

Introdução ao *Tikz* do L^AT_EX

Prof. Angelo Aliano Filho

UTFPR – Universidade Tecnológica Federal do Paraná

27 de Outubro de 2020

Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

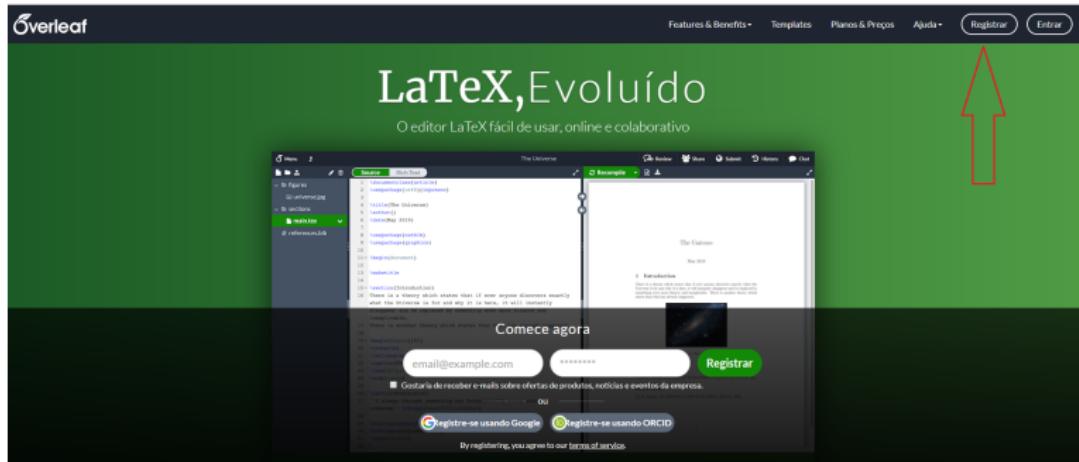
Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Introdução

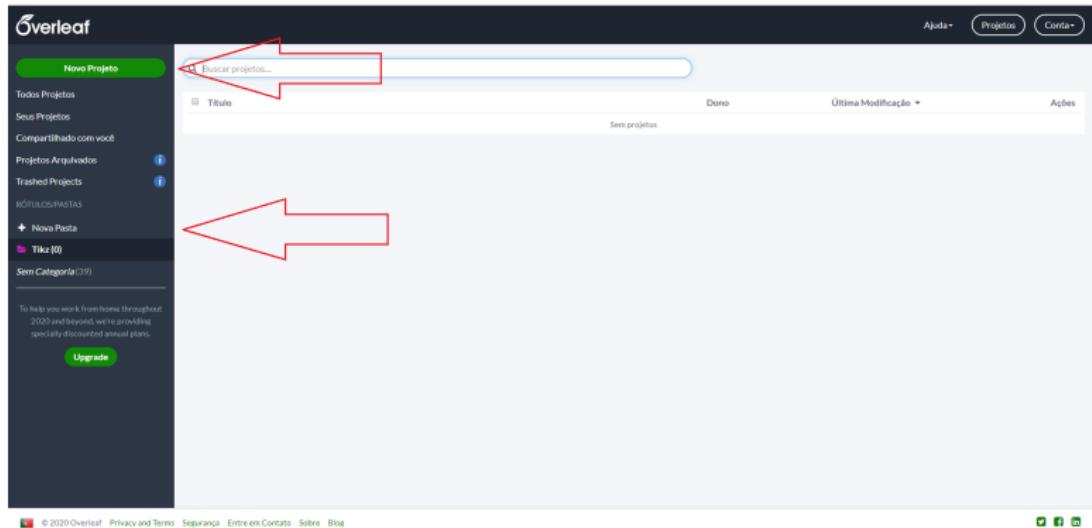
Para brincarmos um pouco com o L^AT_EXe o Tikz, vamos criar uma conta no Overleaf.

- Acesse em: <https://pt.overleaf.com/>
- Se não tiver uma conta, clique em “Registrar”



Introdução

- Crie uma pasta qualquer, depois clique em “Novo Projeto”
- Agora é só utilizar os códigos e compilar!



Introdução

O que é o Tikz?

- TikZ é um conjunto de linguagens para produzir imagens vetoriais, isto é, imagens descritas a partir da sua descrição geométrica ou algébrica. TikZ é um conjunto de macros de alto nível.
- Uma imagem vetorial é composta por pontos, curvas, polígonos, texto, entre outros elementos, isto é, utilizam vetores matemáticos para sua descrição. As imagens vetoriais são usualmente mais leves e não perdem qualidade ao serem ampliadas.

Introdução

Onde encontrar seus manuais?

- Manual completo do Tikz em PDF:

<https://www.ctan.org/pkg/pgf>

- Exemplos de figuras com respectivos códigos:

<https://texample.net/tikz/examples/>

- Demais referências em [1, 2]

Introdução

Porque usar o Tikz?

O TikZ é provavelmente a ferramenta mais complexa e poderosa para criar elementos gráficos em L^AT_EX. Neste curso serão explicados alguns dos elementos básicos: como desenhar linhas, pontos, curvas, círculos, rectângulos, etc., por meio de exemplos simples.

Introdução

Porque usar o Tikz?

Com ele podemos criar diagramas de alta qualidade - e muitas vezes também bastante complexos. Neste curso vamos começar com o básico, mostrando como desenhar formas simples, introduzindo alguns comandos simples mas que construir figuras/diagramas bastante interessantes.

Vantagens

- Nitidez/qualidade dos objetos construídos
- Facilidade de uso em outros documentos
- Intuitiva
- Fácil edição

Introdução

Porque usar o Tikz?

Com ele podemos criar diagramas de alta qualidade - e muitas vezes também bastante complexos. Neste curso vamos começar com o básico, mostrando como desenhar formas simples, introduzindo alguns comandos simples mas que construir figuras/diagramas bastante interessantes.

Vantagens

- Nitidez/qualidade dos objetos construídos
- Facilidade de uso em outros documentos
- Intuitiva
- Fácil edição

Introdução

Que tal essa figura?

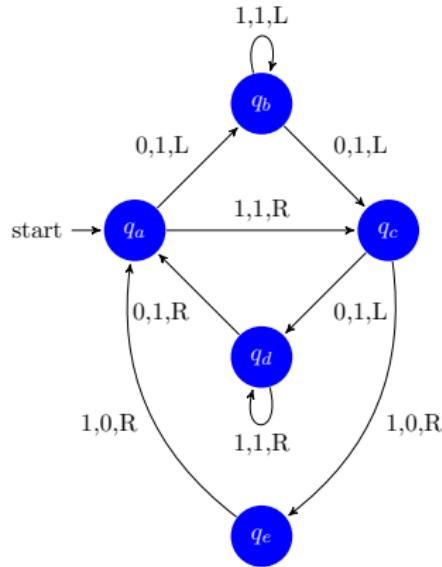


Figura: Desenho construído com Tikz

Introdução

E essa?

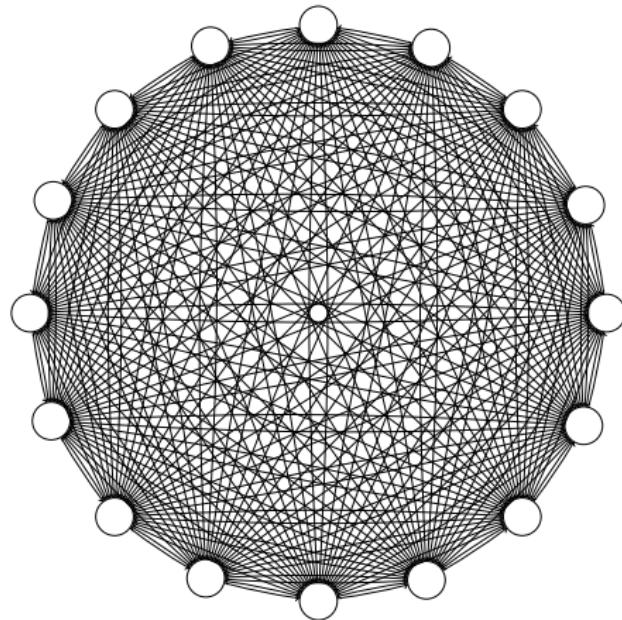


Figura: Desenho construído com *Tikz*

Introdução

E essa?

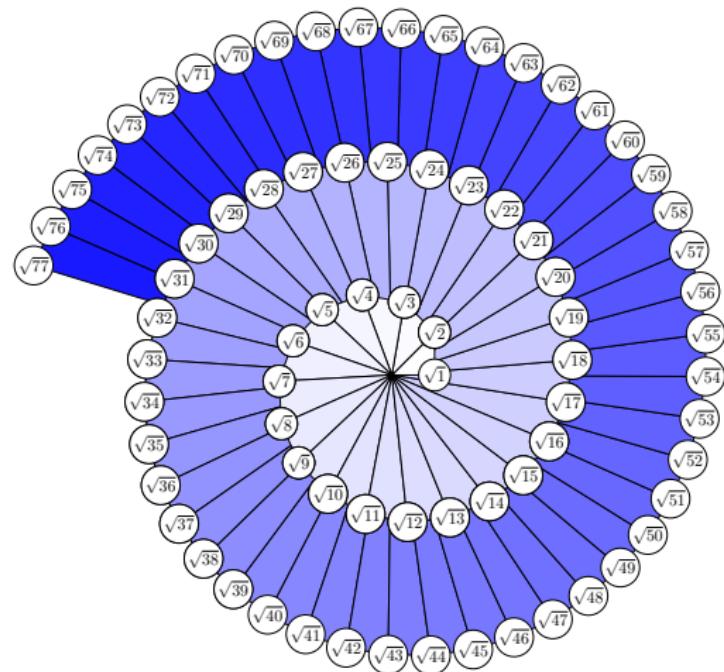


Figura: Desenho construído com Tikz

Introdução

Como usar? Primeiros passos...

- Para começar com o TikZ, precisamos de carregar o pacote tikz:
`\usepackage{tikz}`
- Agora sempre que quisermos criar um diagrama TikZ, precisamos de utilizar o ambiente tikzpicture.
`\begin{tikzpicture}[opções globais da figura]
<os códigos vão aqui...>
\end{tikzpicture}`
- É possível inserir os comandos *inline* da seguinte forma:
`\tikz;`

De agora em diante, veremos como poderemos construir tais objetos

Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Comandos iniciais

Comandos básicos: dentro do ambiente usamos:

- `\draw` : desenhar linhas;
- `\fill` : para preencher áreas com cores sólidas;
- `\shade` : para preencher áreas com cores gradientes;
- `\clip` : para cortar;

sempre terminados em ponto-e-vírgula

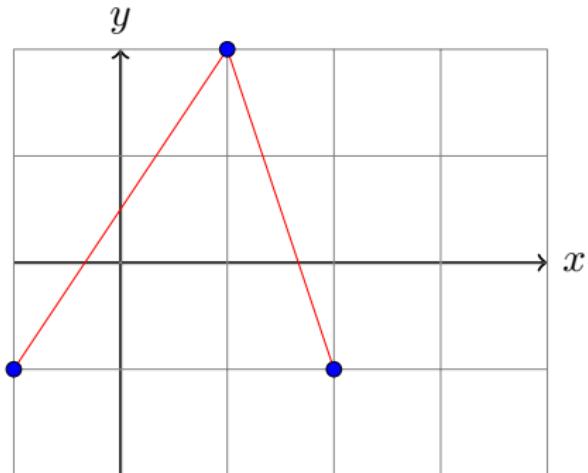
- Com *opções adicionais* dentro de chavetas [...] após os comandos enumerados anteriormente, modificamos os objetos que queremos desenhar

Comandos iniciais

- As coordenadas cartesianas de um ponto são fornecidas no formato (a, b)
- As coordenadas cartesianas de um ponto são fornecidas no formato $(\theta : r)$ onde θ é medido em graus.
- Um sinal de $+$ antes das coordenadas como $+(a, b)$ significa um shift de (a, b) a partir do primeiro ponto.
- Um duplo sinal de $++$ antes das coordenadas como $++(a, b)$ significa um shift de (a, b) a partir do último ponto.

Comandos iniciais

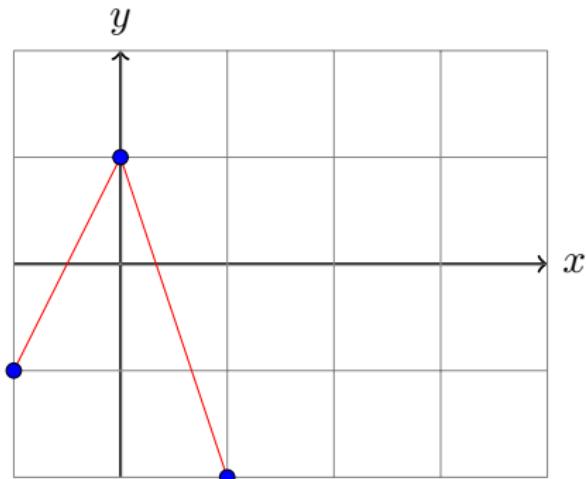
```
\begin{tikzpicture}
\draw [draw=red]
(-1,-1) -- (1,2) -- (2,-1);
\draw [fill=blue]
(-1,-1) circle [radius=2pt]
(1,2) circle [radius=2pt]
(2,-1) circle [radius=2pt];
\end{tikzpicture}
```



O ponto de referência é $(0, 0)$

Comandos iniciais

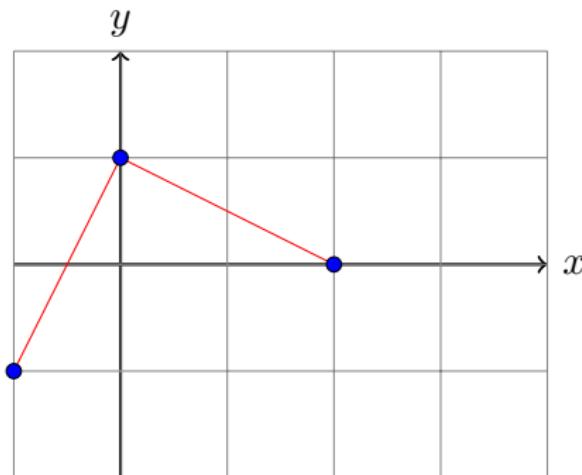
```
\begin{tikzpicture}
\draw [draw=red]
(-1,-1) -- +(1,2) -- +(2,-1);
\draw [fill=blue]
(-1,-1) circle [radius=2pt]
+(1,2) circle [radius=2pt]
+(2,-1) circle [radius=2pt];
\end{tikzpicture}
```



O ponto de referência é $(-1, -1)$

Comandos iniciais

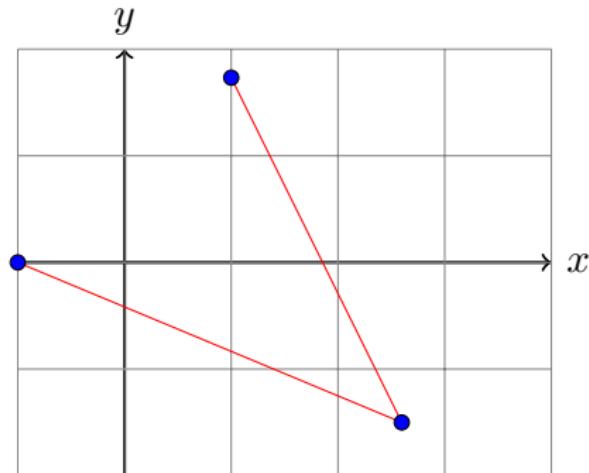
```
\begin{tikzpicture}
\draw [draw=red]
(-1,-1) -- ++(1,2) -- ++(2,-1);
\draw [fill=blue]
(-1,-1) circle [radius=2pt]
++(1,2) circle [radius=2pt]
++(2,-1) circle [radius=2pt];
\end{tikzpicture}
```



O ponto de referência é o último da série.

Comandos iniciais

```
\begin{tikzpicture}
\draw[draw=red]
(0:-1) -- (-30:3) -- (60:2);
\draw[fill=blue]
(0:-1) circle [radius=2pt]
(-30:3) circle [radius=2pt]
(60:2) circle [radius=2pt];
\end{tikzpicture}
```

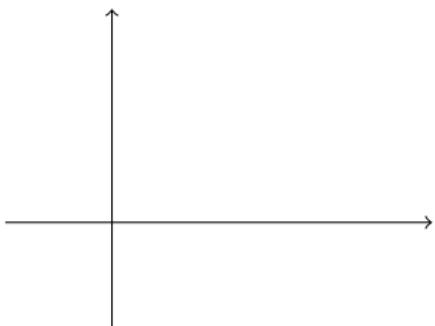


Atenção às coordenadas polares!

Comandos iniciais

Para desenhar os eixos cartesianos com $-1 \leq x \leq 3$ e $-1 \leq y \leq 2$, só usar:

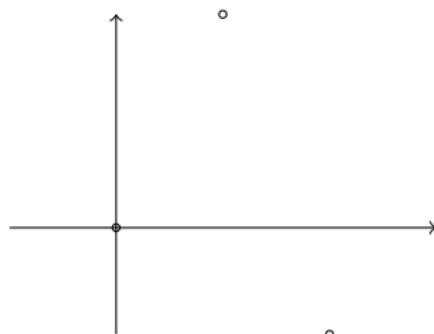
```
\begin{tikzpicture}
\draw[->] (-1,0) -- (3,0);
\draw[->] (0,-1) -- (0,2);
\end{tikzpicture}
```



Comandos iniciais

Juntando dois comandos num só ambiente para plotar pontos neste sistema de eixos é só adotar:

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

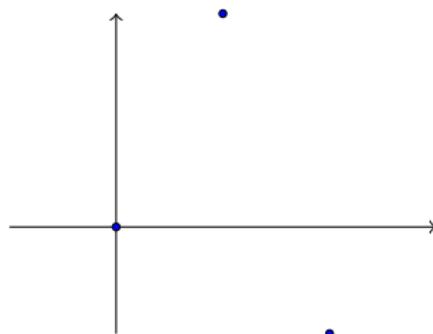


Daqui a pouco veremos como customizar cada objeto construído!

Comandos iniciais

Podemos preencher estes pontos:

```
\begin{tikzpicture}
\draw[->] (-1,0) -- (3,0);
\draw[->] (0,-1) -- (0,2);
\draw[fill=blue] (0,0)
circle [radius=1pt]
(1,2) circle [radius=1pt]
(2,-1) circle [radius=1pt];
\end{tikzpicture}
```



Perceba que já editamos estes pontos!

Comandos iniciais

Desenhando um segmento de $(0, 0)$ a $(1, 2)$, usamos o comando:

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



Desenhando um triângulo de vértices $(0, 0)$, $(1, 2)$ e $(2, 1)$, usamos o comando:

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



Comandos iniciais

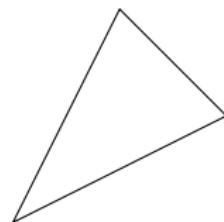
Desenhando um segmento de $(0, 0)$ a $(1, 2)$, usamos o comando:

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



Desenhando um triângulo de vértices $(0, 0)$, $(1, 2)$ e $(2, 1)$, usamos o comando:

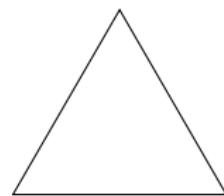
```
\begin{tikzpicture}
\draw (0,0)--(1,2)--(2,1)--cycle;
\end{tikzpicture}
```



Comandos iniciais

Se quisermos um triângulo equilátero de lado 2cm é só usar os pontos em coordenadas polares:

```
\begin{tikzpicture}
\draw (0,0) -- (60:2) --
(0:2) --cycle;
\end{tikzpicture}
```



Observe ao final da série o comando `--cycle` para fechar o polígono.

Comandos iniciais

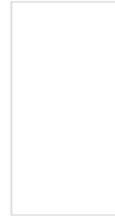
Se quisermos um retângulo cuja diagonal é o segmento $(0, 0)$ e $(1, 2)$ usamos:

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



Ou mais compactamente para ter o mesmo efeito:

```
\begin{tikzpicture}
\draw (0,0) rectangle (1,2);
\end{tikzpicture}
```



Comandos iniciais

Se quisermos um retângulo cuja diagonal é o segmento $(0, 0)$ e $(1, 2)$ usamos:

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



Ou mais compactamente para ter o mesmo efeito:

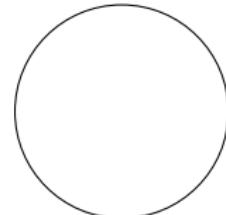
```
\begin{tikzpicture}
\draw (0,0) rectangle (1,2);
\end{tikzpicture}
```



Comandos iniciais

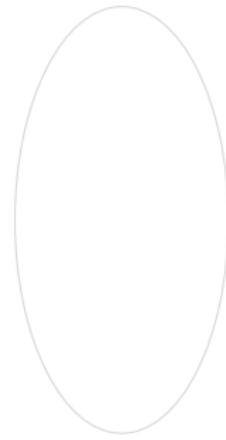
Circunferência de raio 1 cm
centrada em $(2, 2)$ é conseguida
com:

```
\begin{tikzpicture}
\draw (2,2) circle (1cm);
\end{tikzpicture}
```



Já a elipse de centro em $(0, 0)$ e
semi-eixos 1 e 2 é obtida com:

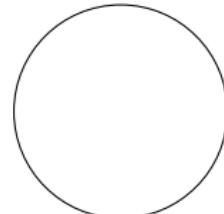
```
\begin{tikzpicture}
\draw (0,0) ellipse (1cm and 2cm);
\end{tikzpicture}
```



Comandos iniciais

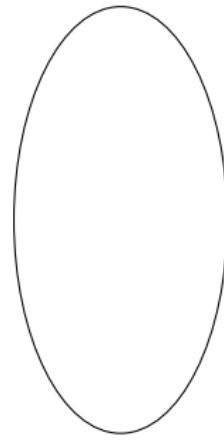
Circunferência de raio 1 cm
centrada em $(2, 2)$ é conseguida
com:

```
\begin{tikzpicture}
\draw (2,2) circle (1cm);
\end{tikzpicture}
```



Já a elipse de centro em $(0, 0)$ e
semi-eixos 1 e 2 é obtida com:

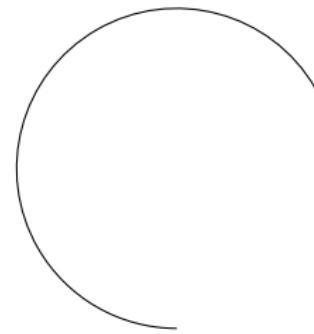
```
\begin{tikzpicture}
\draw (0,0) ellipse (1cm and 2cm);
\end{tikzpicture}
```



Comandos iniciais

Arco de circunferência, iniciando em $(1, 0)$, ângulos inicial e final iguais a 30° e 270° de raio 1.5 cm é feito com:

```
\begin{tikzpicture}
\draw (1,0) arc (30:270:1.5cm);
\end{tikzpicture}
```



Podemos também fazer um arco elíptico, iniciando de $(1, 0)$ ângulos inicial e final iguais a 30° e 270° de raios 1.0 (na direção x) e 1.5 (na direção de y) com:

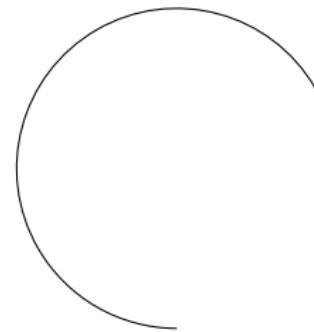
```
\begin{tikzpicture}
\draw (1,0) arc (30:270:1 and 1.5);
\end{tikzpicture}
```



Comandos iniciais

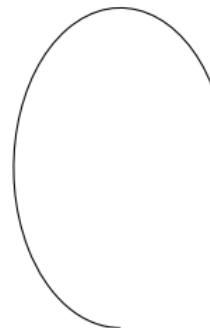
Arco de circunferência, iniciando em $(1, 0)$, ângulos inicial e final iguais a 30° e 270° de raio 1.5 cm é feito com:

```
\begin{tikzpicture}
\draw (1,0) arc (30:270:1.5cm);
\end{tikzpicture}
```



Podemos também fazer um arco elíptico, iniciando de $(1, 0)$ ângulos inicial e final iguais a 30° e 270° de raios 1.0 (na direção x) e 1.5 (na direção de y) com:

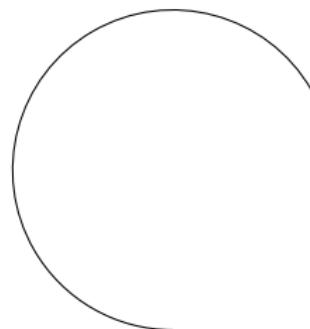
```
\begin{tikzpicture}
\draw (1,0) arc (30:270:1 and 1.5);
\end{tikzpicture}
```



Comandos iniciais

Comandos alternativos para os mesmos efeitos:

```
\begin{tikzpicture}
\draw (1,0) arc [start angle=30,
end angle=270, radius=1.5cm];
\end{tikzpicture}
```



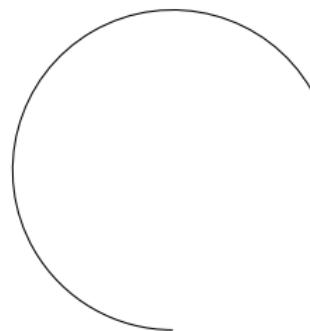
```
\begin{tikzpicture}
\draw (1,0) arc[x radius=1, y radius=1.5,
start angle=30, end angle=270];
\end{tikzpicture}
```



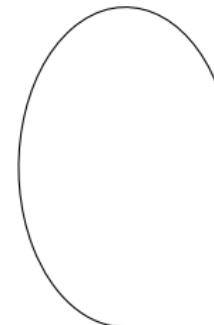
Comandos iniciais

Comandos alternativos para os mesmos efeitos:

```
\begin{tikzpicture}
\draw (1,0) arc [start angle=30,
end angle=270, radius=1.5cm];
\end{tikzpicture}
```



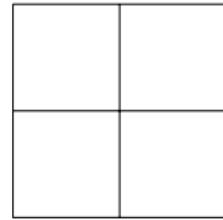
```
\begin{tikzpicture}
\draw (1,0) arc[x radius=1, y radius=1.5,
start angle=30, end angle=270];
\end{tikzpicture}
```



Comandos iniciais

Para desenhar uma grid de ancorada nos pontos $(0,0)$ e $(2,2)$ basta usar:

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



Se quiser uma grade diferente, basta inserir dentro das opções o refinamento da grade:

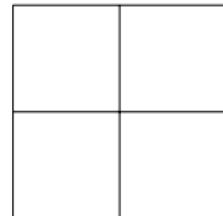
```
\begin{tikzpicture}
\draw[xstep=0.5,ystep=0.25] (0,0)
grid (2,2);
\end{tikzpicture}
```



Comandos iniciais

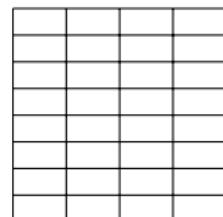
Para desenhar uma grid de ancorada nos pontos $(0,0)$ e $(2,2)$ basta usar:

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



Se quiser uma grade diferente, basta inserir dentro das opções o refinamento da grade:

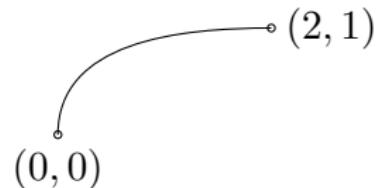
```
\begin{tikzpicture}
\draw[xstep=0.5,ystep=0.25] (0,0)
grid (2,2);
\end{tikzpicture}
```



Comandos iniciais

Curvas definidas por dois pontos, A e B . Definimos os ângulos de saída e entrada em cada ponto. Por exemplo se $A = (0, 0)$ e $B = (2, 1)$, definimos que a curva sai de A com um ângulo de 90° e chega em B com um ângulo de 180° :

```
\draw(0,0) node[below]{$(0,0)$}
to[out=90,in=180] (2,1) node[right]{$(2,1)$};
\draw(0,0) circle[radius=1pt]
(2,1) circle[radius=1pt];
```

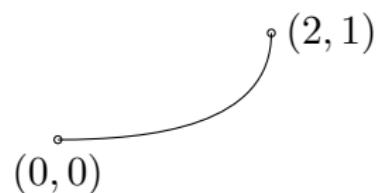


Note que já começamos a inserir as opções, como texto, posicionamento em cada nó!

Comandos iniciais

Curvas definidas por dois pontos, A e B . Definimos os ângulos de saída e entrada em cada ponto. Por exemplo se $A = (0, 0)$ e $B = (2, 1)$, definimos que a curva sai de A com um ângulo de 0° e chega em B com um ângulo de 270° :

```
\draw(0,0) node[below]{$(0,0)$}
      to[out=0,in=270] (2,1) node[right]{$(2,1)$};
\draw(0,0) circle[radius=1pt]
(2,1) circle[radius=1pt];
```



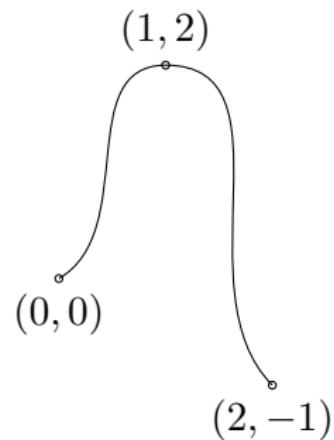
Podemos construir uma curva com mais pontos, bastando imitar estes comandos.

Comandos iniciais

Agora, se $A = (0, 0)$, $B = (2, 1)$ e $C = (3, 1.5)$, definimos que a curva sai de A com um ângulo de 30° e chega em B com um ângulo de 180° , sai de B com 0° e chega em C com 135° :

```
\begin{tikzpicture}
\draw(0,0) node[below]{$(0,0)$}
to[out=30,in=180] (1,2)
node[above]{$(1,2)$} to[out=0,in=135]
(2,-1) node[below]{$(2,-1)$};

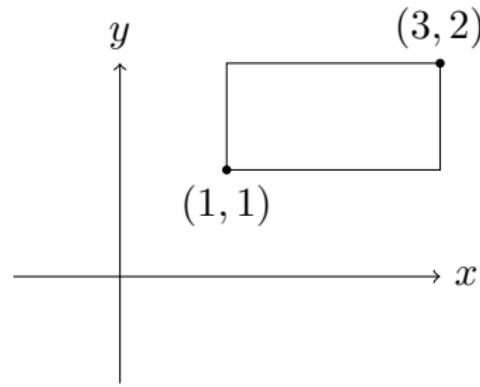
\draw(0,0) circle [radius=1pt]
(1,2) circle [radius=1pt]
(2,-1) circle [radius=1pt];
\end{tikzpicture}
```



Comandos iniciais

Mesclando comandos com algumas opções.

```
\begin{tikzpicture}
\draw[->] (-1,0) -- (3,0) node[right] {$x$};
\draw[->] (0,-1) -- (0,2) node[above] {$y$};
\draw[fill] (1,1) circle (1pt);
\draw[fill] (3,2) circle (1pt);
\draw (1,1) node[below] {$(1,1)$} rectangle (3,2) node[above]
\end{tikzpicture}
```



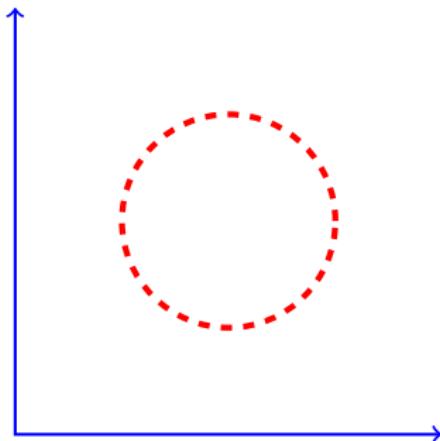
Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Customização dos objetos

Como dito anteriormente, as opções são inseridas dentro de chavetas [...] após o comando `\draw`.

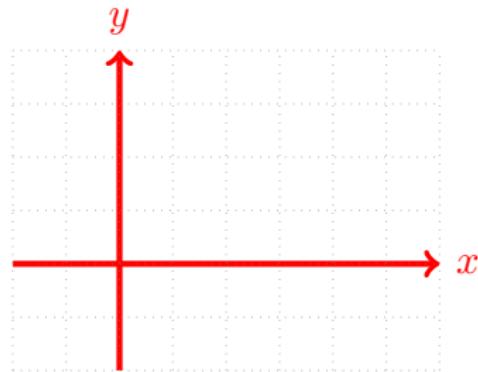
```
\begin{tikzpicture}
\draw[<->, thick, blue]
(0,4)--(0,0) -- (4,0);
\draw[red, ultra thick, dashed]
(2,2) circle (1cm);
\end{tikzpicture}
```



Customização dos objetos

Uso de uma grade e eixos coordenados.

```
\begin{tikzpicture}
\draw[->,red,thick] (-1,0)--(3,0)
node[right]{$x$};
\draw[->,red,thick] (0,-1)--(0,2)
node[above]{$y$};
\draw[step=0.5cm,help lines]
(-1,-1) grid (3,2);
\end{tikzpicture}
```

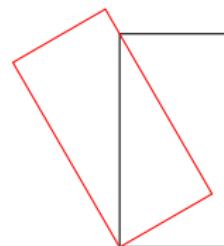


- **thick:** linha mais grossa
- **step:** subdivisão da grade
- **help lines:** linhas cinzas
- **dotted:** pontilha a linha

Customização dos objetos

Rotações de objetos. Rotacionamos o retângulo (preto) em 30° no sentido anti-horário (em vermelho).

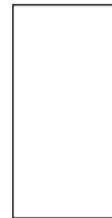
```
\begin{tikzpicture}
\draw (0,0) rectangle (1,2);
\draw[rotate=30,red] (0,0)
    rectangle (1,2);
\end{tikzpicture}
```



Customização dos objetos

Translações de objetos. Arrastamos o retângulo (preto) em na direção do vetor $(1, 3)$ (em vermelho).

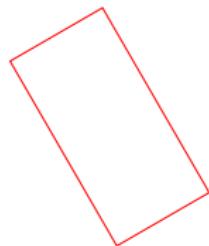
```
\begin{tikzpicture}
\draw(0,0) rectangle (1,2);
\draw[shift={(1,3)},red]
(0,0) rectangle (1,2);
\end{tikzpicture}
```



Customização dos objetos

Translação e rotação. Arrastamos o retângulo (preto) em na direção do vetor $(1, 3)$ e giramos 30° (em vermelho).

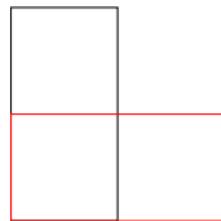
```
\begin{tikzpicture}
\draw(0,0) rectangle (1,2);
\draw[rotate=30,shift={(1,3)},red]
(0,0) rectangle (1,2);
\end{tikzpicture}
```



Customização dos objetos

Mudança na escala. Dobramos a escala na direção x e reduzimos à metade no eixo y .

```
\begin{tikzpicture}
\draw(0,0) rectangle (1,2);
\draw[xscale=2,yscale=0.5,red]
(0,0) rectangle (1,2);
\end{tikzpicture}
```



Customização dos objetos

Estilo das linhas:

- **solid**: _____
- **dotted**:
- **loosely dotted**:
- **densely dotted**:
- **dashed**: -----
- **loosely dashed**: - - - -
- **densely dashed**: -----
- **dashdotted**: - - - -

Grossura das linhas:

- **ultra thin**: _____
- **very thin**: _____
- **thin**: _____
- **semithick**: _____
- **thick**: _____
- **very thick**: _____
- **ultra thick**: _____
- **line width=5pt**: 

Customização dos objetos

Cores:

- black 
- blue 
- brown 
- cyan 
- darkgray 
- gray 
- green 
- lightgray 
- lime 
- magenta 
- olive 
- orange 
- pink 
- purple 
- red 
- teal 
- violet 
- white 
- yellow 

- Podemos misturar duas cores fornecendo os comandos `cor1!X!cor2!` onde X representa a porcentagem da cor `cor1`
- Se `cor2` for branca, poderemos omití-la.

Customização dos objetos

Forma de flechas:

- [->]: 
- [<->]: 
- [|->]: 
- [|->>]: 
- [>-]: 
- [-o]: 
- [diamond-]: 
- [-latex]: 

Customização dos objetos

Preenchimento se dá com o comando `\fill`. O contorno se dá com o comando `\draw` dentro das opções.

```
\begin{tikzpicture}
\fill[blue] (0,0) rectangle (1,1);
\fill[blue!40!] (1.5,0)
rectangle (2.5,1);
\fill[blue!40!lime,draw=yellow]
(3,0) rectangle (4,1);
\end{tikzpicture}
```



Customização dos objetos

Poderemos mudar a opacidade do preenchimento com o comando
`opacity=x`

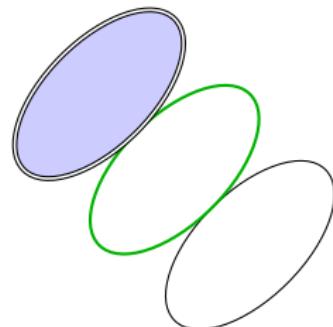
```
\begin{tikzpicture}
\fill[blue] (0,0) rectangle (1,1);
\fill[blue!40!] (1.5,0)
rectangle (2.5,1);
\fill[blue!40!lime,draw=yellow]
(3,0) rectangle (4,1);
\end{tikzpicture}
```



Customização dos objetos

Outros comandos e outputs...

```
\begin{tikzpicture}[rotate=45]
\draw (0,0) ellipse (1cm and 0.5cm);
\draw[green!70!black, thick]
(0,1) ellipse (1cm and 0.5cm);
\draw[fill=blue!20, double]
(0,2) ellipse (1cm and 0.5cm);
\end{tikzpicture}
```



Note que `rotate=45` está mas configurações globais do desenho.

Customização dos objetos

Sombreamentos são conseguidos com `shade`. Veja os exemplos:

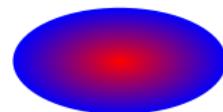
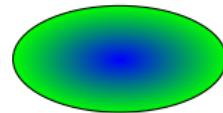
```
\begin{tikzpicture}
\shade [top color=blue,bottom color=red]
(0,1) ellipse (1cm and 0.5cm);
\shade [left color=blue,right color=red]
(0,0) ellipse (1cm and 0.5cm);
\end{tikzpicture}
```



Customização dos objetos

Sombreamentos são conseguidos com `shade`. Veja os exemplos:

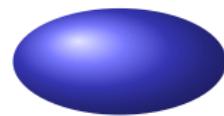
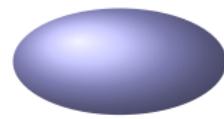
```
\begin{tikzpicture}
\shade[inner color=blue,
outer color=green,draw=black]
(0,1.5) ellipse (1cm and 0.5cm);
\shade[inner color=red,outer color=blue]
(0,0) ellipse (1cm and 0.5cm);
\end{tikzpicture}
```



Customização dos objetos

Sombreamentos são conseguidos com `shade`. Veja os exemplos:

```
\begin{tikzpicture}
\shade [ball color=blue!30!] 
(0,1.5) ellipse (1cm and 0.5cm);
\shade [ball color=blue!70!] 
(0,0) ellipse (1cm and 0.5cm);
\end{tikzpicture}
```



Customização dos objetos

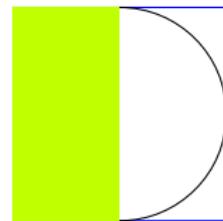
O comando `clip` é muito útil, e serve para cortar objetos.

- Tudo que estiver fora dele (que fora definido depois) não é exibido.
- Efeitos de múltiplos `clip` é a exibição da intersecção das regiões recordadas
- O ambiente `scope` delimita a área que será afetada pelos comandos `clip`

Customização dos objetos

Desenhamos um retângulo azul, uma circunferência preta e um retângulo em cor limão.

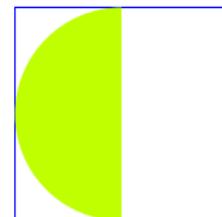
```
\begin{tikzpicture}
\draw[blue] (-1,-1) rectangle (1,1);
\draw (0,0) circle (1cm);
\fill[lime] (0,1) rectangle (-1,-1);
\end{tikzpicture}
```



Customização dos objetos

Agora, cortamos em formato circular e então a região retangular fora desta circunferência não mais aparece.

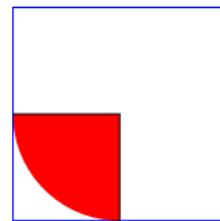
```
\begin{tikzpicture}
\draw[blue] (-1,-1) rectangle (1,1);
\clip (0,0) circle (1cm);
\fill[lime] (0,1) rectangle (-1,-1);
\end{tikzpicture}
```



Customização dos objetos

Agora desenhamos um retângulo menor em vermelho. Claro que só vai aparecer o que está dentro da circunferência.

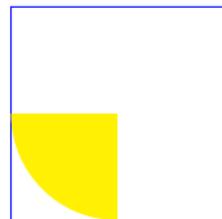
```
\begin{tikzpicture}
\draw[blue] (-1,-1) rectangle (1,1);
\clip (0,0) circle (1cm);
\draw[fill=red] (-1,-1) rectangle (0,0);
\end{tikzpicture}
```



Customização dos objetos

Cortamos o que está fora deste retângulo menor...

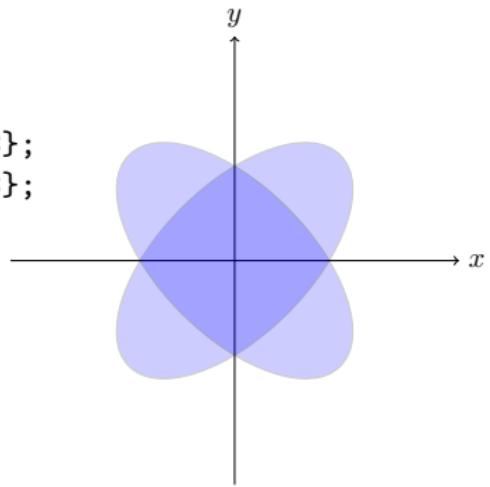
```
\begin{tikzpicture}
\draw[blue] (-1,-1) rectangle (1,1);
\clip (0,0) circle (1cm);
\clip (-1,-1) rectangle (0,0);
\fill[yellow] (0,1) rectangle (-1,-1);
\end{tikzpicture}
```



Customização dos objetos

Outro exemplo passo-a-passo...

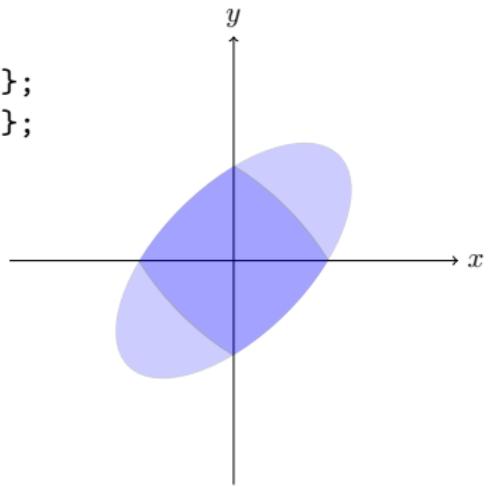
```
\begin{tikzpicture}
\draw[->] (-3,0) -- (3,0) node[right] {$x$};
\draw[->] (0,-3) -- (0,3) node[above] {$y$};
\draw[rotate=45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\draw[rotate=-45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\end{tikzpicture}
```



Customização dos objetos

Outro exemplo passo-a-passo...

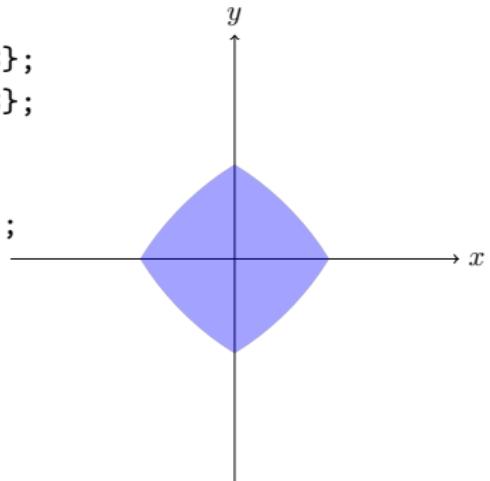
```
\begin{tikzpicture}
\draw[->] (-3,0) -- (3,0) node[right] {$x$};
\draw[->] (0,-3) -- (0,3) node[above] {$y$};
\begin{scope}
\clip[rotate=45] (0,0) ellipse(2cm and 1cm);
\draw[rotate=45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\draw[rotate=-45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\end{scope}
\end{tikzpicture}
```



Customização dos objetos

Outro exemplo passo-a-passo...

```
\begin{tikzpicture}
\draw[->] (-3,0) -- (3,0) node[right] {$x$};
\draw[->] (0,-3) -- (0,3) node[above] {$y$};
\begin{scope}
\clip[rotate=45] (0,0) ellipse(2cm and 1cm);
\clip[rotate=-45] (0,0) ellipse(2cm and 1cm);
\draw[rotate=45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\draw[rotate=-45,fill=blue,opacity=0.2]
(0,0) ellipse (2cm and 1cm);
\end{scope}
\end{tikzpicture}
```



Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Nós em tikz

- Um nó serve basicamente pra inserir um texto em alguma coordenada específica.
- Estrutura do comando:
`node [opções] (nome) at (x,y) {texto};`
- O texto no nó pode ser mostrado no interior de uma figura geométrica se for acrescentado antes de `node` o comando `\draw`
- A forma da figura do nó é definida usando os comandos do tipo `rectangle`, `circle`, `diamond`, etc.

Nós em tikz

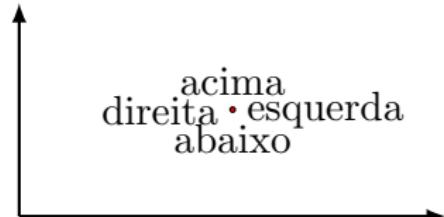
Alguns opcionais de posição do comando `node`:

- `above`, `left`, `right`, `below`, `above right`, etc...
- `xshift=x` e `yshift=y` denotam deslocamentos do objeto
- `pos=n` onde $n \in [0, 1]$ é a posição relativa do objeto ao longo de um caminho.
- Para `pos=0.5` equivale `midway`
- Para inclinar o texto, use `sloped`.

Nós em tikz

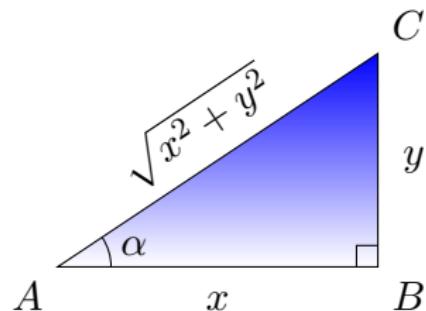
Texto e figuras

```
\begin{tikzpicture}
\begin{tikzpicture}
\def\x{2}
\def\y{1}
\draw [thick, ->,-latex] (0,0) -- (2,0);
\draw [thick, ->,-latex] (0,0) -- (0,2);
\node (Ponto) at (\x,\y) {};
\draw[fill] (Ponto) circle [radius=0.025];
\node [below] at (Ponto) {abaixo};
\node [above] at (Ponto) {acima};
\node [left] at (Ponto) {direita};
\node [right] at (Ponto) {esquerda};
\end{tikzpicture}
\end{tikzpicture}
```



Nós em tikz

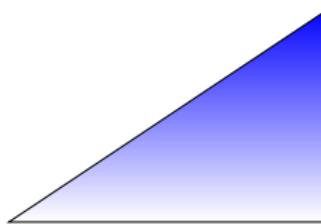
Vamos construir passo-a-passo todos os elementos da figura:



Nós em tikz

Primeiro dizemos que iremos sombrear o polígono de vértices $A = (0, 0)$, $B = (3, 0)$ e $C = (3, 2)$ fornecendo os comandos:

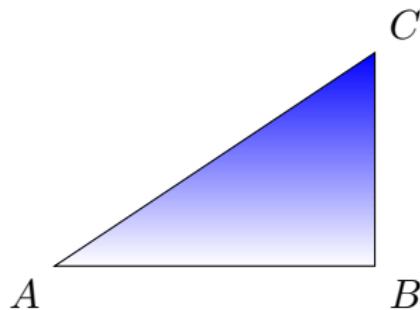
```
\begin{tikzpicture}
\shade[top color=blue,bottom color=white,draw=black]
(0,0) -- (3,0) -- (3,2) -- cycle;
\end{tikzpicture}
```



Nós em tikz

Agora inserimos os labels de cada vértices nas posições indicadas:

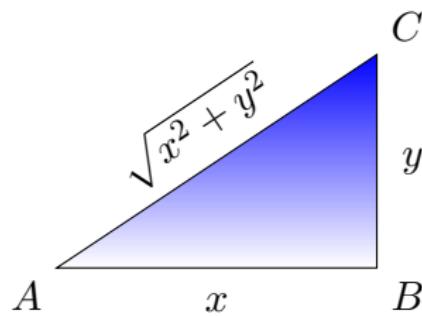
```
\begin{tikzpicture}
\shade[top color=blue,bottom color=white,draw=black]
(0,0) node[below left]{$A$} -- (3,0) node[below right] {$B$}
-- (3,2) node[above right]{$C$} -- cycle;
\end{tikzpicture}
```



Nós em tikz

Inserimos as medidas dos lados indicados:

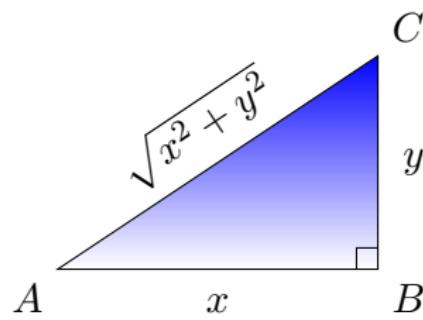
```
\begin{tikzpicture}
\shade[top color=blue,bottom color=white,draw=black]
(0,0) node[below left]{$A$} -- (3,0)
node[below, pos=0.5, yshift=-0.1cm]{$x$} node[below right] {$B$}
-- (3,2) node[right, pos=0.5, xshift=0.1cm]{$y$}
node[above right]{$C$} --
node[above, midway, sloped]{$\sqrt{x^2+y^2}$} cycle;
\end{tikzpicture}
```



Nós em tikz

Desenhamos o ângulo reto:

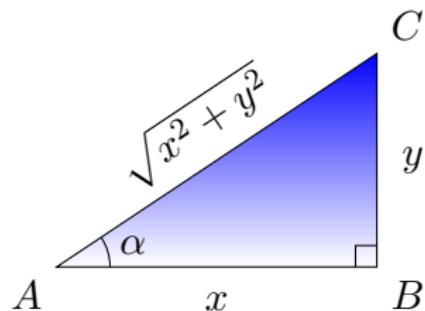
```
\begin{tikzpicture}
\shade[top color=blue,bottom color=white,draw=black]
(0,0) node[below left]{$A$} -- (3,0)
node[below, pos=0.5, yshift=-0.1cm]{$x$} node[below right] {$B$}
-- (3,2) node[right, pos=0.5, xshift=0.1cm]{$y$}
node[above right]{$C$} --
node[above, midway, sloped]{$\sqrt{x^2+y^2}$} cycle;
\draw (3,0.2) -- (2.8,0.2) -- (2.8,0);
\end{tikzpicture}
```



Nós em tikz

Finalmente o ângulo marcado α :

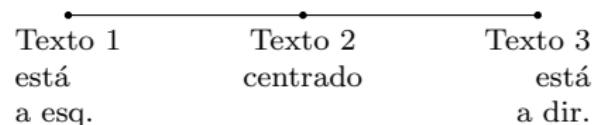
```
\begin{tikzpicture}
\shade[top color=blue,bottom color=white,draw=black]
(0,0) node[below left]{$A$} -- (3,0)
node[below, pos=0.5, yshift=-0.1cm]{$x$} node[below right] {$B$}
-- (3,2) node[right, pos=0.5, xshift=0.1cm]{$y$}
node[above right]{$C$} --
node[above, midway, sloped]{$\sqrt{x^2+y^2}$} cycle;
\draw (3,0.2) -- (2.8,0.2) -- (2.8,0);
\draw (0.5,0) arc (0:33.69:0.5) node[right, pos=0.7] {$\alpha$};
\end{tikzpicture}
```



Nós em tikz

- O texto em múltiplas linhas pode ser incluído dentro de um nó. Uma nova linha é indicada por dupla barra invertida \\, mas adicionalmente é necessário especificar o alinhamento usando a opção de nó \align=. Aqui um exemplo:

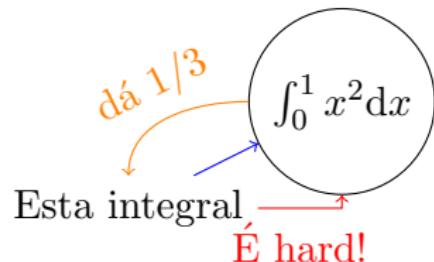
```
\begin{tikzpicture}[scale=0.5]
\filldraw
(0,0) circle (2pt) node[align=left,
below] {Texto 1\\está \\ esq.} --
(4,0) circle (2pt) node[align=center,
below] {Texto 2\\centrado} --
(8,0) circle (2pt) node[align=right,
below] {Texto 3\\ está\\ dir.};
\end{tikzpicture}
```



Nós em tikz

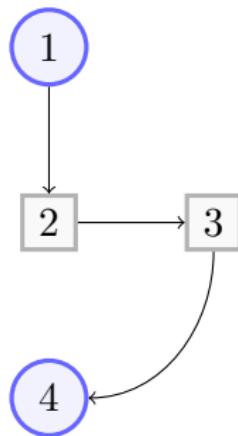
- Veja este outro exemplo com vários comandos.

```
\begin{tikzpicture}[scale=0.5]
\path (0,0) node(Texto) {Esta integral}
(4,2) node[circle,draw](Int)
{$\int_0^1 x^2 \mathrm{d} x$};
\draw[->,blue] (Texto) -- (Int);
\draw[->,red] (Texto.east) -|
node[near start,below]
{É hard!} (Int.south);
\draw[<-,orange] (Texto.north)
to[out=90,in=180]
node[above,sloped] {dá $1/3$} (Int.west);
\end{tikzpicture}
```



Nós em tikz

Vamos produzir esta figura usando os **nodes** e as posições relativas entre eles:



Nós em tikz

Vamos construir passo-a-passo um pequeno diagrama.

- Primeiro, definimos as configurações globais da figura ao inserirmos os comandos:

```
\begin{tikzpicture}[
NoCircular/.style={circle, draw=blue!60, fill=blue!5,
very thick, minimum size=7mm},
NoQuadrado/.style={rectangle, draw=gray!60, fill=gray!5,
very thick, minimum size=5mm},
]
\end{tikzpicture}
```

definindo como cada nó é.

- Em seguida, vamos a construção do diagrama

Nós em tikz

Desenhamos os nós:

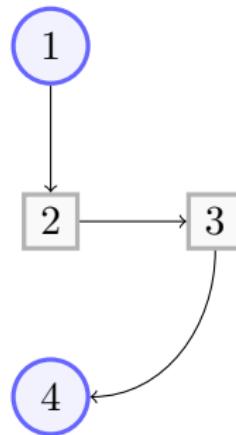
```
\node[NoQuadrado](inicio)  
    {2};  
  
\node[NoCircular] (Cima)  
[above=of inicio] {1};  
  
\node[NoQuadrado](Direita)  
[right=of inicio] {3};  
  
\node[NoCircular] (Abaixo)  
[below=of inicio] {4};
```



Nós em tikz

Agora desenhamos os caminhos usando os comandos vistos e posicionando de onde partem/chegam as flechas.

```
\draw[->] (cima.south) --  
            (inicio.north);  
\draw[->] (inicio.east) --  
            (Direita.west);  
\draw[->] (Direita.south)  
          to[out=270,in=0] (Abaixo.east);
```



Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Estruturas de repetição

Podemos criar estruturas incríveis com este comando que tem a seguinte estrutura

```
\foreach <variables> in {<list>} {<commands>}
```

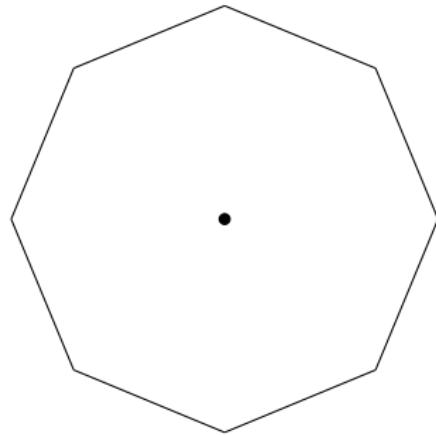
- No caso mais fácil, <variables> é um único comando TEX como \i.
- <list> é uma lista de valores separados por vírgulas. Qualquer coisa pode ser usada como um valor, mas os números são mais prováveis.
- Finalmente, no caso mais fácil, {<commands>} é um conjunto de comandos a ser executado, sempre entre colchetes.

Vamos dar alguns exemplos de como essas construções são feitas.

Estruturas de repetição

Construindo um polígono regular de n lados inscrito numa circunferência de raio r

```
\newcommand{\poligonoregular}[2]{  
\\begin{tikzpicture}  
\\draw[fill] (0,0) circle (0.05);  
\\foreach \\i in {1,2,...,#1}{  
\\draw({(360/#1)*\\i}:#2)--  
({(360/#1)*(\\i+1)}:#2);}  
\\end{tikzpicture}}  
\\begin{document}  
\\poligonoregular{8}{2}  
\\end{document}}
```



Estruturas de repetição

Laços simples...

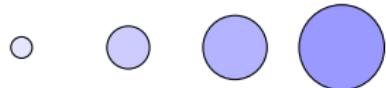
```
\begin{tikzpicture}
\foreach \x in {0,1,2,3}
{
    \draw (\x,0) circle (0.2cm); } \end{tikzpicture}
```



Estruturas de repetição

Laços simples... e comandando a tonalidade das cores e o raio das circunferências

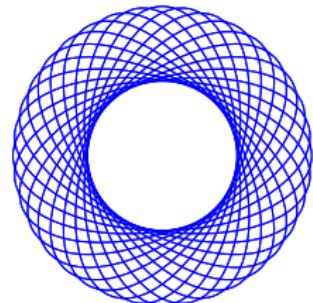
```
\begin{tikzpicture}
\foreach \x in {1,2,3,4}
{
    \draw[fill=blue!\x0]
        (\x,0) circle (0.\x0cm);
}
\end{tikzpicture}
```



Estruturas de repetição

Outro laço simples com controle da rotação:

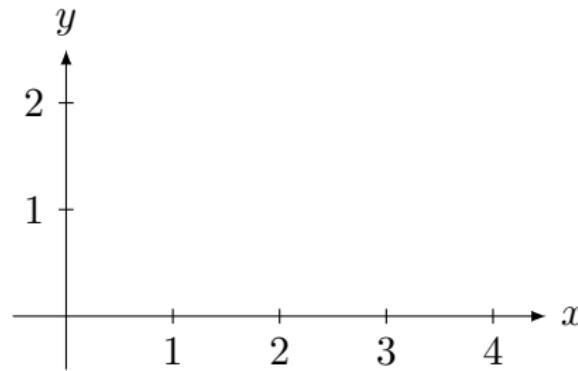
```
\begin{tikzpicture}[scale=1]
\clip (-3.5,-2) rectangle (3,2);
\foreach \i in {10,20,\dots,360}{
\begin{scope}[rotate=\i]
\draw[blue!] (0,0) ellipse (1 and 2);
\end{scope}}
\end{tikzpicture}
```



Estruturas de repetição

Veja este exemplo:

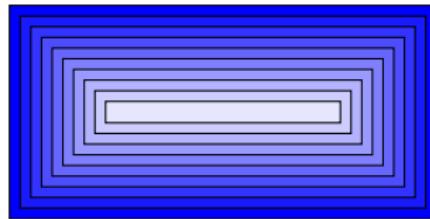
```
\begin{tikzpicture}
\draw (-.5,0)--(4.5,0) node[right]{$x$};
\draw (0,-1.5)--(0,2.5) node[above]{$y$};
\foreach \x in {1,2,3,4}
    \draw (\x,2pt)--(\x,-2pt) node[below] {$\x$};
\foreach \y/\ytext in {1,2}
    \draw (2pt,\y)--(-2pt,\y) node[left] {$\ytext$};
\end{tikzpicture}
```



Estruturas de repetição

Outro exemplo:

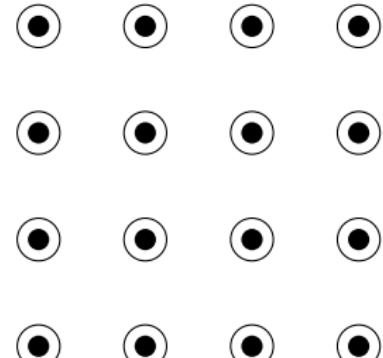
```
\begin{tikzpicture}
\foreach \y in {10, ..., 1}
\draw[fill=brown!\y0] (-0.1*\y-1, -0.1*\y)
rectangle (0.1*\y+1, 0.1*\y);
\end{tikzpicture}
```



Estruturas de repetição

Laços aninhados...

```
\begin{tikzpicture}
    \foreach \x in {0,1,2,3}{
        \foreach \y in {0,1,2,3}{
            \draw (\x,\y) circle (0.2cm);
            \fill (\x,\y) circle (0.1cm);
        }
    }
\end{tikzpicture}
```



Estruturas de repetição

Laços simples... e comandando a tonalidade das cores e o raio das circunferências e fazendo cálculos

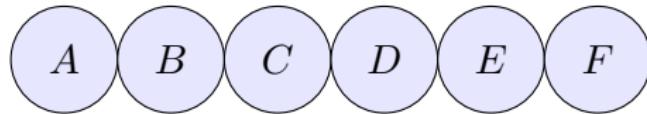
```
\begin{tikzpicture}
\draw[->, -latex'] (0,0) to (4 ,0);
\foreach \x/\txt in
{0,1,0.5*pi/\frac{\pi}{2},pi/\pi$,2.71/e$}{
\draw (\x,0.1) to (\x ,-0.1);
\node at (\x,-0.3) {\footnotesize {\txt}};}
```



Estruturas de repetição

Uso de duas variáveis por looping pode também ser feito, como a seguir.

```
\begin{tikzpicture}
\foreach \i\j in {1/A,2/B,3/C,4/D,5/E,6/F}{
\draw[fill=blue!10] (\i,0) circle (0.5cm)
node[anchor=center] {$\j$};}
\end{tikzpicture}
```



`\i` está associado à posição enquanto `\j` ao texto

Estruturas de repetição

Desenhando uma régua...

```
\begin{tikzpicture}
\draw[fill=gray!10!] (-0.2, 0) rectangle (10.2,1);
\end{tikzpicture}
```



Estruturas de repetição

Desenhandando uma régua...

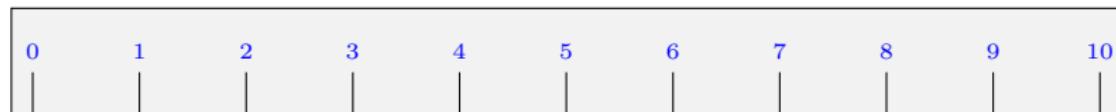
```
\begin{tikzpicture}
\draw[fill=gray!10!] (-0.2, 0) rectangle (10.2,1);
\foreach \x in {0,...,10} {
    \draw (\x,0) -- (\x,0.4);
}
\end{tikzpicture}
```



Estruturas de repetição

Desenhando uma régua...

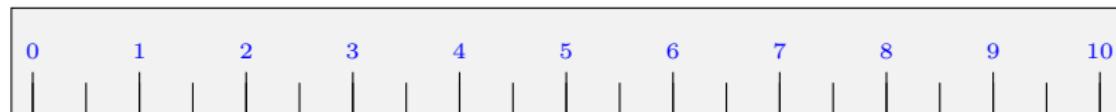
```
\begin{tikzpicture}
\draw[fill=gray!10!] (-0.2, 0) rectangle (10.2,1);
\foreach \x in {0,...,10}{
    \draw (\x,0) -- (\x,0.4);
    \draw (\x,0.6) node{\tiny\color{blue}\x};
}
\end{tikzpicture}
```



Estruturas de repetição

Desenhandando uma régua...

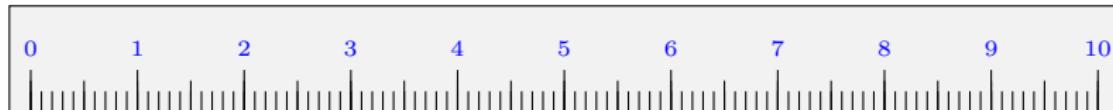
```
\begin{tikzpicture}
\draw[fill=gray!10!] (-0.2, 0) rectangle (10.2,1);
\foreach \x in {0,...,10}{
    \draw (\x,0) -- (\x,0.4);
    \draw (\x,0.6) node{\tiny\color{blue}\x};
}
\foreach \y in {0,...,20} {
    \draw (0.5*\y,0) -- (0.5*\y,0.3);
}
\end{tikzpicture}
```



Estruturas de repetição

Desenhando uma régua...

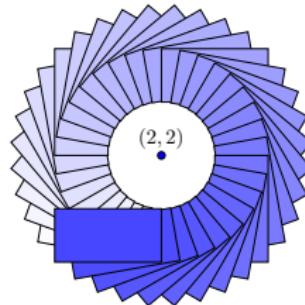
```
\begin{tikzpicture}
\draw[fill=gray!10!] (-0.2, 0) rectangle (10.2,1);
\foreach \x in {0,...,10} {
    \draw (\x,0) -- (\x,0.4);
    \draw (\x,0.6) node{\tiny\color{blue}\x};
}
\foreach \y in {0,...,20} {
    \draw (0.5*\y,0) -- (0.5*\y,0.3);
}
\foreach \z in {0,...,100} {
    \draw (0.1*\z,0) -- (0.1*\z,0.2);
}
\end{tikzpicture}
```



Estruturas de repetição

Rotacionando retângulos...note que cálculos com a variável x podem ser utilizados dentro do comando `foreach` como a seguir.

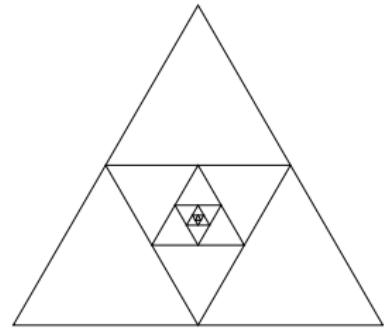
```
\begin{tikzpicture}
\foreach \x [evaluate=\x as \y using \x/5] in {10,20,...,360}{
\draw[fill=blue!\y,rotate around={-\x:(2,2)}] (0,0) rectangle (2,1);
}
\end{tikzpicture}
```



Estruturas de repetição

Veja a figura abaixo que faz uma série de triângulos equiláteros:

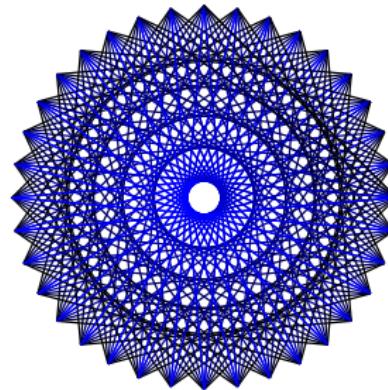
```
\begin{tikzpicture}[scale=1]
\foreach \i in {0,1,...,6}{
\begin{scope}[rotate=60*\i]
\foreach \j in {90,210,330}{
\coordinate (A\j) at (\j:1/2^\i);
}
\draw[ultra thin] (A90) -- (A210)
                  -- (A330) -- cycle;
\end{scope}
}
\end{tikzpicture}
```



Estruturas de repetição

Comandos simples produzem estruturas maravilhosas!

```
\begin{tikzpicture}
\foreach \t in {0,10,\dots,360}{
    \foreach \x in {0,20,40,60,80,100}{
        \draw[blue!\x!black,thick] (\t:3) -- (\t+\x+90:3);
    }
}
\end{tikzpicture}
```



Estruturas de repetição

Suponha que queiramos avaliar a função $f(x) = \frac{x^2-1}{x^2+1}$ para vários valores de x , digamos $x \in \{1, \dots, 100\}$ com 4 casas decimais exatas.

- Primeiro, criamos um arquivo `calculos.tex` com as seguintes instruções:

```
\documentclass [border=5mm,tikz]{standalone}
\usepackage{tkz-euclide}
\usepackage{amsmath}
\usepackage{xfp}
\begin{document}
\foreach \x in {1,2,\dots,100}{
\begin{tikzpicture}[scale=1]
\clip (-3.5,-2) rectangle (3,2);
\node (Texto) at (0,0) {0 valor da função
$f(x)=\frac{x^2-1}{x^2+1}$ em $x= \x$};
\node (Texto) [below= 0.3cm of Texto] {é igual a
\boxed{\fpeval{round((\x^2-1)/(\x^2+1),3)}}};
\end{tikzpicture}
}
```

e o salvamos na mesma pasta que estamos trabalhando.

Estruturas de repetição

- Em seguida, salvamos um novo arquivo com o nome `animacao.tex` no mesmo diretório e usamos a instrução:

```
\begin{figure}[H]
\animategraphics
[controls={step,stop,speed=0.5},buttonsize=5mm,scale=0.85,autoplay,pal
{3}]{integral}{% nome do ficheiro que tem as imagens
{}%primeiro frame
{}%último frame
\end{figure}
```

carregando previamente o pacote `animate`. Note que L^AT_EX faz cálculos matemáticos implicitamente!

Estruturas de repetição

O resultado é animação a seguir:

Estruturas de repetição

Um relógio em funcionamento:

Estruturas de repetição

Parte analógica:

```
\foreach \h in {0,1,...,1800}{
\begin{tikzpicture}
\draw[top color=blue!20,opacity=0.5] (0,0) circle(2);
\foreach \i in {1,2,...,12}{
\node (A\i) at (-270-\i*30:1.6) {\i};
}
\foreach \j in {1,2,...,60}{
\draw (6*\j:1.9) -- (6*\j:2);
}
\foreach \j in {1,...,12}{
\draw[thick] (30*\j:1.8) -- (30*\j:2);
}
\draw[->,-latex',thick] (0,0) -- (90-6*\h:1.4);
\draw[->,-latex',blue,thick] (0,0) -- (90-0.1*\h:1.2);
\draw[->,-latex',red,thick] (0,0) -- (60- 0.00833333*\h:1.0);
\draw[fill=black] (0,0) circle (1pt);
(...)
```

\end{tikzpicture}}

Estruturas de repetição

Parte digital com a estrutura \if-else-\fi:

```
\foreach \h in {0,1,...,1800} {
(...)
\ifnum\h>599
\node (H) at (0,-2.5) [below] {\text{0 relógio marca}
\fpeval{13 + floor(\h/3600,0)}h:\fpeval{floor(\h/60,0)}
minutos e \fpeval{\h - 60*floor(\h/60)} segundos};
\else
\node (H) at (0,-2.5) [below] {\text{0 relógio marca}
\fpeval{13 + floor(\h/3600,0)}h:0\fpeval{floor(\h/60,0)}
minutos e \fpeval{\h - 60*floor(\h/60)} segundos};
\fi
\end{tikzpicture}}
```

Estruturas de repetição

- Em seguida, salvamos um novo arquivo com o nome `relogio.tex` no mesmo diretório e usamos a instrução:

```
\begin{figure}[H]
\animategraphics
[controls={step,stop,play},buttonsize=5mm,scale=0.85,
autoplay]
{1} %# frames/segundo
{relogio} %nome do ficheiro que tem as imagens
{} %primeiro frame
{} %último frame
\end{figure}
```

carregando previamente o pacote `animate`. Note que L^AT_EX faz cálculos matemáticos implicitamente!

Estruturas de repetição

Um relógio em funcionamento (horas e minutos), feito de forma análoga:

Sumário

- 1 Introdução
- 2 Comandos iniciais
- 3 Customização dos objetos
- 4 Nós em tikz
- 5 Estruturas de Repetição
- 6 Esboço de curvas

Esboço de curvas

Comando principal

Para esboçar uma curva parametrizada por $\mathbf{r}(x) = (f(x), g(x))$ com $a \leq x \leq b$ basta utilizar o comando

```
plot({f(\x)}, {g(\x)})
```

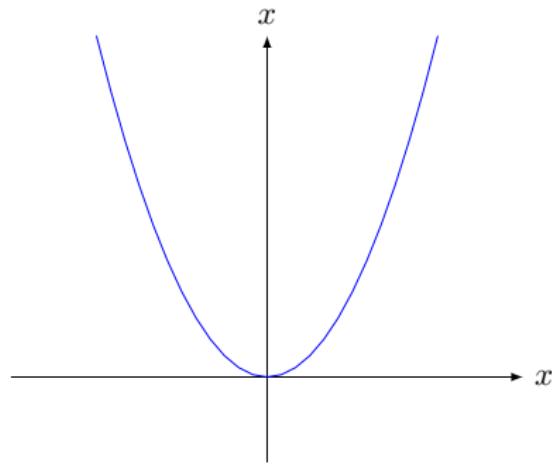
onde **x** é a variável e **f** e **g** são as componentes.

Obs.: no caso de funções trigonométricas, como **sin**, devemos informar que o argumento é em radianos, usando o comando **sin(\x r)**

Esboço de curvas

Exemplos:

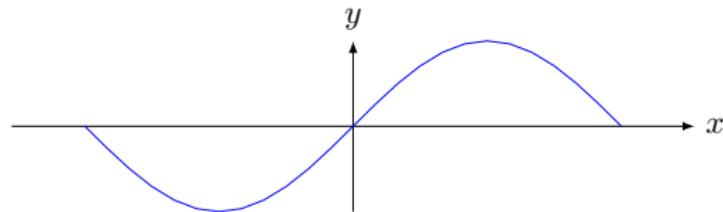
```
\begin{tikzpicture}
\draw[->, -latex] (-3,0)--(3,0) node[right]{$x$};
\draw[->, -latex] (0,-1)--(0,4) node[above]{$x$};
\draw[blue, domain=-2:2] plot({\x}, {(\x)^2});
\end{tikzpicture}
```



Esboço de curvas

Exemplos:

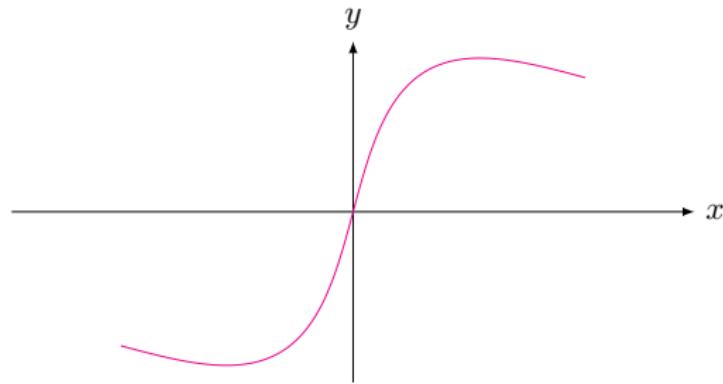
```
\begin{tikzpicture}
\draw[->,-latex] (-4,0)--(4,0) node[right]{$x$};
\draw[->,-latex] (0,-1)--(0,1) node[above]{$y$};
\draw[blue,domain=-pi:pi,rotate=0] plot({\x},{sin(\x r)});
\end{tikzpicture}
```



Esboço de curvas

Exemplos: aumentando o número de amostras com `samples`, suavizando a curva com `smooth` e rotacionando-a

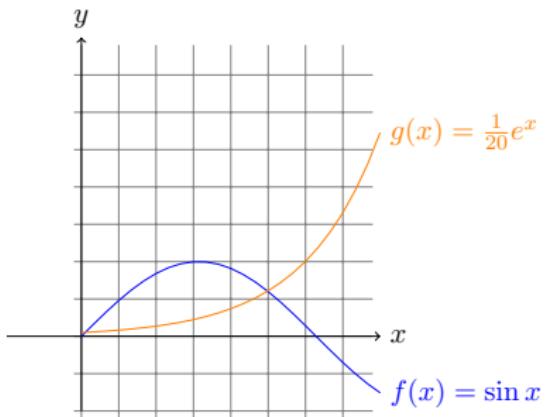
```
\begin{tikzpicture}
\draw[->, -latex] (-4,0)--(4,0) node[right]{$x$};
\draw[->, -latex] (0,-2)--(0,2) node[above]{$y$};
\draw[magenta, domain=-pi:pi, rotate=30, samples=500, smooth]
plot({\x}, {\sin(\x r)} );
\end{tikzpicture}
```



Esboço de curvas

Exemplos: inserindo grade e labels

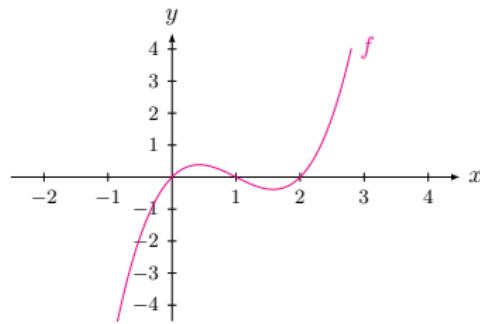
```
\begin{tikzpicture}[domain=0:4]
\draw[help lines] (-0.1,-1.1) grid (3.9,3.9);
\draw[->] (-1,0) -- (4,0) node[right] {$x$};
\draw[->] (0,-1) -- (0, 4) node[above] {$y$};
\draw[blue] plot (\x,{sin(\x r)})
node[right] {$f(x) = \sin x$};
\draw[orange] plot (\x,{0.05*exp(\x)})
node[right] {$g(x) = \frac{1}{20}e^x$};
\end{tikzpicture}
```



Esboço de curvas

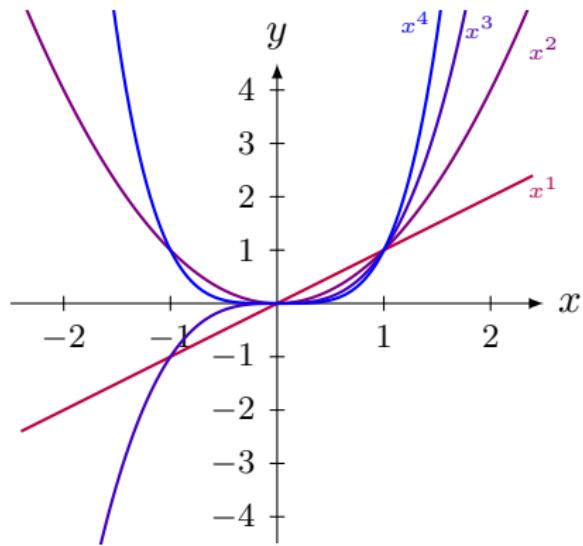
Exemplos:

```
\begin{tikzpicture}[yscale=0.5]
\clip (-2.5,-4.5) rectangle (5,5.5);
\draw[->,-latex] (-4,0)--(4.5,0) node[right]{$x$};
\draw[->,-latex] (0,-4.5)--(0,4.5) node[above]{$y$};
\foreach \i in {-4,-3,-2,-1,1,2,3,4}{
\draw (\i,4pt)--(\i,-4pt) node[below]{{\footnotesize $\i$}};
}
\foreach \i in {-4,-3,-2,-1,1,2,3,4}{
\draw (2pt,\i)--(-2pt,\i) node[left]{{\footnotesize $\i$}};
}
\draw[magenta,domain=-4:2.8,samples=500,smooth]
plot({\x},{(\x)^3 - 3*(\x)^2 +2*\x}) node[right]{$f$};
\end{tikzpicture}
```



Esboço de curvas

Exemplo: plot com \foreach. Para obter o desenho abaixo....



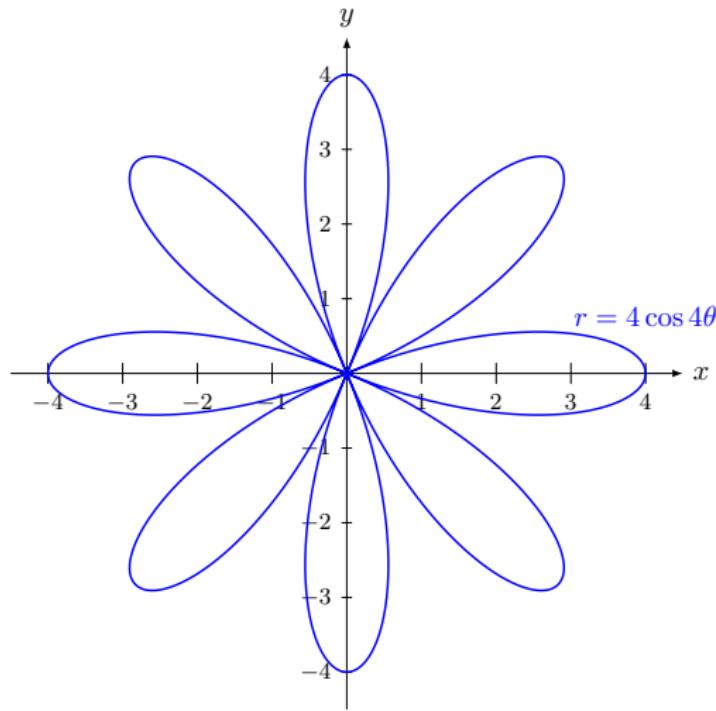
Esboço de curvas

Usamos o seguinte código (completo):

```
\begin{tikzpicture}[yscale=0.5]
\clip (-2.5,-4.5) rectangle (3,5.5);
\draw[->, -latex] (-2.5,0)--(2.5,0) node[right]{$x$};
\draw[->, -latex] (0,-4.5)--(0,4.5) node[above]{$y$};
\foreach \i in {-4,-3,-2,-1,1,2}{
\draw (\i,4pt)--(\i,-4pt) node[below]{{\tiny \footnotesize $\i$}}; }
\foreach \i in {-4,-3,-2,-1,1,2,3,4} {
\draw (2pt,\i)--(-2pt,\i) node[left]{{\tiny \footnotesize $\i$}}; }
\foreach \n\x\y [evaluate=\n as \m using 25*\n] in
{1/2.5/1.7,2/2.5/4.3,3/1.9/4.7,4/1.3/4.8} {
\draw[blue!\m!red, domain=-2.4:2.4, samples=500, smooth, thick] plot({\x}, {(\x)^{\n}});
\draw[blue!\m!red] (\x,\y) node[above]{{\tiny \footnotesize $x^{\n}$}}; }
\end{tikzpicture}
```

Esboço de curvas

Exemplo: podemos também traçar curvas em coordenadas polares usando a opção `xy polar cs:angle=\x r, radius={equação polar}`



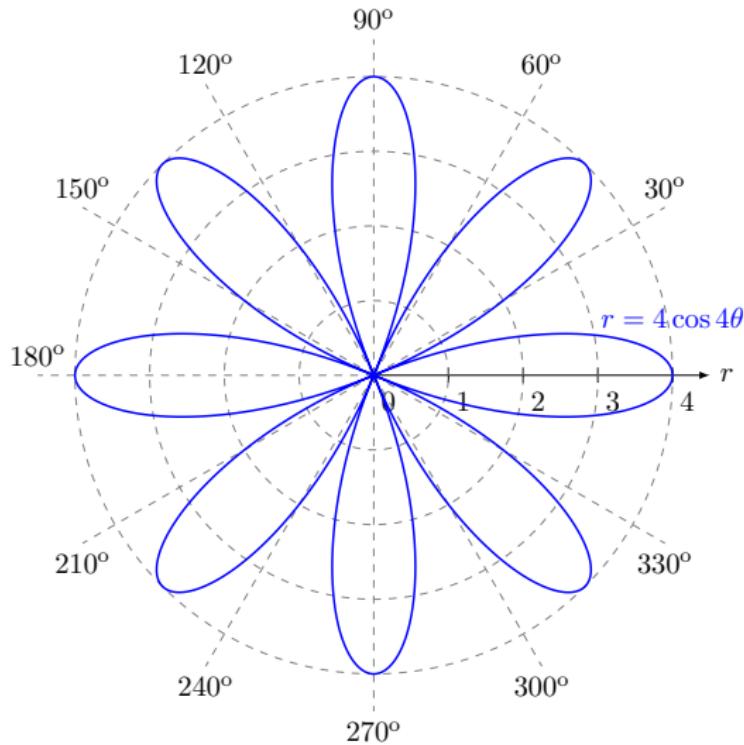
Esboço de curvas

Feito com:

```
\begin{tikzpicture}
\clip (-5,-5) rectangle (5,5);
\draw[->,-latex] (-4.5,0)--(4.5,0) node[right]{$x$};
\draw[->,-latex] (0,-4.5)--(0,4.5) node[above]{$y$};
\foreach \i in {-4,-3,-2,-1,1,2,3,4}{
\draw (\i,4pt)--(\i,-4pt) node[below]{{\footnotesize $\i$}}; }
\foreach \i in {-4,-3,-2,-1,1,2,3,4}{
\draw (2pt,\i)--(-2pt,\i) node[left]{{\footnotesize $\i$}}; }
\draw[blue,domain=0:2*pi,samples=500,smooth,thick]
plot (xy polar cs:angle=\x r, radius={4*cos(4*(\x r))})
node[above,yshift=0.5cm] {$r=4 \cos 4 \theta$};
\end{tikzpicture}
```

Esboço de curvas

Exemplo: claro que podemos mudar o sistema de coordenadas para polar. Fizemos ua construção manualmente:



Esboço de curvas

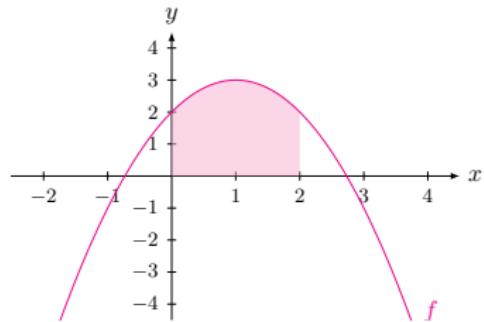
Feito com:

```
\begin{tikzpicture}
\clip (-5,-5) rectangle (5,5);
\node (origem) at (0,0) {};\filldraw(origem) circle (1pt);
\draw[->, -latex] (0,0) -- (4.5,0) node[right]{$r$};
\foreach \i in {0,1,2,3,4}{
\draw (\i,-0.1) node[below,xshift=0.2cm]{$\i$} -- (\i,0.1);
\draw[gray,dashed] (0,0) circle (\i);
}
\foreach \t in {30,60,\dots,180}{
\draw[rotate=\t,gray,dashed] (0,0)
-- (4.5,0) node[above,sloped,black] {$\t^\circ$};
}
\foreach \t in {210,240,\dots,330}{
\draw[rotate=\t,gray,dashed] (0,0)
-- (4.5,0) node[below,sloped,black] {$\t^\circ$};
}
\draw[blue,domain=0:2*pi,samples=500,smooth,thick]
plot (xy polar cs:angle=\x r, radius={4*cos(4*(\x r))})
node[above,yshift=0.5cm] {$r=4 \cos 4 \theta$};
\end{tikzpicture}
```

Esboço de curvas

Podemos hachurar regiões no plano. Veja o exemplo:

```
\begin{tikzpicture}[yscale=0.5]
\fill[magenta!40!,opacity=0.5] (0,0) --
plot [domain=0:2] (\x, {-(\x)^2+2*\x+2})
-- (2,0) -- cycle;
\draw[magenta,domain=-3.8:3.8,smooth]
plot(\x),{-(\x)^2+2*\x+2}
node[above right]{$f$};
\end{tikzpicture}
```

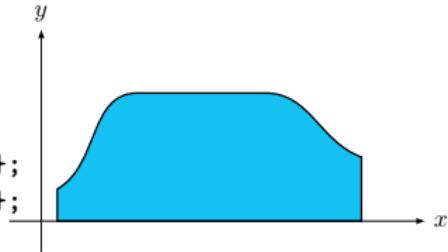


Não exibimos o código para desenhar os eixos.

Esboço de curvas

Podemos hachurar curvas genéricas também. Veja o código abaixo:

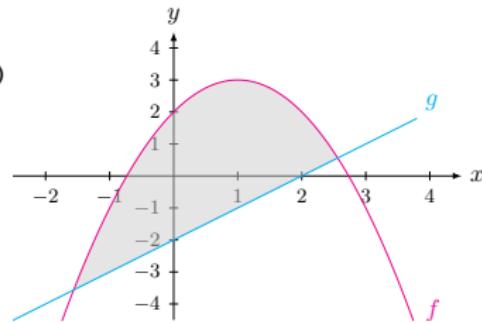
```
\begin{tikzpicture}
\draw[thick, fill=cyan!70] (0.25,0)--(0.25,0.5)
to [out=30,in=180] (1.5,2)to[out=0,in=180]
(3.5,2) to [out=0,in=160](5,1)--(5,0)--cycle;
\draw[->,-latex] (-0.5,0)--(6,0)node[right]{$x$};
\draw[->,-latex] (0,-0.5)--(0,3)node[above]{$y$};
\end{tikzpicture}
```



Esboço de curvas

Podemos hachurar regiões entre duas curvas. Veja o exemplo:

```
\begin{tikzpicture}
\fill[gray!40!,opacity=0.5] (-1.56,-3.56) --
plot [domain=-1.56:2.56] (\x, {-(\x)^2+2*\x+2})
-- (2.56,0.56) -- cycle;
\draw[magenta,domain=-3.8:3.8,smooth]
plot({\x},{-(\x)^2+2*\x+2})
node[above right]{$f$};
\draw[cyan,domain=-3.8:3.8,smooth]
plot({\x},{\x-2}) node[above right]{$g$};
\end{tikzpicture}
```

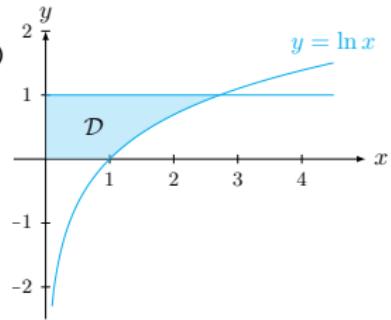


Basta conhecermos as coordenadas dos pontos de intersecção.

Esboço de curvas

Mais um exemplo:

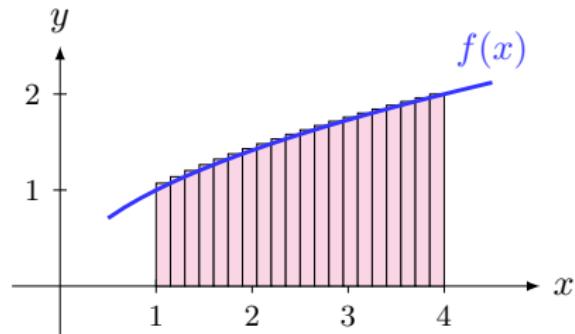
```
\fill[cyan!40!,opacity=0.5](0,0)--(0,1)--(2.718,1)
--plot [domain=2.718:1] (\x,{ln(\x)})-- cycle;
\draw[cyan] (0,1)--(4.5,1);
\draw[cyan, domain=0.1:4.5, samples=500, smooth]
plot({\x},{ln(\x)}) node[above]{$y=\ln x$};
\node at (.75,.5) {$\mathcal{D}$};
\end{tikzpicture}
```



Basta conhecermos as coordenadas dos pontos de intersecção.

Esboço de curvas

Mais um exemplo: somas de Riemann. Definimos previamente a função $f(x)$, os extremos de a e b e o número de retângulos n .



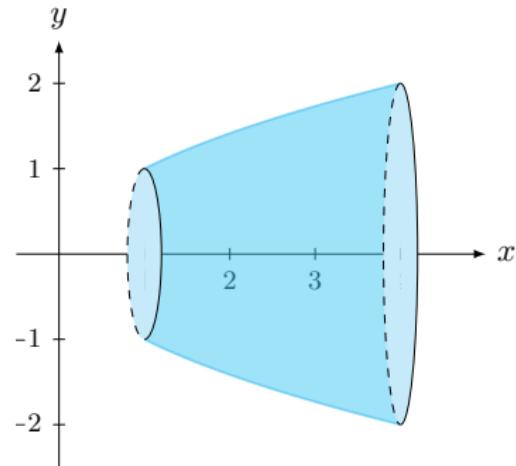
Esboço de curvas

Foi conseguido com:

```
\begin{tikzpicture}[declare function = {f(\x) = \x^0.5;}]
\def\aa{1},\def\bb{4},\def\nn{20}
\clip (-.5,-2.5) rectangle (5.5,3);
\draw[->, -latex] (-.5,0)--(\bb+1,0) node[right]{$x$};
\draw[->, -latex] (0,-.5)--(0,2.5) node[above]{$y$};
\foreach \i in {1,2,3,4}{
\draw (\i,2pt)--(\i,-2pt) node[below]{{\footnotesize $\i$}}; }
\foreach \i in {1,2}{
\draw (2pt,\i)--(-2pt,\i) node[left]{{\footnotesize $\i$}}; }
\foreach \i in {1,2,...,\nn}{
\draw[very thin,fill=magenta!20] (\aa+(\i-1)*((\bb-\aa))/\nn,0)
rectangle (\aa+\i*((\bb-\aa))/\nn,{f(\aa+\i*((\bb-\aa))/\nn)}); }
\draw[blue!80,very thick] plot [domain=\aa-0.5:\bb+0.5]
(\x,{f(\x)}) node[above]{$f(x)$}; }
\end{tikzpicture}
```

Esboço de curvas

Uma tentativa de esboçar objetos tridimensionais...



Esboço de curvas

Com o seguinte código:

```
\begin{tikzpicture}[declare function = {f(\x) = \x^0.5;}]
\def\aa{1},\def\bb{4}
\clip (-.5,-2.5) rectangle (5.5,3);
\draw[cyan!40,thick] plot [domain=\aa:\bb] (\x,{f(\x)});
\draw[cyan!40,yscale=-1,thick] plot [domain=\aa:\bb] (\x,{f(\x)});
\fill[cyan!60!,opacity=0.5] (\aa,0)--(\aa,{f(\aa)})--plot
[domain=\aa:\bb] (\x,{f(\x)})--(\bb,0)--cycle;
\fill[cyan!60!,opacity=0.5,yscale=-1] (\aa,0)--(\aa,{f(\aa)})
--plot [domain=\aa:\bb] (\x,{f(\x)})--(\bb,0)--cycle;
\draw[fill=cyan!20,dashed] (\aa,{f(\aa)}) arc (90:270:0.2 and {f(\aa)});
\draw[fill=cyan!20] (\aa,{f(\aa)}) arc (90:-90:0.2 and {f(\aa)});
\draw[fill=cyan!20,dashed] (\bb,{f(\bb)}) arc (90:270:0.2 and {f(\bb)});
\draw[fill=cyan!20] (\bb,{f(\bb)}) arc (90:-90:0.2 and {f(\bb)});
\end{tikzpicture}
```

Note que fizemos a definição da função usando o comando
declare function = {}

Esboço de curvas

Área abaixo de uma curva animada.

Esboço de curvas

```
\foreach \b in {0,0.1,...,4.1}{  
    \begin{tikzpicture}[font=\scriptsize,scale=1,  
        declare function={f(\x)=\x^2/4;}]  
        \clip (-1.5,-1.5) rectangle (5,5);  
        \draw[->,-latex'] (-1,0) -- (4.5,0) node[right] {$x$};  
        \draw[->,-latex'] (0,-1) -- (0,4.5) node[above] {$y$};  
        \foreach \i in {-1,...,4}{  
            \draw (\i,-2pt) -- (\i,2pt) node[below=2mm,fill=white] {$\i$};}  
        \foreach \j in {-1,...,4}{  
            \draw (-2pt,\j) -- (2pt,\j) node[left=2mm,fill=white] {$\j$};}  
        \draw[magenta,thick,domain=0:\b,samples=100,smooth]  
            plot({\x},{f(\x)}) node[right]{$f$};  
        \fill[magenta,opacity=0.5,domain=0:\b,samples=100,smooth] (0,0) --  
            plot({\x},{f(\x)}) -- (\b,0) -- cycle;  
        \node[anchor=west] (Texto) at (0.5,4.5) {0 valor da área é:};  
        \node[anchor=west] (Texto1) at (0.5,4)  
            {$\displaystyle A=\int_0^{\b} f(x)\, dx =$}  
            \fpeval{round(\b^3/12,5)}$};  
    \end{tikzpicture}  
}
```

Esboço de curvas

O código anterior é salvo como `Integral_animada.tex` e depois vem as instruções para a animação:

```
\begin{figure}[H]
\animategraphics
[controls={step,stop,speed=0.5},buttonsize=5mm,scale=0.85,autoplay]
{3}%frame rate
{Integral_animada}%nome do ficheiro que tem as imagens
{}%primeiro frame
{}%último frame
\end{figure}
```

Referências I

-  J. Cr emer, “A very minimal introduction to tikz,” *Tantau et al.: TikZ & PGF Manual, both available at <https://www.ctan.org/pkg/pgf>, 2011.*
-  T. Tantau, “Graph drawing in tikz,” in *International Symposium on Graph Drawing*, pp. 517–528, Springer, 2012.