâmetro `help lines` cria uma malha semelhante ao Exemplo 47.

Exemplo 50 *Malha utilizando o parâmetro help lines.*

```
\begin{tikzpicture}
  \draw[help lines] (-3,-1) grid (3,2);
\end{tikzpicture}
```



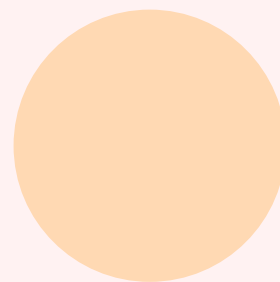
2.4 O comando fill

O comando `fill` permite criar apenas o preenchimento de uma figura.

```
\begin{tikzpicture}
  \fill[<cor desejada>] <...>;
\end{tikzpicture}
```

Exemplo 51 *Círculo de raio 18mm com preenchimento utilizando o comando fill.*

```
\begin{tikzpicture}
  \fill[orange!30] (0,0) circle[radius=18mm];
\end{tikzpicture}
```



Outra forma de aplicar preenchimento em um objeto é utilizando o comando `draw` com um parâmetro opcional `fill`.

Exemplo 52 *Retângulo do Exemplo 42 com preenchimento utilizando o comando draw e o parâmetro fill.*

```
\begin{tikzpicture}
  \draw[fill=blue!20] (0,0) rectangle (4,2);
\end{tikzpicture}
```

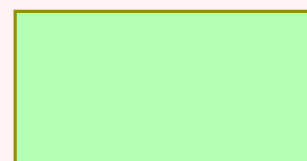


Observação 21 *Os comandos draw e fill podem ser utilizados com parâmetros.*

No caso do Exemplo 42 o contorno do objeto aparece. Isso acontece porque foi utilizado o comando `draw` e, por padrão, o TikZ adiciona uma borda preta fina.

Exemplo 53 *Retângulo com borda grossa verde-oliva.*

```
\begin{tikzpicture}
  \draw[fill=green!30, draw=olive, very thick] (0,0)
  \rightarrow rectangle (4,2);
\end{tikzpicture}
```



2.5 O comando path

O comando `path` cria apenas o caminho de um desenho. Em outras palavras, ele não cria o preenchimento nem a borda de uma figura. A sintaxe do comando `path` é semelhante a dos comandos `draw` e `fill`. Uma aplicação do comando `path` está no Exemplo 86.

2.6 O comando coordinate

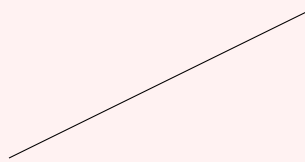
O pacote TikZ possui um comando chamado `coordinate` que permite identificar uma coordenada. Este comando é útil quando utiliza-se o mesmo par ordenado mais de uma vez em

uma mesma figura. Esta identificação é apenas utilizada pelo ambiente, ou seja, ela não ficará visível na tela.

```
\begin{tikzpicture}
  \coordinate (<nome>) at (<coordenada>);
\end{tikzpicture}
```

Exemplo 54 Identificando um ponto $(5,2)$ com a letra A , um ponto $(2 : 1cm)$ como B e ligando A com B .

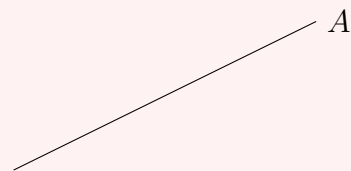
```
\begin{tikzpicture}
  \coordinate (A) at (5,2);
  \coordinate (B) at (2:1cm);
  \draw (A) -- (B);
\end{tikzpicture}
```



É possível nomear um ponto com o comando *label* (etiqueta, em português). Para isso, utiliza-se o parâmetro *label* junto ao comando *coordinate*.

Exemplo 55 Utilizando o Exemplo 54 para nomear o ponto A .

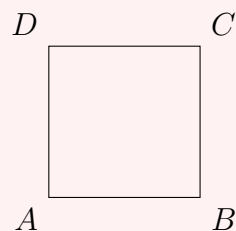
```
\begin{tikzpicture}
  \coordinate[label=right:$A$] (A) at (5,2);
  \coordinate (B) at (2:1cm);
  \draw (A) -- (B);
\end{tikzpicture}
```



O parâmetro *right* indica que o nome do ponto, neste caso A , aparecerá à direita. Existem outras opções pré-definidas, como: *above*, *below*, *left*, *above right*, *above left* e assim por diante.

Exemplo 56 Quadrado de lado $2cm$ com os quatro vértices identificados.

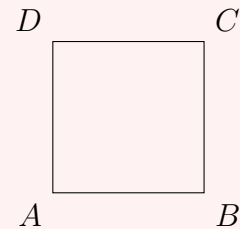
```
\begin{tikzpicture}
  \coordinate[label=below left:$A$] (A) at (0,0);
  \coordinate[label=below right:$B$] (B) at (2,0);
  \coordinate[label=above right:$C$] (C) at (2,2);
  \coordinate[label=above left:$D$] (D) at (0,2);
  \draw (A) -- (B) -- (C) -- (D) -- cycle;
\end{tikzpicture}
```



Existe a possibilidade de nomear uma coordenada utilizando o posicionamento graus. Esse modo é mais preciso do que os comandos apresentados anteriormente.

Exemplo 57 *Quadrado do Exemplo 56 com os quatro vértices identificados, utilizando ângulos.*

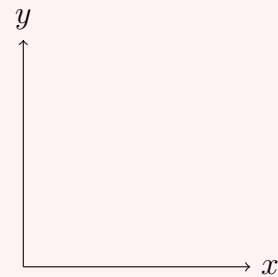
```
\begin{tikzpicture}
  \coordinate[label=-135:$A$] (A) at (0,0);
  \coordinate[label=-45:$B$] (B) at (2,0);
  \coordinate[label=45:$C$] (C) at (2,2);
  \coordinate[label=135:$D$] (D) at (0,2);
  \draw (A) -- (B) -- (C) -- (D) -- cycle;
\end{tikzpicture}
```



Exemplo 58 *Plano cartesiano simples.*

```
\begin{tikzpicture}
  \coordinate (0) at (0,0);
  \coordinate[label=90:$y$] (A) at (0,3);
  \coordinate[label=0:$x$] (B) at (3,0);

  \draw[->] (0) -- (A);
  \draw[->] (0) -- (B);
\end{tikzpicture}
```



2.7 O comando node

O comando `node` cria um nó em um ponto selecionado, permitindo inserir desde um texto até um objeto. O seu comportamento é semelhante ao do comando `label`. O `node` é inserido como um comando ou dentro de um comando (como o `draw`).

```
node[<opções>](<nome>){<texto do nó>}
```

Em `opções` são inseridos parâmetros opcionais, como localização do texto do nó e alinhamento. Além disso, é possível inserir uma forma diretamente em um nó e isso é definido também em `opções`. O nome é utilizado para referenciar esse nó. E o `texto do nó` é, literalmente, o texto que ficará visível neste nó.

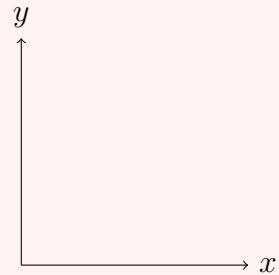
Exemplo 59 *Plano cartesiano simples, utilizando o comando `node` para nomear os eixos.*


```

\begin{tikzpicture}
  \coordinate (O) at (0,0);
  \coordinate (A) at (0,3);
  \coordinate (B) at (3,0);

  \draw[->] (O) -- (A) node[above]{$y$};
  \draw[->] (O) -- (B) node[right]{$x$};
\end{tikzpicture}

```



Observação 22 Para inserir uma forma em nó basta inserir dois parâmetros opcionais. Por exemplo, para inserir um círculo com preenchimento azul, insere-se `circle, fill=blue`.

2.8 O parâmetro scale

Como nas imagens, é possível utilizar o parâmetro `scale` no TikZ. Esse parâmetro pode ser muito útil quando for necessário adaptar um grande conjunto de comandos, sem que seja necessário modificar comando por comando.

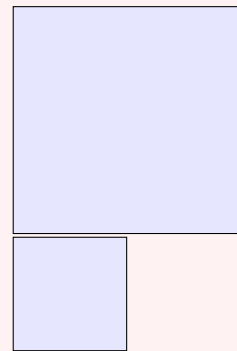
Exemplo 60 Quadrado de lado 3cm (acima) e em escala menor (abaixo).

```

\begin{tikzpicture}
  \draw[fill=blue!10] (0,0) -- (0,3) -- (3,3) --
  → (3,0) -- cycle;
\end{tikzpicture}

\begin{tikzpicture}[scale=0.5]
  \draw[fill=blue!10] (0,0) -- (0,3) -- (3,3) --
  → (3,0) -- cycle;
\end{tikzpicture}

```



Observação 23 É possível modificar a escala individualmente, ou seja, em x e em y de forma separada. Para isso, utilizam-se os parâmetros `xscale` e `yscale`.

2.9 O comando tikzset

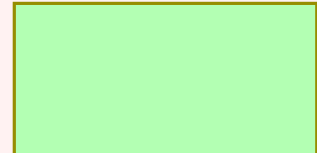
O comando `tikzset` permite definir um estilo com apenas um parâmetro. Desse modo, é possível agrupar vários parâmetros em um só.

```
\tikzstyle{<nome do estilo>}=[<parâmetros>]
```

O nome do estilo é o nome definido pelo usuário. Dentro de parâmetros é possível utilizar qualquer parâmetro que possa ser inserido dentro do comando `draw`, por exemplo.

Exemplo 61 *Figura do Exemplo 53 utilizando o comando `tikzset`.*

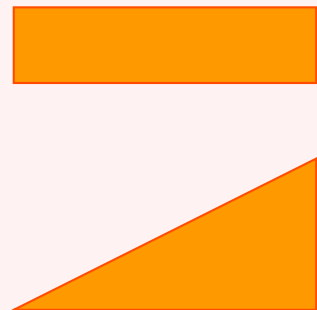
```
\tikzstyle{verde}=[fill=green!30, draw=olive, very  
→ thick]  
\begin{tikzpicture}  
  \draw[verde] (0,0) rectangle (4,2);  
\end{tikzpicture}
```



A principal vantagem de definir um estilo é a facilidade de alterá-lo fora do ambiente `tikzpicture`. Assim, ao utilizar os mesmos parâmetros várias vezes é necessário fazer apenas uma alteração.

Exemplo 62 *Dois polígonos utilizando o mesmo estilo pré-definido com o comando `tikzstyle`.*

```
\tikzstyle{estilo_um}=[fill=red!40!yellow,  
→ draw=red!70!yellow, thick]  
\begin{tikzpicture}  
  \draw[estilo_um] (0,0) -- (4,0) -- (4,2) -- cycle;  
  \draw[estilo_um] (0,3) rectangle ++(4,1);  
\end{tikzpicture}
```



Observação 24 *Ao adicionar “++” ao lado de um ponto, o ponto anterior é somado ao ponto atual. No Exemplo 62, o ponto inicial do retângulo é $(0,3)$ e o ponto final é $(0,3) + (4,1) = (4,4)$.*

2.10 O comando `clip`

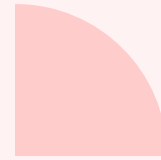
O comando `clip` recorta uma figura. Assim como o comando `path`, o comando `clip` apenas traça um caminho para recorte, sem criar um preenchimento ou uma borda. A sintaxe do comando `clip` é semelhante a dos comandos `draw` e `fill`.

Exemplo 63 *1/4 de círculo criado a partir de um recorte.*

```

\begin{tikzpicture}
  \clip (0,0) rectangle (2,2);
  \fill[red!20] (0,0) circle[radius=2cm];
\end{tikzpicture}

```



2.11 O ambiente scope

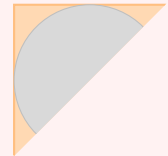
O ambiente `scope` serve como um delimitador utilizado dentro do ambiente `tikzpicture`. O comando `clip`, por exemplo, atua sobre todos os objetos criados depois dele. Assim, o ambiente `scope` delimita a ação do comando `clip`.

Exemplo 64 *Recortando um quadrado e um círculo com o comando clip.*

```

\begin{tikzpicture}
\clip[rotate=45] (0,0) rectangle (3,3);
\draw[fill=orange!30, draw=orange!50, thick] (0,0)
  \to rectangle (2,2);
\draw[fill=gray!30, draw=gray!50] (1,1)
  \to circle[radius=1cm];
\end{tikzpicture}

```

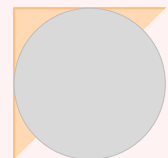


Exemplo 65 *Recortando um quadrado com o comando clip, utilizando também o ambiente scope.*

```

\begin{tikzpicture}
\begin{scope}
\clip[rotate=45] (0,0) rectangle (3,3);
\draw[fill=orange!30, draw=orange!50, thick] (0,0)
  \to rectangle (2,2);
\end{scope}
\draw[fill=gray!30, draw=gray!50] (1,1)
  \to circle[radius=1cm];
\end{tikzpicture}

```



Capítulo 3

Representação gráfica de figuras bidimensionais e tridimensionais

Neste capítulo serão abordadas duas formas de criar uma representação gráfica bidimensional e uma para criar uma representação gráfica tridimensional. A primeira delas é utilizando o pacote `tikz` e o comando `plot`. A outra forma é por meio do pacote `pgfplots`, que utiliza o ambiente `axis` e o comando `addplot`. Além de gráficos em duas dimensões, será trabalhada uma forma de criar gráficos tridimensionais, utilizando o comando `addplot3` do pacote `pgfplots`.

3.1 O comando `plot`

Para plotar um gráfico com o TikZ é necessário preparar o local, isto porque o comando `plot` apenas insere o gráfico. Utilizando os comandos e parâmetros do Capítulo 2 é possível criar os eixos x e y e nomeá-los, criar uma malha e identificar as marcações dos eixos. Após isso, é criado o gráfico da função.

Exemplo 66 *Com o comando `plot`, construa o gráfico da função definida por $f(x) = 0.3x^3$, no intervalo $I = \{x, y \in \mathbb{R}; -3 \leq x \leq 3 \text{ e } -3 \leq y \leq 3\}$.*

O passo a passo é o seguinte:

1. Criar uma malha com auxílio do comando `draw` e do parâmetro `help lines`;
2. Traçar os eixos x e y e utilizar o parâmetro `thick`;
3. Inserir as setas indicando o sentido positivo dos eixos;
4. Identificar os eixos utilizando o comando `node` e posicionar a identificação;
5. Identificar as marcações dos pontos nos eixos.

```

\begin{tikzpicture}
\draw[help lines] (-2.99,-2.99) grid
→ (2.99,2.99);
\draw[thick, ->] (-3,0) --(3.2,0)
→ node[right]{$x$};
\draw[thick, ->] (0,-3)--(0,3.2)
→ node[above]{$y$};

\draw (1,2pt) --(1,-2pt) node[above]{$1$};
\draw (2,2pt) --(2,-2pt) node[above]{$2$};
\draw (3,2pt) --(3,-2pt) node[above]{$3$};

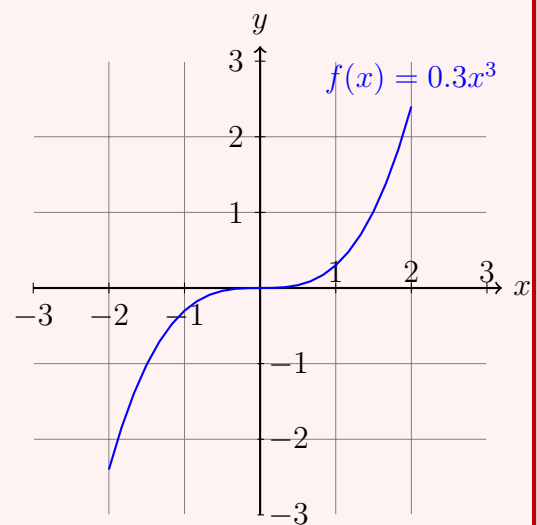
\draw (-1,2pt) --(-1,-2pt)
→ node[below]{$-1$};
\draw (-2,2pt) --(-2,-2pt)
→ node[below]{$-2$};
\draw (-3,2pt) --(-3,-2pt)
→ node[below]{$-3$};

\draw (2pt,1) --(-2pt,1) node[left]{$1$};
\draw (2pt,2) --(-2pt,2) node[left]{$2$};
\draw (2pt,3) --(-2pt,3) node[left]{$3$};

\draw (1.5pt,-1) --(-1pt,-1)
→ node[right]{$-1$};
\draw (1.5pt,-2) --(-1pt,-2)
→ node[right]{$-2$};
\draw (1.5pt,-3) --(-1pt,-3)
→ node[right]{$-3$};

\draw[color=blue, thick, domain=-2:2] plot
→ (\x,0.3*\x^3)
→ node[above]{$f(x)=0.3x^3$};
\end{tikzpicture}

```



O comando `samples` (amostras, em português) utilizado no Exemplo 66 aumenta o número de “segmentos” no gráfico. O \LaTeX não calcula a imagem para todos os pontos no intervalo definido. Por esse motivo, funções que possuem curvaturas (como é o caso de $f(x) = \sin(x)$) podem aparecer como um conjunto de linhas poligonais. Ao aumentar o número de *samples*, essas linhas poligonais ficam cada vez menores e, por consequência, o gráfico da função apresenta

um traço mais suave.

A vantagem do comando `plot` é o alto nível de personalização, visto que cada elemento pode ser definido de forma manual e precisa. Por outro lado, caso seja necessário criar um gráfico simples, é necessário muito tempo e muitas linhas de comando para realizar esta ação. A próxima seção apresenta o `pgfplots`, um pacote que simplifica o processo realizado no Exemplo 66.

3.2 O pacote `pgfplots`

O pacote `pgfplots` é uma ferramenta que desenha gráficos de alta qualidade utilizando comandos simples. O gráfico do Exemplo 66 utilizou 18 linhas de comando para ser criado. Ao utilizar o pacote `pgfplots`, o mesmo gráfico utiliza apenas 5 linhas de comando.

```
\usepackage{pgfplots}
```

Ainda, é preciso identificar a versão do `pgfplots` que será utilizada.

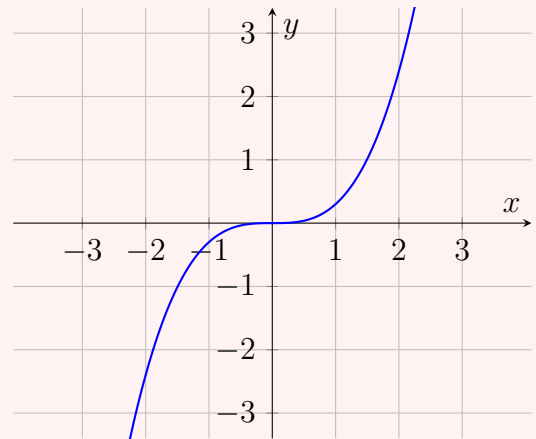
```
\usepackage{newest}
```

Exemplo 67 *Gráfico da função definida por $f(x) = 0.3x^3$ utilizando comandos do pacote `pgfplots`. O resultado é semelhante ao Exemplo 66.*

```

\begin{tikzpicture}
\begin{axis}[
  axis lines=middle,
  xmin=-3, xmax=3,
  ymin=-3.4, ymax=3.4,
  xtick={-3,...,3},
  ytick={-3,...,3},
  xlabel=$x$,
  ylabel=$y$,
  grid=major,
  axis equal=true,
]
\addplot[
  domain=-3:3,
  samples=200,
  blue,
  thick]{0.3*(x^3)};
\end{axis}
\end{tikzpicture}

```



3.2.1 O ambiente axis

O ambiente `axis` cria a estrutura do plano cartesiano, contendo os eixos x e y . Esse ambiente possui vários parâmetros que definem o “plano de fundo” do gráfico, como o seu intervalo e o posicionamento dos eixos. Ele é utilizado dentro do ambiente `tikzpicture`.

```

\begin{tikzpicture}
  \begin{axis}[<parâmetros=opções>]
    <...>
  \end{axis}
\end{tikzpicture}

```

Os parâmetros mais utilizados são apresentados na Tabela 3.1.

Tabela 3.1: Parâmetros do ambiente `axis`.

parâmetro	função	opções
<code>axis lines</code>	posicionamento dos eixos	<code>top, middle, center, bottom, none</code>
<code>axis x line</code>	posicionamento do eixo x	<code>top, middle, center, bottom, none</code>
<code>axis y line</code>	posicionamento do eixo y	<code>top, middle, center, bottom, none</code>
<code>xmin, ymin, zmin</code>	intervalo mínimo do gráfico	qualquer valor real
<code>xmax, ymax, zmax</code>	intervalo máximo do gráfico	qualquer valor real
<code>xtick, ytick</code>	marcação dos n^{os} nos eixos	intervalo real
<code>xlabel, ylabel</code>	nomeia os eixos	qualquer texto
<code>grid</code>	cria uma malha	<code>minor, major, both, none</code>
<code>axis equal</code>	igual a escala entre os eixos	<code>true, false</code>

3.2.2 Gráficos bidimensionais com o comando `addplot`

O comando `addplot` é inserido dentro do ambiente `axis`. Em tradução livre, o termo *add plot* significa “adicionar gráfico”. Ou seja, é o comando que interpreta e cria o gráfico da função desejada.

```
\addplot[<parâmetros>]{<função f(x)>}
```

Os parâmetros mais utilizados do comando `addplot` são apresentados na Tabela 3.2.

Tabela 3.2: Parâmetros do comando `addplot`.

parâmetro	função	opções
<code>domain</code>	define o domínio da função	qualquer intervalo real
<code>samples</code>	número de elementos do domínio	qualquer valor natural

Além disso, os parâmetros que alteram a espessura da linha, a cor da linha e o tipo de linha da Seção 2.3 podem ser utilizados juntamente ao comando `addplot`.

Observação 25 *Os comandos `xmin` e `xmax` definem o intervalo dos eixos. Já o parâmetro `domain` define o domínio da função.*

Ao digitar a lei da função no parâmetro obrigatório do comando `addplot`, é necessário muito cuidado com a notação utilizada na escrita matemática.

Exemplo 68 *A função $f(x) = 5x^2$ é escrita como `5*x^2`.*

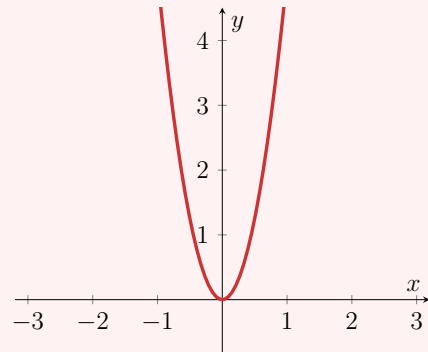
As representações matemáticas mais utilizadas para construção gráfica no \LaTeX são: `+`, `-`, `*`, `/`, `abs`, `sin`, `cos`, `tan`, `atan`, `asin`, `acos`, `cot`, `sec`, `cosec`, `exp`, `ln`, `^`, `factorial`, `pi` e `e`.

Exemplo 69 Gráfico da função definida por $f(x) = 5x^2$.

```

\begin{tikzpicture}[scale=0.8]
\begin{axis}[
axis equal=true,
xmin=-3.2, xmax=3.2,
ymin=-0.5, ymax=4.2,
axis lines=middle,
xlabel=$x$,
ylabel=$y$
]
\addplot[
domain=-3:3,
samples=100,
ultra thick,
red!60!gray
] {5*x^2};
\end{axis}
\end{tikzpicture}

```

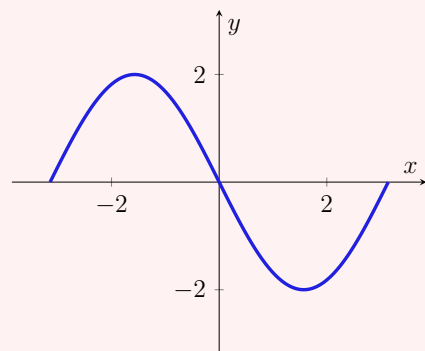


Exemplo 70 Gráfico da função definida por $f(x) = 2 \sin(x + \pi)$.

```

\begin{tikzpicture}[scale=0.8]
\begin{axis}[
axis equal=true,
xmin=-3.2, xmax=3.2,
ymin=-3.2, ymax=3.2,
axis lines=middle,
xlabel=$x$,
ylabel=$y$
]
\addplot[
domain=-3.14159:3.14159,
samples=100,
ultra thick,
blue!75!gray
] {2*sin(deg(x+pi))};
\end{axis}
\end{tikzpicture}

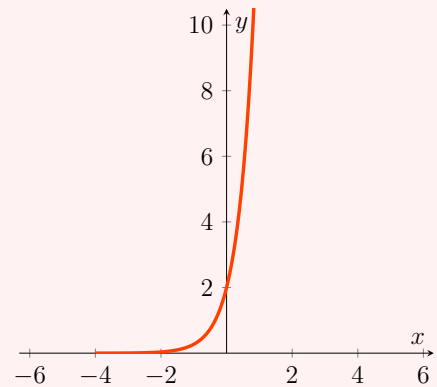
```



Observação 26 O comando `deg` converte o ângulo de radianos para graus.

Exemplo 71 Gráfico da função definida por $f(x) = 2e^{2x}$.

```
\begin{tikzpicture}[scale=0.8]
\begin{axis}[
axis equal=true,
xmin=-4, xmax=4,
ymin=0, ymax=10.5,
axis lines=middle,
xlabel=$x$,
ylabel=$y$
]
\addplot[
domain=-4:3,
samples=100,
ultra thick,
red!75!yellow
] {2*e^(2*x)};
\end{axis}
\end{tikzpicture}
```



3.2.3 Funções definidas por partes

Por padrão, não é possível plotar uma função definida por partes utilizando o pacote `pgfplots`. No entanto, é possível criar um estilo com o comando `pgfplotsset` para utilizar esta função.

```
\pgfplotsset{<nome do estilo>/.style={<parâmetros>}}
```

O parâmetro `nome do estilo` é o nome de chamada. Ele pode ser utilizado como parâmetro opcional do comando `addplot`. Este tópico não será abordado com profundidade neste material. As duas linhas de comando abaixo, quando inseridas no preâmbulo, permitem a criação de intervalos abertos e fechados em uma determinada coordenada. O código abaixo foi adaptado de um fórum de $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ¹.

```
\pgfplotsset{fechado/.style={color=black,only marks,mark=*}}
\pgfplotsset{aberto/.style={color=black,fill=white,only marks,mark=*}}
```

¹Adaptado de: tex.stackexchange.com/questions/76418/plot-non-continuous-function-with-tikz

Exemplo 72 *Gráfico definido por*

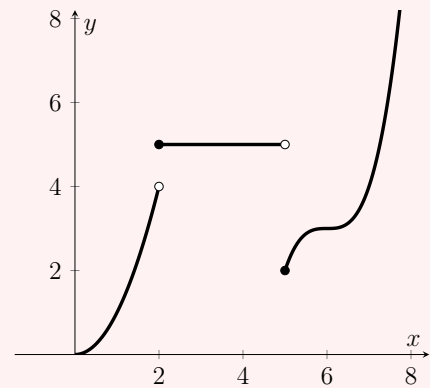
$$f(x) = \begin{cases} x^2, & \text{se } 0 \leq x < 2 \\ 5, & \text{se } 2 \leq x < 5 \\ (x-6)^3 + 3, & \text{se } x \geq 5 \end{cases}$$

```

\begin{tikzpicture}[scale=0.8]
\begin{axis}[
axis equal=true,
xmin=0, xmax=7,
ymin=0, ymax=8.2,
axis lines=middle,
xlabel=$x$,
ylabel=$y$
]
\addplot[
domain=0:2,
samples=100,
ultra thick,
] {x^2};
\addplot[
domain=2:5,
samples=100,
ultra thick,
] {5};
\addplot[
domain=5:8,
samples=100,
ultra thick,
] {(x-6)^3+3};

\addplot[aberto] coordinates {(2,4) (5,5)};
\addplot[fechado] coordinates {(2,5) (5,2)};
\end{axis}
\end{tikzpicture}

```



3.2.4 Superfícies com o comando addplot3

Para plotar uma superfície tridimensional, utiliza-se o comando `addplot3`.

```
\addplot3 [<parâmetros>] {função f(x,y)};
```

A estrutura do `addplot3` é semelhante à estrutura do comando `addplot`. Os parâmetros mais importantes do ambiente `axis`, quando utilizados juntamente com o comando `addplot3`, encontram-se na Tabela 3.3.

Tabela 3.3: Parâmetros do ambiente `axis`.

Parâmetro	Função	Opções
<code>xlabel, ylabel, zlabel</code>	nomeia os eixos	qualquer texto
<code>view/az</code>	ângulo de visão da superfície	ângulo em graus
<code>colormap</code>	tipo do mapa de cores	<code>hot, hot2, jet, blackwhite, greenyellow, bluered, cool, redyellow, violet, viridis</code>

Os parâmetros mais importantes do comando `addplot3` encontram-se na Tabela 3.4.

Tabela 3.4: Parâmetros do comando `addplot3`.

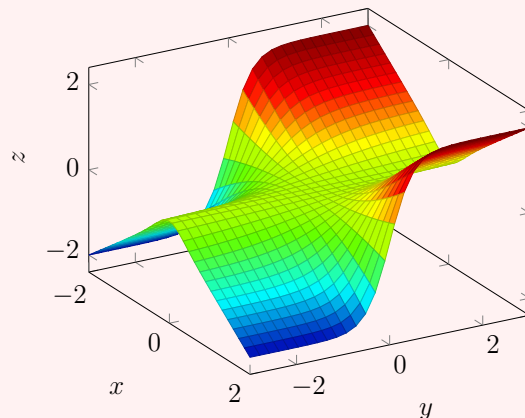
Parâmetro	Função	Opções
<code>domain, y domain</code>	define o domínio do gráfico	qualquer intervalo real
<code>surf</code>	cria a superfície	
<code>samples</code>	número de amostras	qualquer valor natural
<code>draw</code>	cor das linhas da superfície	ver Subseção 2.3.2
<code>shader</code>	shader do gráfico	<code>flat, interp, faceted, flat mean, flat corner, faceted interp</code>

Exemplo 73 Superfície definida por $f(x, y) = x \cdot \tanh(xy)$.

```

\begin{tikzpicture}[scale=0.85]
  \begin{axis}[
    xlabel={x},
    ylabel={y},
    zlabel={z},
    view/az=60,
    samples=30,
    colormap/bluered
  ]
  \addplot3[
    surf,
    domain=-2:2,
    y domain=-3:3
  ]{x*tanh(x*y)};
  \end{axis}
\end{tikzpicture}

```

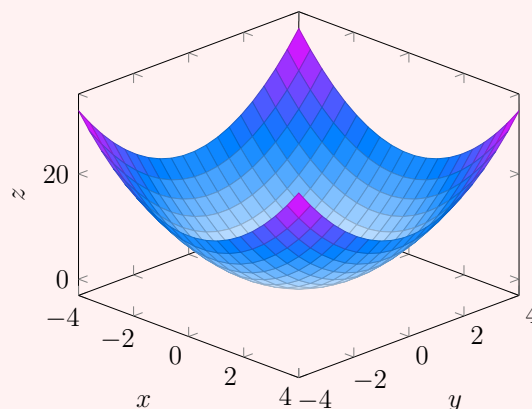


Exemplo 74 Paraboloides definido por $f(x, y) = x^2 + y^2$.

```

\begin{tikzpicture}[scale=0.85]
  \begin{axis}[
    xlabel={x},
    ylabel={y},
    zlabel={z},
    view/az=45,
    samples=20,
    colormap/cool,
  ]
  \addplot3[
    surf,
    domain=-4:4,
    y domain=-4:4
  ]{x^2+ y^2};
  \end{axis}
\end{tikzpicture}

```

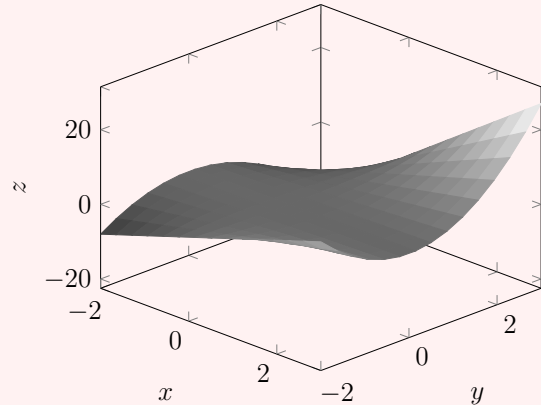


Exemplo 75 Superfície definida por $f(x, y) = xy^2$.

```

\begin{tikzpicture}[scale=0.85]
  \begin{axis}[
    xlabel={x}, ylabel={y},
    zlabel={z}, view/az=45,
    samples=15,
    colormap/blackwhite,
    shader=flat
  ]
  \addplot3[
    surf,
    domain=-2:3,
    y domain=-2:3,
    ]{x*y^2};
  \end{axis}
\end{tikzpicture}

```



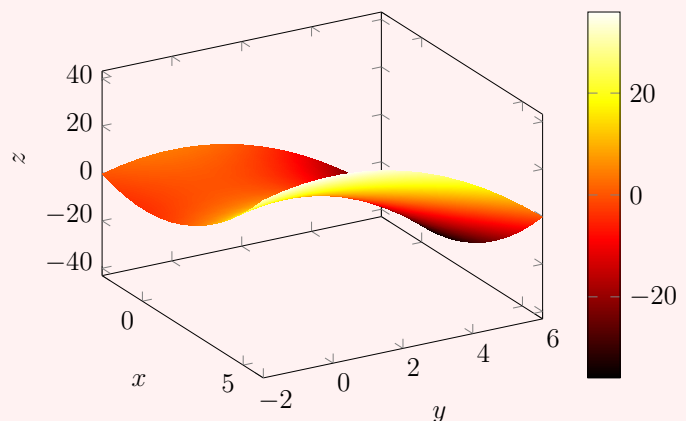
Para adicionar uma barra de cores vertical ao lado da superfície utiliza-se o parâmetro `colorbar` no ambiente `axis`.

Exemplo 76 Parabolóide hiperbólico definido por $f(x, y) = x^2 - y^2$, utilizando uma barra de cores.

```

\begin{tikzpicture}[scale=0.85]
  \begin{axis}[
    xlabel={x}, ylabel={y},
    zlabel={z}, view/az=60,
    samples=15,
    colormap/hot2,
    shader=interp,
    colorbar,
  ]
  \addplot3[
    surf,
    domain=-2:6,
    y domain=-2:6,
    ]{x^2-y^2};
  \end{axis}
\end{tikzpicture}

```



Observação 27 *Para inserir uma barra de cores na parte inferior do gráfico, utiliza-se o parâmetro `colorbar horizontal`.*

Capítulo 4

Mais representações

Neste capítulo é possível encontrar métodos para desenhar objetos matemáticos frequentemente utilizados, como:

- Polígonos regulares;
- Retas paralelas interceptadas por uma transversal;
- Relação entre conjuntos;
- Diagramas de Venn.

Para alguns destes desenhos são necessárias bibliotecas. As bibliotecas introduzem ferramentas novas que facilitam alguns procedimentos que dariam muito trabalho se feitos passo a passo. Algumas bibliotecas que serão abordadas neste capítulo são: `angles`, `intersections` e `babel`.

4.1 Polígonos Regulares

Esta seção apresenta dois exemplos específicos de como desenhar polígonos regulares (pentágono e hexágono). Em seguida, apresenta uma generalização muito útil para desenhar um polígono regular qualquer de n lados.

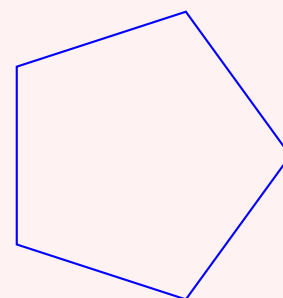
4.1.1 Pentágono regular

Para desenhar um pentágono regular definem-se as coordenadas de seus vértices. Isso pode ser feito utilizando a medida do ângulo central desse polígono (nesse caso, 72°) e coordenadas polares. Após isso, utilizando o comando `draw`, é feita a ligação de todos os vértices adjacentes.

Exemplo 77 *Pentágono regular de lado 2cm.*

```
\begin{tikzpicture}[scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (72:1);
  \coordinate (C) at (144:1);
  \coordinate (D) at (216:1);
  \coordinate (E) at (288:1);

  \draw[thick, blue] (A) -- (B) -- (C) -- (D) -- (E)
  \to -- cycle;
\end{tikzpicture}
```



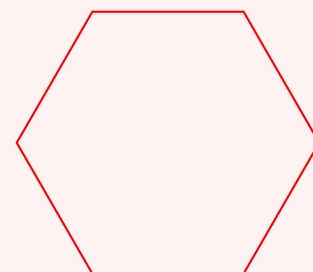
4.1.2 Hexágono regular

Da mesma forma que na Subseção 4.1.1, utiliza-se a medida do ângulo central do hexágono, que é 60° .

Exemplo 78 *Hexágono regular de lado 2cm.*

```
\begin{tikzpicture}[scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (60:1);
  \coordinate (C) at (120:1);
  \coordinate (D) at (180:1);
  \coordinate (E) at (240:1);
  \coordinate (F) at (300:1);

  \draw[thick, red] (A) -- (B) -- (C) -- (D) -- (E) --
  \to (F) -- cycle;
\end{tikzpicture}
```



4.1.3 Polígono regular de n lados

É possível criar uma generalização para desenhar polígonos regulares. No caso do pentágono, existe uma variação de 72° nas coordenadas polares. No caso do hexágono, essa variação é de 60° . Além disso, o número de vértices é igual ao número de lados do polígono.

Exemplo 79 *Quadrado de lado 2cm.*

```
\begin{tikzpicture}[rotate=45, scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (90:1);
  \coordinate (C) at (180:1);
  \coordinate (D) at (270:1);

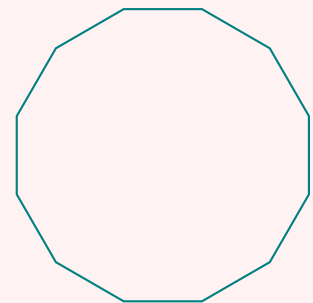
  \draw[thick, red!30!yellow] (A) -- (B) -- (C) -- (D)
  \to -- cycle;
\end{tikzpicture}
```



Exemplo 80 *Dodecágono regular de lado 2cm.*

```
\begin{tikzpicture}[rotate=45, scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (30:1);
  \coordinate (C) at (60:1);
  \coordinate (D) at (90:1);
  \coordinate (E) at (120:1);
  \coordinate (F) at (150:1);
  \coordinate (G) at (180:1);
  \coordinate (H) at (210:1);
  \coordinate (I) at (240:1);
  \coordinate (J) at (270:1);
  \coordinate (K) at (300:1);
  \coordinate (L) at (330:1);

  \draw[thick, blue!50!green] (A) -- (B) -- (C) --
  \to (D) -- (E) -- (F) -- (G) -- (H) -- (I) -- (J)
  \to -- (K) -- (L) -- cycle;
\end{tikzpicture}
```



Generalizando:

$$4 \cdot 90^\circ = 360^\circ$$

$$5 \cdot 72^\circ = 360^\circ$$

$$6 \cdot 60^\circ = 360^\circ$$

⋮

$$n \cdot a_c = 360^\circ$$

onde a_c é o ângulo central do polígono regular. Isolando a_c ,

$$a_c = \frac{360^\circ}{n}$$

Desse modo, para desenhar um polígono de n lados, utiliza-se o caso geral disposto no quadro abaixo.

```
\begin{tikzpicture}
  \coordinate (A) at ((360/n)*0:1);
  \coordinate (B) at ((360/n)*1:1);
  \coordinate (C) at ((360/n)*2:1);
  \coordinate (D) at ((360/n)*3:1);

  <...>

  \coordinate (Z) at ((360/n)*(n-1):1);
  \draw (A) -- (B) -- ... -- (Z) -- cycle;
\end{tikzpicture}
```

4.1.4 Ângulos e vértices

Para identificar os vértices, recomenda-se a utilização de nós (comando `node`). O posicionamento dos nós pode ser feito com os parâmetros `above`, `below`, `left` e `right` e suas combinações. Para alterar o tamanho do texto posicionado neste nó, utilizam-se os comandos `tiny`, `scriptsize`, `footnotesize`, `small`, `large`, `Large`, `LARGE`, `huge` e `HUGE`.

Para identificar um ângulo reto, cria-se um pequeno quadrado a partir de um dos vértices. Isso pode ser feito utilizando os comandos `draw` e `rectangle`. Ainda, para representar o pequeno círculo utilizado em ângulos retos, utilizam-se os comandos `draw` e `circle`.

Exemplo 81 *Quadrado do Exemplo 79 com os vértices e dois ângulos retos identificados.*

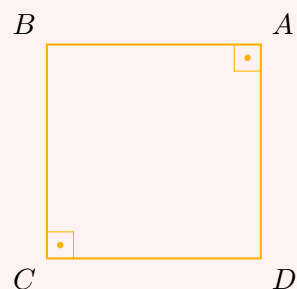
```
\begin{tikzpicture}[rotate=45, scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (90:1);
  \coordinate (C) at (180:1);
  \coordinate (D) at (270:1);

  \draw[thick, red!30!yellow] (A) -- (B) -- (C)
  → -- (D) -- cycle;

  \node at (A) [above right]{\small{$A$}};
  \node at (B) [above left]{\small{$B$}};
  \node at (C) [below left]{\small{$C$}};
  \node at (D) [below right]{\small{$D$}};

  \draw[thin, red!30!yellow, rotate=45] (A)
  → rectangle ++(-5pt, 5pt);
  \draw[thin, fill=red!30!yellow,
  → draw=red!30!yellow, rotate=45] (A) ++
  → (-2.5pt, 2.5pt) circle[radius=0.5pt];

  \draw[thin, red!30!yellow, rotate=45] (C)
  → rectangle ++(5pt, -5pt);
  \draw[thin, fill=red!30!yellow,
  → draw=red!30!yellow, rotate=45] (C) ++
  → (2.5pt, -2.5pt) circle[radius=0.5pt];
\end{tikzpicture}
```



4.1.5 A biblioteca angles

O pacote TikZ possui uma biblioteca que facilita a inserção de ângulos em um desenho.

```
\usetikzlibrary{angles}
```

Para definir a representação de um ângulo, utiliza-se a seguinte linha de comando:

```
\draw pic[<parâmetros opcionais>]{angle=<coordenadas>};
```

Os parâmetros opcionais mais importantes se encontram na Tabela 4.1.

Tabela 4.1: Parâmetros da biblioteca `angles`.

parâmetro	função	opções
<code>angle radius</code>	medida do raio do ângulo	qualquer medida real
<code>"<texto>"</code>	nomenclatura do ângulo	qualquer texto
<code>angle eccentricity</code>	distância de posicionamento do texto	qualquer valor real

Exemplo 82 *Hexágono regular do Exemplo 78 com os vértices e ângulos identificados.*

```

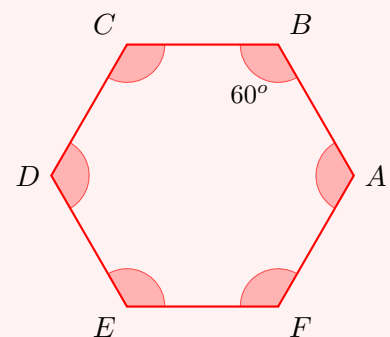
\begin{tikzpicture}[scale=2]
  \coordinate (A) at (0:1);
  \coordinate (B) at (60:1);
  \coordinate (C) at (120:1);
  \coordinate (D) at (180:1);
  \coordinate (E) at (240:1);
  \coordinate (F) at (300:1);

  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm, "\footnotesize{\$60^o\$}", angle
  ↪ eccentricity=1.5]{angle=C--B--A};
  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm]{angle=B--A--F};
  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm]{angle=A--F--E};
  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm]{angle=F--E--D};
  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm]{angle=E--D--C};
  \draw pic[draw=red!70, fill=red!30, angle
  ↪ radius=0.5cm]{angle=D--C--B};

  \draw[thick,red] (A) -- (B) -- (C) -- (D) --
  ↪ (E) -- (F) -- cycle;

  \node at (A) [right]{\small{\$A\$}};
  \node at (B) [above right]{\small{\$B\$}};
  \node at (C) [above left]{\small{\$C\$}};
  \node at (D) [left]{\small{\$D\$}};
  \node at (E) [below left]{\small{\$E\$}};
  \node at (F) [below right]{\small{\$F\$}};
\end{tikzpicture}

```



Observação 28 *Recomenda-se a utilização da biblioteca `babel` ao utilizar o parâmetro que nomeia os ângulos.*

```
\usetikzlibrary{babel}
```

4.2 Retas paralelas interceptadas por uma transversal

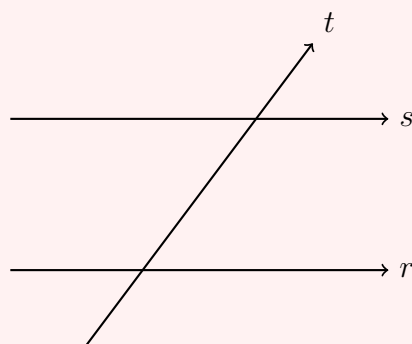
Em primeiro lugar, traça-se duas retas paralelas entre si utilizando o comando `draw` e os parâmetros `thick` e `->`. A seguir, nomeia-se estas retas utilizando o comando `node`, posicionando os nós à direita. Agora, cria-se um terceira reta que intercepte as outras duas, com os mesmos parâmetros, nomeando-a com o comando `node`.

Exemplo 83 *Duas retas r e s paralelas entre si e uma transversal t .*

```
\begin{tikzpicture}
  \draw[thick, ->] (0,0) -- (5,0);
  \draw[thick, ->] (0,2) -- (5,2);

  \node at (5,0)[right]{$r$};
  \node at (5,2)[right]{$s$};

  \draw[thick,->] (1,-1) -- (4,3);
  \node at (4,3)[above right]{$t$};
\end{tikzpicture}
```



4.2.1 A biblioteca intersections

A biblioteca `intersections` permite criar um ponto de interseção entre duas retas concorrentes de forma simples, sem a necessidade de fazer isto manualmente.

```
\usetikzlibrary{intersections}
```

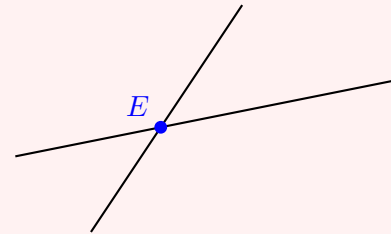
Em primeiro lugar, é necessário nomear o segmento desejado com o parâmetro `name path`. Este parâmetro pode ser utilizado dentro dos comandos `draw`, `fill` e `path`. Após ter nomeado dois segmentos, insere-se outro parâmetro `name intersections`. A estrutura necessária se encontra no Exemplo 84.

Exemplo 84 *Ponto de interseção E entre dois segmentos quaisquer.*

```
\begin{tikzpicture}
  \coordinate (A) at (0,0);
  \coordinate (B) at (2,3);
  \coordinate (C) at (4,2);
  \coordinate (D) at (-1,1);

  \draw[thick, name path=segmento1] (A) -- (B);
  \draw[thick, name path=segmento2] (C) -- (D);

  \draw[name intersections={of=segmento1 and
  ↪ segmento2, by=E}];
  \draw[thick, blue, fill=blue] (E) node[above
  ↪ left]{\small{ $E$ }} circle[radius=2pt];
\end{tikzpicture}
```



Exemplo 85 *Duas retas r e s , paralelas entre si, e uma transversal t com um dos ângulos identificados.*


```

\begin{tikzpicture}
  \coordinate (A) at (0,0);
  \coordinate (B) at (5,0);
  \coordinate (C) at (0,2);
  \coordinate (D) at (5,2);
  \coordinate (E) at (1,-1);
  \coordinate (F) at (4,3);

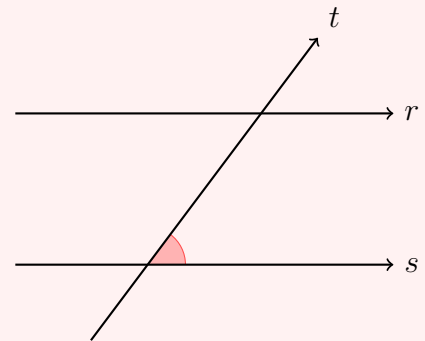
  \path[name path=reta_s] (A) -- (B);
  \path[name path=reta_r] (C) -- (D);
  \path[name path=reta_t] (E) -- (F);

  \draw[name intersections={of=reta_t and
  ↪ reta_s, by=I}];
  \draw pic[draw=red!70,
  ↪ fill=red!30]{angle=B--I--F};

  \draw[thick, ->] (A) -- (B);
  \draw[thick, ->] (C) -- (D);
  \draw[thick,->] (E) -- (F);

  \node at (B)[right]{$s$};
  \node at (D)[right]{$r$};
  \node at (F)[above right]{$t$};
\end{tikzpicture}

```



Exemplo 86 Três retas r , s e t , paralelas entre si, e uma transversal u . Os ângulos foram identificados utilizando dois estilos personalizados.

```

\tikzstyle{vermelho}=[draw=red!70, fill=red!30]
\tikzstyle{azul}=[draw=blue!70, fill=blue!30]
\begin{tikzpicture}
  \coordinate (A) at (0,0);
  \coordinate (B) at (5,0);
  \coordinate (C) at (0,1);
  \coordinate (D) at (5,1);
  \coordinate (E) at (0,2);
  \coordinate (F) at (5,2);
  \coordinate (G) at (1,-1);
  \coordinate (H) at (3.5,3.2);

  \path[name path=reta_t] (A) -- (B);
  \path[name path=reta_s] (C) -- (D);
  \path[name path=reta_r] (E) -- (F);
  \path[name path=reta_u] (G) -- (H);

  \draw[name intersections={of=reta_r and
  ↪ reta_u, by=r_u}];
  \draw[name intersections={of=reta_s and
  ↪ reta_u, by=s_u}];
  \draw[name intersections={of=reta_t and
  ↪ reta_u, by=t_u}];

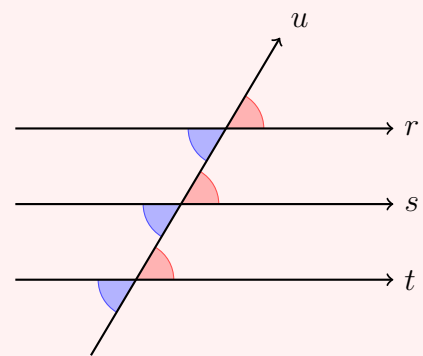
  \draw pic[vermelho]{angle=B--t_u--H};
  \draw pic[vermelho]{angle=D--s_u--H};
  \draw pic[vermelho]{angle=F--r_u--H};

  \draw pic[azul]{angle=A--t_u--G};
  \draw pic[azul]{angle=C--s_u--G};
  \draw pic[azul]{angle=E--r_u--G};

  \draw[thick, ->] (A) -- (B);
  \draw[thick, ->] (C) -- (D);
  \draw[thick, ->] (E) -- (F);
  \draw[thick, ->] (G) -- (H);

  \node at (B) [right] {\$t\$};
  \node at (D) [right] {\$s\$};
  \node at (F) [right] {\$r\$};
  \node at (H) [above right] {\$u\$};
\end{tikzpicture}

```



Observação 29 No caso do Exemplo 86 foi necessário definir os caminhos das retas r , s e t com o comando `path`, e só no final foi utilizado o comando `draw` para desenhá-las. Isso acontece porque o TikZ insere as figuras de modo ordenado, de cima para baixo. Nesse sentido, o desenho do ângulo deve ficar atrás do desenho das retas por questões estéticas.

4.3 Relação entre conjuntos

Para representar a relação entre dois conjuntos A e B utilizando o `tikz` desenham-se duas elipses com o comando `draw`. A seguir, nomeiam-se os elementos de cada conjunto utilizando o parâmetro `label` dentro do comando `node`. Ainda neste comando criam-se nomes para as coordenadas dos elementos. Por fim, criam-se as setas que relacionam os dois conjuntos utilizando o comando `draw`.

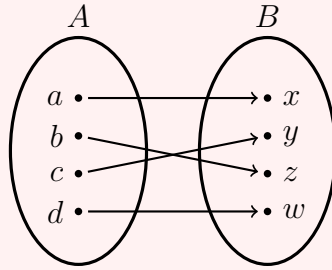
Exemplo 87 Relação entre dois conjuntos A e B , cada um com quatro elementos.

```

\begin{tikzpicture}
\tikzstyle{conjuntos}=[very thick]
\tikzstyle{elementos}=[fill=black, circle, inner sep=1pt]

\node[above] at (0, 1.5){$A$};
\node[above] at (2.5, 1.5){$B$};
\draw[conjuntos] (0,0) circle[x radius=0.9, y radius=1.5];
\draw[conjuntos] (2.5,0) circle[x radius=0.9, y radius=1.5];
\node[elementos, label=left:$a$] (A1) at (0,0.7) {};
\node[elementos, label=left:$b$] (A2) at (0,0.2) {};
\node[elementos, label=left:$c$] (A3) at (0,-0.3) {};
\node[elementos, label=left:$d$] (A4) at (0,-0.8) {};
\node[elementos, label=right:$x$] (B1) at (2.5,0.7) {};
\node[elementos, label=right:$y$] (B2) at (2.5,0.2) {};
\node[elementos, label=right:$z$] (B3) at (2.5,-0.3) {};
\node[elementos, label=right:$w$] (B4) at (2.5,-0.8) {};
\draw[->,thick,shorten <=2pt, shorten >=2pt] (A1) -- (B1);
\draw[->,thick,shorten <=2pt, shorten >=2pt] (A2) -- (B3);
\draw[->,thick,shorten <=2pt, shorten >=2pt] (A3) -- (B2);
\draw[->,thick,shorten <=2pt, shorten >=2pt] (A4) -- (B4);
\end{tikzpicture}

```



4.4 Diagramas de Venn

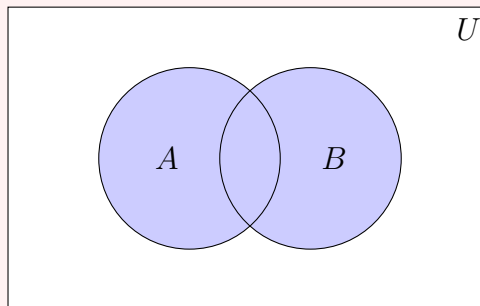
Para criar Diagramas de Venn com o `tikz` utilizam-se os comandos `clip`, `draw` e `fill` e os ambientes `tikzpicture` e `scope`. Uma técnica muito utilizada para desenhar os Diagramas de Venn é desenhar separadamente a borda e o preenchimento da figura.

Exemplo 88 *União entre dois conjuntos ($A \cup B$).*

```
\begin{tikzpicture}[scale=0.8]
  \fill[white] (0,0) rectangle (8,5);
  \draw (0,0) rectangle (8,5) node[below left]{$U$};

  \fill[blue!20] (3,2.5) circle[radius=1.5] node[left]{$A$};
  \fill[blue!20] (5,2.5) circle[radius=1.5] node[right]{$B$};

  \draw[thin] (3,2.5) circle[radius=1.5] node[left]{$A$};
  \draw[thin] (5,2.5) circle[radius=1.5] node[right]{$B$};
\end{tikzpicture}
```



Exemplo 89 *Interseção entre dois conjuntos ($A \cap B$).*

```
{ }
\begin{tikzpicture}[scale=0.8]
  \fill[white] (0,0) rectangle (8,5);
```

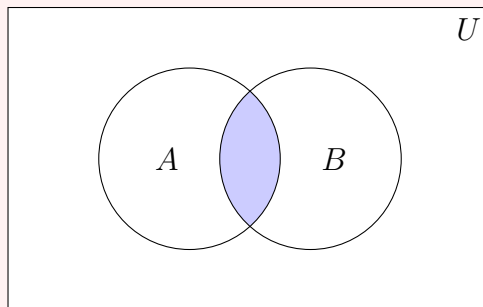
```

\draw (0,0) rectangle (8,5) node[below left]{$U$};

\begin{scope}
  \clip (3,2.5) circle[radius=1.5];
  \fill[blue!20] (5,2.5) circle[radius=1.5];
\end{scope}

\draw[thin] (3,2.5) circle[radius=1.5] node[left]{$A$};
\draw[thin] (5,2.5) circle[radius=1.5] node[right]{$B$};
\end{tikzpicture}

```



Exemplo 90 *Complemento de um conjunto (A^c).*

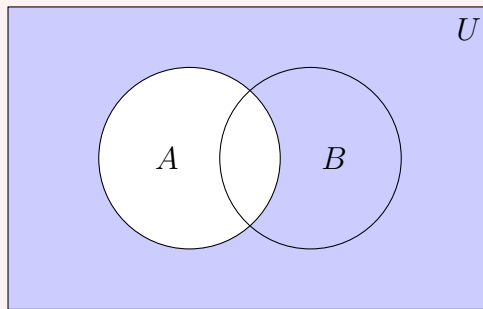
```

{}
\begin{tikzpicture}[scale=0.8]
\fill[blue!20] (0,0) rectangle (8,5);
\draw (0,0) rectangle (8,5) node[below left]{$U$};

\begin{scope}
  \clip (3,2.5) circle[radius=1.5];
  \fill[white] (5,2.5) circle[radius=1.5];
  \fill[white] (3,2.5) circle[radius=1.5];
\end{scope}

\draw[thin] (3,2.5) circle[radius=1.5] node[left]{$A$};
\draw[thin] (5,2.5) circle[radius=1.5] node[right]{$B$};
\end{tikzpicture}

```



4.5 Ciclo Trigonométrico

Para criar um Ciclo Trigonométrico utilizam-se os comandos `draw`, `foreach` e `tikzstyle`. Apesar de ser possível fazer o mesmo procedimento apenas com o comando `draw`, os comandos `foreach` e `tikzstyle` facilitam alguns passos. Em primeiro lugar, definem-se duas cores para os segmentos que ligam o arco ao seu valor nos eixos x e y , respectivamente. Dentro do ambiente `tikzpicture` inicia-se o desenho pela circunferência de raio 1. Em seguida, criam-se os segmentos que ligam o arco aos seus valores em x e y e os eixos $\cos(x)$ e $\sin(x)$. Utilizando o comando `foreach`, criam-se todos os segmentos que ligam a origem aos arcos dos quatro quadrantes, nomeando-os. A seguir, utiliza-se a notação de graus em radianos para nomear quatro ângulos. Por fim, os pontos nos eixos x e y são identificados.

Exemplo 91 *Ciclo Trigonométrico com o primeiro quadrante completo.*

```

\tikzstyle{parax}=[draw=blue!50, thick]
\tikzstyle{paray}=[draw=red!50, thick]

\begin{tikzpicture}[scale=5.5]
\draw[very thick, gray] (0,0) circle[radius=1];

\draw[parax] (30:1cm) -- (0.866,0);
\draw[parax] (45:1cm) -- (0.707,0);
\draw[parax] (60:1cm) -- (0.5,0);

\draw[paray] (30:1cm) -- (0,0.5);
\draw[paray] (45:1cm) -- (0, 0.707);
\draw[paray] (60:1cm) -- (0,0.866);

\draw[->, very thick] (-1.2,0) --(1.2,0);
\draw[->, very thick] (0,-1.2) --(0,1.2);
\draw (1.32,0) node{\$cos(x)\$};

```

```

\draw (0,1.25) node{\$sen(x)};

\foreach \i in {30,45,60,90,120,135,150,180,210,225,240,270,300,315,330}{\draw
→ (0,0) -- (\i:1cm);
\draw (\i:1.12cm) node[fill=red!5!white] {\$i^o};}

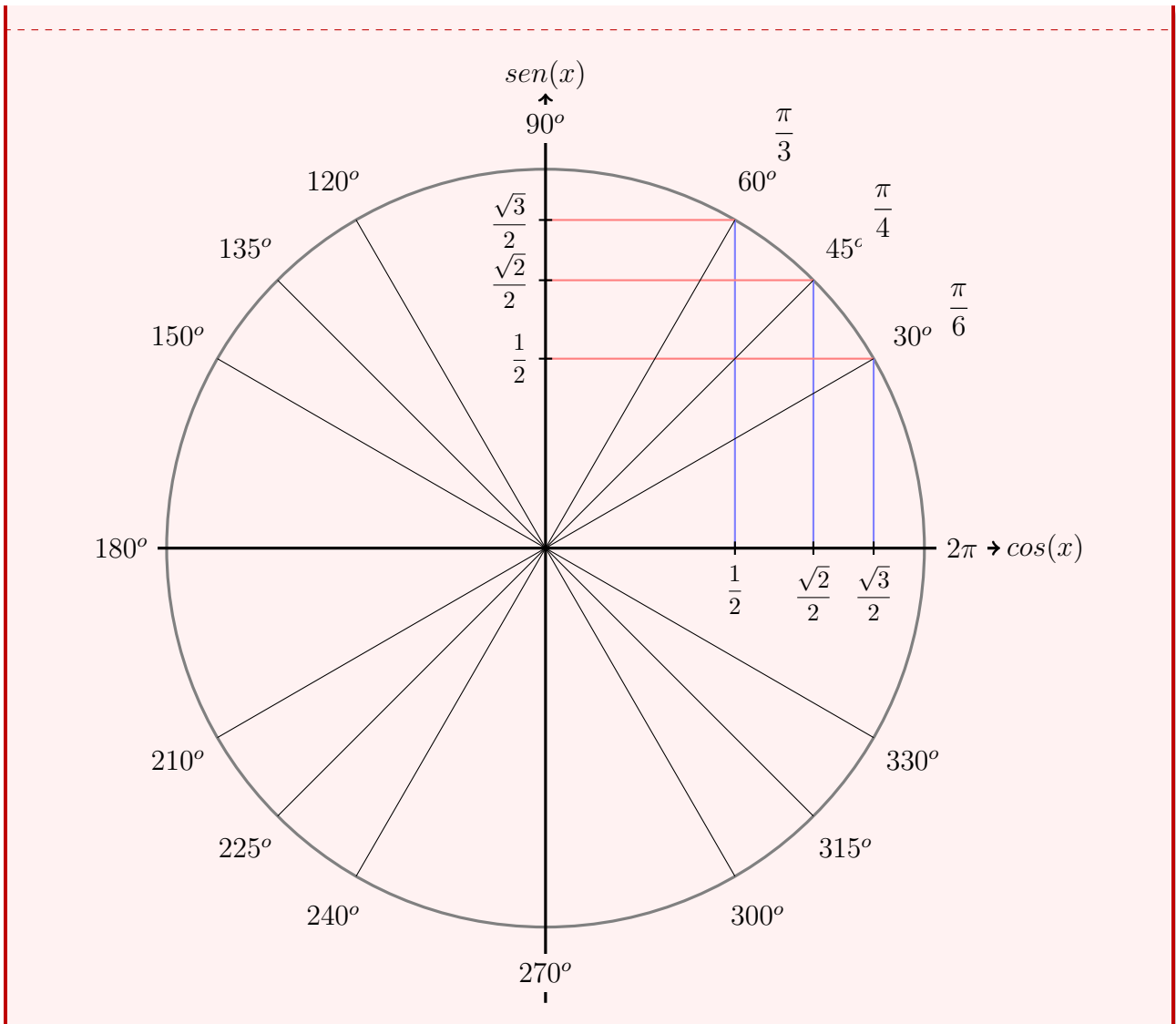
\draw (0:1.1cm) node[fill=red!5!white]{$2\pi$};
\draw (30:1.26cm) node[fill=red!5!white]{$\dfrac{\pi}{6}$};
\draw (45:1.26cm) node[fill=red!5!white]{$\dfrac{\pi}{4}$};
\draw (60:1.26cm) node[fill=red!5!white]{$\dfrac{\pi}{3}$};

\draw[thick] (0.5, -0.5pt) -- (0.5, 0.5pt) ++ (0,-1pt)
→ node[below]{\footnotesize{\dfrac{1}{2}}};
\draw[thick] (0.707, -0.5pt) -- (0.707, 0.5pt) ++ (0,-1pt)
→ node[below]{\footnotesize{\dfrac{\sqrt{2}}{2}}};
\draw[thick] (0.866, -0.5pt) -- (0.866, 0.5pt) ++ (0,-1pt)
→ node[below]{\footnotesize{\dfrac{\sqrt{3}}{2}}};

\draw[thick] (-0.5pt, 0.5) -- (0.5pt, 0.5) ++ (-1pt,0)
→ node[left]{\footnotesize{\dfrac{1}{2}}};
\draw[thick] (-0.5pt, 0.707) -- (0.5pt, 0.707) ++ (-1pt,0)
→ node[left]{\footnotesize{\dfrac{\sqrt{2}}{2}}};
\draw[thick] (-0.5pt, 0.866) -- (0.5pt, 0.866) ++ (-1pt,0)
→ node[left]{\footnotesize{\dfrac{\sqrt{3}}{2}}};

\end{tikzpicture}

```



4.6 Intervalos na Reta

Para delimitar um intervalo na reta (por exemplo, um intervalo $(2, 3]$), inicialmente, criam-se dois estilos. Esses estilos definem a representação gráfica dos intervalos aberto e fechado. Após, indicam-se as coordenadas que serão utilizadas e, com o comando `draw`, cria-se a reta com setas nas extremidades e o segmento que está contido no intervalo indicado. Por fim, desenham-se os intervalos.

Exemplo 92 *Intervalo $(2, 3]$ representado na reta.*

```

\begin{tikzpicture}
\tikzstyle{aberto}=[fill=white, draw=black, circle, inner sep=1.2pt]
\tikzstyle{fechado}=[fill=black, draw=black, circle, inner sep=1.2pt]
\coordinate (A) at (-2,0);
\coordinate (B) at (0,0);
\coordinate (C) at (2,0);

```



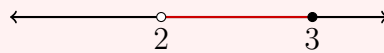
```

\coordinate (D) at (3,0);
\draw[thick, <->] (A)--(B) --(D);
\draw[red!80!black, thick] (B)--(C);

\node at (B)[aberto]{};
\node at (B)[below]{$2$};

\node at (C)[fechado]{};
\node at (C)[below]{$3$};
\end{tikzpicture}

```



Capítulo 5

Pôster com a classe tikzposter

Existem várias formas de criar um pôster científico no L^AT_EX. No entanto, o enfoque deste material será a utilização da classe `tikzposter`. A classe `tikzposter` permite fazer a criação de pôsteres de forma simples. Esta classe utiliza o pacote `tikz` para gerar o *layout* do pôster.

```
\documentclass[<parâmetros opcionais>]{tikzposter}
```

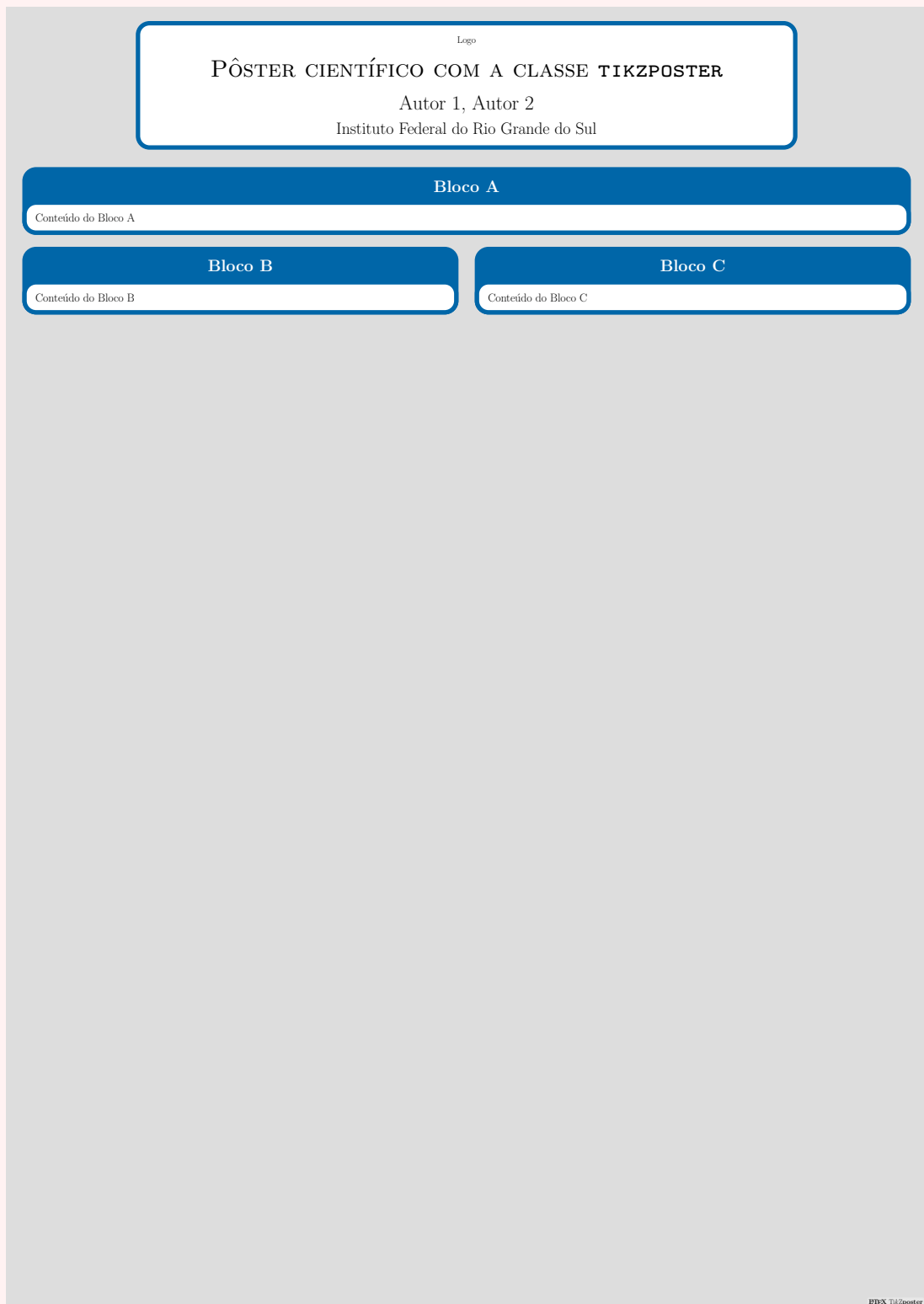
Os parâmetros opcionais serão abordados na Seção 5.1. O código-fonte básico para gerar um pôster é apresentado no Exemplo 93.

Exemplo 93 *Código-fonte básico para gerar um pôster utilizando a classe `tikzpicture`.*

```
\documentclass{tikzposter}
\usepackage[utf8]{inputenc}
\usepackage[portuguese]{babel}
\title{Pôster científico com a classe \texttt{tikzposter}}
\institute{Instituto Federal do Rio Grande do Sul}
\author{Autor 1, Autor 2}
\titlegraphic{Logo}
\begin{document}
  \maketitle
  \block{Bloco A}{Conteúdo do Bloco A}

  \begin{columns}
    \column{0.5}
    \block{Bloco B}{Conteúdo do Bloco B}
    \column{0.5}
    \block{Bloco C}{Conteúdo do Bloco C}
  \end{columns}
\end{document}
```

```
\end{document}
```



5.1 Os parâmetros opcionais da classe tikzposter

Os parâmetros opcionais permitem personalizar a orientação e o tamanho da página, as margens e outras configurações do documento. Alguns desses parâmetros são próprios do

L^AT_EX e outros foram criados para ser utilizados com a classe `tikzposter`. Os parâmetros já conhecidos são aqueles que alteram o tamanho da fonte, o tamanho da página e sua orientação.

Tabela 5.1: Parâmetros opcionais do L^AT_EX.

parâmetro	valores
tamanho da fonte	12pt, 14pt, 17pt, 20pt ou 25pt
tamanho da página	a0paper, a1paper ou a2paper
orientação	landscape (paisagem) ou portrait (retrato)

Exemplo 94 *Definindo o tamanho da fonte como 25pt, o tamanho da página como a0paper e a orientação da página como retrato.*

```
\documentclass[25pt, a0paper, portrait]{tikzposter}
```

Os parâmetros opcionais específicos da classe `tikzposter` se referem à configuração de espaçamento dos elementos do pôster.

Tabela 5.2: Parâmetros opcionais da classe `tikzposter`.

parâmetro	função
<code>margin</code>	margem entre a borda externa do pôster e a borda do papel
<code>innermargin</code>	margem entre a borda do pôster e a borda dos blocos
<code>colspace</code>	espaçamento horizontal entre colunas
<code>subcolspace</code>	espaçamento horizontal entre colunas no ambiente <code>subcolumn</code>
<code>blockverticalspace</code>	espaçamento vertical entre os blocos

Para utilizar os parâmetros da Tabela 5.2 é necessário atribuir valores. Já no caso dos parâmetros da Tabela 5.1 os valores são inseridos de forma direta.

Exemplo 95 *Definindo valores para os parâmetros da Tabela 5.2. Os parâmetros do Exemplo 94 foram mantidos.*

```
\documentclass[margin=2cm, colspace=10pt, portrait, margin=0cm,
→ innermargin=15mm, blockverticalspace=15mm, colspace=15mm,
→ subcolspace=8mm]{tikzposter}
```

Observação 30 *Os valores utilizados nos Exemplos 94 e 95 são os valores padrão da classe `tikzposter`.*

5.2 Inserção de dados de identificação

Para identificar as informações essenciais do pôster, como autor, título, instituição e logotipo, utiliza-se um conjunto de comandos no preâmbulo.

```
\title{<Título do documento>}
\institute{<Instituição de ensino>}
\author{<Autores>}
\titlegraphic{<Logotipo>}
```

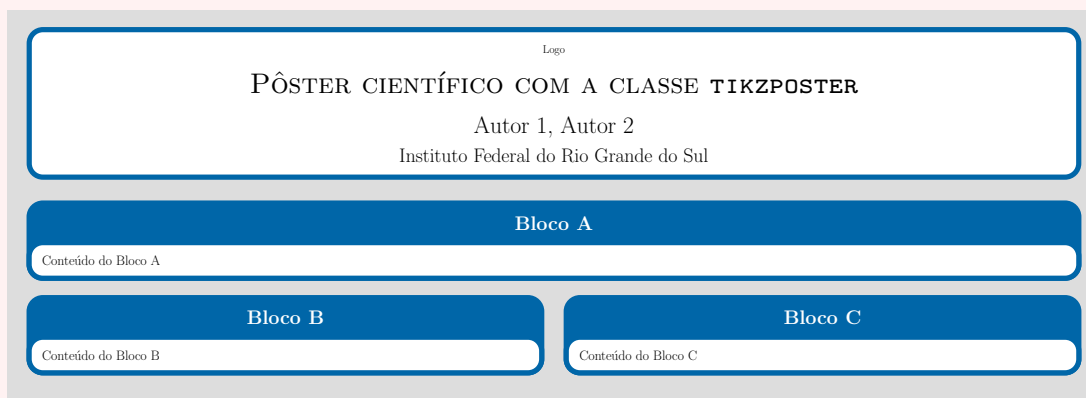
Para inserir o bloco do título utiliza-se o comando `\maketitle` no início do ambiente `document`.

```
\maketitle
```

Para alterar a largura da caixa de título produzida pelo comando `\maketitle`, utiliza-se o parâmetro `width`.

Exemplo 96 *Alterando a largura da caixa de título para 807mm. O código-fonte utilizado aqui é uma adaptação do Exemplo 93.*

```
\maketitle[width=807mm]
```



5.3 O comando block

Os blocos da classe `tikzposter` são semelhantes aos blocos do BEAMER. Para inserir um bloco, utiliza-se o comando `block`.

```
\block[<parâmetros opcionais>]{<título>}{<conteúdo do bloco>}
```

É possível inserir um bloco apenas com título e conteúdo. No entanto, para modificar o posicionamento do título, do conteúdo ou até do bloco em si, é necessário utilizar os parâmetros

opcionais. A Tabela 5.3 apresenta esses parâmetros.

Tabela 5.3: Parâmetros opcionais do comando `block`.

parâmetro	função	exemplos de valores
<code>titleleft</code>	posiciona o título do bloco à esquerda	não admite
<code>titlecenter</code>	posiciona o título do bloco ao centro	não admite
<code>titleright</code>	posiciona o título do bloco à direita	não admite
<code>titleoffsetx</code>	deslocamento do título do bloco em x	$1cm$, $15pt$
<code>titleoffsety</code>	deslocamento do título do bloco em y	$1cm$, $15pt$
<code>bodyoffsetx</code>	deslocamento do corpo do bloco em x	$1cm$, $15pt$
<code>bodyoffsety</code>	deslocamento do corpo do bloco em y	$1cm$, $15pt$
<code>roundedcorners</code>	raio dos cantos do bloco	$12pt$, $5mm$
<code>linewidth</code>	espessura da linha externa bloco	$3mm$, $1cm$
<code>titleinnersep</code>	adiciona espaço no campo do título	$20pt$, $2cm$
<code>bodyinnersep</code>	adiciona espaço no corpo do bloco	$20pt$, $2cm$

5.4 Separando o texto em colunas

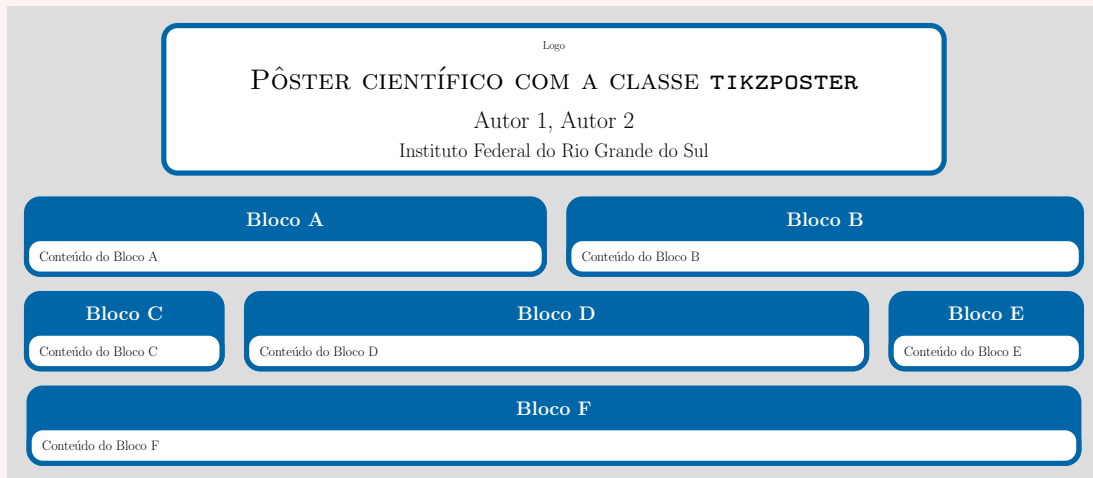
Para separar um texto em duas ou mais colunas utiliza-se o ambiente `columns`. A utilização deste ambiente está na Seção 1.9.

Exemplo 97 *Pôster com blocos em duas e três colunas.*

```
\begin{columns}
  \column{0.5}
  \block{Bloco A}{Conteúdo do Bloco A}
  \column{0.5}
  \block{Bloco B}{Conteúdo do Bloco B}
\end{columns}

\begin{columns}
  \column{0.2}
  \block{Bloco C}{Conteúdo do Bloco C}
  \column{0.6}
  \block{Bloco D}{Conteúdo do Bloco D}
  \column{0.2}
  \block{Bloco E}{Conteúdo do Bloco E}
\end{columns}
```

```
\block{Bloco F}{Conteúdo do Bloco F}
```



5.5 Temas

A classe `tikzposter` possui 9 temas pré-definidos. São eles:

- Default
- Rays
- Basic
- Simple
- Envelope
- Wave
- Board
- Autumn
- Desert

Observação 31 *Todo pôster criado esta classe insere automaticamente um logo na margem inferior. Para removê-lo, insere-se o comando `tikzposterlatexaffectionproofoff` no preâmbulo.*

Capítulo 6

Modelo de pôster

Este capítulo apresenta um modelo de pôster criado utilizando a classe `tikzposter`. O pacote `lipsum` foi utilizado para gerar parágrafos com textos fictícios.

```
\documentclass[blockverticalspace=20mm]{tikzposter}
\usepackage[utf8]{inputenc}
\usepackage[portuguese]{babel}
\usepackage{lipsum}
\usepackage{graphicx}

\title{Modelo de pôster científico utilizando a classe \texttt{tikzposter}}
\author{Luís Henrique Ribeiro da Silva \ \ Greice da Silva Lorenzetti Andreis}
\institute{Instituto Federal do Rio Grande do Sul - \textit{Campus} Caxias do
\to Sul}
\usetheme{Rays}
\tikzposterlatexaffectionproofoff

\begin{document}
  \maketitle[width=807mm]
  \begin{columns}

  \column{0.5}
  \block{Resumo}{
  \lipsum[1]
  }

  \block{Introdução}{
  \lipsum[2]
  }
  }
```



```

\block{Metodologia}{
\lipsum[6]
}

\block{}{
\centering
\textbf{Figura 1}: Consectetuer adipiscing elit.
\includegraphics[scale=2]{dados1} \\
{\small Fonte: Disponível em: pixabay.com (2018).}
}

\column{0.5}

\block{Resultados}{
\lipsum[2]
}

\block{}{
\centering
\textbf{Figura 2}: Aliquam tincidunt urna.
\includegraphics[scale=0.5]{dados3} \\
{\small Fonte: Disponível em: pixabay.com (2018).}
}

\block{Considerações Finais}{
\lipsum[2]
}

\block{Referências}{
\lipsum[1]
}

\end{columns}

\block{}{
\centering
\textbf{Figura 3}: Suspendisse ut massa.
\includegraphics[scale=1.7]{dados2}\\
{\small Fonte: Disponível em: pixabay.com (2018).}
}

```

}

\end{document}

MODELO DE POSTER CIENTÍFICO UTILIZANDO A CLASSE TIKZPOSTER

Luís Henrique Ribeiro da Silva
Greice da Silva Lorenzetti Andreis
Instituto Federal do Rio Grande do Sul - *Campus* Caxias do Sul

Resumo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

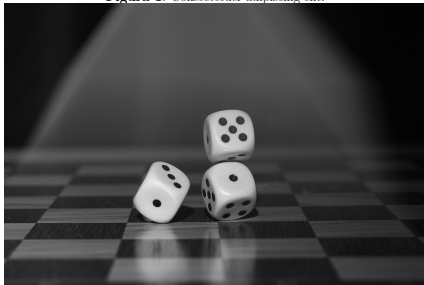
Introdução

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Metodologia

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Figura 1: Consectetur adipiscing elit.



Fonte: Disponível em: pexels.com (2018).

Resultados

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Figura 2: Aliquam tincidunt urna.



Fonte: Disponível em: pexels.com (2018).

Considerações Finais

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Referências

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Figura 3: Suspendisse ut massa.



Fonte: Disponível em: pexels.com (2018).

Capítulo 7

Introdução aos Macros

Ao utilizar macros é possível atingir um alto grau de personalização do documento. Os macros permitem ao usuário a criação de comandos e ambientes que, muitas vezes, tornam o código-fonte menor e mais prático.

7.1 O comando `newcommand`

Para criar comandos dentro do L^AT_EX utiliza-se o comando `newcommand`.

```
\newcommand{<nome>}[<número de argumentos>]{<comando a executar>}
```

O parâmetro `nome` define como esse comando será chamado. O parâmetro `número de argumentos` define o número de parâmetros que serão necessários na execução do comando. E, por último, o parâmetro `comando a executar` define o comando ou os comandos que serão utilizados.

Exemplo 98 *Definindo um comando mais simples para utilizar o símbolo \mathbb{R} dentro do ambiente matemático.*

```
\newcommand{\R}{\mathbb{R}}
```

`$$\R$`

\mathbb{R}

É possível utilizar o comando `newcommand` em qualquer parte do documento. No entanto, recomenda-se utilizá-lo no preâmbulo.

Exemplo 99 Comando utilizando três argumentos.

```
\newcommand{\cadastro}[3]{\textbf{\#1}: \textit{\#2} - \textit{Campus \#3}}
```

```
\cadastro{Vitória}{Instituto Federal do Rio Grande do Sul}{Caxias do Sul} \\
\cadastro{João}{Instituto Federal do Rio Grande do Sul}{Bento Gonçalves} \\
\cadastro{Luís}{Instituto Federal do Rio Grande do Sul}{Osório}
```

Vitória: *Instituto Federal do Rio Grande do Sul - Campus Caxias do Sul.*

João: *Instituto Federal do Rio Grande do Sul - Campus Bento Gonçalves.*

Luís: *Instituto Federal do Rio Grande do Sul - Campus Osório.*

Exemplo 100 Definição de comandos para representação de derivadas de ordem $n = 1, 2, 3, 4, 5$.

```
\newcommand{\dpri}[1]{f'(x)=#1}
\newcommand{\dseg}[1]{f''(x)=#1}
\newcommand{\dter}[1]{f'''(x)=#1}
\newcommand{\dqua}[1]{f^{(4)}(x)=#1}
\newcommand{\dqui}[1]{f^{(5)}(x)=#1}
```

`$$\dpri{x^2}$$`

`$$\dseg{x^2}$$`

`$$\dter{x^2}$$`

`$$\dqua{x^2}$$`

`$$\dqui{x^2}$$`

$$f'(x) = x^2$$

$$f''(x) = x^2$$

$$f'''(x) = x^2$$

$$f^{(4)}(x) = x^2$$

$$f^{(5)}(x) = x^2$$

Exemplo 101 Definindo dois comandos para simplificar a representação de produtos notáveis (quadrado e cubo).

```
\newcommand{\quadrado}[3]{\left(\#1 \#2 \#3\right)^2={\#1}^2 \#2^2 \cdot \#1 \#3 +
\to {\#3}^2}
```

```
\newcommand{\cubo}[3]{\left(\#1 \#2 \#3\right)^3 = {\#1}^3 \#2^3 \cdot {\#1}^2 \cdot
\to \#3 + 3 \cdot \#1 \cdot {\#3}^2 \#2 {\#3}^3}
```

```
$$\quadrado{x}{+}{y}$$
```

```
$$\quadrado{x}{-}{y}$$
```

```
$$\cubo{a}{+}{b}$$
```

```
$$\cubo{a}{-}{b}$$
```

$$(x + y)^2 = x^2 + 2 \cdot xy + y^2$$

$$(x - y)^2 = x^2 - 2 \cdot xy + y^2$$

$$(a + b)^3 = a^3 + 3 \cdot a^2 \cdot b + 3 \cdot a \cdot b^2 + b^3$$

$$(a - b)^3 = a^3 - 3 \cdot a^2 \cdot b + 3 \cdot a \cdot b^2 - b^3$$

7.2 O comando newenvironment

O comando `newenvironment` permite criar novos ambientes.

```
\newenvironment{<nome>}{<início>}{<fim>}
```

O parâmetro `nome` é o nome dado ao ambiente. O parâmetro `início` é o código a ser executado logo no início do ambiente e o parâmetro `fim` é o código a ser executado ao final do ambiente.

Exemplo 102 *Definindo um ambiente que centraliza e transforma em negrito um texto qualquer.*

```
\newenvironment{destaque}{\begin{center} \bf}{\end{center}}
```

```
\begin{destaque}
```

Texto em destaque.

```
\end{destaque}
```

Texto em destaque.

Exemplo 103 *Definindo um ambiente para uma lista de questões não enumeradas.*

```
\newenvironment{questao}{\textbf{Questão}. \it}{}
```

```
\begin{questao}
```

Mostre que $a^2 = a \cdot a$.

```
\end{questao} \\
```

```
\begin{questao}
```

Mostre que $a + (-a) = 0$.

```
\end{questao}
```

Questão. *Mostre que $a^2 = a \cdot a$.*

Questão. *Mostre que $a + (-a) = 0$.*

Capítulo 8

O pacote `tcolorbox`

O pacote `tcolorbox` permite criar caixas coloridas personalizadas. As caixas utilizadas neste material foram criadas com esse pacote. Além de colorir as caixas, é possível adicionar sombras e arredondar os cantos. Existem duas formas de criar essas caixas: com o ambiente `tcolorbox` e o comando `tcbox`. Além disso, é possível criar ambientes para utilizar as caixas personalizadas.

8.1 Introdução

Para criar caixas personalizadas utilizando o `tcolorbox`, é necessário inserir o pacote `tcolorbox` no preâmbulo do documento.

```
\usepackage{tcolorbox}
```

O ambiente `tcolorbox` cria uma caixa simples com cantos arredondados. A sua largura é fixa, definida pelas dimensões das margens.

Exemplo 104 *Caixa utilizando o comando `tcolorbox`.*

```
\begin{tcolorbox}
Conteúdo da caixa.
\end{tcolorbox}
```

Conteúdo da caixa.

Além disso, é possível separar o conteúdo da caixa horizontalmente, dividindo-a em duas partes. Para isso, utiliza-se o comando `tcblower`. Este comando cria uma linha pontilhada.

Exemplo 105 *Caixa utilizando o comando `tcblower`.*

```
\begin{tcolorbox}
Conteúdo da caixa - \textbf{superior}.
\tcblower
Conteúdo da caixa - \textbf{inferior}.
\end{tcolorbox}
```

Conteúdo da caixa - **superior**.

Conteúdo da caixa - **inferior**.

O comando `tcbox` cria uma caixa que se ajusta conforme seu conteúdo.

Exemplo 106 *Caixa utilizando o comando `tcbox`.*

```
\tcbox{Conteúdo da caixa.}
```

Conteúdo da caixa.

8.2 Personalizando as caixas

O ambiente `tcolorbox` e o comando `tcbox` são personalizáveis a partir de um parâmetro opcional.

```
\begin{tcolorbox}[<opções>]
Conteúdo da caixa.
\end{tcolorbox}

\tcbox[<opções>]{Conteúdo da caixa.}
```

Por exemplo, para alterar a cor da borda da caixa, utiliza-se o comando `colframe`. A Tabela 8.1 apresenta os principais parâmetros que podem ser inseridos em `opções`.

Tabela 8.1: Parâmetros opcionais do ambiente `tcolorbox`.

parâmetro	ação
<code>colback</code>	altera a cor do fundo da caixa
<code>colframe</code>	altera a cor da borda da caixa
<code>title</code>	adiciona um título
<code>sidebyside</code>	caixa lado a lado
<code>sharpish corners</code>	deixa os cantos quadrados
<code>lower separated=false</code>	remove a linha tracejada (quando usar <code>tcblower</code>)
<code>toprule</code>	cria uma borda superior
<code>bottomrule</code>	cria uma borda inferior
<code>leftrule</code>	cria uma borda à esquerda
<code>rightrule</code>	cria uma borda à direita

Exemplo 107 *Caixas utilizando os parâmetros `colback`, `colframe` e `title`.*

```
\tcbox[colback=magenta!5, colframe=magenta!40!black, title=Caixa com
→ \texttt{tcbox}.]{Conteúdo da caixa.}

\begin{tcolorbox}[colback=magenta!5, colframe=magenta!40!black, title=Caixa com
→ \texttt{tcbox}.]
Conteúdo da caixa.
\end{tcolorbox}
```

Caixa com `tcbox`.

Conteúdo da caixa.

Caixa com `tcbox`.

Conteúdo da caixa.

Exemplo 108 *Caixas utilizando os parâmetros colback, colframe, sidebyside e sharpish corners.*

```
\begin{tcolorbox}[colback=cyan!5, colframe=cyan!40!black, sidebyside, sharpish
→ corners]
Conteúdo da caixa - \textbf{esquerda}.
\tcblower
Conteúdo da caixa - \textbf{direita}.
\end{tcolorbox}
```

Conteúdo da caixa - **esquerda**.

Conteúdo da caixa - **direita**.

Exemplo 109 *Caixa utilizando os parâmetros colback, colframe, leftrule e bottomrule.*

```
\begin{tcolorbox}[colback=cyan!5, colframe=cyan!40!black, leftrule=2mm,
→ bottomrule=2mm]
Conteúdo da caixa.
\end{tcolorbox}
```

Conteúdo da caixa.

Para alterar a aparência de todas as caixas, utiliza-se o comando `tcbset`. No caso do exemplo anterior, foi necessário repetir todos os parâmetros opcionais. Ao utilizar o comando `tcbset`, isso é definido só uma vez.

Exemplo 110 *Caixas do Exemplo 107 utilizando o comando tcbset.*

```
\tcbset{colback=magenta!5, colframe=magenta!40!black}
\tcbbox[title=Caixa com \texttt{tcbbox}.]{Conteúdo da caixa.}

\begin{tcolorbox}[title=Caixa com \texttt{tcbbox}.]
Conteúdo da caixa.
\end{tcolorbox}
```

8.3 Espaçamento

Para ajustar manualmente o tamanho caixa ou o espaçamento interno desta caixa, utilizam-se os parâmetros da Tabela 8.2.

Tabela 8.2: Parâmetros de espaçamento.

parâmetro	ação
<code>halign</code>	alinhamento horizontal do conteúdo da caixa
<code>width</code>	altera a largura da caixa
<code>text width</code>	altera a largura da caixa em relação ao texto
<code>boxsep</code>	adiciona um espaçamento vertical
<code>lefttitle</code>	adiciona um espaço à esquerda do título

Observação 32 As opções para o parâmetro `halign` são: `center`, `left` e `right`.

Exemplo 111 Caixa utilizando os parâmetros de espaçamento `boxsep` e `lefttitle`.

```
\begin{tcolorbox}[colback=yellow!5, colframe=yellow!40!black, title=Título,  
→ boxsep=5mm, lefttitle=2cm]  
Conteúdo da caixa.  
\end{tcolorbox}
```

Título

Conteúdo da caixa.

Exemplo 112 Caixas utilizando o parâmetro de espaçamento `halign`.

```
\begin{tcolorbox}[colback=red!5, colframe=red!40!black, title=Título,  
→ halign=center]  
Conteúdo da caixa.  
\end{tcolorbox}  
\begin{tcolorbox}[colback=red!5, colframe=red!40!black, title=Título,  
→ halign=right]  
Conteúdo da caixa.  
\end{tcolorbox}
```

Título

Conteúdo da caixa.

Título

Conteúdo da caixa.

Exemplo 113 Caixa utilizando o parâmetro de espaçamento `width`.

```
\begin{tcolorbox}[colback=olive!5, colframe=olive!40!black, title=Título,  
→ width=5cm  
Conteúdo da caixa.  
\end{tcolorbox}
```

Título

Conteúdo da caixa.

8.4 O comando `newtcolorbox`

O comando `newtcolorbox` permite criar um novo ambiente baseado no ambiente padrão, o `tcolorbox`.

```
\newtcolorbox{<nome>}{<opções>}
```

O parâmetro `nome` define o nome do ambiente da caixa criada. O parâmetro `opções` admite qualquer parâmetro da Tabela 8.1.

Exemplo 114 Caixa personalizada criada utilizando o ambiente `tcolorbox`.

```
\newtcolorbox{cx}{colback=orange!10, colframe=orange!60!black}  
\begin{cx}  
Conteúdo da caixa.  
\end{cx}
```

Conteúdo da caixa.

Apesar de o comando `newcolorbox` poder ser inserido em qualquer parte do documento, aconselha-se que ele seja inserido no preâmbulo do documento. A justificativa para isso é que, ao criar muitas caixas, todas elas se localizarão na mesma parte do documento, evitando que o usuário fique muito tempo procurando. No entanto, por questões práticas, neste material o comando `newtcolorbox` será disposto junto à caixa criada.

8.5 O ambiente `tcblisting`

O ambiente `tcblisting` permite escrever e executar um código-fonte em \LaTeX .

```
\begin{tcblisting}{<parâmetros opcionais>}
Conteúdo da caixa.
\end{tcblisting}
```

Os parâmetros opcionais deste comando são os mesmos utilizados até o momento. Além disso, é possível utilizar um novo parâmetro para exibir apenas o código-fonte. Esse parâmetro é o `listing only`.

Para poder utilizar este ambiente, é necessário utilizar a biblioteca `listings`.

```
\tcbuselibrary{listings}
```

Exemplo 115 *Utilização básica do ambiente `tcblisting`.*

```
\begin{tcblisting}{colback=cyan!10!, colframe=magenta!60!black}
Este é um \texttt{exemplo} de utilização do ambiente \texttt{tcblisting}.
\end{tcblisting}
```

Este é um `\texttt{exemplo}` de utilização do ambiente `\texttt{tcblisting}`.

Este é um exemplo de utilização do ambiente `tcblisting`.

Exemplo 116 *Utilizando o ambiente `tcblisting` com o parâmetro `listing only`.*

```
\begin{tcblisting}{colback=cyan!10!, colframe=magenta!60!black, listing only}
Este é um \texttt{exemplo} de utilização do ambiente \texttt{tcblisting}.
\end{tcblisting}
```

Este é um `\texttt{exemplo}` de utilização do ambiente `\texttt{tcblisting}`.

Esse ambiente, quando utilizando com o parâmetro `listing only`, é muito útil para construir documentos sobre linguagens de programação, como *Python*, *Java* e *C++*.

8.6 Criando caixas enumeradas

Para criar caixas com numeração automática utiliza-se o parâmetro `use counter`. Também, é necessário criar um novo contador com o comando `newcounter`.

```
\newcounter{<nome do contador>}
```

Exemplo 117 *Criando um comando de caixas enumeradas.*

```
\newcounter{contador}
\newtcolorbox[use counter=contador]{enumera}{title={Exercício \thetcbcounter}}

\begin{enumera}
Calcule as raízes da função definida por  $f(x)=x^2+6x$ .
\end{enumera}

\begin{enumera}
Calcule as raízes da função definida por  $f(x)=x^2+3x-5$ .
\end{enumera}
```

Exercício 1

Calcule as raízes da função definida por $f(x) = x^2 + 6x$.

Exercício 2

Calcule as raízes da função definida por $f(x) = x^2 + 3x - 5$.

Capítulo 9

Bibliografia com o pacote `abntex2cite`

Para qualquer texto acadêmico é de grande importância a incorporação de referências no documento redigido. A forma mais comum de criar uma lista de referências no \LaTeX é utilizando o ambiente `thebibliography`. No entanto, caso seja necessário utilizar essa mesma lista em vários documentos ou até mesmo adaptá-la, esse ambiente pode se tornar pouco eficiente. Assim, existe um pacote chamado `abntex2cite` que pode não só facilitar a criação de uma lista de referências, mas também fazê-la de modo mais organizado.

9.1 O ambiente `thebibliography`

O \LaTeX possui um ambiente chamado `thebibliography` que permite a criação de referências de forma rápida. Este ambiente é muito recomendado quando se utiliza uma lista pequena de referências. Dentro desse ambiente, para cada referência criada, insere-se o comando `bibitem`.

```
\begin{thebibliography}{}  
  \bibitem{<chamada da referência>}  
  <dados da referência>  
\end{thebibliography}
```

Exemplo 118 *Lista de referência contendo um livro.*

```
\begin{thebibliography}{}  
  \bibitem{anton1}  
  ANTON, H.; BIVENS, I.; DAVIS, S. \textbf{Cálculo}. v. 2., 10. ed. Porto  
  ↪ Alegre: Bookman, 2014.  
\end{thebibliography}
```

Bibliografia

[1] ANTON, H.; BIVENS, I.; DAVIS, S. *Cálculo*. v. 2., 10. ed. Porto Alegre: Bookman, 2014.

9.2 O pacote `abntex2cite`

Por padrão, o \LaTeX não possui ferramentas que permitam formatar as referências de um documento conforme as normas brasileiras, apenas conforme algumas normas estrangeiras. Por isso, ao utilizar o ambiente `thebibliography`, isso é feito de forma manual. Uma ótima alternativa é o pacote `abntex2cite`, que permite criar uma lista de referências conforme a Norma 6023 de 2002 (NBR 6023) da Associação Brasileira de Normas Técnicas (ABNT).

A NBR 6023 permite duas formas de ordenação das referências em um documento. A primeira forma é o sistema alfabético, no qual as referências aparecem em ordem alfabética conforme nome do autor. A segunda forma é o sistema numérico, no qual a lista de referências está organizada de acordo com o aparecimento dos autores no decorrer do documento.

```
\usepackage[alf]{abntex2cite}
```

```
\usepackage[num]{abntex2cite}
```

O parâmetro `alf` se refere ao sistema alfabético e o parâmetro `num` ao sistema numérico.

Observação 33 *A utilização desse pacote implica em criar uma lista externa com as referências no formato `.bib`. Esse tipo de arquivo também é utilizado pelo pacote `bibtex`, o qual não tem suporte direto às normas brasileiras.*

9.3 O arquivo de referências - `.bib`

O arquivo com extensão `.BIB` irá conter todas as referências a serem utilizadas no documento. Para fazer a chamada desse arquivo utiliza-se o comando `bibliography` juntamente com o nome do arquivo de referências.

```
\bibliography{arquivo.bib}
```

Exemplo 119 *Lista com duas referências criada utilizando o sistema alfabético do abntex2cite.*

```
@book{anton1,
author={Howard Anton and Irl Bivens and Stephen Davis},
title={Cálculo},
volume={2},
edition={10},
publisher={Bookman},
year={2014},
address={Porto Alegre},
}
@article{barreto,
number={9},
author={Marcelo Chelminski Barreto and Monique Antunes de Souza Chelminski
→ Barreto},
title={A prevenção dos problemas de aprendizagem e as capacidades e
→ competências mínimas para a participação produtiva no século {XXI}},
journal={Revista Psicopedagogia},
address={São Paulo},
volume={22},
year={2005},
pages={154-161},
}
@phdthesis{kawasaki,
author={Teresinha Fumi Kawasaki},
title={Tecnologia na sala de aula de matemática},
subtitle={resistência e mudanças na formação continuada de professores},
year={2008},
address={Belo Horizonte},
pages={212},
school={Universidade Federal de Minas Gerais},
type={Doutorado em Educação}
}
```


Referências

ANTON, H.; BIVENS, I.; DAVIS, S. *Cálculo*. 10. ed. Porto Alegre: Bookman, 2014. v. 2.

BARRETO, M. C.; BARRETO, M. A. de S. C. A prevenção dos problemas de aprendizagem e as capacidades e competências mínimas para a participação produtiva no século XXI. *Revista Psicopedagogia*, São Paulo, v. 22, n. 9, p. 154–161, 2005.

KAWASAKI, T. F. *Tecnologia na sala de aula de matemática: resistência e mudanças na formação continuada de professores*. 212 p. Tese (Doutorado em Educação) — Universidade Federal de Minas Gerais, Belo Horizonte, 2008.

Observação 34 O `TEXmaker` cria a lista de referências com base nos autores citados utilizando o comando `cite`. Caso exista autores não citados, estes não aparecerão na lista de referências.

9.4 Criando a primeira lista de referências

Como já foi comentado, a criação de uma lista de referências exige a criação de um arquivo externo `.BIB`. Assim, cria-se um novo arquivo no `TEXMaker` com esta extensão, salvando-o na mesma pasta do arquivo `.TEX`. Cada tipo de referência (livro, artigo de periódico, monografia, ...) possui um comando específico iniciado por `@`.

```
@<tipo de referência>{<entradas>}
```

A Tabela 9.1 apresenta os comandos de declaração dos tipos de referências.

Tabela 9.1: Tipos de referências do pacote `abntex2cite`.

comando	tipo de referência
<code>@article</code>	artigo de periódico
<code>@book</code>	livro
<code>@booklet</code>	livro sem editora
<code>@conference</code>	artigo de anais de uma conferência
<code>@inbook</code>	parte de livro
<code>@incollection</code>	parte de coletânea
<code>@inproceedings</code>	igual ao <code>@conference</code>
<code>@journalpart</code>	parte de periódico
<code>@manual</code>	documentação técnica
<code>@mastersthesis</code>	dissertação de mestrado
<code>@misc</code>	outros tipos de publicação
<code>@monography</code>	monografias em geral
<code>@patent</code>	patentes
<code>@phdthesis</code>	tese de doutorado
<code>@proceedings</code>	anais de uma conferência
<code>@techreport</code>	relatório técnico
<code>@thesis</code>	semelhante a <code>@phdthesis</code>
<code>@unpublished</code>	documento não publicado

Junto ao comando iniciado por `@` é necessário informar os dados da obra. Os comandos de identificação são chamados de entradas. Uma entrada é toda informação sobre autoria, título, ano de publicação (entre outros) de uma obra. A Tabela 9.2 apresenta as entradas do pacote `abntex2cite`.

Tabela 9.2: Entradas do pacote `abntex2cite`.

comando	tipo de referência
<code>address</code>	local de publicação
<code>author</code>	nome do autor
<code>booktitle</code>	título do livro
<code>booksubtitle</code>	subtítulo do livro
<code>chapter</code>	capítulo
<code>conference-location</code>	local da conferência
<code>conference-number</code>	número da conferência
<code>conference-year</code>	ano da conferência
<code>dimensions</code>	dimensões
<code>edition</code>	edição
<code>editor</code>	editor
<code>editortype</code>	tipo de editor
<code>furtherresp</code>	informações adicionais sobre autoria
<code>howpublished</code>	informações adicionais sobre a publicação
<code>illustrated</code>	indica as ilustrações
<code>institution</code>	instituição
<code>isbn</code>	ISBN ¹
<code>issn</code>	ISSN ²
<code>journal</code>	nome do periódico
<code>month</code>	mês de publicação
<code>note</code>	nota
<code>number</code>	número da publicação
<code>organization</code>	organização
<code>pagename</code>	altera o nome da página
<code>pages</code>	número de páginas
<code>publisher</code>	editora
<code>school</code>	vinculação acadêmica
<code>section</code>	seção
<code>series</code>	coleção
<code>subtitle</code>	subtítulo da obra
<code>title</code>	título da obra
<code>type</code>	tipo
<code>url</code>	URL ³
<code>urlaccessdate</code>	Data de acesso à URL
<code>volume</code>	volume
<code>year</code>	ano de publicação
<code>year-presented</code>	ano de apresentação

Observação 35 A primeira entrada de uma referência do arquivo BIB sempre será o nome de como essa obra será citada no documento TEX.

A Tabela 9.3 apresenta o tipo de referência com suas entradas exclusivas, isto é, aquelas entradas que só são compatíveis com o tipo de referência indicado.

Tabela 9.3: Entradas do pacote `abntex2cite`.

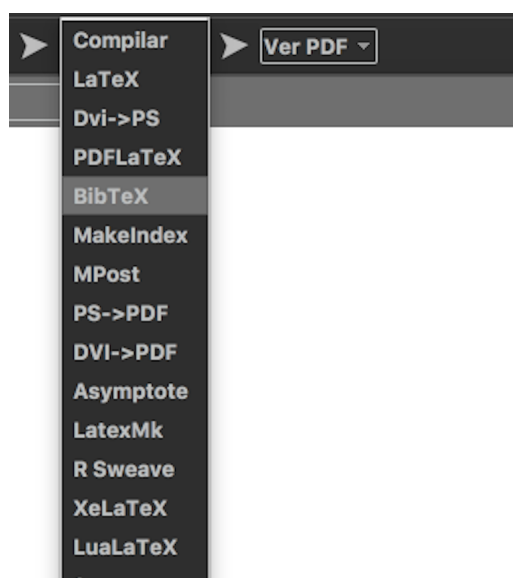
entrada exclusiva	tipos de referências
<code>conference-location</code>	<code>@proceedings</code> , <code>@inproceedings</code> , <code>@conference</code>
<code>conference-number</code>	<code>@proceedings</code> , <code>@inproceedings</code> , <code>@conference</code>
<code>conference-year</code>	<code>@proceedings</code> , <code>@inproceedings</code> , <code>@conference</code>
<code>year-presented</code>	<code>@thesis</code> , <code>@monography</code> , <code>@phdthesis</code> , <code>@mastersthesis</code>

Uma vez que o arquivo de referências está pronto, os comandos abaixo devem ser executados na seguinte ordem:

1. PDFLaTeX
2. BibTeX
3. PDFLaTeX
4. PDFLaTeX

Ao executar os comandos acima, a lista de referências será incorporada ao documento.

Observação 36 Estes comandos podem ser encontrados na barra superior do `TEXmaker`.



Exemplo 120 Lista com 5 referências.

```
@article{artigo,  
address={Bento Gonçalves},  
author={Naylene Fraccanabbia and André Luvisa and Delair Bavaresco},  
journal={REMAT: Revista Eletrônica da Matemática},  
month={ago.},  
number={1},  
pages={229-244},  
title={Planejamento de trajetórias polinomiais para robótica com Arduino},  
volume={4},  
year={2018}}
```

```
@book{livro,  
address={São Paulo},  
author={Hamilton Luiz Guidorizzi},  
edition={5},  
year={2001},  
pages={496},  
publisher={LTC Editora},  
title={Um Curso de Cálculo},  
volume={2}}
```

```
@booklet{livrob,  
address={Brasília, DF},  
edition={2},  
organization={IBICT},  
page={41},  
title={Manual de normas de editoração do {IBICT}},  
year={1993}}
```

```
@inproceedings{anais,  
address={Porto Alegre},  
author={Mônica Marques and Simone Cavalheiro and Luciana Foss and Adriana  
↪ Bordini and Christiano Ávila},  
booktitle={Anais...},  
conference-number={XXVIII},  
conference-year={2017},  
conference-location={Recife},  
title={Uma Proposta para o Desenvolvimento do Pensamento Computacional  
↪ Integrado ao Ensino de Matemática},  
organization={SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO},
```

```

pages={314-323},
publisher={Sociedade Brasileira de Computação},
year={2017}}

@phdthesis{tese,
address={Belo Horizonte},
author={Teresinha Fumi Kawasaki},
institution={Universidade Federal de Minas Gerais},
pages={212},
school={Programa de Pós-Graduação em Educação},
subtitle={resistência e mudanças na formação continuada de professores},
title={Tecnologia na sala de aula de matemática},
type={Doutorado em Educação},
year={2008},
year-presented={2008}}

```

Referências

- FRACCANABBIA, N.; LUVISA, A.; BAVARESCO, D. Planejamento de trajetórias polinomiais para robótica com arduino. *REMAT: Revista Eletrônica da Matemática*, Bento Gonçalves, v. 4, n. 1, p. 229-244, ago. 2018.
- GUIDORIZZI, H. L. *Um Curso de Cálculo*. 5. ed. São Paulo: LTC Editora, 2001. v. 2. 496 p.
- IBICT. *Manual de normas de editoração do IBICT*. 2. ed. Brasília, DF, 1993.
- KAWASAKI, T. F. *Tecnologia na sala de aula de matemática: resistência e mudanças na formação continuada de professores*. 2008. 212 p. Tese (Doutorado em Educação) — Programa de Pós-Graduação em Educação, Belo Horizonte, 2008.
- MARQUES, M. et al. Uma proposta para o desenvolvimento do pensamento computacional integrado ao ensino de matemática. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, XXVIII., 2017, Recife. *Anais...* Porto Alegre: Sociedade Brasileira de Computação, 2017. p. 314-323.

9.5 O comando citeoption

O comando `citeoption` permite personalizar o comportamento das referências. A Tabela 9.4 apresenta os principais parâmetros desse comando.

Tabela 9.4: Parâmetros do comando `citeoption`

parâmetro	tipos de referências
<code>abnt-full-initials=yes</code>	mantém os nomes completos, sem abreviá-los
<code>abnt-emphasize=bf</code>	altera o estilo do grifo para negrito
<code>abnt-etal-list=0</code>	não abrevia a lista de autores
<code>abnt-etal-text=emph</code>	deixa o <i>et al.</i> em itálico
<code>abnt-repeated-author-omit=yes</code>	omite a repetição dos nomes de autores

Os parâmetros da Tabela 9.4 podem ser inseridos logo antes do comando `bibliography`.

Exemplo 121 *Lista de referências do Exemplo 120 com os parâmetros da Tabela 9.4.*

```
<...>
\citeoption{abnt-full-initials=yes}
\citeoption{abnt-emphasize=bf}
\citeoption{abnt-etal-list=0}
\citeoption{abnt-etal-text=emph}
\citeoption{abnt-repeated-author-omit=yes}
\bibliography{referencias.bib}
```

Referências

FRACCANABBIA, Naylene; LUVISA, André; BAVARESCO, Delair. Planejamento de trajetórias polinomiais para robótica com arduino. **REMAT: Revista Eletrônica da Matemática**, Bento Gonçalves, v. 4, n. 1, p. 229-244, ago. 2018.

GUIDORIZZI, Hamilton Luiz. **Um Curso de Cálculo**. 5. ed. São Paulo: LTC Editora, 2001. v. 2. 496 p.

IBICT. **Manual de normas de editoração do IBICT**. 2. ed. Brasília, DF, 1993.

KAWASAKI, Teresinha Fumi. **Tecnologia na sala de aula de matemática**: resistência e mudanças na formação continuada de professores. 2008. 212 p. Tese (Doutorado em Educação) — Programa de Pós-Graduação em Educação, Belo Horizonte, 2008.

MARQUES, Mônica; CAVALHEIRO, Simone; FOSS, Luciana; BORDINI, Adriana; ÁVILA, Christiano. Uma proposta para o desenvolvimento do pensamento computacional integrado ao ensino de matemática. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, XXVIII., 2017, Recife. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 2017. p. 314-323.

9.6 Criando as citações

Para criar uma citação, utilizam-se os comandos `cite` e `citeonline`. O comando `cite` cria uma citação na forma (AUTOR, 2018) e o comando `citeonline` cria uma citação da forma

Autor (2018). É importante ressaltar que somente os autores citados aparecem na lista de referências.

Referências

ARAÚJO, L. C. **The abntex2 package**. Disponível em: <https://www.ctan.org/pkg/abntex2>. Acesso em: 4 maio 2019.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: Informação e documentação: referências: elaboração. Rio de Janeiro, 2002

RICHTER, P.; BOTOEVA, E.; BARNARD, R.; SURMANN, D. **The tikzposter class**. Disponível em: <https://www.ctan.org/pkg/tikzposter>. Acesso em: 4 maio 2019.

STURM, T. F. **The tcolorbox package**. Disponível em <https://www.ctan.org/pkg/tcolorbox>. Acesso em: 4 maio 2019.

TANTAU, T. **The TikZ and PGF packages**. Disponível em: <https://www.ctan.org/pkg/pgf>. Acesso em: 4 maio 2019.

TANTAU, T.; WRIGHT, J.; MILETIĆ, V. **The beamer class**. Disponível em: <https://www.ctan.org/pkg/beamer>. Acesso em: 4 maio 2019.

WIKIBOOKS CONTRIBUTORS. **The Free Textbook Project**. 2016. Disponível em: <https://en.wikibooks.org/wiki/LaTeX>. Acesso em: 4 maio 2019