

Uma breve introdução ao TikZ

GABRIEL O. ALVES

Conteúdo

1	Introdução	3
1.1	Carregando o pacote e o ambiente	3
2	Comandos básicos	3
2.1	Desenhando linhas	4
2.2	Retângulos	4
2.3	Adicionando alguns parâmetros	5
2.3.1	Cores	5
2.3.2	Setas	8
2.3.3	Espessura, linhas tracejadas e pontilhadas	10
2.3.4	Alterando a escala	11
2.4	Adicionando uma malha	12
2.5	Arcos	13
2.6	Parábolas e senóides	15
2.7	Curvas	16
3	Colorindo áreas	18
3.1	Tipos básicos	18
3.2	Formas diversas	21
4	Plotando gráficos	23
5	Adicionando textos às figuras	26
6	Decorações e padrões	28
6.1	Decorações	28
6.2	Padrões	30
7	Desenhos em três dimensões	31
8	Desenhando circuitos	32
9	Referências e links úteis	34

1 Introdução

O TikZ é um pacote criado por Till Tantau e é utilizado para a confecção de figuras técnicas por meio de códigos escritos em \TeX . O nome vem do alemão: "Tikz ist kein Zeichenprogramm" (Tikz não é um programa de desenhos).

Este trabalho tem como objetivo apresentar as funcionalidades básicas do pacote, portanto a abordagem será superficial. O leitor interessado em se aprofundar no assunto deve consultar as referências listadas na última seção, em especial a extensa documentação oficial.

Caso você encontre algum erro, algum código mais adequado, tenha alguma crítica ou sugestão, por favor entre em contato conosco.

1.1 Carregando o pacote e o ambiente

Para carregar o pacote basta escrever `\usepackage{tikz}` no preâmbulo do documento. Para iniciar o ambiente é necessário escrever:

```
1 \begin{tikzpicture}
2
3 %Seu código aqui
4
5 \end{tikzpicture}
```

Colocando seu código entre `\begin{tikzpicture}` e `\end{tikzpicture}`.

Também é possível iniciar o ambiente do TikZ como uma figura, assim é possível utilizar alguns comandos como `\center`, forçar a imagem a ficar em alguma posição desejada da página ou incluir legendas, dentre outras possibilidades:

```
1 \begin{figure}
2 \center %Centraliza a figura
3 \begin{tikzpicture}
4
5 %Seu código aqui
6
7 \end{tikzpicture}
8 \caption{Esta é uma legenda} %Inclui a legenda
9 \end{figure}
```

2 Comandos básicos

O comando básico para realizar algum tipo de desenho é `\draw`, em seguida são adicionados parâmetros que irão determinar que tipo de figura será feita.

2.1 Desenhando linhas

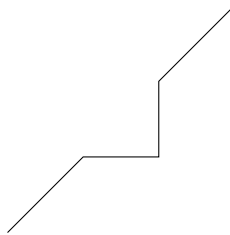
Para desenhar linhas simples podemos utilizar:

```
1 \begin{tikzpicture}
2
3 \draw (x0,y0)--(x1,y1); %Desenha uma linha simples de (x0,y0) ate (x1
  ,y1)
4
5 \end{tikzpicture}
```

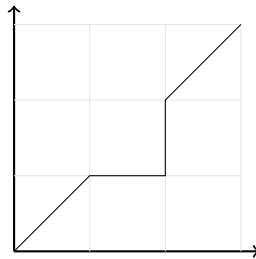
O termo $(x_0,y_0)--(x_1,y_1)$ indica que será desenhada uma linha da posição (x_0, y_0) até (x_1, y_1) em coordenadas cartesianas. Também é possível desenhar linhas consecutivas, basta adicionar a expressão $--(x,y)$ ao comando para cada coordenada adicional. O código seguinte ilustra o processo:

```
1 \begin{tikzpicture}
2
3 \draw (0,0)--(1,1)--(2,1)--(2,2)--(3,3);
4
5 \end{tikzpicture}
```

A figura gerada é:



Adicionando uma malha para facilitar a visualização:



Veja que o caminho percorrido é exatamente aquele descrito no código anterior!

2.2 Retângulos

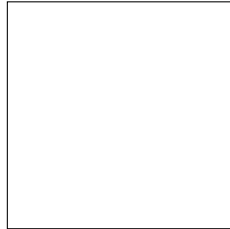
Retângulos podem ser desenhados especificando-se as coordenadas de seus vértices a partir do comando `\draw (x0,y0) rectangle (x1,y1)`. (x_0, y_0) representa a localização

do seu vértice inferior esquerdo e (x_1, y_1) representa a localização de seu vértice superior direito. Segue o exemplo:

```

1 \begin{tikzpicture}
2
3 \draw (0,0) rectangle (3,3);
4
5 \end{tikzpicture}

```



2.3 Adicionando alguns parâmetros

O comando `\draw` apresentado anteriormente aceita alguns argumentos que podem modificar o desenho da linha, como a mudança de cores, espessura etc. E alguns desses parâmetros ainda serão válidos ou análogos para outros comandos, portanto é importante aprender como utilizá-los. A estrutura geral é basicamente esta:

```

1 \begin{tikzpicture}
2
3 \draw [parametros] ...;
4
5 \end{tikzpicture}

```

2.3.1 Cores

Adicionar cores aos desenhos é simples, basta inserir `color=...` entre os delimitadores `[e]`, como no exemplo:

```

1 \begin{tikzpicture}
2
3 \draw [color=red] (0,0) -- (1,2);
4
5 \end{tikzpicture}

```

As cores padrão são as cores dos sistemas RGB e CMY (Pode haver outras cores dependendo da instalação, contudo as cores listadas a seguir devem estar presentes):

- Branco: `white`



- Preto: black
- Vermelho: red
- Verde: green
- Azul: blue
- Ciano: cyan
- Magenta: magenta
- Amarelo: yellow

Há mais cores pré-definidas provenientes do pacote `xcolor`, que podem ser utilizadas pelo TikZ (Para incluir o pacote adicione `\usepackage[usenames,dvipsnames]{xcolor}` ao preâmbulo. Veja a figura ??).

Também é possível definir novas cores pelo sistema RGB, com o comando `\definecolor{nomedacor}{sistemadecores}{valor1,valor2,valor3}`. Podemos definir a cor verde como:

```
1 \definecolor{verde}{rgb}{0,1,0}
```

Veja que a escala para cada cor vai de 0 até 1. Para utilizar a cor em algum desenho:

```
1 \definecolor{verde}{rgb}{0,1,0}
2
3 \begin{tikzpicture}
4
5 \draw [verde] (0,0) rectangle (1,2);
6
7 \end{tikzpicture}
```

Criando a cor magenta, pelo CMY:

```
1 \definecolor{magenta}{cmy}{0,1,0}
2
3 \begin{tikzpicture}
4
5 \draw [magenta] (0,0) rectangle (1,2);
6
7 \end{tikzpicture}
```








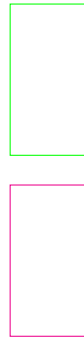
Name	Color	Name	Color
Apricot		Aquamarine	
Bittersweet		Black	
Blue		BlueGreen	
BlueViolet		BrickRed	
Brown		BurntOrange	
CadetBlue		CarnationPink	
Cerulean		CornflowerBlue	
Cyan		Dandelion	
DarkOrchid		Emerald	
ForestGreen		Fuchsia	
Goldenrod		Gray	
Green		GreenYellow	
JungleGreen		Lavender	
LimeGreen		Magenta	
Mahogany		Maroon	
Melon		MidnightBlue	
Mulberry		NavyBlue	
OliveGreen		Orange	
OrangeRed		Orchid	
Peach		Periwinkle	
PineGreen		Plum	
ProcessBlue		Purple	
RawSienna		Red	
RedOrange		RedViolet	
Rhodamine		RoyalBlue	
RoyalPurple		RubineRed	
Salmon		SeaGreen	
Sepia		SkyBlue	
SpringGreen		Tan	
TealBlue		Thistle	
Turquoise		Violet	
VioletRed		White	
WildStrawberry		Yellow	
YellowGreen		YellowOrange	

Figura 1: Lista de cores provenientes do pacote xcolor. Fonte: <https://en.wikibooks.org/wiki/LaTeX/Colors>



Também é possível alterar a intensidade das cores adicionando !intensidade após o nome desejada (A intensidade varia numa escala de 0 a 100), por exemplo:

```
1 \begin{tikzpicture}
2
3 \draw [orange] (-2,0) -- (-1,2);
4 \draw [orange!50] (0,0) -- (1,2);
5
6 \end{tikzpicture}
```



Também é possível misturá-las:

```
1 \begin{tikzpicture}
2
3 \draw [green!50!blue] (0,0) -- (1,2);
4
5 \end{tikzpicture}
```



2.3.2 Setas

Outra funcionalidade muito útil é a possibilidade de transformar os segmentos em setas. Para desenhar um vetor simples é necessário adicionar \rightarrow aos parâmetros do desenho:


```

1 \begin{tikzpicture}
2
3
4 \draw [->] (0,0) -- (1,2);
5
6 \end{tikzpicture}

```



É possível adicionar mais de um parâmetro adicionando uma vírgula entre cada um deles:

```

1 \begin{tikzpicture}
2
3 \draw [->,color=blue] (0,0) -- (1,2);
4
5 \end{tikzpicture}

```

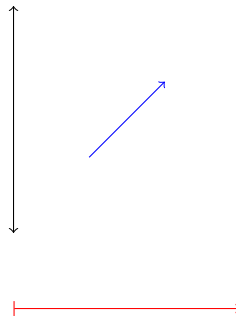


Além disso há outras tipos de setas, como $\mid\rightarrow$ e $\leftarrow\rightarrow$:

```

1 \begin{tikzpicture}
2
3 \draw [->,color=blue] (1,1) -- (2,2);
4 \draw [<->,color=black] (0,0) -- (0,3);
5 \draw [|->,color=red] (-1,-1) -- (3,-1);
6
7 \end{tikzpicture}

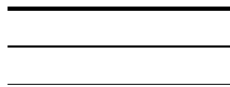
```



2.3.3 Espessura, linhas tracejadas e pontilhadas

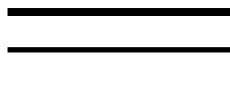
Você pode alterar a espessura das linhas adicionando `thick`, `semithick`, `very thick`, `ultra thick`, `thin`, `very thin` e `ultra thin`:

```
1 \begin{tikzpicture}
2
3 \draw [ultra thin] (0,-3)--(3,-3);
4 \draw (0,0)--(3,0);
5 \draw [ultra thick] (0,4)--(3,4);
6
7 \end{tikzpicture}
```



Para definir um valor específico de espessura utilize `line width=valor`:

```
1 \begin{tikzpicture}
2
3 \draw [line width=1] (0,-.5)--(3,-.5);
4 \draw [line width=2] (0,0)--(3,0);
5 \draw [line width=3] (0,.5)--(3,.5);
6
7 \end{tikzpicture}
8 \end{figure}
```



Para fazer uma linha dupla adicione o parâmetro `double` (A espessura da linha deve ser grande o suficiente para que o efeito seja perceptível):

```
1 \begin{tikzpicture}
2
```

```

3 \draw [ultra thick,double] (0,0) -- (3,0);
4
5 \end{tikzpicture}

```



Adicionando os parâmetros `dashed` e `dotted`, é possível criar linhas tracejadas e pontilhadas, respectivamente :

```

1 \begin{tikzpicture}
2
3 \draw [dashed] (0,0) -- (3,0); %Linha tracejada
4 \draw [dotted] (0,0.5) -- (3,.5); %Linha pontilhada
5
6 \end{tikzpicture}

```



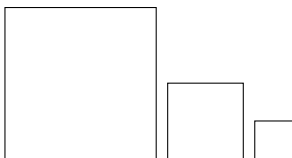
2.3.4 Alterando a escala

O TikZ permite que você altere a escala da figura adicionando `[scale=...]` logo após iniciar o ambiente, como no exemplo a seguir:

```

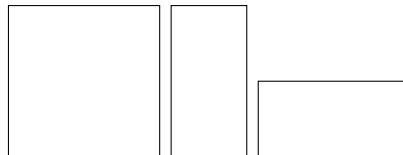
1 \begin{tikzpicture}
2
3 \draw (0,0) rectangle (2,2);
4
5 \end{tikzpicture}
6 \begin{tikzpicture} [scale=.5]
7
8 \draw (0,0) rectangle (2,2);
9
10 \end{tikzpicture}
11 \begin{tikzpicture} [scale=.25]
12
13 \draw (0,0) rectangle (2,2);
14
15 \end{tikzpicture}

```



Também é possível alterar a escala em um único eixo:

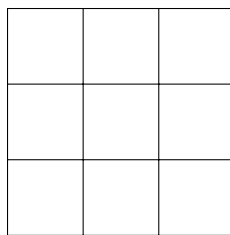
```
1 \begin{tikzpicture}
2
3 \draw (0,0) rectangle (2,2);
4
5 \end{tikzpicture}
6 \begin{tikzpicture}[xscale=.5]
7
8 \draw (0,0) rectangle (2,2);
9
10 \end{tikzpicture}
11 \begin{tikzpicture}[yscale=.5]
12
13 \draw (0,0) rectangle (2,2);
14
15 \end{tikzpicture}
```



2.4 Adicionando uma malha

Para algumas figuras é conveniente criar malhas, uma maneira de fazer isso é a seguinte:

```
1 \begin{tikzpicture}
2
3 \draw (0,0) grid (3,3);
4
5 \end{tikzpicture}
```



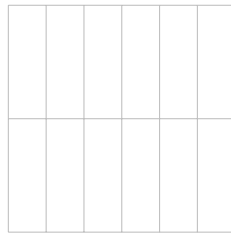
O primeiro par de coordenadas indica a posição do canto inferior esquerdo da malha, o segundo par indica a posição do canto superior direito.

Você provavelmente irá querer usar essa malha como um auxílio na confecção das suas figuras, por isso pode ser conveniente deixá-la em uma cor clara ou alterar o valor do passo. Para isto basta alterar os parâmetros:

```

1 \begin{tikzpicture}
2
3 \draw [color=gray!50,xstep=0.5,ystep=1.5] (0,0) grid (3,3);
4
5 \end{tikzpicture}

```



Os termos `xstep` e `ystep` mudam o comprimento de cada divisão nos eixos x e y , respectivamente. Logo, no exemplo anterior o espaçamento de cada retângulo no eixo x vale $.5$, já no eixo y vale 1.5 .

2.5 Arcos

Arcos podem ser desenhados a partir do comando `\draw (x0,y0) arc [start angle=t0, end angle=t1, radius=r]`, onde os termos entre colchetes indicam o ângulo inicial, final e raio, respectivamente. As coordenadas (x_0, y_0) indicam a posição na qual o arco está centrado:

```

1 \begin{tikzpicture}
2
3 \draw [gray!50] (-2,0) grid (0,2);
4 \draw (0,0) arc [start angle=0,end angle=45, radius=1];
5
6 \end{tikzpicture}

```



Figura 2: Veja que o centro do arco não está centrado na origem, e sim na posição $(-2,0)$

Uma maneira mais simplificada de escrever isto é:

```

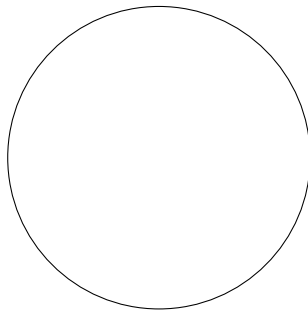
1 \begin{tikzpicture}
2
3 \draw (0,0) arc (0:45:2);

```

```
4  
5 \end{tikzpicture}
```

Que gera o mesmo resultado. Também é possível desenhar círculos (Só o raio deve ser especificado):

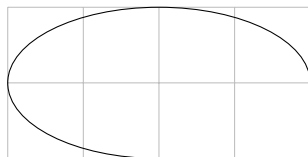
```
1 \begin{tikzpicture}  
2  
3 \draw (0,0) circle [radius=2];  
4  
5 \end{tikzpicture}
```



Também é possível escrever o comando de uma forma mais compacta: `\draw (0,0) circle (2)`, que gera uma figura idêntica à anterior.

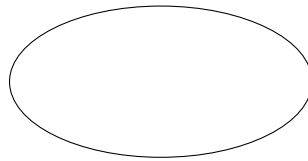
Ou elipses, especificando `x radius` e `y radius`:

```
1 \begin{tikzpicture}  
2  
3 \draw [gray!50] (-4,-1) grid (0,1);  
4 \draw (0,0) arc [start angle=0,end angle=270, x radius=2, y radius=1]  
5 ;  
6 \end{tikzpicture}
```



Uma maneira alternativa de desenhar elipses é escrevendo:

```
1 \begin{tikzpicture}  
2  
3 \draw (0,0) ellipse (2 and 1);  
4  
5 \end{tikzpicture}
```



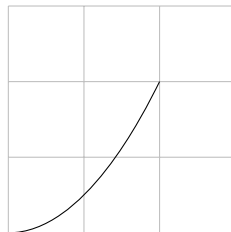
2.6 Parábolas e senóides

A sintaxe do comando que cria uma parábola que passa pelos pontos (x_0, y_0) e (x_1, y_1) é `(x0,y0) parabola (x1,y1)`, conforme o exemplo:

```

1 \begin{tikzpicture}
2
3 \draw [gray!50] (0,0) grid (3,3);
4 \draw (0,0) parabola (2,2);
5
6 \end{tikzpicture}

```

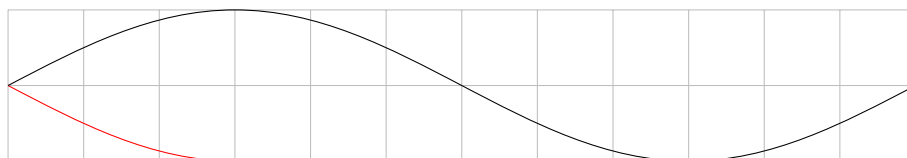


Utilizando os comandos `(x0,y0) sin (x1,y1)` e `(x0,y0) cos (x1,y1)` é possível desenhar senóides:

```

1 \begin{tikzpicture}
2
3 %Desenha a malha
4 \draw [gray!50] (0,-1) grid (12,1);
5 %Desenha a senoide vermelha
6 \draw [color=red] (0,0) sin (3,-1);
7 %Desenha uma senoide preta de x=0 ate x=12
8 \draw (0,0) sin (3,1) cos (6,0) sin (9,-1) cos (12,0);
9
10 \end{tikzpicture}

```

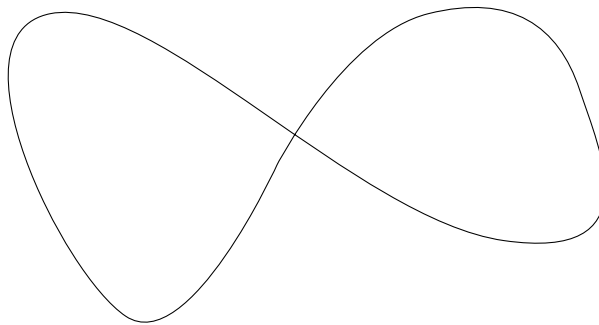


Para desenhar uma senóide contínua, como no exemplo anterior, é necessário alternar os comandos `sin` e `cos`, conforme foi feito.

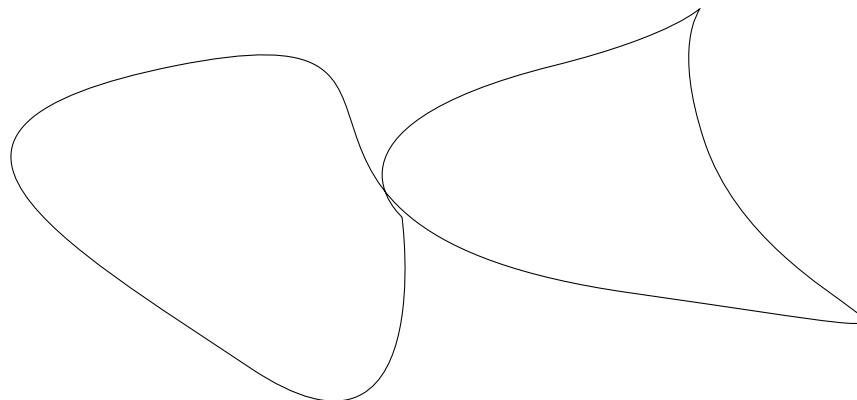
2.7 Curvas

Há diversas maneiras de desenhar curvas, serão introduzidas aqui quatro delas. A primeira consiste em especificar um conjunto de pontos e adicionar os parâmetros `smooth`, `tension` e utilizar o comando `\draw` de maneira um pouco diferente :

```
1 \begin{tikzpicture}
2
3 \draw plot [smooth,tension=1] coordinates { (0,0) (2,2) (4,1) (3,-1)
4        (-3,2) (-2,-2) (0,0) };
5 \end{tikzpicture}
```



Os valores entre `coordinates {...}` representam o conjunto de pontos, o parâmetro `tension` altera a suavidade das curvas. Por exemplo, se fizermos com que ele seja o triplo na figura anterior obtemos uma figura com curvas mais acentuadas:

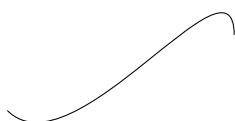


A segunda consiste em definir um ângulo inicial e um ângulo final tangentes à curvatura:

```

1 \begin{tikzpicture}
2
3 \draw (0,0) to [out=-45, in=90] (3,1);
4
5 \end{tikzpicture}

```



Também é possível utilizar o parâmetro bend:

```

1 \begin{tikzpicture}
2
3 \draw (0,0) to [bend right] (3,1);
4
5 \end{tikzpicture}

```

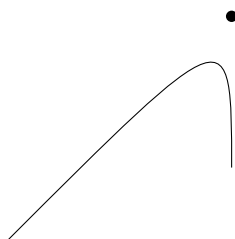


Por fim, também é possível curvar uma reta adicionando 'pontos de controle'. Eles fazem com que o segmento 'se curve' em sua direção. Basta adicionar o termo `.. controls (x,y) ..` entre as coordenadas de uma linha:

```

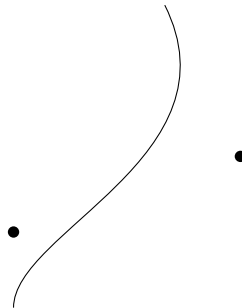
1 \begin{tikzpicture}
2
3 \draw (0,0) .. controls (3,3) .. (3,1);
4
5 \end{tikzpicture}

```



A bolinha preta foi utilizada para ilustrar a localização do ponto de controle, localizado em (3,3). Também é possível adicionar um segundo ponto de controle, conforme o exemplo:

```
1 \begin{tikzpicture}
2
3 \draw (0,0) .. controls (0,1) and (3,2) .. (2,4);
4
5 \end{tikzpicture}
```

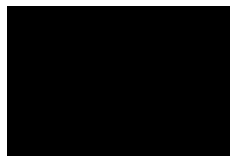


3 Colorindo áreas

3.1 Tipos básicos

É bem provável que você precise colorir certas regiões em suas figuras, para isso você pode utilizar o comando `\fill`. Para utilizar este comando é necessário especificar a área que será preenchida. Para preencher a região compreendida por um retângulo por exemplo, basta fazer o seguinte:

```
1 \begin{tikzpicture}
2
3 \fill (0,0) rectangle (3,2);
4
5 \end{tikzpicture}
```

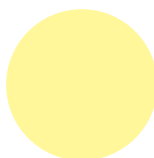


A sintaxe é basicamente aquela apresentada nas seções anteriores, a diferença é que o comando `\draw` foi substituído pelo comando `\fill`. A cor padrão de preenchimento é a cor preta, para mudá-la é necessário adicionar a cor desejada da mesma maneira feita nas seções anteriores para retas, retângulos, etc:

```

1 \begin{tikzpicture}
2
3 \fill [color=yellow!50] (0,0) circle (1);
4
5 \end{tikzpicture}

```

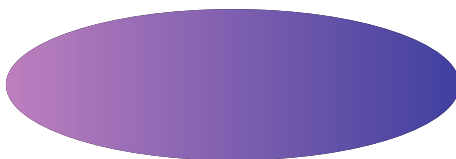


Agora, que tal adicionar um gradiente de cores? Você pode criar uma figura que apresente um gradiente da esquerda para a direita:

```

1 \begin{tikzpicture}
2
3 \fill [left color=Purple!50, right color=NavyBlue!75] (0,0) ellipse
   (3 and 1);
4
5 \end{tikzpicture}

```



De cima para baixo:

```

1 \begin{tikzpicture}
2
3 \fill [top color=Cyan!100, bottom color=Teal!75] (0,0) rectangle
   (2,1);
4
5 \end{tikzpicture}

```



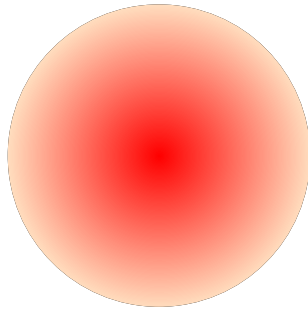
Ou do interior para o exterior:

```

1 \begin{tikzpicture}
2

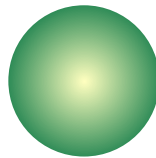
```

```
3 \fill [inner color=Red!100, outer color=BurntOrange!25] (0,0) circle  
   (2);  
4  
5 \end{tikzpicture}
```



Nós também podemos utilizar o comando `\shade`, que é capaz de fazer a mesma coisa, contudo é obrigatório especificar as cores do gradiente, caso contrário ele gera um gradiente cinza:

```
1 \begin{tikzpicture}  
2  
3 \shade [inner color=yellow!25,outer color=SeaGreen!100] (0,0) circle  
   (1);  
4  
5 \end{tikzpicture}
```



Também é possível desenhar o contorno da figura e preenchê-la simultaneamente utilizando o comando `\draw` e especificando-se a cor de preenchimento:

```
1 \begin{tikzpicture}  
2  
3 %Desenha um quadrado azul com contorno preto  
4 \draw [fill=blue!50,draw=black] (0,0) rectangle (3,2);  
5  
6 \end{tikzpicture}
```



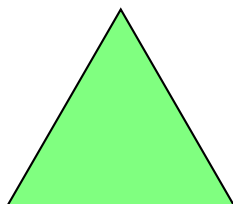
3.2 Formas diversas

O preenchimento de áreas não se restringe somente às figuras mais simples, o TikZ também é capaz de preencher áreas arbitrárias, como a do seguinte exemplo:

```

1 \begin{tikzpicture}
2
3 \draw [fill=green!50, thick] (0,0) -- ({3*cos(60)},{3*sin(60)}) -- (3,0)
   -- cycle;
4 %Para utilizar funcoes matematicas como sin e cos e necessario isolar
   cada coorenada entre os delimitadores {}, deste modo o codigo e
   compilado corretamente
5 %A expressao cycle faz com que o comando draw trace uma figura "
   fechada", voltando para a posicao inicial
6
7 \end{tikzpicture}

```

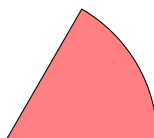


Basta desenhar o caminho que se deseja com o comando `\draw` e adicionar o parâmetro `fill = ...`, com a cor desejada. Para colorir um arco, podemos fazer o seguinte:

```

1 \begin{tikzpicture}
2
3 \draw [fill=red!50] (0,0) -- (2,0) arc (0:60:2) -- cycle;
4
5 \end{tikzpicture}

```



Para preencher áreas um pouco mais complicadas você pode utilizar o comando `plot`, mostrado nas seções anteriores, e adicionar parâmetros (Veja que os parâmetros relativos à cor, espessura da linha, etc aparecem *antes* do comando `plot`):

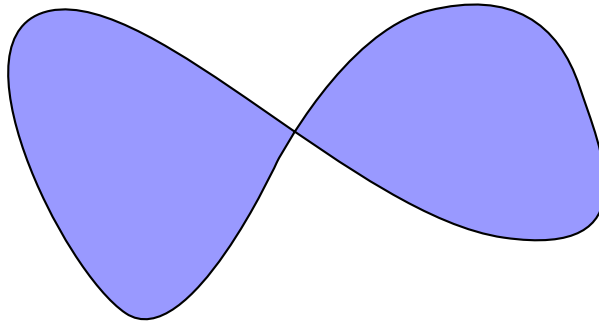
```

1 \begin{tikzpicture}
2

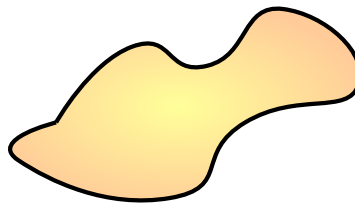
```

3. COLORINDO ÁREAS

```
3 \draw [fill=blue!40, thick] plot [smooth,tension=1] coordinates {
    (0,0) (2,2) (4,1) (3,-1) (-3,2) (-2,-2) (0,0)};
4
5 \end{tikzpicture}
```

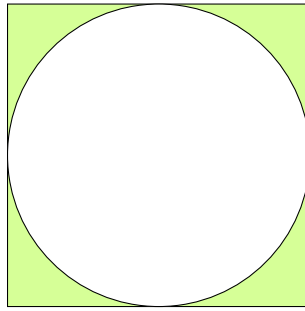


```
1 \begin{tikzpicture} [scale=.5]
2
3 \draw [inner color=yellow!40, outer color=orange!40, ultra thick]
    plot [smooth,tension=1] coordinates { (0,0) (2,2) (4,1.5) (6,3)
    (8,1) (5,0) (3,-2) (-1,-1) (0,0)};
4
5 \end{tikzpicture}
```



Um detalhe importante do tikz é que as figuras são desenhadas na ordem em que aparecem no código. Você pode se aproveitar disso para realizar certos tipos de desenhos, como na seguinte imagem:

```
1 \begin{tikzpicture}
2
3 %Desenha um quadrado verde
4 \draw [fill=GreenYellow!50] (0,0) rectangle (4,4);
5 %Em seguida desenha um círculo branco
6 \draw [fill=white] (2,2) circle (2);
7
8 \end{tikzpicture}
```



Preencher a parte destacada fica fácil por meio desta estratégia. Basta desenhar um retângulo verde e em seguida desenhar um círculo branco dentro do quadrado, apagando a parte indesejada. Tentar fazer a mesma figura por outros métodos seria um pouco mais complicado.

4 Plotando gráficos

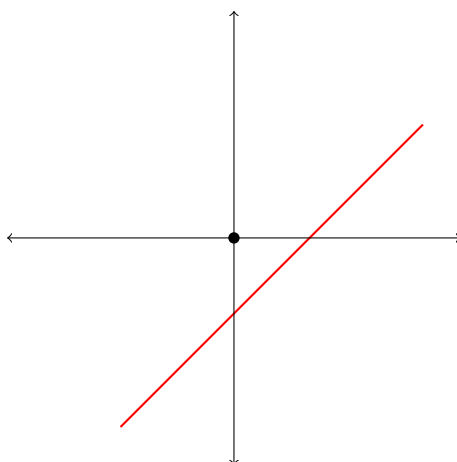
Durante o seu trabalho você provavelmente precisará plotar gráficos. Apesar de alguns programas como o Octave e Mathematica serem mais adequados para este tipo de tarefa, o TikZ é capaz de realizar tarefas mais simples muito bem, com a vantagem de criar figuras similares ao resto daquelas utilizadas no documento e tornar possível a integração de gráficos com outros tipos de figuras, tudo isso no mesmo código.

A maneira mais simples de desenhar gráficos no TikZ é utilizar a seguinte estrutura:

```
1 \draw [parametros] plot (variavel,funcao);
```

Como no exemplo a seguir, no qual é plotada a função $y = x - 1$, no intervalo $x = [-1.5, 2.5]$:

```
1 \begin{tikzpicture}
2
3 \draw [domain=-1.5:2.5,variable=\x,color=red,thick] plot ({\x},{\x
4 -1});
5 %Desenho dos eixos x e y
6 \draw [<->] (-3,0)--(3,0);
7 \draw [<->] (0,-3)--(0,3);
8 \fill (0,0) circle (.075);
9
10 \end{tikzpicture}
```

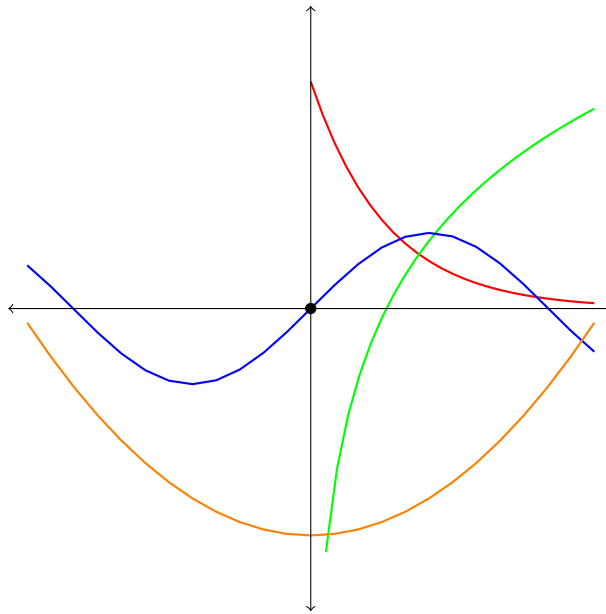


No exemplo anterior a variável utilizada é x . Podemos introduzir uma variável digitando `\` e em seguida a letra ou nome desejado (No caso foi feito `\x`). O domínio é especificado entre os delimitadores `[e]`, como parâmetro, utilizando `domain=x0:x1`. É possível utilizar muitas outras funções matemáticas, como as seguintes:

```

1 \begin{tikzpicture}
2
3 %Exponencial negativa
4 \draw [domain=-0:3.75,variable=\x,color=red,thick] plot ({\x},3*exp{-
   \x});
5 %Logaritmo natural
6 \draw [domain=0.2:3.75,variable=\x,color=green,thick] plot ({\x},2*ln
   {\x});
7 %Senoide
8 \draw [domain=-3.75:3.75,variable=\x,color=blue,thick] plot ({\x},sin
   {deg(\x)});
9 \draw [domain=-3.75:3.75,variable=\x,color=blue,thick] plot ({\x},sin{
   deg(\x)});
10 %Parabola (Caso o código não compile, tente escrever a função y entre
   { e }, como foi feito na expressão a seguir. Isto pode ocorrer
   devido ao uso de parenteses, que pode fazer com que o compilador
   interprete o caractere ')' utilizado em algumas funções
   matemáticas como um indicador do fim do comando plot)
11 \draw [domain=-3.75:3.75,variable=\x,color=orange,thick] plot ({\x
   },{1/5*pow(\x,2)-3});
12
13 \draw [<->] (-4,0)--(4,0);
14 \draw [<->] (0,-4)--(0,4);
15 \fill (0,0) circle (.075);
16
17 \end{tikzpicture}

```

Veja que para plotar a função $\sin x$ a sintaxe utilizada foi $\sin\{\deg(\backslash x)\}$. O termo $\deg(\backslash x)$ indica que a variável x deve ser tratada como um ângulo (em radianos).

Você também pode criar funções paramétricas. No exemplo a seguir foi desenhado um cicloide, parametrizado por:

$$x(t) = \frac{1}{2}(t - \sin t)$$

$$y(t) = \frac{1}{2}(1 - \cos t)$$

E uma curva de Lissajous, parametrizada por:

$$x(t) = 3 \sin(2t)$$

$$y(t) = 2 \cos(t)$$

```

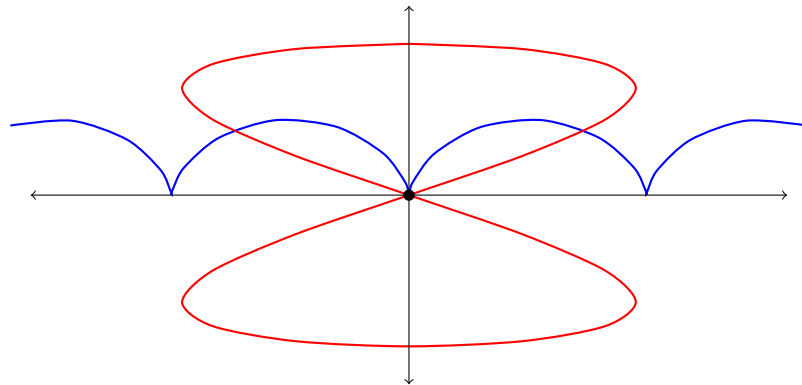
1 \begin{tikzpicture}
2
3 %Cicloide
4 \draw[domain=-10:10,variable=\t,color=blue,thick,smooth] plot ({.5*(\
   t-sin{deg(\t)}}),{.5*(1-cos{deg(\t)}}));
5 %Curva de lissajous
6 \draw[domain=0:2*pi,variable=@\t@>>,color=red,thick,smooth] plot
   (3*sin{deg(2*\t)},2*cos{deg(1*\t)});
7
8 \draw [<->] (-5,0)--(5,0);

```

```

9 \draw [<->] (0,-2.5) -- (0,2.5);
10 \fill (0,0) circle (.075);
11
12 \end{tikzpicture}

```



Na plotagem do ciclóide foi adicionado o parâmetro `smooth` para suavizar a curva.

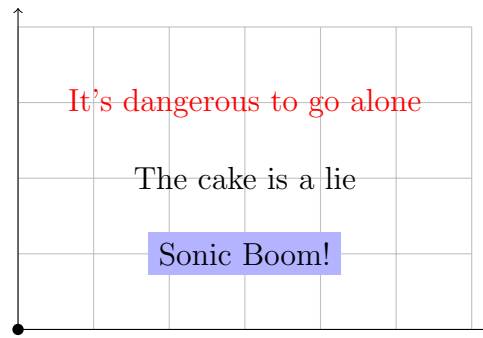
5 Adicionando textos às figuras

Outra funcionalidade conveniente do TikZ é a adição de textos às figuras. Você pode fazer isso utilizando o comando `\node at (x,y) {texto}`, conforme o exemplo:

```

1 \begin{tikzpicture}
2
3 \draw [gray!50] (0,0) grid (4,4);
4 \draw [->] (0,0) -- (4.25,0);
5 \draw [->] (0,0) -- (0,4.25);
6 \fill (0,0) circle (.075);
7
8 \node at (2,2) {The cake is a lie};
9 \node [fill=blue!30] at (3,1) {Sonic Boom!};
10 \node [color=red] at (3,3) {It's dangerous to go alone};
11
12 \end{tikzpicture}

```

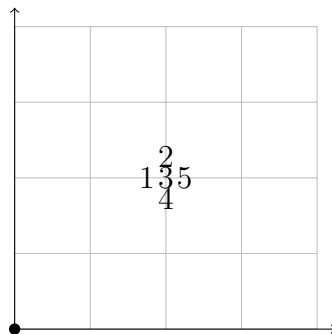


Também é possível colocar o texto imediatamente acima, abaixo ou ao lado do ponto desejado:

```

1 \begin{tikzpicture}
2
3 \draw [gray!50] (0,0) grid (4,4);
4 \draw [->] (0,0) -- (4.25,0);
5 \draw [->] (0,0) -- (0,4.25);
6 \fill (0,0) circle (.075);
7
8 \node [left] at (2,2) {1};
9 \node [above] at (2,2) {2};
10 \node at (2,2) {3};
11 \node [below] at (2,2) {4};
12 \node [right] at (2,2) {5};
13
14 \end{tikzpicture}

```



Ou até mesmo combinar os comandos anteriores:

```

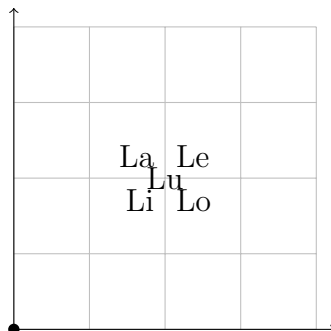
1 \begin{tikzpicture}
2
3 \draw [gray!50] (0,0) grid (4,4);
4 \draw [->] (0,0) -- (4.25,0);
5 \draw [->] (0,0) -- (0,4.25);

```

```

6 \fill (0,0) circle (.075);
7
8 \node [above left] at (2,2) {La};
9 \node [below left] at (2,2) {Li};
10 \node at (2,2) {Lu};
11 \node [above right] at (2,2) {Le};
12 \node [below right] at (2,2) {Lo};
13 \end{tikzpicture}

```



6 Decorações e padrões

6.1 Decorações

As decorações são certos padrões utilizados para confeccionar desenhos, dados os caminhos desejados. Antes de mais nada, adicione algumas das bibliotecas necessárias para as decorações escrevendo `\usetikzlibrary{decorations.pathmorphing,patterns,decorations.shapes}` no preâmbulo do documento. Agora iremos iniciar com um exemplo simples, no qual desenharemos um zigue-zague:

```

1 \begin{tikzpicture}
2
3 \draw [decorate,decoration=zigzag] (0,0) -- (5,0);
4
5 \end{tikzpicture}

```



Ou seja, estrutura básica consiste em adicionar `decorate` e `decoration=...` aos parâmetros do comando. Além da decoração em zigue-zague há muitas outras, como:

```

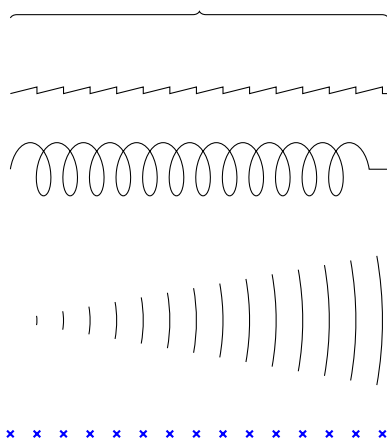
1 \begin{tikzpicture}
2

```

```

3 \draw [decorate,decoration=brace] (0,0) --(5,0);
4 \draw [decorate,decoration=saw] (0,-1) --(5,-1);
5 \draw [decorate,decoration={coil,segment length=10,amplitude=10}]
  (0,-2) --(5,-2);
6 \draw [decorate,decoration={expanding waves, angle=10}] (0,-4)
  --(5,-4);
7 \draw [decorate,decoration={crosses}] [fill=blue!20,draw=blue,thick]
  (0,-5.5) -- (5,-5.5);
8
9 \end{tikzpicture}

```

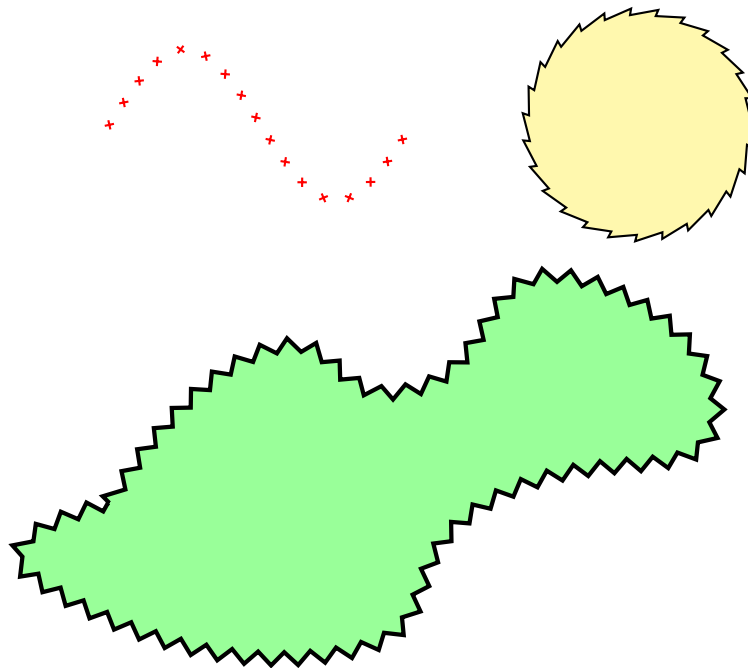


As decorações não precisam ser feitas somente em linhas retas, você também pode utilizá-las em caminhos arbitrários:

```

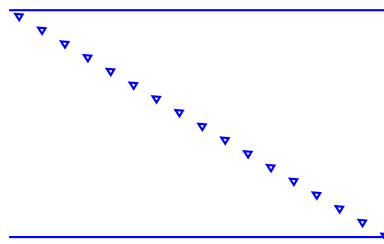
1 \begin{tikzpicture}
2
3 \draw [decorate,decoration={crosses},fill=red!20,draw=red,thick]
  (0,5) sin (1,6) cos (2,5) sin (3,4) cos (4,5);
4
5 \draw [decorate, decoration=saw,fill=yellow!40,thick] (7,5) circle
  (1.5);
6
7 \draw [decorate,decoration=zigzag, ultra thick, fill=green!40] plot [
  smooth,tension=1] coordinates { (0,0) (2,2) (4,1.5) (6,3) (8,1)
  (5,0) (3,-2) (-1,-1) (0,0) };
8
9 \end{tikzpicture}

```



Também é possível decorar somente uma parte do caminho:

```
1 \begin{tikzpicture}
2
3 \draw [decoration={triangles},draw=blue,thick] (0,0) -- (5,0) decorate
4         {--(0,3)}--(5,3);
5 \end{tikzpicture}
```



6.2 Padrões

Para adicionar a biblioteca de padrões escreva `\usetikzlibrary{patterns}` no preâmbulo, assim como foi feito para a biblioteca de decorações. Os padrões são desenhos similares às decorações e se repetem. Os padrões podem ser utilizados utilizando os comandos `\fill` ou `\draw`, como no exemplo a seguir:

```
1 \begin{tikzpicture}
```

```

2
3 \fill [pattern=fivepointed stars] (0,0) rectangle (2,2);
4
5 \end{tikzpicture}

```

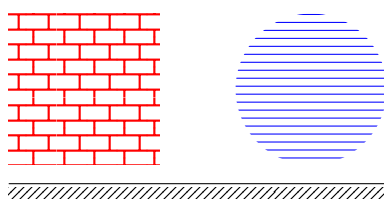


Também é possível utilizar outras cores, formas e tipos de padrões:

```

1 \begin{tikzpicture}
2
3 \fill [pattern=bricks ,pattern color=red] (0,0) rectangle (2,2);
4
5 \fill [pattern=horizontal lines,pattern color=blue] (4,1) circle (1);
6
7 \fill [pattern=north east lines] (0,-.5) rectangle (5,-.3);
8 \draw (0,-.25)--(5,-.25);
9
10 \end{tikzpicture}

```



7 Desenhos em três dimensões

Até agora todos os exemplos foram feitos utilizando figuras bidimensionais, contudo é possível utilizar o pacote `3dplot` para confeccionar figuras tridimensionais. Para isso adicione `\usepackage{tikz-3dplot}` ao preâmbulo e garanta que o pacote esteja na mesma pasta do seu projeto ou já tenha sido instalado.

No ambiente 3D, o uso de muitas funções utilizadas é análogo ao das funções mais simples no ambiente 2D:

```

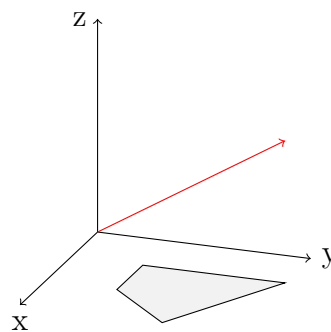
1 \tdplotsetmaincoords{70}{110}
2 \begin{tikzpicture} [tdplot_main_coords]
3
4 \draw [->] (0,0,0) -- (3,0,0);
5 \draw [->] (0,0,0) -- (0,3,0);

```

```

6 \draw [->] (0,0,0) --(0,0,3);
7
8 \draw [color=red,->] (0,0,0) --(1,3,2);
9
10 \draw [fill=gray!10] (1,1,0) --(1,3,0) --(3,2,0) --(2,1,0) --cycle;
11
12 \node [left] at (0,0,3) {z};
13 \node [right] at (0,3,0) {y};
14 \node [below] at (3,0,0) {x};
15
16 \end{tikzpicture}

```



Contudo, note um detalhe importante no exemplo anterior: é necessário adicionar a expressão `\tdplotsetmaincoords{70}{110}` antes de iniciar o ambiente TikZ e também escrever `[tdplot_main_coords]` após `\begin{tikzpicture}`. Isso faz com que as coordenadas sejam transformadas corretamente, se esses comandos não fossem incluídos muitas intruções relativas à 3ª coordenada seriam ignorados e algumas figuras seriam desenhadas como se estivessem no ambiente 2D. Os valores 70 e 110 referem-se aos ângulos de rotação, que podem ser alterados.

8 Desenhando circuitos

A biblioteca `circuitikz` foi escrita por Massimo Redaelli que pode ser utilizada para confeccionar diagramas de circuito elétricos. Para incluir a biblioteca escreva `\usepackage{circuitikz}` no preâmbulo. A sintaxe básica para desenhar um componente é:

```

1 \begin{circuitikz}
2
3 \draw [...] (x0,y0) to (x1,y1);
4
5 \end{circuitikz}

```


Onde \dots é substituído pelo nome que designa um componente. Para desenhar um resistor, que é designado pela letra R, escrevemos:

```

1 \begin{circuitikz}
2
3 \draw [R=$470 \Omega$] (0,0) to (2,0); % Resistor, R designa um
   resistor
4
5 \end{circuitikz}

```

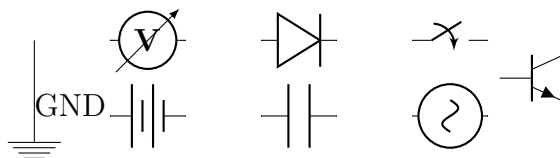


Segue um exemplo contendo alguns componentes simples,

```

1 \begin{circuitikz}
2
3 \draw (0,0) to (0,-1) node[ground]{GND};
4 \draw [voltmeter] (1,0) to (2,0) ;
5 \draw [battery] (1,-1) to (2,-1) ;
6 \draw [diode] (3,0) to (4,0) ;
7 \draw [C] (3,-1) to (4,-1);
8 \draw [sV] (5,-1) to (6,-1) ; %Fonte AC
9 \draw [closing switch] (5,0) to (6,0) ;
10 \draw (7,-.5) node [npn]{}; $Transistor NPN
11
12 \end{circuitikz}

```

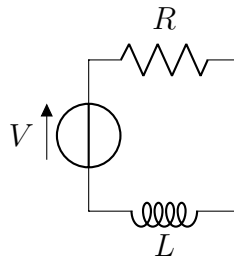


E um exemplo contendo um circuito $R - L$:

```

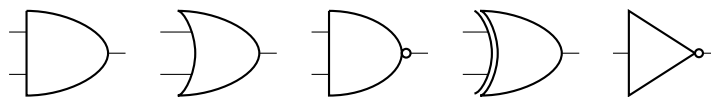
1 \begin{circuitikz}
2   \draw (0,0)
3     to [V,v=$V$] (0,2) % Fonte de tensao
4     to [R=$R$] (2,2) % Resistor
5     to [short] (2,0) %Desenha um fio
6     to [L=$L$] (0,0); %Indutor
7 \end{circuitikz}

```



Também é possível utilizar a biblioteca para desenhar portas lógicas:

```
1 \begin{circuitikz}
2
3 \draw (0,0) node[and port] {};%Porta AND
4 \draw (2,0) node[or port] {};%Porta OR
5 \draw (4,0) node[nand port] {};%Porta NAND
6 \draw (6,0) node[xor port] {};%Porta XOR
7 \draw (7.3,0) node[not port] {};%Porta NOT
8
9 \end{circuitikz}
```



9 Referências e links úteis

- Documentação do pacote: <http://ftp.fau.de/ctan/graphics/pgf/base/doc/pgfmanual.pdf>
- Um pequeno manual do TikZ, no qual este trabalho foi inspirado: <http://cremeronline.com/LaTeX/minimaltikz.pdf>
- Documentação do Circuitikz: <http://texdoc.net/texmf-dist/doc/latex/circuitikz/circuitikzmanual.pdf>
- Documentação do pacote 3dplot: ftp://ftp.yzu.edu.tw/CTAN/graphics/pgf/contrib/tikz-3dplot/tikz-3dplot_documentation.pdf
- Stack Exchange, site muito útil para tirar dúvidas: <http://tex.stackexchange.com/>
- Exemplos de figuras feitas em TikZ: <http://www.texample.net/tikz/examples/>