

---

MINISTÉRIO DA EDUCAÇÃO – MEC

SECRETARIA DE EDUCAÇÃO SUPERIOR – SESU

PROGRAMA DE EDUCAÇÃO TUTORIAL – PET

UNIVERSIDADE FEDERAL FLUMINENSE – UFF

ESCOLA DE ENGENHARIA – TCE

GRUPO PET DO CURSO DE ENG. DE TELECOMUNICAÇÕES – PET-TELE

## Tutoriais PET-Tele

# Elaboração de objetos não textuais diretamente no L<sup>A</sup>T<sub>E</sub>X

(Versão: A2023M02D01)

Autores: João Guilherme Coutinho Beltrão  
Pedro Henryque Barbosa da Silva e Kriz  
Alexandre Santos de la Vega (jan/2022)

Tutor: Alexandre Santos de la Vega

Niterói – RJ

Fevereiro / 2023

---

# Sumário

<b>I</b>	<b>Introdução</b>	<b>3</b>
<b>1</b>	<b>Introdução</b>	<b>4</b>
1.1	PET e grupo PET-Tele . . . . .	4
1.2	Motivações . . . . .	4
1.3	Objetivo . . . . .	5
1.4	Resultados esperados . . . . .	5
<b>2</b>	<b>Sistema de preparação de documentos L<sup>A</sup>T<sub>E</sub>X</b>	<b>6</b>
2.1	Introdução . . . . .	6
2.2	Inserção de objetos não textuais . . . . .	6
2.3	Inserção de objetos não textuais por importação . . . . .	7
2.3.1	Pacote gráfico . . . . .	7
2.3.2	Ambiente <i>figure</i> . . . . .	7
2.3.3	Comando <code>\centering</code> e ambiente <i>center</i> . . . . .	7
2.3.4	Comando <code>\includegraphics</code> . . . . .	7
2.3.5	Comando <code>\caption</code> . . . . .	7
2.3.6	Comando <code>\label</code> . . . . .	8
2.4	Referências . . . . .	8
<b>3</b>	<b>Geração e uso de objetos não textuais</b>	<b>9</b>
3.1	Comandos nativos: ambiente <i>picture</i> . . . . .	9
3.2	Uso de bibliotecas de comandos (pacote ou <i>package</i> ) . . . . .	9
3.3	Outros pacotes e editores para objetos não textuais . . . . .	9
3.4	Referências . . . . .	10
<b>4</b>	<b>Ambiente <i>picture</i></b>	<b>11</b>
4.1	Unidade de comprimento . . . . .	11
4.2	Ambiente <i>picture</i> . . . . .	11
4.3	Exemplos genéricos com o ambiente <i>picture</i> . . . . .	11
4.4	Exemplos com o ambiente <i>picture</i> . . . . .	13
<b>II</b>	<b>Pacotes gráficos</b>	<b>20</b>
<b>5</b>	<b>Pacote <i>Xy-pic</i></b>	<b>21</b>
5.1	Exemplos com o pacote <i>Xy-pic</i> . . . . .	21

<b>6</b>	<b>Pacote TikZ</b>	<b>23</b>
6.1	Definição dos pacotes <i>TikZ</i> e <i>pgf</i> . . . . .	23
6.2	Importação do pacote . . . . .	23
6.3	Comando <code>\tikz</code> . . . . .	23
6.4	Ambiente <i>tikzpicture</i> . . . . .	23
6.5	Alguns comandos básicos . . . . .	24
6.5.1	Comando <code>\draw</code> . . . . .	24
6.5.2	Comando <code>\fill</code> . . . . .	24
6.5.3	Comando <code>\shade</code> . . . . .	24
6.6	Exemplos com o pacote <i>TikZ/pgf</i> . . . . .	25
6.6.1	Exemplos simples . . . . .	25
6.6.2	Exemplos usando bibliotecas . . . . .	27
6.7	Outros exemplos com o pacote <i>TikZ/pgf</i> . . . . .	28
6.8	Exemplos para Apostila de Processamento Digital de Sinais . . . . .	37
6.9	Referências . . . . .	40
<b>7</b>	<b>Pacote <i>PgfPlots</i></b>	<b>41</b>
7.1	Exemplos com o pacote <i>PgfPlots</i> . . . . .	41
<b>III</b>	<b>Aplicações</b>	<b>43</b>
<b>8</b>	<b>Mapa de Karnaugh</b>	<b>44</b>
8.1	Circuitos digitais . . . . .	44
8.2	Funções lógicas . . . . .	44
8.3	Mapa de Karnaugh . . . . .	45
8.4	Geração de mapas para textos didáticos . . . . .	45
8.5	Exemplos de pacotes especializados . . . . .	45
8.6	Exemplos com o pacote <i>askmaps</i> . . . . .	47
8.7	Exemplos com o pacote <i>kvmap</i> . . . . .	48
8.8	Exemplos para Apostila de Circuitos Digitais . . . . .	50
8.9	Referências . . . . .	55
	<b>Referências bibliográficas</b>	<b>59</b>

# Parte I

## Introdução

# Capítulo 1

## Introdução

Este capítulo trata da introdução do presente documento. Inicialmente, o PET e o grupo PET-Tele são brevemente descritos. Em seguida, são apresentados as motivações, o objetivo e os resultados esperados, deste trabalho.

### 1.1 PET e grupo PET-Tele

O Programa de Educação Tutorial (PET) [Prof] exige que os bolsistas dos seus grupos, ao serem submetidos a uma formação complementar, desenvolvam atividades que possuam, conjuntamente, itens relativos às áreas de Pesquisa, Ensino e Extensão, que consigam algum tipo de penetração no curso ao qual pertencem e que realizem trabalhos de cooperação com outros grupos, ligados ou não ao seu curso de origem. Logo, o PET busca atitudes inovadoras em Educação. Procurando atender aos requisitos do Programa, o PET-Tele [Gru], grupo PET do Curso de Engenharia de Telecomunicações da Universidade Federal Fluminense, realiza atividades em diversas linhas do conhecimento, de acordo com o interesse e as competências de seus integrantes.

### 1.2 Motivações

Seguindo a filosofia de uso de *software* livre, público e gratuito, o grupo PET-Tele emprega o  $\LaTeX$  para a produção do seu material textual, além de oferecer oficinas/minicursos/cursos de introdução ao  $\LaTeX$ . O Sistema de Preparação de Documentos  $\LaTeX$  teve origem no ambiente acadêmico, com o intuito de oferecer facilidades para estruturação e padronização do texto, digitação de equações, construção de tabelas, importação de objetos gráficos, além de apresentar uma portabilidade direta entre plataformas computacionais diferentes (*hardware* e *software*). Além disso, ainda é possível uma interação direta com outras linguagens e ferramentas computacionais, tais como Julia, Python, R e Jupyter Notebook. O  $\LaTeX$  baseia-se no uso de uma linguagem de marcação (*markup language*), por meio da qual o texto original tem seus elementos estruturais identificados por comandos específicos. O texto marcado é passado por um *software* de formatação, gerando o documento final, adequadamente formatado. Nativamente, o  $\LaTeX$  oferece a possibilidade de construção de desenhos simples, por meio do ambiente de marcação *picture* e de seus respectivos comandos. Com a evolução das bibliotecas criadas por usuários, tornou-se possível a elaboração de desenhos e gráficos com elevada complexidade, diretamente no texto original, sem a necessidade de importação de tais objetos não textuais. Embora alguns padrões básicos tenham sido estabelecidos, os documentos do grupo têm sido preparados com total liberdade de criação. Porém, o grupo visualizou uma forma de inovar,

amadurecer e expandir a preparação dos seus documentos, empregando bibliotecas  $\text{\LaTeX}$  para a elaboração de elementos não textuais (desenhos e gráficos).

### 1.3 Objetivo

O objetivo deste trabalho é realizar estudos e experimentos relativos a elaboração de desenhos e gráficos por meio de código nativo e de bibliotecas do Sistema de Preparação de Documentos  $\text{\LaTeX}$ .

### 1.4 Resultados esperados

Espera-se que, de posse deste documento, o leitor consiga elaborar desenhos e gráficos simples por meio de código nativo e de bibliotecas do Sistema de Preparação de Documentos  $\text{\LaTeX}$ .

# Capítulo 2

## Sistema de preparação de documentos $\text{\LaTeX}$

Este capítulo apresenta brevemente o Sistema de Preparação de Documentos  $\text{\LaTeX}$ , destacando a inserção de objetos não textuais nos textos nele produzidos.

### 2.1 Introdução

Seguindo a filosofia de uso de *software* livre, público e gratuito, o grupo PET-Tele emprega o  $\text{\LaTeX}$  para a produção do seu material textual, além de oferecer oficinas/minicursos/cursos de introdução ao  $\text{\LaTeX}$ .

O Sistema de Preparação de Documentos  $\text{\LaTeX}$  teve origem no ambiente acadêmico, com o intuito de oferecer facilidades para estruturação e padronização do texto, digitação de equações, construção de tabelas, importação de objetos gráficos, além de apresentar uma portabilidade direta entre plataformas computacionais diferentes (*hardware* e *software*).

O  $\text{\LaTeX}$  baseia-se no uso de uma linguagem de marcação (*markup language*), por meio da qual o texto original tem seus elementos estruturais identificados por comandos específicos. O texto marcado é passado por um *software* de formatação, gerando o documento final, adequadamente formatado.

Nativamente, o  $\text{\LaTeX}$  oferece a possibilidade de construção de desenhos simples, por meio do ambiente de marcação *picture* e de seus respectivos comandos. Com a evolução das bibliotecas criadas por usuários, tornou-se possível a elaboração de desenhos e gráficos com elevada complexidade, diretamente no texto original, sem a necessidade de importação de tais objetos não textuais.

### 2.2 Inserção de objetos não textuais

Basicamente, há dois modos de inserção de objetos não textuais, em um texto produzido no  $\text{\LaTeX}$ . Primeiramente, pode-se adotar o procedimento comum aos editores de texto conhecidos, que é o seguinte: gerar o objeto não textual, salvar o objeto em um arquivo com formato padrão, importar o arquivo e ancorá-lo no ponto desejado do texto. Alternativamente, pode-se gerar o objeto não textual por meio de comandos  $\text{\LaTeX}$ .

A produção do código  $\text{\LaTeX}$  que irá gerar o objeto não textual pode-se dar de duas maneiras. Os comandos podem ser digitados manualmente, no caso de objetos simples ou no caso do uso de bibliotecas de funções usadas para objetos mais complexos. Além disso, alguns

*softwares* especializados na geração dos objetos não textuais, já apresentam a possibilidade de armazenamento em arquivo contendo código L<sup>A</sup>T<sub>E</sub>X, facilitando a tarefa.

A inserção de objetos não textuais por importação de arquivo em formato padrão é brevemente apresentada na próxima seção. A geração por meio de código L<sup>A</sup>T<sub>E</sub>X é o tema principal deste documento.

## 2.3 Inserção de objetos não textuais por importação

A seguir, são citados os itens básicos para a inserção de objetos não textuais por importação de arquivo em formato padrão.

### 2.3.1 Pacote gráfico

Para realizar a inserção de figuras em um texto L<sup>A</sup>T<sub>E</sub>X, o primeiro passo é importar o pacote gráfico *graphicx*. O pacote deve ser importado no início do arquivo principal, antes de utilizar o comando `\begin{document}`. O comando para a importação é `\usepackage{graphicx}`.

### 2.3.2 Ambiente *figure*

Embora não seja estritamente necessário, os objetos não textuais são aninhados dentro de um ambiente *figure*, para possibilitar uma correta identificação, referência e posição.

O ambiente *figure* é criado com o par de comandos `\begin{figure}[pos]`, para iniciar o ambiente, e do comando `\end{figure}`, para encerrar o ambiente. Os parâmetros opcionais indicados em “pos” (!,h,t,b,p) são usados para indicar onde a figura deve ser inserida.

### 2.3.3 Comando `\centering` e ambiente *center*

Para centralizar uma figura, pode-se utilizar o comando `\centering`, dentro do ambiente *figure*. Como alternativa, pode-se criar um ambiente “*center*”, com a utilização do comando `\begin{center}`, para iniciar o ambiente, e do comando `\end{center}`, para encerrar o ambiente.

### 2.3.4 Comando `\includegraphics`

A inserção propriamente dita do arquivo contendo a figura em um formato padrão é resultado da execução do comando `\includegraphics{nome_arq}`.

Embora “nome\_arq” possa conter a extensão do arquivo, é recomendável não incluí-la. Dessa forma, o próprio L<sup>A</sup>T<sub>E</sub>X realiza a busca por um arquivo em um dos formatos padrões, tais como: ps, eps, png, jpg, gif, entre outros.

### 2.3.5 Comando `\caption`

O comando `\caption{legenda_desejada}` é utilizado para adicionar uma legenda à figura, contendo o texto “legenda\_desejada”.

### 2.3.6 Comando $\backslash\text{label}$

O comando  $\backslash\text{label}\{\text{ref\_interna}\}$  é utilizado para definir uma referência para a figura, internamente ao código. Desta maneira, para se referir à figura, em alguma parte do código do texto, basta utilizar o comando  $\backslash\text{ref}\{\text{ref\_interna}\}$ . Assim não é necessário saber o número da figura, que é controlado automaticamente pelo  $\text{\LaTeX}$ .

## 2.4 Referências

Mais detalhes sobre os tópicos abordados neste capítulo podem ser encontrados nas seguintes referências: [Bera], [Berb], [Berc], [Berd], [Pesa], [Pesb], [Pesc], [Pesd], [Pese], [Pesf] e [MiK].

# Capítulo 3

## Geração e uso de objetos não textuais

A seguir, são brevemente comentados os mecanismos que podem ser empregados para a geração e o uso de objetos não textuais no  $\text{\LaTeX}$ .

### 3.1 Comandos nativos: ambiente *picture*

No sistema  $\text{\LaTeX}$ , assim como em outros ambientes de preparação de documentos, objetos não textuais são comumente importados a partir de arquivos específicos, os quais são gerados em ferramentas especializadas. Por outro lado, no  $\text{\LaTeX}$ , objetos não textuais com baixa complexidade também podem ser gerados internamente, empregando comandos nativos. Para isso, o  $\text{\LaTeX}$  disponibiliza o ambiente *picture* e seus respectivos comandos.

### 3.2 Uso de bibliotecas de comandos (pacote ou *package*)

Bibliotecas de comandos para o  $\text{\LaTeX}$ , específicas para determinadas aplicações, costumam ser disponibilizadas gratuitamente pela comunidade de usuários do sistema, sob a denominação de pacotes (*packages*).

Pacotes especializados são capazes de gerar objetos não textuais de alta complexidade.

Alguns pacotes estendem o ambiente *picture*, tais como: *epic*, *eepic*, *PSTricks*.

O pacote *PSTricks* conta com a ferramenta de desenho *PSPicture*.

Existem ainda pacotes de uso específico, tal como o *XY-pic*, que é capaz de gerar desenhos de diagramas. Um manual para o pacote *XY-pic* pode ser encontrado no seguinte URL:

<http://www.ctan.org/tex-archive/macros/generic/diagrams/xypic/xy/doc/xyguide.pdf>.

O ambiente *picture* também pode ser usado em conjunto com o utilitário *gnuplot*, que é uma ferramenta capaz de gerar de gráficos de duas e três dimensões.

### 3.3 Outros pacotes e editores para objetos não textuais

Podem ainda ser citados os seguintes pacotes e editores para objetos não textuais:

- Pacote *PiCTex*, que é um pacote com macros  $\text{\TeX}$  para gráficos.
- Pacote *TikZ/pgf*, que é um pacote capaz de gerar gráficos vetoriais. Diversas ferramentas de edição gráfica são capazes de exportar seus desenhos na forma de código *TikZ/pgf*, tais como: Cirkuit, GeoGebra, Inkscape, Octave, matplotlib/Python, Gnuplot, R.

- Pacote *PSTricks*, similar ao pacote *TikZ/pgf*. Porém, o código deve ser compilado com o comando *latex*.
- Pacote DITAA (*Diagrams Through Ascii Art*), que é um pacote para geração de diagramas a partir de código ASCII (*American Standard Code for Information Interchange*).
- Editor LaTeXDraw, que gera código  $\text{\LaTeX}$ , com padrão *PSTricks*, para as figuras nele desenhadas.
- Editor gráfico Inkscape, que permite exportar figuras no padrão *PSTricks*.
- Editores gráficos Inkscape, ImageMagik e The Gimp, que permitem conversão entre formatos de imagem (PDF, PNG, JPG, GIF, PS, EPS).
- Utilitários jpeg2ps, bmeps e epstopdf, que permitem conversão entre formatos de imagem (PDF, PNG, JPG, GIF, PS, EPS).
- TeXText, que é uma extensão do Inkscape.
- Shutter, que é uma ferramenta de captura.

## 3.4 Referências

Mais detalhes sobre os tópicos abordados neste capítulo podem ser encontrados nas seguintes referências: [Stab], [Staa], [dak], [Hir], [Hir18], [Pesj], [Pesi], e [Peg].

# Capítulo 4

## Ambiente *picture*

O ambiente *picture* é utilizado para criar figuras em qualquer lugar do texto gerado pelo código  $\text{\LaTeX}$  correspondente. O objetivo deste capítulo é explicar as principais funções do presente ambiente, bem como seus respectivos comandos.

### 4.1 Unidade de comprimento

Para controlar o tamanho final da figura criada pelo ambiente *picture*, é utilizada a variável associada ao comando `\unitlength`, que define o valor da unidade de comprimento do desenho (*drawing unit*). O valor da unidade de comprimento pode ser ajustado por meio do comando `\setlength{\unitlength}{value}`.

### 4.2 Ambiente *picture*

O ambiente *picture* é criado com o par de comandos `\begin{picture}(width, height)`, para iniciar o ambiente, e do comando `\end{picture}`, para encerrar o ambiente. Os parâmetros *width* e *height* são usados, respectivamente, para calcular a largura ( $width * \unitlength$ ) e a altura ( $height * \unitlength$ ) da figura a ser criada.

Vale ressaltar que parâmetros de opções podem ser acrescentados ao comando de início, mas eles não serão tratados aqui.

### 4.3 Exemplos genéricos com o ambiente *picture*

Um exemplo genérico de uso do ambiente *picture* é o seguinte:

```
\setlength{\unitlength}{value}
\begin{picture}(width, height)
  comandos_picture_para_gerar_a_figura_desejada
  ...
  comandos_picture_para_gerar_a_figura_desejada
\end{picture}
```

Um exemplo genérico de uso do ambiente *picture*, dentro do ambiente *figure*, com centralização por comando, é dado por:

```
\begin{figure}
%
\centering
%
\setlength{\unitlength}{value}
\begin{picture}(width, height)
  comandos_picture_para_gerar_a_figura_desejada
  ...
  comandos_picture_para_gerar_a_figura_desejada
\end{picture}
%
\caption{legenda_desejada}
%
\label{ref_interna}
%
\end{figure}
```

Um exemplo com a centralização por ambiente é o seguinte:

```
\begin{figure}
%
\begin{center}
%
\setlength{\unitlength}{value}
\begin{picture}(width, height)
  comandos_picture_para_gerar_a_figura_desejada
  ...
  comandos_picture_para_gerar_a_figura_desejada
\end{picture}
%
\end{center}
%
\caption{legenda_desejada}
%
\label{ref_interna}
%
\end{figure}
```

## 4.4 Exemplos com o ambiente *picture*

A seguir, são realizados alguns testes, usando o ambiente *picture*, com códigos encontrados em diferentes referências.

Código:

```
\begin{figure}[!ht]
\setlength{\unitlength}{2cm}
\centering
\begin{picture}(3,1.6)
\put(1,1){\circle*{0.2}}
\put(1,1){\circle{1.2}}
\put(1,1){\vector(0,1){0.6}}
\put(2.5,1){\circle*{0.5}}
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Some circles.} \label{flpic:circles}
\end{figure}
```

Resultado:

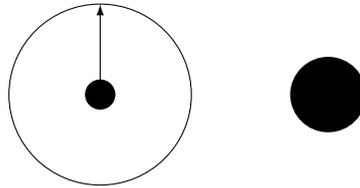


Figura 4.1: Some circles.

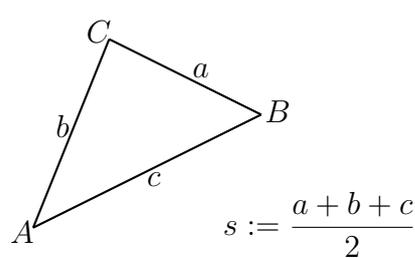
Código:

```

\begin{figure}[!ht]
\setlength{\unitlength}{1cm}
\centering
\begin{picture}(6,5)
\thicklines
\put(1,0.5){\line(2,1){3}}
\put(4,2){\line(-2,1){2}}
\put(2,3){\line(-2,-5){1}}
\put(0.7,0.3){\mathcal{A}}
\put(4.05,1.9){\mathcal{B}}
\put(1.7,2.95){\mathcal{C}}
\put(3.1,2.5){\mathcal{a}}
\put(1.3,1.7){\mathcal{b}}
\put(2.5,1.05){\mathcal{c}}
\put(0.3,4){\mathcal{F}=}
\sqrt{s(s-a)(s-b)(s-c)}
\put(3.5,0.4){\displaystyle
s:=\frac{a+b+c}{2}}
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Text and formulas.} \label{flpic:TextAndFormulas}
\end{figure}

```

Resultado:

$$F = \sqrt{s(s-a)(s-b)(s-c)}$$


$$s := \frac{a+b+c}{2}$$

Figura 4.2: Text and formulas.

Código:

```

\begin{figure}[!ht]
\setlength{\unitlength}{2cm}
\centering
\begin{picture}(6,4)
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}%
{\line(0,1){4}}
\multiput(0,0)(0,1){5}%
{\line(1,0){6}}
\thicklines
\put(2,3){\oval(3,1.8)}
\thinlines
\put(3,2){\oval(3,1.8)}
\thicklines
\put(2,1){\oval(3,1.8)[t1]}
\put(4,1){\oval(3,1.8)[b]}
\put(4,3){\oval(3,1.8)[r]}
\put(3,1.5){\oval(1.8,0.4)}
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Squares and ovals.} \label{flpic:SquaresAndOvals}
\end{figure}

```

Resultado:

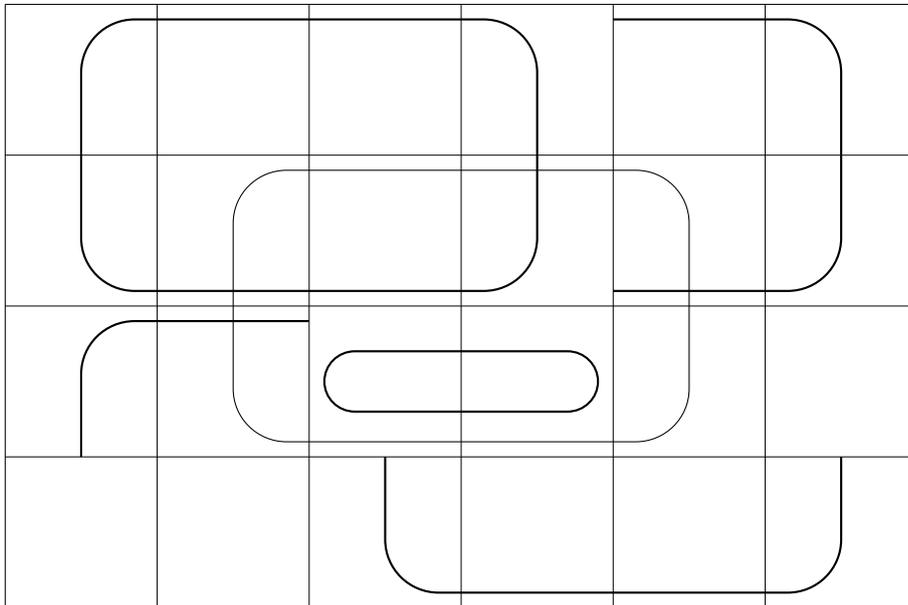


Figura 4.3: Squares and ovals.

Código:

```

\begin{figure}[!ht]
\centering
\setlength{\unitlength}{0.3mm}
\begin{picture}(120,168)
\newsavebox{\foldera}
\savebox{\foldera}
(40,32)[bl]{% definition
\multiput(0,0)(0,28){2}
{\line(1,0){40}}
\multiput(0,0)(40,0){2}
{\line(0,1){28}}
\put(1,28){\oval(2,2)[t1]}
\put(1,29){\line(1,0){5}}
\put(9,29){\oval(6,6)[t1]}
\put(9,32){\line(1,0){8}}
\put(17,29){\oval(6,6)[tr]}
\put(20,29){\line(1,0){19}}
\put(39,28){\oval(2,2)[tr]}
}
\newsavebox{\folderb}
\savebox{\folderb}
(40,32)[l]{% definition
\put(0,14){\line(1,0){8}}
\put(8,0){\usebox{\foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
{\usebox{\folderb}}
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Squares and ovals.} \label{flpic:SquaresAndOvals}
\end{figure}

```

Resultado:

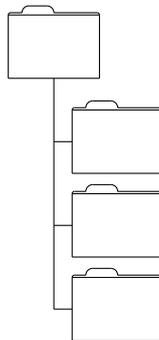


Figura 4.4: Folders.

Código:

```

\begin{figure}[!ht]
\centering
\setlength{\unitlength}{1.8cm}
\begin{picture}(6,4)(-3,-2)
\put(-2.5,0){\vector(1,0){5}}
\put(2.7,-0.1){$\chi$}
\put(0,-1.5){\vector(0,1){3}}
\multiput(-2.5,1)(0.4,0){13}
{\line(1,0){0.2}}
\multiput(-2.5,-1)(0.4,0){13}
{\line(1,0){0.2}}
\put(0.2,1.4)
{${\beta=v/c=\tanh\chi}$}
\qbezier(0,0)(0.8853,0.8853)
(2,0.9640)
\qbezier(0,0)(-0.8853,-0.8853)
(-2,-0.9640)
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Plotting graphs with \textit{qbezier}.} \label{flpic:PlotGraphQBezier}
\end{figure}

```

Resultado:

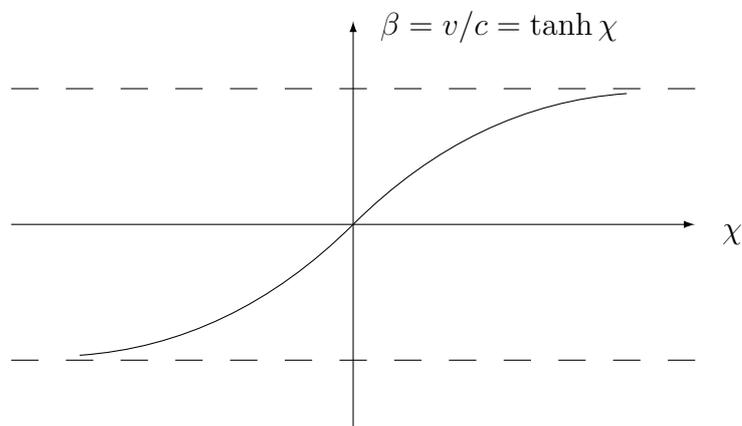


Figura 4.5: Plotting graphs with *qbezier*.

Código:

```

\begin{figure}[!ht]
\setlength{\unitlength}{1pc}
\centering
\begin{picture}(21,26)
% Draw Pica scale
\put(2,2){\begin{picture}(5,24)
\put(0,-0.5){\makebox(0,0)[t]{\textbf{Picas}}}
\thicklines \put(0,0){\line(0,1){24.0}}
\thinlines \multiput(0,0)(0,1){25}{\line(1,0){1}}
\multiput(0,0)(0,10){3}{\line(1,0){2}}
\put(-1,0){\makebox(0,0)[br]{0}}
\put(-1,10){\makebox(0,0)[br]{10}}
\put(-1,20){\makebox(0,0)[br]{20}}
\end{picture}}
% Draw Points scale
\put(7,2){\begin{picture}(5,24)
\put(0,-0.5){\makebox(0,0)[t]{\textbf{Points}}}
\thicklines \put(0,0){\line(0,1){24.2}}
\thinlines \multiput(0,0)(0,0.8333){30}{\line(1,0){1}}
\multiput(0,0)(0,8.333){3}{\line(1,0){2}}
\put(-1,0){\makebox(0,0)[br]{0}}
\put(-1,8.333){\makebox(0,0)[br]{100}}
\put(-1,16.667){\makebox(0,0)[br]{200}}
\end{picture}}
% Draw Millimeter scale
\put(12,2){\begin{picture}(5,24)
\put(0,-0.5){\makebox(0,0)[t]{\textbf{Millimeters}}}
\thicklines \put(0,0){\line(0,1){24.2}}
\thinlines \multiput(0,0)(0,0.4742){15}{\line(1,0){1}}
\multiput(0,0)(0,2.3711){11}{\line(1,0){2}}
\put(-1,0){\makebox(0,0)[br]{0}}
\put(-1,4.742){\makebox(0,0)[br]{20}}
\put(-1,9.484){\makebox(0,0)[br]{40}}
\put(-1,14.226){\makebox(0,0)[br]{60}}
\put(-1,18.968){\makebox(0,0)[br]{80}}
\put(-1,23.71){\makebox(0,0)[br]{100}}
\end{picture}}
% Draw Inch scale
\put(17,2){\begin{picture}(5,24)
\put(0,-0.5){\makebox(0,0)[t]{\textbf{Inches}}}
\thicklines \put(0,0){\line(0,1){24.1}}
\thinlines \multiput(0,0)(0,0.60225){41}{\line(1,0){1}}
\multiput(0,0)(0,6.0225){5}{\line(1,0){2}}
\put(-1,0){\makebox(0,0)[br]{0}}
\put(-1,6.0225){\makebox(0,0)[br]{1}}
\put(-1,12.045){\makebox(0,0)[br]{2}}
\put(-1,18.0675){\makebox(0,0)[br]{3}}

```

```

\put(-1,24.09){\makebox(0,0)[br]{4}}
\end{picture}}
\end{picture}
\setlength{\unitlength}{1pt} % back to the default value
\caption{Some measuring scales.} \label{flpic:scales}
\end{figure}

```

Resultado:

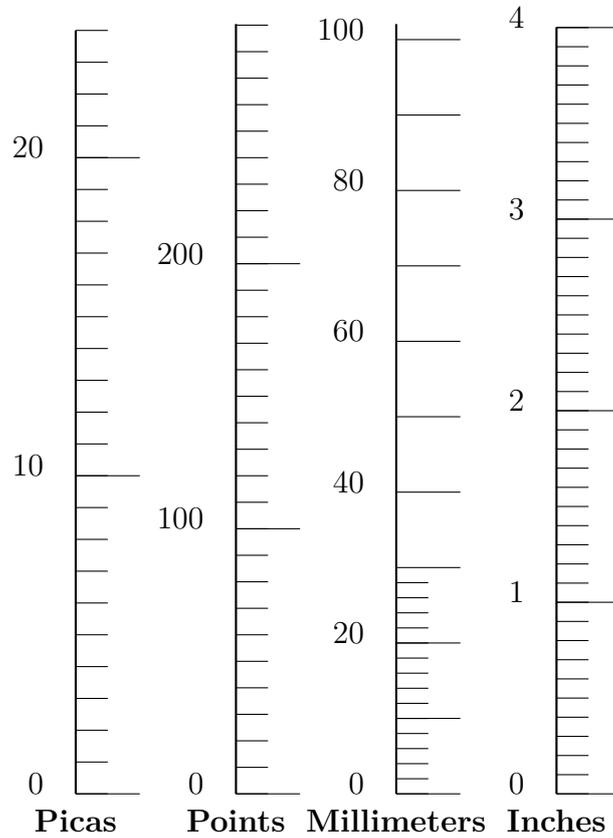


Figura 4.6: Some measuring scales.

# Parte II

## Pacotes gráficos

# Capítulo 5

## Pacote *Xy-pic*

Para gerar grafos e diagramas, pode-se utilizar o pacote *Xy-pic*.

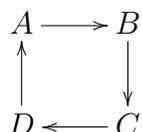
### 5.1 Exemplos com o pacote *Xy-pic*

A seguir, são realizados alguns testes, usando o pacote *Xy-pic*, com códigos encontrados em diferentes referências.

Código:

```
\begin{displaymath}
\xymatrix{ A \ar[r] & B \ar[d] \\
           D \ar[u] & C \ar[l] }
\end{displaymath}
```

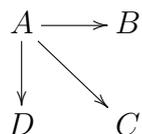
Resultado:



Código:

```
\begin{displaymath}
\xymatrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C }
\end{displaymath}
```

Resultado:



Código:

```
\begin{displaymath}
\mathrm{xmatrix}
{
A \mathrm{ar}[r]^f \mathrm{ar}[d]_g & B \mathrm{ar}[d]^{g'} \\
D \mathrm{ar}[r]_{f'} & C }
\end{displaymath}
```

Resultado:

$$\begin{array}{ccc}
A & \xrightarrow{f} & B \\
g \downarrow & & \downarrow g' \\
D & \xrightarrow{f'} & C
\end{array}$$

Código:

```
\begin{displaymath}
\mathrm{xmatrix}
{ \bullet \mathrm{ar}@\sim/[r] \mathrm{ar}@\_/@{.>}[r] & \bullet }
\end{displaymath}
```

Resultado:



# Capítulo 6

## Pacote TikZ

O pacote *TikZ/pgf* é um par de linguagens para produção de ilustrações técnicas e desenhos por meio de uma descrição geométrica que permite que o usuário produza imagens gráficas da mesma forma que se produz textos usando  $\text{\TeX}$ . Isso significa que, com *TikZ/pgf*, o usuário ganha todas as vantagens que o  $\text{\TeX}$  traz para criação de documentos, só que para usá-las em gráficos, como a velocidade para criar imagens simples, o uso de macros e a precisão no posicionamento. Porém, o pacote também herda as desvantagens do  $\text{\TeX}$ , como a difícil curva de aprendizado, o longo tempo de recompilação após fazer pequenas mudanças e a dificuldade de visualizar a imagem pelo código.

### 6.1 Definição dos pacotes *TikZ* e *pgf*

O *pgf* é um pacote que criou a base de comandos em  $\text{\TeX}$  para a construção de gráficos “*inline*” e o pacote *TikZ* é um conjunto de macros, construídos a partir do *pgf*, para expandir as possibilidades dos pacotes e simplificar a usabilidade deles.

### 6.2 Importação do pacote

Como todo pacote em  $\text{\LaTeX}$ , para usá-lo primeiro é necessário importá-lo. Para isso, é preciso inserir o comando `\usepackage{tikz}` antes do comando `\begin{document}`, no início do arquivo principal.

### 6.3 Comando `\tikz`

Uma das formas de gerar gráficos simples pelo *TikZ* é através do comando `\tikz`. Para criar imagens dessa forma deve-se usar o comando `\tikz` seguido dos comandos e parâmetros básicos do pacote na mesma linha.

### 6.4 Ambiente *tikzpicture*

Uma forma alternativa, que possibilita criar imagens mais complexas com o *TikZ*, é por meio do ambiente *tikzpicture*. Para iniciá-lo, assim como em outros ambientes em  $\text{\LaTeX}$ , deve-se usar o comando `\begin{tikzpicture}`. Para finalizá-lo, usa-se o comando `\end{tikzpicture}`.

## 6.5 Alguns comandos básicos

A seguir, são brevemente descritos alguns dos comandos mais básicos do pacote.

### 6.5.1 Comando `\draw`

O comando básico em um ambiente *tikzpicture* é o comando `\draw`. Esse comando irá desenhar uma linha entre duas coordenadas fornecidas pelo usuário. Essa linha poderá ser uma linha reta, se as coordenadas estiverem separadas por “-”. Aproximada por uma Curva de Bézier [Wik], pode-se obter uma linha curva, se as coordenadas estiverem separadas por “. *controls*”, seguido de uma ou de duas outras coordenadas, que servirão como pontos de controle, e, finalmente, finalizado por “.”.

O *TikZ* também possui parâmetros que facilitam a criação de estruturas mais comuns. Por exemplo, para construir um círculo, basta colocar o comando “*circle*” depois de uma coordenada “(x,y)” no comando `\draw`, que será o centro do círculo, seguido de “[radius=Rpt]”, onde “R” é o tamanho do raio do círculo, na unidade *point* (pt). Seguindo essa mesma regra, também se consegue construir elipses, colocando “*ellipse*” no lugar de “*circle*” e acrescentado mais um parâmetro no raio. Da mesma forma, pode-se construir retângulos, usando “*rectangle*” no lugar de “*circle*” e colocando duas coordenadas extras, que serão os cantos do retângulo, no lugar do raio.

Outros parâmetros podem ser adicionados ao comando `\draw`, para alterar a estrutura das linhas. Esses parâmetros podem alterar a espessura da linha, como o *width*, ou a cor, como o *color*, ou ainda outras parâmetros que controlam as estruturas das linhas.

### 6.5.2 Comando `\fill`

O comando `\fill` pode ser usado como uma alternativa para o comando `\draw`, quando se quer preencher as formas geométricas criadas. Para isso deve-se colocar `\fill` seguido da mistura de cores que se quer preencher, entre colchetes e, depois, parâmetros geométricos, como feito com o comando `\draw`.

Um comando similar ao `\fill` é o comando `\filldraw` que desenha a imagem e já a preenche com cor. Esse comando permite ao usuário usar diferentes cores nas bordas, feitas pela parte do *draw* do comando, e na parte interna, feita pela parte do *fill* do comando.

### 6.5.3 Comando `\shade`

O comando `\shade` é usado para criar uma transição entre cores dentro das imagens geométricas. Para usá-lo, o usuário deve usar o `\shade` seguido de comandos e parâmetros geométricos como no `\draw`. Dessa forma, será criada a imagem e, dentro dela, será feita uma transição de cinza para branco, como padrão. Para trocar as cores, é preciso especificar as cores e, como elas, misturam-se dentro de colchetes, após o `\shade`.

Como no comando `\fill`, também é possível usar o comando `\shadedraw` para conseguir controlar as bordas da figura separadamente de seu interior.

## 6.6 Exemplos com o pacote *TikZ/pgf*

Os códigos apresentados a seguir foram obtidos a partir de [Chr11].

### 6.6.1 Exemplos simples

A seguir, são apresentados alguns exemplos simples, usando o pacote *TikZ/pgf*.

Código:

```
\begin{tikzpicture}
\draw[line width=2pt,color=red]
(1,0) -- (0,0) -- (0,1);
\end{tikzpicture}
```

Resultado:



Código:

```
\begin{tikzpicture}
\draw[line width=2pt,color=red]
(1,0) -- (0,0) -- (0,1) -- cycle;
\end{tikzpicture}
```

Resultado:



Código:

```
\begin{tikzpicture}
\fill[line width=1pt,color=blue]
(0,0) {[rounded corners]
-- (0,1)} -- (1,0);
\end{tikzpicture}
```

Resultado:



Código:

```
\begin{tikzpicture}
\fill[line width=1pt,color=blue]
(0,0) {[rounded corners]
-- (0,1) -- (1,0)} -- cycle;
\end{tikzpicture}
```

Resultado:



Código:

```
\begin{tikzpicture}
\fill[line width=1pt,color=blue]
(0,0) rectangle (2,1);
\end{tikzpicture}
```

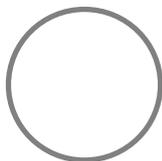
Resultado:



Código:

```
\begin{tikzpicture}
\draw[line width=2pt,color=gray]
(0,0) circle[radius=1];
\end{tikzpicture}
```

Resultado:

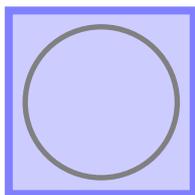


### 6.6.2 Exemplos usando bibliotecas

A seguir, são apresentados alguns exemplos, usando bibliotecas do pacote *TikZ*, com códigos encontrados em diferentes referências.

Código:

```
\begin{tikzpicture}
[background rectangle/.style={fill=blue!20,
draw=blue!50,line width=3pt},
show background rectangle]
\draw[line width=2pt,color=gray]
(2,2) circle[radius=1];
\end{tikzpicture}
```



Resultado:

Código:

```
\begin{tikzpicture}[decoration=Koch
snowflake,draw=blue,fill=blue!20,thick]
\filldraw decorate{ decorate{ (0,0) --
++(60:1) -- ++(-60:1) -- cycle}};
\end{tikzpicture}
```



Resultado:

## 6.7 Outros exemplos com o pacote *TikZ/pgf*

A seguir, são apresentados alguns exemplos, usando o pacote *Tikz*, com códigos encontrados em diferentes referências.

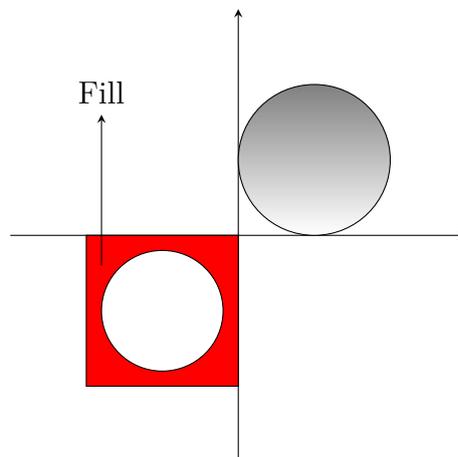
Código:

```
Bola azul
%
\tikz \fill[blue] (1ex,1ex) circle (1ex);
%
no meio do texto.
```

Resultado: Bola azul ● no meio do texto.

Código:

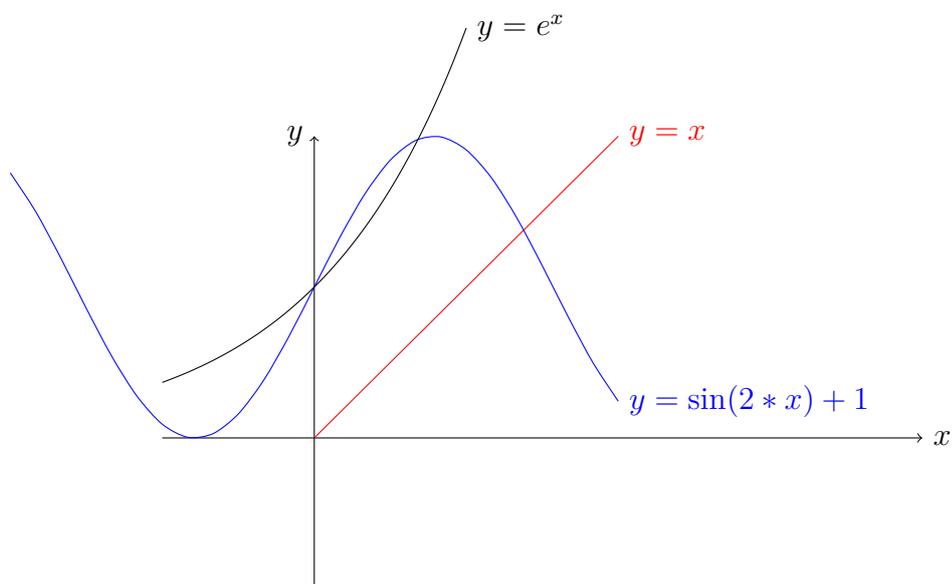
```
\begin{tikzpicture}[scale=2, >=stealth]
\draw [->] (-1.5,0) -- (1.5,0);
\draw [->] (0,-1.5) -- (0,1.5);
\shadedraw (0.5,0.5) circle (0.5cm);
%Filling
\filldraw[fill=red,even odd rule]
(-1,-1) rectangle (0,0)
(-0.5,-0.5) circle (0.4cm);
\draw[->] (-0.9,-0.2) -- +(0,1)
[above] node{Fill};
\end{tikzpicture}
```



Resultado:

Código:

```
\begin{tikzpicture}[scale=2]
\draw[>-] (-1,0) -- (4,0) node[right] {$x$};
\draw[>-] (0,-1) -- (0,2) node[left] {$y$};
% Domain: domain = a:b
\draw[smooth, domain = 0:2, color=red]
plot (\x,\x) node[right] {$y = x$};
\draw[smooth, domain = -2:2, color=blue]
%\x r means radian
plot (\x,{sin(2*\x r)+1}) node[right] {$y = \sin(2*x)+1$};
\draw[smooth, domain = -1:1, color=black]
plot (\x,{exp(\x)}) node[right] {$y = e^x$};
\end{tikzpicture}
```

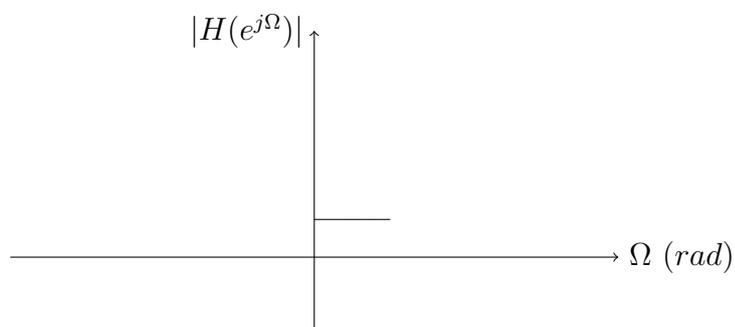


Resultado:

Código:

```
\begin{tikzpicture}[%scale=2,
  declare function={
    num(\w) = 0.5 + 0.1*exp(1*\w);
    den(\w) = 1.0 + 0.2*exp(1*\w);
    h_mod(\w) = abs(num(\w) / den(\w));
%    h_ang(\w) = angle(num(\w) / den(\w));
  }
]
%
\draw[->] (-4, 0) -- (4,0) node[right] {\Omega \ (rad)};
\draw[->] ( 0,-1) -- (0,3) node[left]  {\| H(e^{j \Omega}) |};
% Domain: domain = a:b
\draw[smooth, domain = 0:1, samples= 120, color=black]
plot (\x,{h_mod(\x)});
\end{tikzpicture}
```

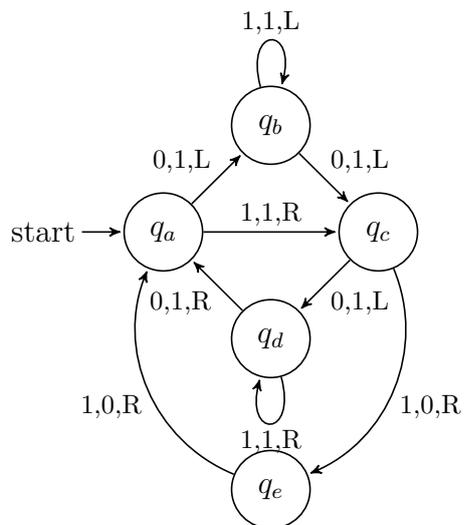
Resultado:



Código:

```
\begin{tikzpicture}[->,>=stealth',shorten >=1pt,%
                    auto,node distance=2cm,on grid,semithick,
                    inner sep=2pt,bend angle=45]
\node[initial,state] (A) {$q_a$};
\node[state] (B) [above right=of A] {$q_b$};
\node[state] (D) [below right=of A] {$q_d$};
\node[state] (C) [below right=of B] {$q_c$};
\node[state] (E) [below=of D] {$q_e$};

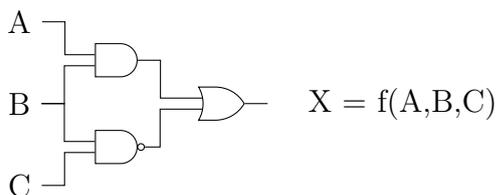
\path [every node/.style={font=\footnotesize}]
(A) edge node {0,1,L} (B)
    edge node {1,1,R} (C)
(B) edge [loop above] node {1,1,L} (B)
    edge node {0,1,L} (C)
(C) edge node {0,1,L} (D)
    edge [bend left] node {1,0,R} (E)
(D) edge [loop below] node {1,1,R} (D)
    edge node {0,1,R} (A)
(E) edge [bend left] node {1,0,R} (A);
\end{tikzpicture}
```



Resultado:

Código:

```
\begin{tikzpicture}[%circuit logic CDH,
                    circuit logic US,
                    tiny circuit symbols,
                    every circuit symbol/.style={
                    fill=white,draw}]
    \matrix[column sep=7mm]
    {
    \node (i0) {A}; & & & \\
    & \node [and gate] (a1) {}; & & \\
    \node (i1) {B}; & & \node [or gate] (o) {}; & \\
    & & \node (out) {X = f(A,B,C)}; & \\
    & \node [nand gate] (a2) {}; & & \\
    \node (i2) {C}; & & & \\
    };
    \draw (i0.east) -- ++(right:3mm) |- (a1.input 1);
    \draw (i1.east) -- ++(right:3mm) |- (a1.input 2);
    \draw (i1.east) -- ++(right:3mm) |- (a2.input 1);
    \draw (i2.east) -- ++(right:3mm) |- (a2.input 2);
    \draw (a1.output) -- ++(right:3mm) |- (o.input 1);
    \draw (a2.output) -- ++(right:2mm) |- (o.input 2);
    \draw (o.output) -- ++(right:3mm);
```



Resultado:

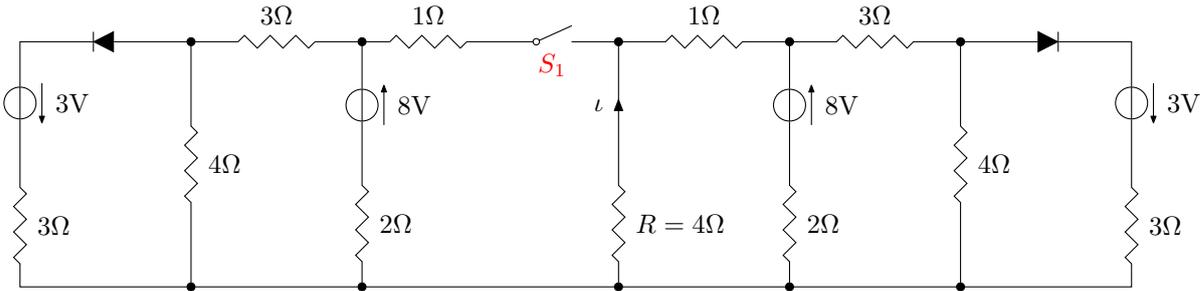
Código:

```

\begin{tikzpicture}[
    %circuit ee IEC,x=3cm,y=2cm,semithick,
    rotate=-90,circuit ee IEC,x=3.25cm,y=2.25cm,
    every info/.style={font=\footnotesize},
    small circuit symbols,
    set resistor graphic=var resistor IEC graphic,
    set diode graphic=var diode IEC graphic,
    set make contact graphic= var make contact IEC graphic]
% Let us start with some contacts:
\foreach \contact/\y in {1/1,2/2,3/3.5,4/4.5,5/5.5}
{
    \node [contact] (left contact \contact) at (0,\y) {};
    \node [contact] (right contact \contact) at (1,\y) {};
}
\draw (right contact 1) -- (right contact 2) -- (right contact 3)
      -- (right contact 4) -- (right contact 5);
\draw (left contact 1) to [diode] ++(down:1)
      to [voltage source={near start,
        direction info={volt=3}},
        resistor={near end,ohm=3}] ++(right:1)
      to (right contact 1);
\draw (left contact 1) to [resistor={ohm=4}] (right contact 1);
\draw (left contact 1) to [resistor={ohm=3}] (left contact 2);
\draw (left contact 2) to [voltage source={near start,
        direction info={<-,volt=8}},
        resistor={ohm=2,near end}] (right contact 2);
\draw (left contact 2) to [resistor={near start,ohm=1},
        make contact={near end,info'={[red]$S_1$}}]
      (left contact 3);
\draw (left contact 3) to [current direction'={near start,info=$\iota$},
        resistor={near end,info={\$R=4\Omega$}}]
      (right contact 3);
\draw (left contact 4) to [voltage source={near start,
        direction info={<-,volt=8}},
        resistor={ohm=2,near end}] (right contact 4);
\draw (left contact 3) to [resistor={ohm=1}] (left contact 4);
\draw (left contact 4) to [resistor={ohm=3}] (left contact 5);
\draw (left contact 5) to [resistor={ohm=4}] (right contact 5);
\draw (left contact 5) to [diode] ++(up:1)
      to [voltage source={near start,
        direction info={volt=3}},
        resistor={near end,ohm=3}] ++(right:1)
      to (right contact 5);
\end{tikzpicture}

```

Resultado:



Código:

```

\begin{tikzpicture}
  [point/.style={coordinate},>stealth',thick,draw=black!50,
  tip/.style={->,shorten >=1pt},
  every join/.style={rounded corners},
  hv path/.style={to path={|-| (\tikztotarget)}}},
  vh path/.style={to path={|- (\tikztotarget)}}}]

\matrix[column sep=4mm] {
  % First row:
  & & & & & & & & & \node (plus) [terminal] {+};\\
  % Second row:
  \node (p1) [point] {}; & \node (ui1) [nonterminal] {unsigned integer}; & & & & & & & & & \\
  \node (p2) [point] {}; & \node (dot) [terminal] {.}; & & & & & & & & & \\
  \node (p3) [point] {}; & \node (digit) [terminal] {digit}; & & & & & & & & & \\
  \node (p4) [point] {}; & \node (p5) [point] {}; & & & & & & & & & \\
  \node (p6) [point] {}; & \node (e) [terminal] {E}; & & & & & & & & & \\
  \node (p7) [point] {}; & & & & & & & & & & \\
  \node (p8) [point] {}; & \node (ui2) [nonterminal] {unsigned integer}; & & & & & & & & & \\
  \node (p9) [point] {}; & \node (p10) [point] {};\\
  % Third row:
  & & & & & & & & & & \node (minus)[terminal] {-};\\
};

{ [start chain]
  \chainin (p1);
  \chainin (ui1) [join=by tip];
  \chainin (p2) [join];
  \chainin (dot) [join=by tip];
  \chainin (p3) [join];
  \chainin (digit) [join=by tip];
  \chainin (p4) [join];
  { [start branch=digit loop]
    \chainin (p3) [join=by {skip loop=-6mm,tip}];
  }
  \chainin (p5) [join,join=with p2 by {skip loop=6mm,tip}];
  \chainin (p6) [join];
  \chainin (e) [join=by tip];
  \chainin (p7) [join];
  { [start branch=plus]
    \chainin (plus) [join=by {vh path,tip}];
    \chainin (p8) [join=by {hv path,tip}];
  }
  { [start branch=minus]
    \chainin (minus) [join=by {vh path,tip}];
    \chainin (p8) [join=by {hv path,tip}];
  }
  \chainin (p8) [join];
}

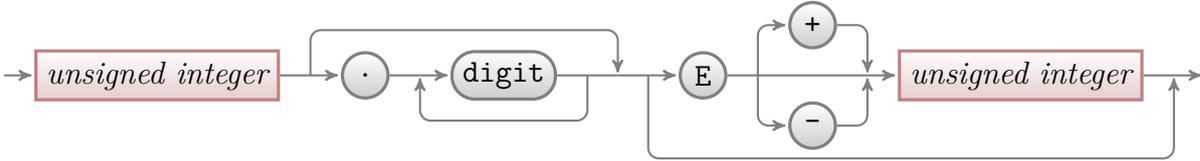
```

```

\chainin (ui2) [join=by tip];
\chainin (p9) [join,join=with p6 by {skip loop=-11mm,tip}];
\chainin (p10) [join=by tip];
}
\end{tikzpicture}

```

Resultado:



## 6.8 Exemplos para Apostila de Processamento Digital de Sinais

A seguir, são apresentados alguns testes, que foram elaborados para sua inserção em [Ale23b], usando o pacote *TikZ*.

Código:

```
\begin{tikzpicture}
  [point/.style={coordinate},>=stealth',thick,draw=black!50,
  tip/.style={->,shorten >=1pt},
  every join/.style={rounded corners},
  hv path/.style={to path={|-| (\tikztotarget)}}},
  vh path/.style={to path={|- (\tikztotarget)}}]

\matrix[column sep=6mm] {
  % First row:
  \node (inp) [terminal] {x[n]}; & \node (p100) [point] {}; &
  \node (b0) [sqr_box] {b0}; & \node (p101) [point] {}; &
  \node (sum01) [terminal] {+}; & \node (p102) [point] {}; &
  \node (out) [terminal] {y[n]};
  \\
  % Second row:
  & \node (z1) [sqr_box] {$D^{-1}$}; &
  & &
  & \node (p201) [point] {}; &
  & &

  \\
  % Third row:
  & \node (p300) [point] {}; &
  \node (b1) [sqr_box] {b1}; & \node (p301) [point] {}; &
  \node (sum12) [terminal] {+}; & &

  \\
  % Forth row:
  & \node (z2) [sqr_box] {$D^{-1}$}; &
  & &
  & \node (p401) [point] {}; &
  & &

  \\
  % Fifth row:
  & \node (p500) [point] {}; &
  \node (b2) [sqr_box] {b2}; & \node (p501) [point] {}; &
  \node (no_sum) [point] {}; & &

  \\
};
```

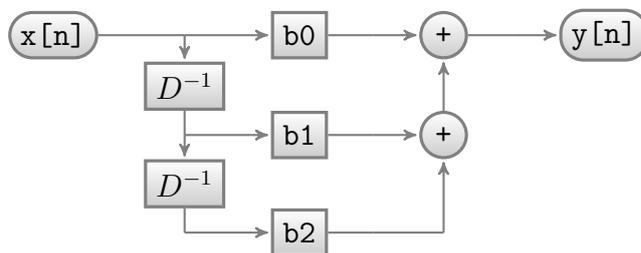
```

{ [start chain]
  \chainin (inp)      ;
  \chainin (p100)    [join];
  \chainin (b0)      [join=by tip];
  %
  \chainin (p101)    [join];
  \chainin (sum01)   [join=by tip];
  %
  \chainin (p102)    [join];
  \chainin (out)     [join=by tip];

  \chainin (p100)    ;
  \chainin (z1)      [join=by tip];
  \chainin (p300)    [join];
  \chainin (b1)      [join=by tip];
  %
  \chainin (p301)    [join];
  \chainin (sum12)   [join=by tip];
  %
  \chainin (sum01)   [join=by tip];

  \chainin (p300)    ;
  \chainin (z2)      [join=by tip];
  \chainin (p500)    [join];
  \chainin (b2)      [join=by tip];
  %
  \chainin (p501)    [join];
  \chainin (no_sum)  [join];
  %
  \chainin (sum12)   [join=by tip];
}
\end{tikzpicture}

```



Resultado:

Código:

```

\begin{tikzpicture}
  [point/.style={coordinate},>=stealth',thick,draw=black!50,
  tip/.style={->,shorten >=1pt},
  every join/.style={rounded corners},
  hv path/.style={to path={|-| (\tikztotarget)}}},
  vh path/.style={to path={|- (\tikztotarget)}}}]

\matrix[column sep=6mm] {
  % First row:
  \node (inp)    [terminal] {x[n]}; & \node (p100) [point] {}; &
  \node (b0)    [sqr_box] {b0}; & \node (p101) [point] {}; &
  \node (sum01) [terminal] {+}; & \node (p102) [point] {}; &
  \node (out)   [terminal] {y[n]};
  \\
  % Second row:
  & \node (z1)    [sqr_box] {$D^{-1}$}; &
  & &
  & \node (p201) [point] {}; &
  & &

  \\
  % Third row:
  & \node (p300) [point] {}; &
  \node (b1)    [sqr_box] {b1}; & \node (p301) [point] {}; &
  \node (sum12) [terminal] {+}; & &

  \\
  % Forth row:
  & \node (z2)    [sqr_box] {$D^{-1}$}; &
  & &
  & \node (p401) [point] {}; &
  & &

  \\
  % Fifth row:
  & \node (p500) [point] {}; &
  \node (b2)    [sqr_box] {b2}; & \node (p501) [point] {}; &
  \node (no_sum) [point] {}; & &

  \\
};

{ [start chain]
  \chainin (inp)    ;
  \chainin (p100)   [join];
  \chainin (b0)     [join=by tip];
  %

```

```

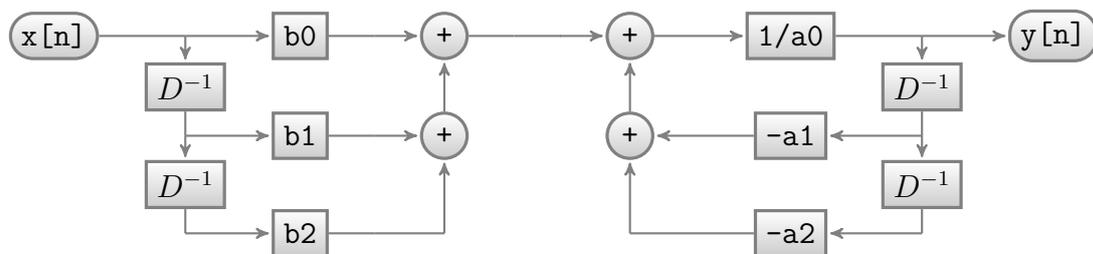
\chainin (p101) [join];
\chainin (sum01) [join=by tip];
%
\chainin (p102) [join];
\chainin (out) [join=by tip];

\chainin (p100) ;
\chainin (z1) [join=by tip];
\chainin (p300) [join];
\chainin (b1) [join=by tip];
%
\chainin (p301) [join];
\chainin (sum12) [join=by tip];
%
\chainin (sum01) [join=by tip];

\chainin (p300) ;
\chainin (z2) [join=by tip];
\chainin (p500) [join];
\chainin (b2) [join=by tip];
%
\chainin (p501) [join];
\chainin (no_sum) [join];
%
\chainin (sum12) [join=by tip];
}
\end{tikzpicture}

```

Resultado:



## 6.9 Referências

Mais detalhes sobre os tópicos abordados neste capítulo podem ser encontrados nas seguintes referências: [Ovea], [Oveb], [Ovec], [Oved], [Ovee], [Proa], [Prob], [Proc], [Prod], e [Proe].

# Capítulo 7

## Pacote *PgfPlots*

O pacote *PgfPlots* é baseado no pacote *TikZ*, sendo dedicado à geração de gráficos.

### 7.1 Exemplos com o pacote *PgfPlots*

A seguir, são realizados alguns testes, usando o pacote *PgfPlots*, com códigos encontrados em diferentes referências.

Código:

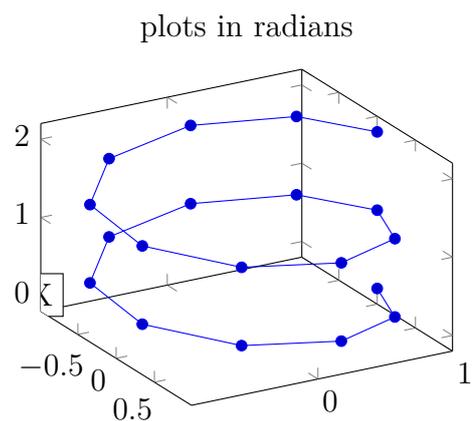
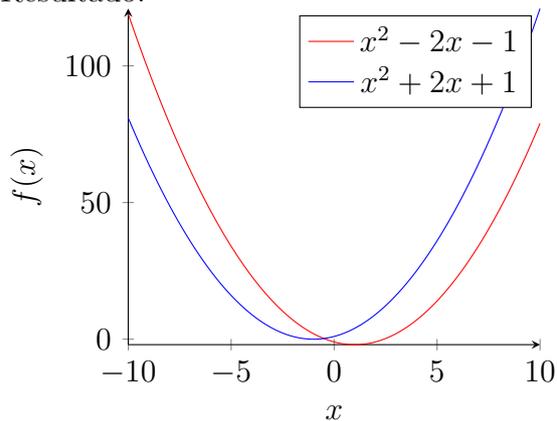
```
\begin{tikzpicture}
%
\begin{axis}[
axis lines = left ,
xlabel = $x$ ,
ylabel = {$f(x)$} ,
]
%
% Aqui definimos a parabola vermelha
%
\addplot[
domain = -10:10 ,
samples = 100 ,
color = red ,
]
{x^2 - 2*x - 1};
%
\addlegendentry{$x^2 - 2x - 1$}
%
% Aqui definimos a parabola azul
%
\addplot[
domain = -10:10 ,
samples =100 ,
color = blue ,
]
{x^2 + 2*x + 1};
```

```

%
\addlegendentry{ $x^2 + 2x + 1$ }
%
\end{axis}
%
\end{tikzpicture}
%
%
\hskip 1.5cm
%
%
\begin{tikzpicture}
%
\begin{axis}[view={60}{30},trig format plots=rad,
title=plots in radians,
]
%
\addplot3+ [domain=0:4*pi,samples=19,samples y=1]
({sin(x)},
{cos(x)},
{2*x/(4*pi)});
%
% drawing instructions still use PGF's default
%
\node [fill=white,draw=black,anchor=center] at
({sin(90)},{cos(90)},1) {X};
\end{axis}
%
\end{tikzpicture}

```

Resultado:



# Parte III

## Aplicações

# Capítulo 8

## Mapa de Karnaugh

Este capítulo trata de pacotes  $\text{\LaTeX}$  especializados para a geração de mapa de Karnaugh, que é uma ferramenta gráfica que pode ser utilizada tanto para apenas representar uma função lógica quanto para realizar um processo de simplificação sobre ela.

### 8.1 Circuitos digitais

Circuitos digitais podem ser classificados, de acordo com a existência de memória, em circuitos digitais combinacionais (sem memória) e em circuitos digitais sequenciais (com memória).

Em circuitos digitais convencionais, as variáveis manipuladas são variáveis lógicas. Mais particularmente, emprega-se a lógica clássica ou binária.

Em um circuito digital combinacional, uma saída simples pode ser pensada como um mapeamento (ou uma função) das suas variáveis de entrada em tal variável de saída. Tais funções podem ser representadas de diversas formas diferentes.

A fim de tentar diminuir o tamanho do circuito que implementa uma função, procura-se tentar simplificá-la, empregando-se algum dos possíveis processos de simplificação aplicáveis neste caso.

### 8.2 Funções lógicas

Uma possível representação gráfica para uma função lógica é uma tabela, denominada de Tabela Verdade, uma vez que os valores envolvidos são VERDADEIRO (V) ou FALSO (F).

Uma possível representação algébrica padrão para uma função lógica, é uma combinação de termos padrões fundamentais, que, por sua vez, são combinações de todas as variáveis de entrada da função.

Os termos padrões fundamentais são denominados de MINTERMOS e de MAXTERMOS.

As combinações padrões possíveis são denominadas de SOMA DE MINTERMOS e de PRODUTO DE MAXTERMOS.

Os MINTERMOS são aqueles que, correspondentemente, geram o valor lógico V na Tabela Verdade. Por sua vez, os MAXTERMOS são aqueles que geram o valor lógico F na Tabela Verdade.

Um processo de simplificação bem conhecido utiliza, como ponto de partida, uma das duas formas padrões. Em seguida, procura-se sistematicamente por pares de combinações, onde todas as variáveis envolvidas são iguais, exceto uma delas. Tais pares são aglutinados em uma única combinação final menor, composta apenas pelas variáveis originalmente iguais. Com isso, elimina-se uma variável e um termo, simplificando-se a equação.

## 8.3 Mapa de Karnaugh

O mapa de Karnaugh que é uma ferramenta gráfica que pode ser utilizada tanto para apenas representar uma função lógica quanto para realizar um processo de simplificação sobre ela.

Em essência, o mapa de Karnaugh é uma Tabela Verdade, reorganizada na forma de uma matriz. Logo, ele é mais uma forma de representar uma função lógica.

No entanto, a reorganização do mapa tem o propósito de colocar lado a lado, tanto na horizontal quanto na vertical, as posições dos termos que produzem aglutinações. Portanto, torna-se mais fácil encontrar visualmente as aglutinações possíveis.

Pelas características apresentadas, o mapa de Karnaugh é utilizado tanto no ensino de circuitos digitais quanto no projeto de circuitos de baixa complexidade.

Na prática, só é humanamente viável utilizar o mapa de Karnaugh para simplificar funções de até seis variáveis. Acima disso, devem-se empregar algoritmos computacionais.

## 8.4 Geração de mapas para textos didáticos

Na produção de textos didáticos, precisa-se não apenas desenhar a matriz do mapa de Karnaugh, ou ainda preenchê-la, mas também identificar todas as variáveis, marcar todas as aglutinações possíveis e adicionar observações pertinentes.

Embora seja possível utilizar o ambiente *picture* para fazer tais desenhos, o custo do processo não é tão baixo.

Vários pacotes já foram propostos para facilitar o desenho de forma automática, usando código L<sup>A</sup>T<sub>E</sub>X. Alguns desses pacotes são apresentados e experimentados a seguir.

## 8.5 Exemplos de pacotes especializados

Alguns exemplos de pacotes especializados para a geração de mapas de Karnaugh, com os comentários originais das suas fontes, são os seguintes:

- Pacote **karnaughmap** [Mic15]: “ *This package provides an easy to use interface to typeset Karnaugh maps using TikZ. Though similar to the “karnaugh” macros, it provides a key-value system to customize karnaughmaps and a proper LaTeX package.*”
- Pacote **karnaugh-map** [Mat17]: “ *This package draws karnaugh maps with 2, 3, 4, 5, and 6 variables. It also contains commands for filling the karnaugh map with terms semi-automatically or manually. Last but not least it contains commands for drawing implicants on top of the map. This package depends on the TikZ, xparse, and xstring packages.*”
- Pacote **cartonaugh** [Jam21] (derivado do pacote **karnaugh-map**): “ *This package, a fork of 2pi’s karnaugh-map package, draws karnaugh maps with 2, 3, 4, 5, and 6 variables. It also contains commands for filling the karnaugh map with terms semi-automatically or manually. Last but not least it contains commands for drawing implicants on top of the map. The name “cartonaugh” is a portmanteau of “cartographer” and “karnaugh”. The package needs LuaLaTeX and depends on TikZ, xparse, and xstring.*”

- Macros **kvmacros** do pacote **karnaugh** [And02]: “*The package provides macros for typesetting Karnaugh-Maps and Veitch-Charts in a simple and user-friendly way. Karnaugh-Maps and Veitch-Charts are used to display and simplify logic functions “manually”. These macros can typeset Karnaugh-Maps and Veitch-Charts with up to ten variables (=1024 entries).*”
- Pacote **askmaps** [Jes20]: “*The package provides 1, 2, 3, 4 and 5 variable Karnaugh maps, in the style used in numerous American textbooks on digital design. The package draws K-maps where the most significant input variables are placed on top of the columns and the least significant variables are placed left of the rows.*”
- Pacote **kvmmap** [Ben20]: “*This LaTeX package allows the creation of (even large) Karnaugh maps. It provides a tabular-like input syntax and support for drawing bundles (implicants) around adjacent values. It is based on an answer at StackExchange.*”
- Pacote **tikz-karnaugh** [Lui21]: “*The tikz-karnaugh package is a LaTeX package used to draw Karnaugh maps. It uses TikZ to produce high quality graph up to 12 variables, but this limit depends on the TeX memory usage and can be different for you. It is very good for presentation since TikZ allows for a better control over the final appearance of the map. You can control colour, styles and distances. It can be considered as an upgrade of Andreas W. Wieland’s “karnaugh” package towards TikZ supporting. Also, complex maps with solution (prime implicants) pointed out can be generated with external java software. It supports both American and traditional styles, though American style requires a little extra effort.*”

Como deve ser esperado, diante de uma lista de pacotes diferentes, especializados na mesma aplicação, cada um desses pacotes possui as suas próprias vantagens e desvantagens. Uma breve comparação entre tais pacotes pode ser encontrada em [Ben20] [Lui21].

## 8.6 Exemplos com o pacote *askmaps*

A seguir, são realizados alguns testes, usando o pacote *askmaps*, encontrados em [Jes20].

Código:

```
\askmapii{F}{xy}{f}{0111}{%
%
\setlength\fbboxsep{0pt}\linethickness{0pt}
%
\put(1.1,0.1)%
{\pgfsetfillopacity{0.2}\colorbox{red}%
{\framebox(0.8,1.8){}}\pgfsetfillopacity{1}}%
%
\put(0.1,0.1)%
{\pgfsetfillopacity{0.2}\colorbox{blue}%
{\framebox(1.8,0.8){}}\pgfsetfillopacity{1}}%
%
}%
```

Resultado:

	x		
		0	1
y	0	0	1
1	1	1	1

Código:

```
\askmapiii{s}{abc}{f}{11100111}{%
\put(2.5,1.0){\oval(0.8,1.8)}%
\put(1.0,1.5){\oval(1.8,0.8)}%
\put(3.5,0.5){\oval(0.8,0.8)[l]}%
\put(3.5,0.9){\line(1,0){0.7}}%
\put(3.5,0.1){\line(1,0){0.7}}%
\put(0.5,0.5){\oval(0.8,0.8)[r]}%
\put(0.5,0.9){\line(-1,0){0.7}}%
\put(0.5,0.1){\line(-1,0){0.7}}%
}%
```

Resultado:

		ab			
		00	01	11	10
c	0	1	1	1	0
1	1	0	1	1	1

## 8.7 Exemplos com o pacote *kvmmap*

A seguir, são realizados alguns testes, usando o pacote *kvmmap*, encontrados em [Ben20].

Código:

```
\kvlist{2}{4}{0,1,0,0,0,0,0,1}{a,b,c}
```

Resultado:

	<i>a</i>		
<i>bc</i>		0	1
00		0	1
01		0	0
11		0	0
10		0	1

Código:

```
\begin{kvmatrix}{a,b,c}
0 & 1 & 1 & 0 \\
0 & 1 & 0 & 1 \\
\end{kvmatrix}
```

Resultado:

	<i>ab</i>				
<i>c</i>		00	01	11	10
0		0	1	1	0
1		0	1	0	1

Código:

```
\kvlist{4}{4}{0,1,1,0,0,1,0,1,0,1,1,1,0,0,1,1}{a,b,c,d}
```

Resultado:

	<i>ab</i>				
<i>cd</i>		00	01	11	10
00		0	1	1	0
01		0	1	0	1
11		0	1	1	1
10		0	0	1	1

Código:

```
\begin{kvmmap}
\kvlist{4}{4}{0,1,1,0,0,1,0,1,0,1,1,1,0,0,1,1}{a,b,c,d,e,f}
\bundle[color=red]{1}{1}{1}{2}
\bundle{3}{2}{2}{3}
\bundle[color=blue, reducespace=3pt]{3}{2}{3}{1}
\end{kvmmap}
```

Resultado:

$cd \backslash ab$	00	01	11	10
00	0	1	1	0
01	0	1	0	1
11	0	1	1	1
10	0	0	1	1

Código:

```
\begin{kvmmap}
\begin{kvmatrix}{a,b,c,d}
0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 1 & 1 & 1 \\
\end{kvmatrix}
\end{kvmmap}
\bundle{3}{3}{2}{3}
\bundle[color=blue]{3}{2}{3}{1}
\bundle[invert=true, reducespace=2pt, overlapmargins=6pt]{1}{0}{2}{3}
\bundle[invert=true, reducespace=2pt]{0}{1}{3}{1}
\end{kvmmap}
```

Resultado:

$cd \backslash ab$	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	0	0	0	1
10	0	1	1	1

## 8.8 Exemplos para Apostila de Circuitos Digitais

A seguir, são apresentados alguns testes, que foram elaborados para sua inserção em [Ale23a], usando o pacote *karnaugh-map* [Mat17] e o pacote *kvmap* [Ben20].

O mapa da Figura 8.1 (Figura 8.2) exemplifica a Equação (8.1), onde são realizadas as combinações  $(m_0 - m_4)$  e  $(m_3 - m_7)$ .

$$\begin{aligned} F(A, B, C) &= (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (A \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot C) + (A \cdot B \cdot C) \\ &= (\bar{B} \cdot \bar{C}) + (B \cdot C). \end{aligned} \quad (8.1)$$

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1		1	
	1	1		1	

Figura 8.1: Mapa de Karnaugh relativo à Equação (8.1), usando o pacote *karnaugh-map*.

		<i>AB</i>			
<i>C</i>		00	01	11	10
0	1	0	0	1	
1	0	1	1	0	

Figura 8.2: Mapa de Karnaugh relativo à Equação (8.1), usando o pacote *kvmap*.

O mapa da Figura 8.3 (Figura 8.4) exemplifica a Equação (8.2), onde o mintermo  $m_0$  é replicado para as combinações  $(m_0 - m_2)$  e  $(m_0 - m_4)$ .

$$\begin{aligned} F(A, B, C) &= (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot \bar{C}) + (A \cdot \bar{B} \cdot \bar{C}) \\ &= (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot \bar{C}) + (A \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot \bar{B} \cdot \bar{C}) \\ &= (\bar{A} \cdot \bar{C}) + (\bar{B} \cdot \bar{C}). \end{aligned} \quad (8.2)$$

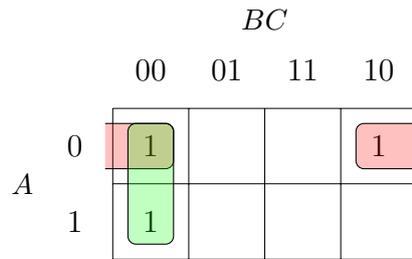


Figura 8.3: Mapa de Karnaugh relativo à Equação (8.2) usando o pacote *karnaugh-map*.

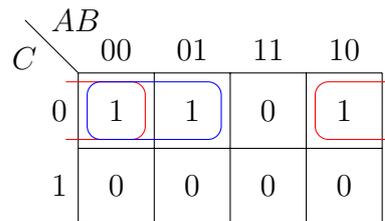


Figura 8.4: Mapa de Karnaugh relativo à Equação (8.2) usando o pacote *kvmap*.

A Figura 8.5 (Figura 8.6) apresenta o mapa da função dada por

$$\begin{aligned}
 F(A, B, C) &= \sum m(0, 2, 3, 5, 6, 7) \\
 &= (\bar{A} \cdot \bar{B} \cdot \bar{C}) + (\bar{A} \cdot B \cdot \bar{C}) + (\bar{A} \cdot B \cdot C) + \\
 &\quad (A \cdot \bar{B} \cdot C) + (A \cdot B \cdot \bar{C}) + (A \cdot B \cdot C) .
 \end{aligned}
 \tag{8.3}$$

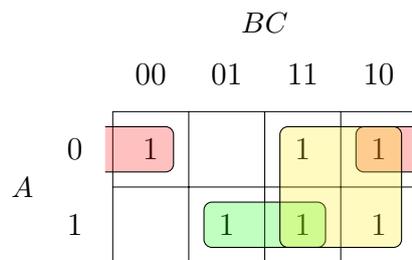


Figura 8.5: Mapa de Karnaugh relativo à Equação (8.3) usando o pacote *karnaugh-map*.

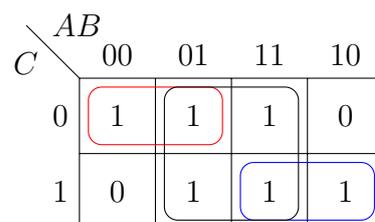


Figura 8.6: Mapa de Karnaugh relativo à Equação (8.3) usando o pacote *kvmap*.

A Figura 8.7 (Figura 8.8) apresenta o mapa de uma função com múltiplas formas mínimas, envolvendo o termo  $m_2$ . As Figuras 8.7a e 8.7b (Figuras 8.8a e 8.8b) respectivamente ilustram as duas soluções de mesmo custo, que são as seguintes: a)  $(m_0 - m_4), (m_3 - m_7), (m_0 - m_2)$  e b)  $(m_0 - m_4), (m_3 - m_7), (m_3 - m_2)$ .

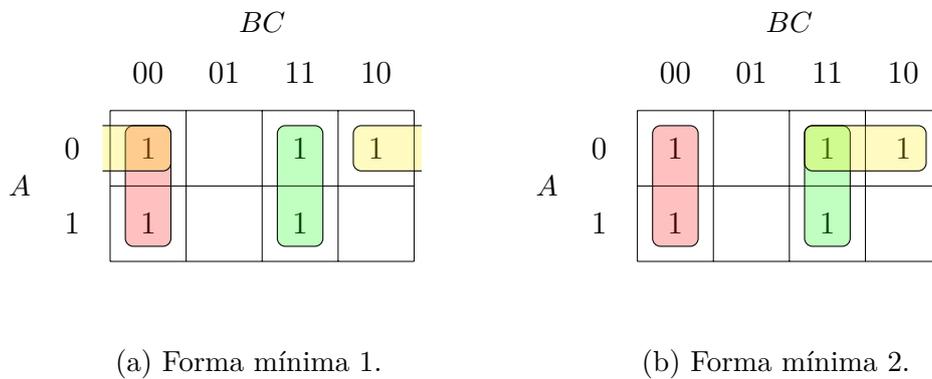


Figura 8.7: Mapa de Karnaugh com múltiplas formas mínimas usando o pacote *karnaugh-map*.

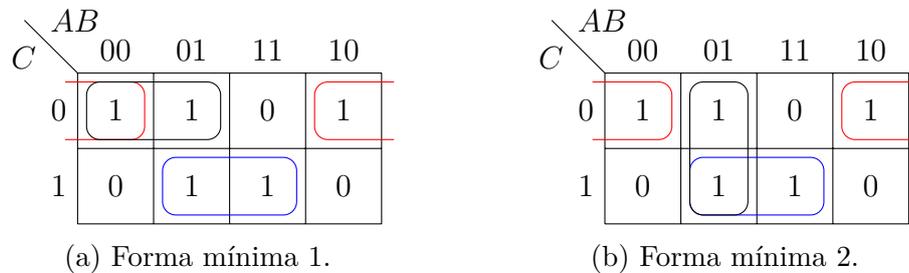


Figura 8.8: Mapa de Karnaugh com múltiplas formas mínimas usando o pacote *kvmmap*.

A Figura 8.9 (Figura 8.10) apresenta um mapa com ciclo, que possui duas soluções de mesmo custo. As Figuras 8.9a e 8.9b (Figuras 8.10a e 8.10b) respectivamente ilustram as duas soluções de mesmo custo, que são as seguintes: a)  $(m_0 - m_2), (m_3 - m_7), (m_4 - m_5)$  e b)  $(m_0 - m_4), (m_2 - m_3), (m_5 - m_7)$ .

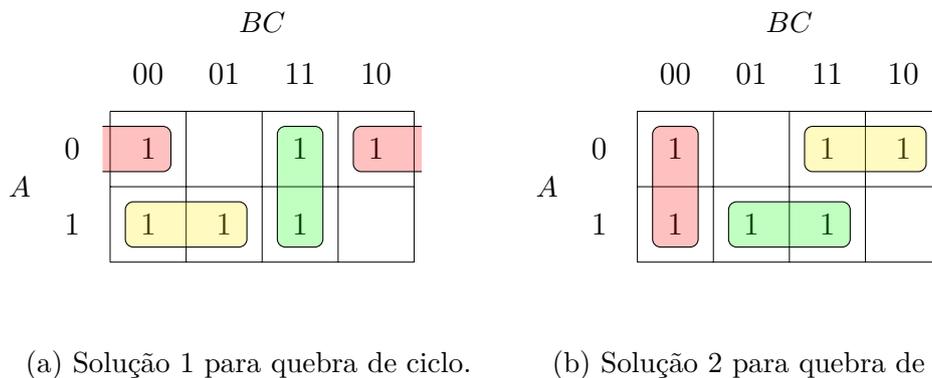
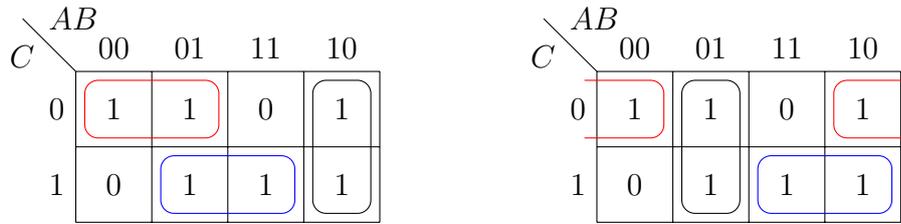


Figura 8.9: Mapa de Karnaugh com ciclo, usando o pacote *karnaugh-map*.



(a) Solução 1 para quebra de ciclo. (b) Solução 2 para quebra de ciclo.

Figura 8.10: Mapa de Karnaugh com ciclo, usando o pacote *kvmmap*.

A Tabela 8.1 exemplifica uma função incompletamente especificada, a qual também pode ser expressa por

$$F(A, B, C) = \sum m(0, 3, 4) + \sum d(2, 7) = \prod M(1, 5, 6) \cdot \prod d(2, 7) . \quad (8.4)$$

<i>Linha</i>	<i>A</i>	<i>B</i>	<i>C</i>	$F(A, B, C)$
0	0	0	0	1
1	0	0	1	0
2	0	1	0	<i>X</i>
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	<i>X</i>

Tabela 8.1: Tabela verdade de função incompletamente especificada.

As Figuras 8.11 a 8.13 (Figuras 8.14 a 8.16) ilustram, respectivamente, os mapas de Karnaugh da função definida na Equação (8.4), dos seus mintermos e dos seus maxtermos.

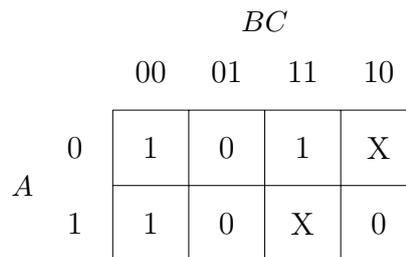


Figura 8.11: Mapa de Karnaugh da Tabela 8.1, usando o pacote *karnaugh-map*.

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1		1	X
	1	1		X	

(a) Forma mínima 1.

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0	1		1	X
	1	1		X	

(b) Forma mínima 2.

Figura 8.12: Mapa de Karnaugh dos mintermos da Tabela 8.1, usando o pacote *karnaugh-map*.

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0		0		X
	1		0	X	0

(a) Forma mínima 1.

		<i>BC</i>			
		00	01	11	10
<i>A</i>	0		0		X
	1		0	X	0

(b) Forma mínima 2.

Figura 8.13: Mapa de Karnaugh dos maxtermos da Tabela 8.1, usando o pacote *karnaugh-map*.

		$AB$			
		00	01	11	10
$C$	0	1	$X$	0	1
	1	0	1	$X$	0

Figura 8.14: Mapa de Karnaugh da Tabela 8.1, usando o pacote *kvmap*.

		$AB$			
		00	01	11	10
$C$	0	1	$X$		1
	1		1	$X$	

(a) Forma mínima 1.

		$AB$			
		00	01	11	10
$C$	0	1	$X$		1
	1		1	$X$	

(b) Forma mínima 2.

Figura 8.15: Mapa de Karnaugh dos mintermos da Tabela 8.1, usando o pacote *kvmap*.

		$AB$			
		00	01	11	10
$C$	0		$X$	0	
	1	0		$X$	0

(a) Forma mínima 1.

		$AB$			
		00	01	11	10
$C$	0		$X$	0	
	1	0		$X$	0

(b) Forma mínima 2.

Figura 8.16: Mapa de Karnaugh dos maxtermos da Tabela 8.1, usando o pacote *kvmap*.

## 8.9 Referências

Mais detalhes sobre os tópicos abordados neste capítulo podem ser encontrados nas seguintes referências: [Mic15], [Mat17], [Jam21], [Pesh], [And02], [Jes20], [Ben20] e [Lui21].

# Referências bibliográficas

- [Ale23a] Alexandre S. de la Vega. Apostila de Teoria para Circuitos Digitais. Disponível em: “[http://www.telecom.uff.br/~delavega/public/CircDig/apostila\\_teo\\_cd.pdf](http://www.telecom.uff.br/~delavega/public/CircDig/apostila_teo_cd.pdf)”, 2023. Acesso em: 01/02/2023.
- [Ale23b] Alexandre S. de la Vega. Apostila de Teoria para Fundamentos de Processamento Digital de Sinais. Disponível em: “[http://www.telecom.uff.br/~delavega/public/DSP/apostila\\_teo\\_dsp.pdf](http://www.telecom.uff.br/~delavega/public/DSP/apostila_teo_dsp.pdf)”, 2023. Acesso em: 01/02/2023.
- [And02] Andreas W. Wieland. **L<sup>A</sup>T<sub>E</sub>X macros *kvmacros* from package *karnaugh* – Typeset Karnaugh-Veitch-maps**. Disponível em: “<http://www.ctan.org/tex-archive/macros/latex/contrib/karnaugh>”, 2002. Acesso em: 01/02/2023.
- [Ben20] Ben Frank. **L<sup>A</sup>T<sub>E</sub>X package *kvmmap* – Create Karnaugh maps with L<sup>A</sup>T<sub>E</sub>X**. Disponível em: “<https://ctan.org/pkg/kvmmap>”, 2020. Acesso em: 01/02/2023.
- [Bera] Berkeley University. Curso de L<sup>A</sup>T<sub>E</sub>X: Berkeley. Disponível em: “<https://www.ocf.berkeley.edu/~latex/>”. Acesso em: 01/02/2023.
- [Berb] Berkeley University. Curso de L<sup>A</sup>T<sub>E</sub>X: Berkeley. Disponível em: “<https://www.ocf.berkeley.edu/~latex/files/syllabus.pdf>”. Acesso em: 01/02/2023.
- [Berc] Berkeley University. Curso de L<sup>A</sup>T<sub>E</sub>X: Berkeley. Disponível em: “<https://www.ocf.berkeley.edu/~latex/files/reader.pdf>”. Acesso em: 01/02/2023.
- [Berd] Berkeley University. Curso de L<sup>A</sup>T<sub>E</sub>X: Berkeley. Disponível em: “<https://www.ocf.berkeley.edu/~latex/files/beameruserguide.pdf>”. Acesso em: 01/02/2023.
- [Chr11] Christian Blanco and Brandon Eltiste. *L<sup>A</sup>T<sub>E</sub>X for Math and Science – Lecture 9 - TikZ and pgf*. Disponível em: “[https://www.ocf.berkeley.edu/~latex/files/tikz\\_lecture.pdf](https://www.ocf.berkeley.edu/~latex/files/tikz_lecture.pdf)”, 2011. Acesso em: 01/02/2023.
- [dak] dakusui HU. DITAA (*DIagrams Through Ascii Art*) (*GitHub*). Disponível em: “<https://github.com/dakusui/latex-ditaa>”. Acesso em: 01/02/2023.
- [Gru] Grupo PET-Tele. **Webpage do Grupo PET-Tele**. Disponível em: “<http://www.telecom.uff.br/pet>”. Acesso em: 01/02/2023.
- [Hir] Hiroshi Ukai. DITAA (*DIagrams Through Ascii Art*) (*GitHub*). Disponível em: “<https://ctan.org/pkg/ditaa>”. Acesso em: 01/02/2023.
- [Hir18] Hiroshi Ukai. DITAA (*DIagrams Through Ascii Art*) (*GitHub*). Disponível em: “<https://ctan.math.illinois.edu/macros/latex/contrib/ditaa/ditaa.pdf>”, 2018. Acesso em: 01/02/2023.

- [Jam21] Jamal Bouajjaj. **L<sup>A</sup>T<sub>E</sub>X package *cartonaugh* – A LuaL<sup>A</sup>T<sub>E</sub>X package for drawing karnaugh ps with up to 6 variables**. Disponível em: “<https://ctan.org/pkg/cartonaugh>”, 2021. Acesso em: 01/02/2023.
- [Jes20] Jesse op den Brouw. **L<sup>A</sup>T<sub>E</sub>X package *askmaps* – Typeset American style Karnaugh maps**. Disponível em: “<https://ctan.org/pkg/askmaps>”, 2020. Acesso em: 01/02/2023.
- [Lui21] Luis Paulo Laus. **L<sup>A</sup>T<sub>E</sub>X package *tikz-karnaugh* – Typeset Karnaugh maps using TikZ**. Disponível em: “<https://ctan.org/pkg/tikz-karnaugh>”, 2021. Acesso em: 01/02/2023.
- [Mat17] Mattias Jacobsson. **L<sup>A</sup>T<sub>E</sub>X package *karnaugh-map* – L<sup>A</sup>T<sub>E</sub>X package for drawing karnaugh maps with up to 6 variables**. Disponível em: “<https://ctan.org/pkg/karnaugh-map>”, 2017. Acesso em: 01/02/2023.
- [Mic15] Michael Vonbun. **L<sup>A</sup>T<sub>E</sub>X package *karnaughmap* – Typeset Karnaugh maps**. Disponível em: “<https://ctan.org/pkg/karnaughmap>”, 2015. Acesso em: 01/02/2023.
- [MiK] MiKTeX (*website*). MiKTeX Packages A-Z. Disponível em: “<https://miktex.org/pkg/az>”. Acesso em: 01/02/2023.
- [Ovea] Overleaf (*website*). Curso sobre Tikz - Parte 1. Disponível em: “[https://pt.overleaf.com/learn/latex/LaTeX\\_Graphics\\_using\\_TikZ%3A\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_1\)%E2%80%94Basic\\_Drawing](https://pt.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_1)%E2%80%94Basic_Drawing)”. Acesso em: 01/02/2023.
- [Oveb] Overleaf (*website*). Curso sobre Tikz - Parte 2. Disponível em: “[https://pt.overleaf.com/learn/latex/LaTeX\\_Graphics\\_using\\_TikZ%3A\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_2\)%E2%80%94Generating\\_TikZ\\_Code\\_from\\_GeoGebra](https://pt.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_2)%E2%80%94Generating_TikZ_Code_from_GeoGebra)”. Acesso em: 01/02/2023.
- [Ovec] Overleaf (*website*). Curso sobre Tikz - Parte 3. Disponível em: “[https://pt.overleaf.com/learn/latex/LaTeX\\_Graphics\\_using\\_TikZ%3A\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_3\)%E2%80%94Creating\\_Flowcharts](https://pt.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_3)%E2%80%94Creating_Flowcharts)”. Acesso em: 01/02/2023.
- [Oved] Overleaf (*website*). Curso sobre Tikz - Parte 4. Disponível em: “[https://pt.overleaf.com/learn/latex/LaTeX\\_Graphics\\_using\\_TikZ%3A\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_4\)%E2%80%94Circuit\\_Diagrams\\_Using\\_Circuitikz](https://pt.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_4)%E2%80%94Circuit_Diagrams_Using_Circuitikz)”. Acesso em: 01/02/2023.
- [Ovee] Overleaf (*website*). Curso sobre Tikz - Parte 5. Disponível em: “[https://pt.overleaf.com/learn/latex/LaTeX\\_Graphics\\_using\\_TikZ%3A\\_A\\_Tutorial\\_for\\_Beginners\\_\(Part\\_5\)%E2%80%94Creating\\_Mind\\_Maps](https://pt.overleaf.com/learn/latex/LaTeX_Graphics_using_TikZ%3A_A_Tutorial_for_Beginners_(Part_5)%E2%80%94Creating_Mind_Maps)”. Acesso em: 01/02/2023.
- [Pesa] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 1. Disponível em: “[https://steeven9.github.io/USI-LaTeX/html/math\\_equations\\_system.html](https://steeven9.github.io/USI-LaTeX/html/math_equations_system.html)”. Acesso em: 01/02/2023.
- [Pesb] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 2. Disponível em: “<https://tex.stackexchange.com/questions/19921/is-there-any-easy-way-to-draw-a-ruled-surface-like-a-hyperbolic-paraboloid-in-ti>”. Acesso em: 01/02/2023.

- [Pesc] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 3. Disponível em: “<https://tex.stackexchange.com/questions/39577/what-are-the-relative-strengths-of-tikz-and-asymptote/39585>”. Acesso em: 01/02/2023.
- [Pesd] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 4. Disponível em: “<https://tex.stackexchange.com/questions/136796/drawing-figures-in-latex>”. Acesso em: 01/02/2023.
- [Pese] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 5. Disponível em: “<https://tex.stackexchange.com/questions/158668/nice-scientific-pictures-show-off>”. Acesso em: 01/02/2023.
- [Pesf] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>X 6. Disponível em: “<https://tex.stackexchange.com/questions/243580/drawing-undirected-graphs-in-latex>”. Acesso em: 01/02/2023.
- [Pesg] Pesquisar o autor. Fórum L<sup>A</sup>T<sub>E</sub>XPython 1. Disponível em: “<https://tex.stackexchange.com/questions/18020/generating-graphics-for-latex-with-pyx?rq=1>”. Acesso em: 01/02/2023.
- [Pesh] Pesquisar o autor. LuaLatex (*website*). Disponível em: “<http://www.luatex.org/index.html>”. Acesso em: 01/02/2023.
- [Pesi] Pesquisar o autor. Matplotlib para Python. Disponível em: “<https://matplotlib.org/stable/tutorials/introductory/usage.html>”. Acesso em: 01/02/2023.
- [Pesj] Pesquisar o autor. PiScript para Python. Disponível em: “<https://personal.math.ubc.ca/~cass/piscript/>”. Acesso em: 01/02/2023.
- [Proa] Procurar o autor. Fórum TikZ 1. Disponível em: “<https://tex.stackexchange.com/questions/205/what-graphics-packages-are-there-for-creating-graphics-in-latex-documents?rq=1>”. Acesso em: 01/02/2023.
- [Prob] Procurar o autor. Fórum TikZ 2. Disponível em: “<https://tex.stackexchange.com/questions/26972/what-gui-applications-are-there-to-assist-in-generating-graphics-for-tex>”. Acesso em: 01/02/2023.
- [Proc] Procurar o autor. Fórum TikZ 3. Disponível em: “<https://tex.stackexchange.com/questions/80418/list-of-tikz-editor?noredirect=1&lq=1>”. Acesso em: 01/02/2023.
- [Prod] Procurar o autor. Fórum TikZ 4. Disponível em: “<https://tex.stackexchange.com/questions/521990/update-on-gui-applications-for-latex-tikz-support?rq=1>”. Acesso em: 01/02/2023.
- [Proe] Procurar o autor. Projeto 1. Disponível em: “<https://project-awesome.org/xiaohanyu/awesome-tikz>”. Acesso em: 01/02/2023.
- [Prof] Programa de Educação Tutorial (PET). **Webpage do Programa PET**. Disponível em: “[http://portal.mec.gov.br/index.php?option=com\\_content&view=article&id=12223&ativo=481&Itemid=480](http://portal.mec.gov.br/index.php?option=com_content&view=article&id=12223&ativo=481&Itemid=480)”. Acesso em: 01/02/2023.
- [Staa] Stathis Sideris. DITAA (*Diagrams Through Ascii Art*) (*GitHub*). Disponível em: “<https://github.com/stathissideris/ditaa>”. Acesso em: 01/02/2023.

- [Stab] Stathis Sideris. DITAA (*DIagrams Through Ascii Art*) (*website*). Disponível em: “<http://ditaa.sourceforge.net/>”. Acesso em: 01/02/2023.
- [Wik] Wikipedia (*website*). Bézier curve. Disponível em: “[https://en.wikipedia.org/wiki/B%C3%A9zier\\_curve](https://en.wikipedia.org/wiki/B%C3%A9zier_curve)”. Acesso em: 01/02/2023.