

CAPTURA E VISUALIZAÇÃO DE DADOS

Diénert Vieira

dienertalencar@gmail.com


(83) 9 8182-1478



VISUALIZAÇÃO DE DADOS

TÓPICOS

► Visualização

- Exemplos na Plataforma observablehq.com
 - Javascript e bibliotecas
 - Como transformar um dado para utilizar uma visualização pronta
 - Fazer as adaptações desejadas na visualização
 - Exemplo de Extensão do Qlikview
 - D3.js
 - SVG
 - Exemplos React.js
 - Trabalho de Visualização
- 

FERRAMENTAS

► Soluções Gratuitas e Abertas:



Google
Developers
CHARTS

matplotlib



► Soluções Pagas:



tableau
SOFTWARE

Desktop (publicação web) (Avaliação 14 dias)



QlikView

Desktop (4 acessos
p/ arquivos de 3^{os})

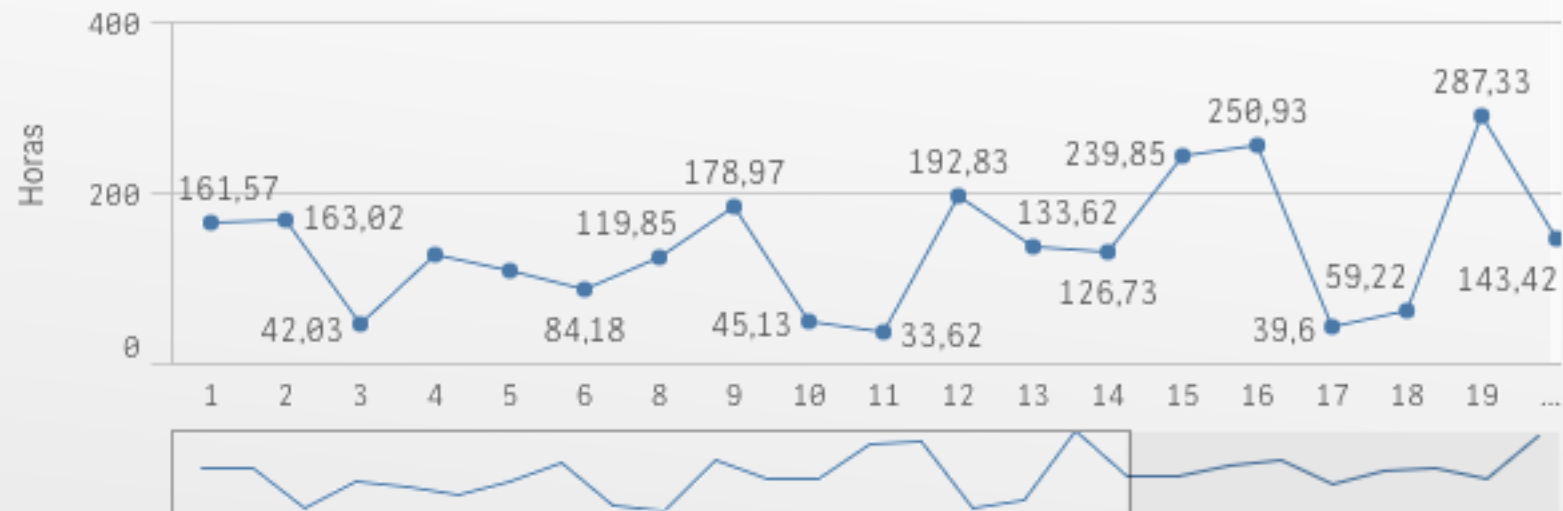


Qlik Sense
Visualização na Cloud
Para até 5 pessoas

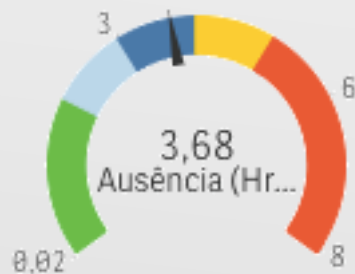
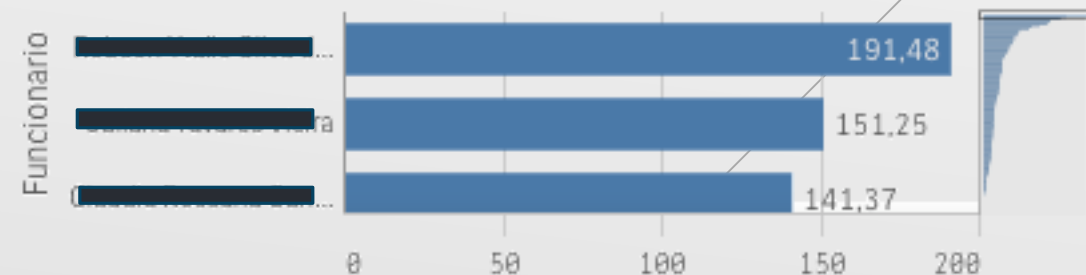
QLIKSENSE – CAPTURA DO SISTEMA DE PONTO



Ausência em Horas por Dia



Ausência em Horas por Funcionário



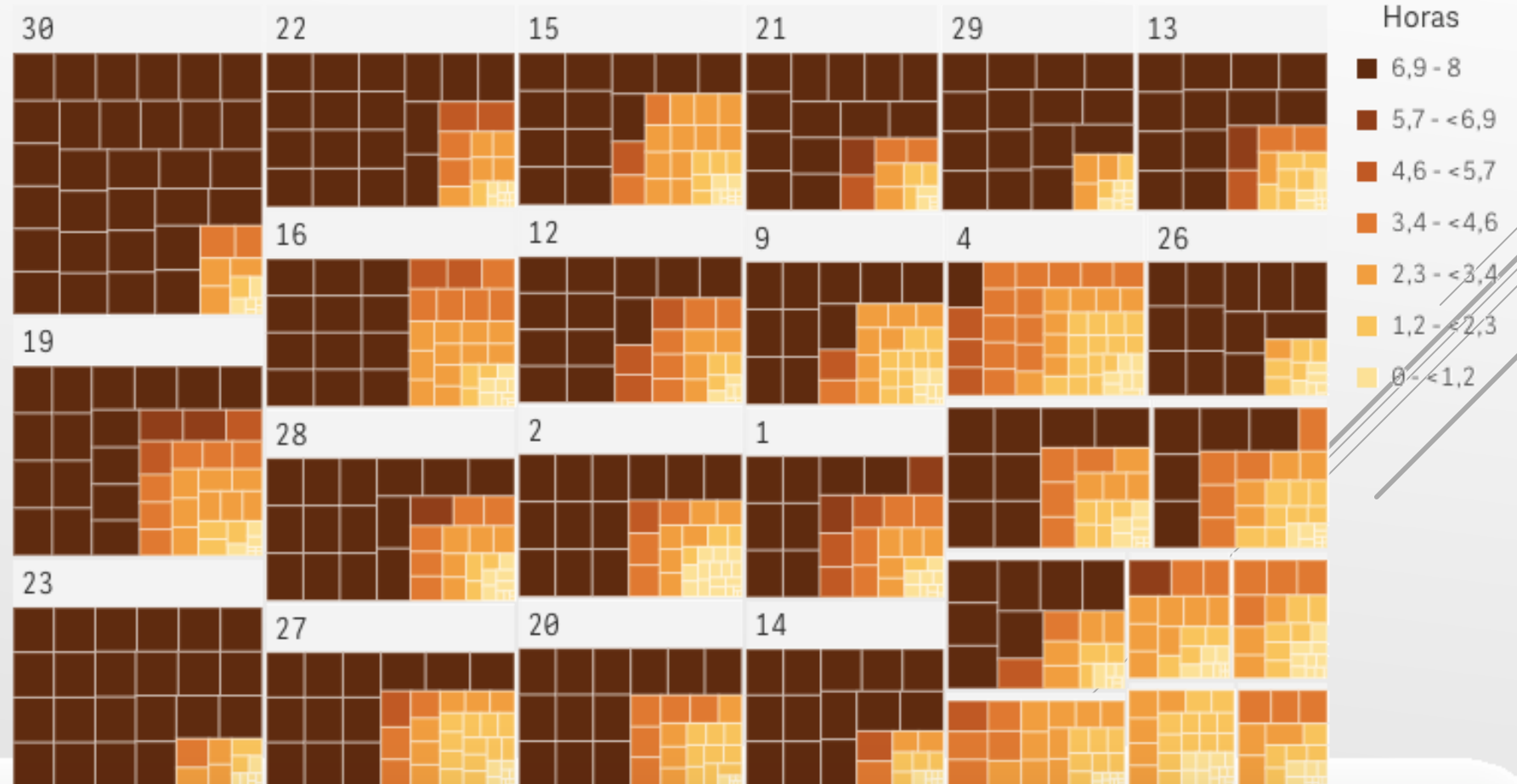
QLIKSENSE - CAPTURA DO SISTEMA DE PONTO

- ▶ Cada retângulo é um funcionário
- ▶ Então o dia que as pessoas estiveram mais ausentes foi o dia 30



Ausências

Horas por Dia



QLIKSENSE

► Vantagens:

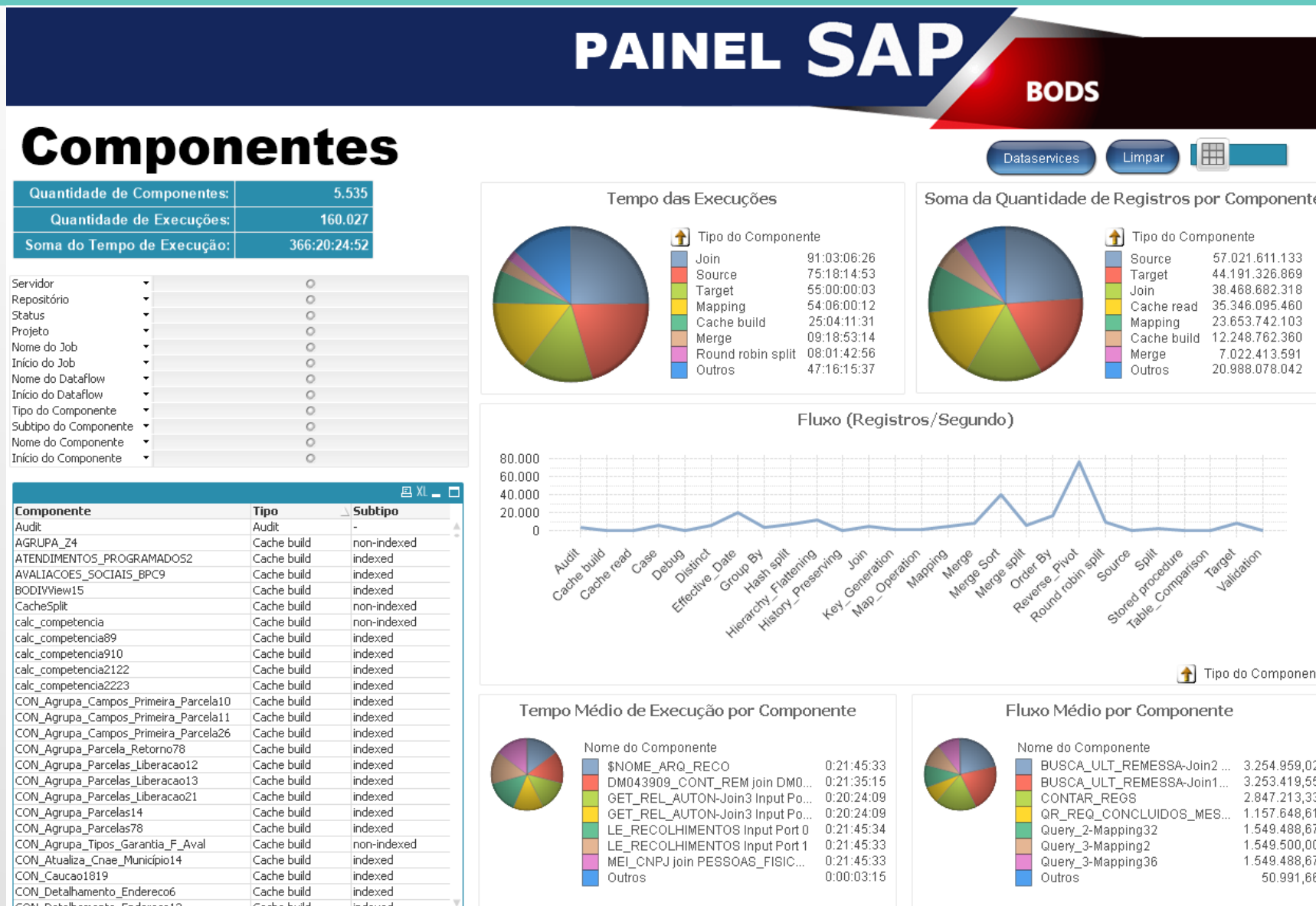
- Modelo Associativo
- Rápida curva de aprendizado
- Facilidade de utilização provê grande produtividade
- Responsivo
- Dinamicidade na construção das interfaces
- Várias opções de gráficos

► Desvantagens

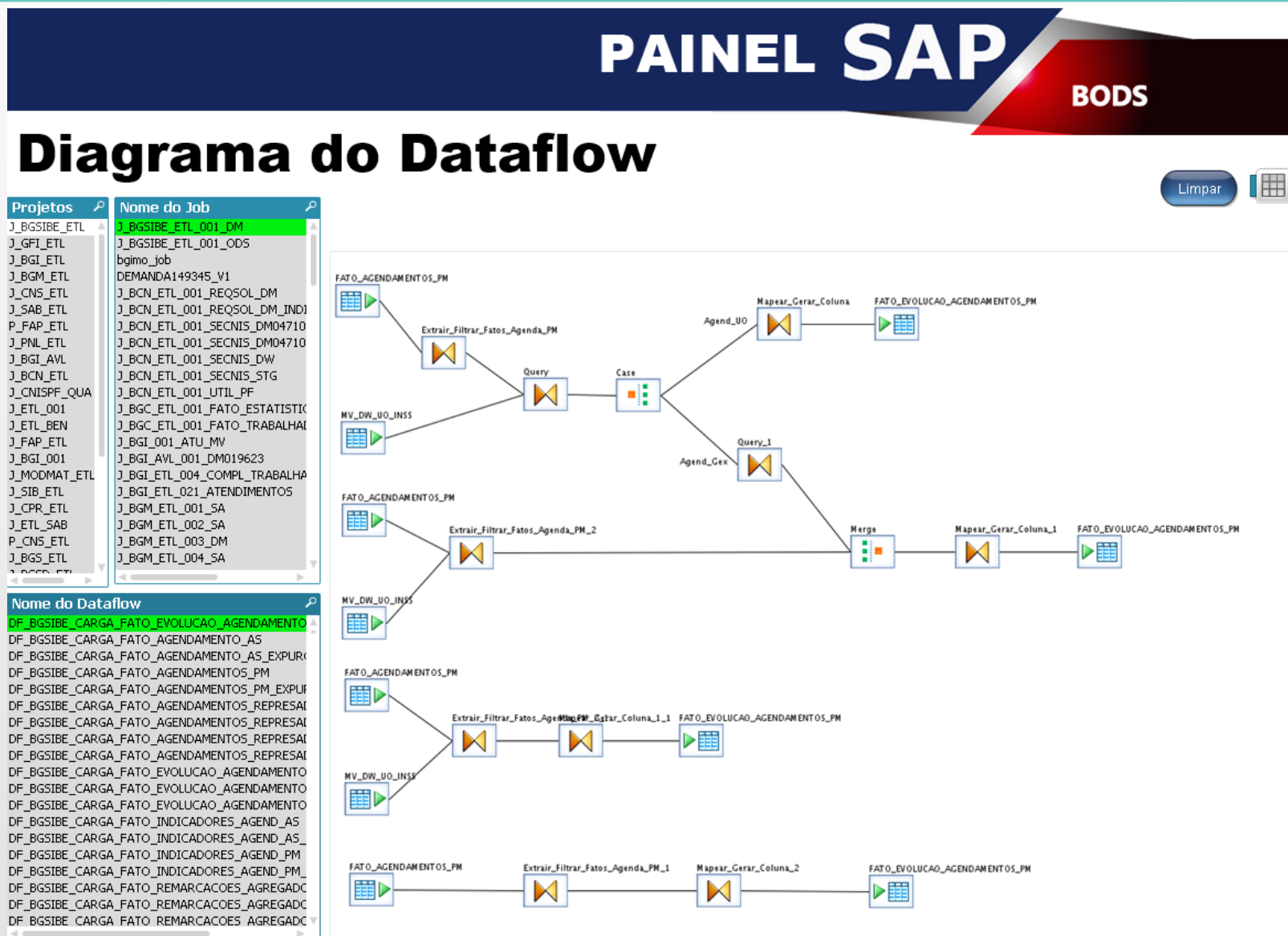
- É pago
- De graça, pode publicar na <https://qlikcloud.com/> para até 5 usuários acessarem
- Uso avançado depende do uso de Set Analysis, que possui uma curva de aprendizado mais longa
- Auxílio para construção de Set Analysis:

<http://tools.qlikblog.at/SetAnalysisWizard/QlikView-SetAnalysis Wizard and Generator.aspx?sa=>

QLIKVIEW - PAINEL DO SISTEMA DE ETL



QLIKVIEW - PAINEL DO SISTEMA DE ETL



QLIKVIEW - PAINEL DO SISTEMA DE ETLs



PAINEL SAP BODS

Erros dos Jobs

Erros:	1.206.727
Tipos de Erros:	17

Servidor	▼	○
Repositório	▼	○
Status	▼	○
Projeto	▼	○
Nome do Job	▼	○
Tempo de Início do Job	▼	○
Nome do Dataflow	▼	○
Tempo de Início do Dataflow	▼	○
Momento do Log	▼	○
Onde ocorreu o erro	▼	○

Limpar

Log Completo:



SQL:



Tipos de Erro

OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DSOL_FK) violated - parent key not found	32
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DUSU_FK) violated - parent key not found	1.616
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	1
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FRR_DSOL_FK) violated - parent key not found	1.230
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FSR_DPEFI_FK) violated - parent key not found	583
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FSR_DUSU_FK) violated - parent key not found	333
OCISmtExecute>: <ORA-02291: restrição de integridade (DMCNIS.FER_DSOL_FK) violada - chave mãe não localizada	12
ORA-00060: deadlock detected while waiting for resource	1.508
ORA-02291: integrity constraint (DMCNIS.FER_DSOL_FK) violated - parent key not found	295
ORA-02291: integrity constraint (DMCNIS.FER_DUSU_FK) violated - parent key not found	477.151
ORA-02291: integrity constraint (DMCNIS.FER_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	2
ORA-02291: integrity constraint (DMCNIS.FRR_DSOL_FK) violated - parent key not found	1.782
ORA-02291: integrity constraint (DMCNIS.FRR_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	45.864
ORA-02291: integrity constraint (DMCNIS.FSR_DPEFI_FK) violated - parent key not found	57
ORA-02291: integrity constraint (DMCNIS.FSR_DUSU_FK) violated - parent key not found	3.570
ORA-02291: restrição de integridade (DMCNIS.FER_DSOL_FK) violada - chave mãe não localizada	54
	672.637

LOGSTASH

QLIKVIEW - PAINEL DO SISTEMA DE ETLs



PAINEL SAP BODS

Erros dos Jobs

Erros:	1.206.727
Tipos de Erros:	17

Servidor	
Repositório	
Status	
Projeto	
Nome do Job	
Tempo de Início do Job	
Nome do Dataflow	
Tempo de Início do Dataflow	
Momento do Log	
Onde ocorreu o erro	

Limpar

Log Completo:

SQL:



Tipos de Erro

OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DSOL_FK) violated - parent key not found	32
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DUSU_FK) violated - parent key not found	1.616
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FER_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	1
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FRR_DSOL_FK) violated - parent key not found	1.230
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FSR_DPEFI_FK) violated - parent key not found	583
OCISmtExecute>: <ORA-02291: integrity constraint (DMCNIS.FSR_DUSU_FK) violated - parent key not found	333
OCISmtExecute>: <ORA-02291: restrição de integridade (DMCNIS.FER_DSOL_FK) violada - chave mãe não localizada	12
ORA-00060: deadlock detected while waiting for resource	1.508
ORA-02291: integrity constraint (DMCNIS.FER_DSOL_FK) violated - parent key not found	295
ORA-02291: integrity constraint (DMCNIS.FER_DUSU_FK) violated - parent key not found	477.151
ORA-02291: integrity constraint (DMCNIS.FER_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	2
ORA-02291: integrity constraint (DMCNIS.FRR_DSOL_FK) violated - parent key not found	1.782
ORA-02291: integrity constraint (DMCNIS.FRR_DUSU_ULTIMA_SITUACAO_FK) violated - parent key not found	45.864
ORA-02291: integrity constraint (DMCNIS.FSR_DPEFI_FK) violated - parent key not found	57
ORA-02291: integrity constraint (DMCNIS.FSR_DUSU_FK) violated - parent key not found	3.570
ORA-02291: restrição de integridade (DMCNIS.FER_DSOL_FK) violada - chave mãe não localizada	54
	672.637

LOGSTASH

QLIKVIEW

► Vantagens:

- Modelo Associativo
- Rápida curva de aprendizado

► Desvantagens

- É pago
- Não é responsivo
- Não é tão produtivo quanto o QlikSense
- Uso avançado depende do uso de Set Analysis, que possui uma curva de aprendizado mais longa
- Auxílio para construção de Set Analysis:

http://tools.qlikblog.at/SetAnalysisWizard/QlikView-SetAnalysis_Wizard_and_Generator.aspx?sa=

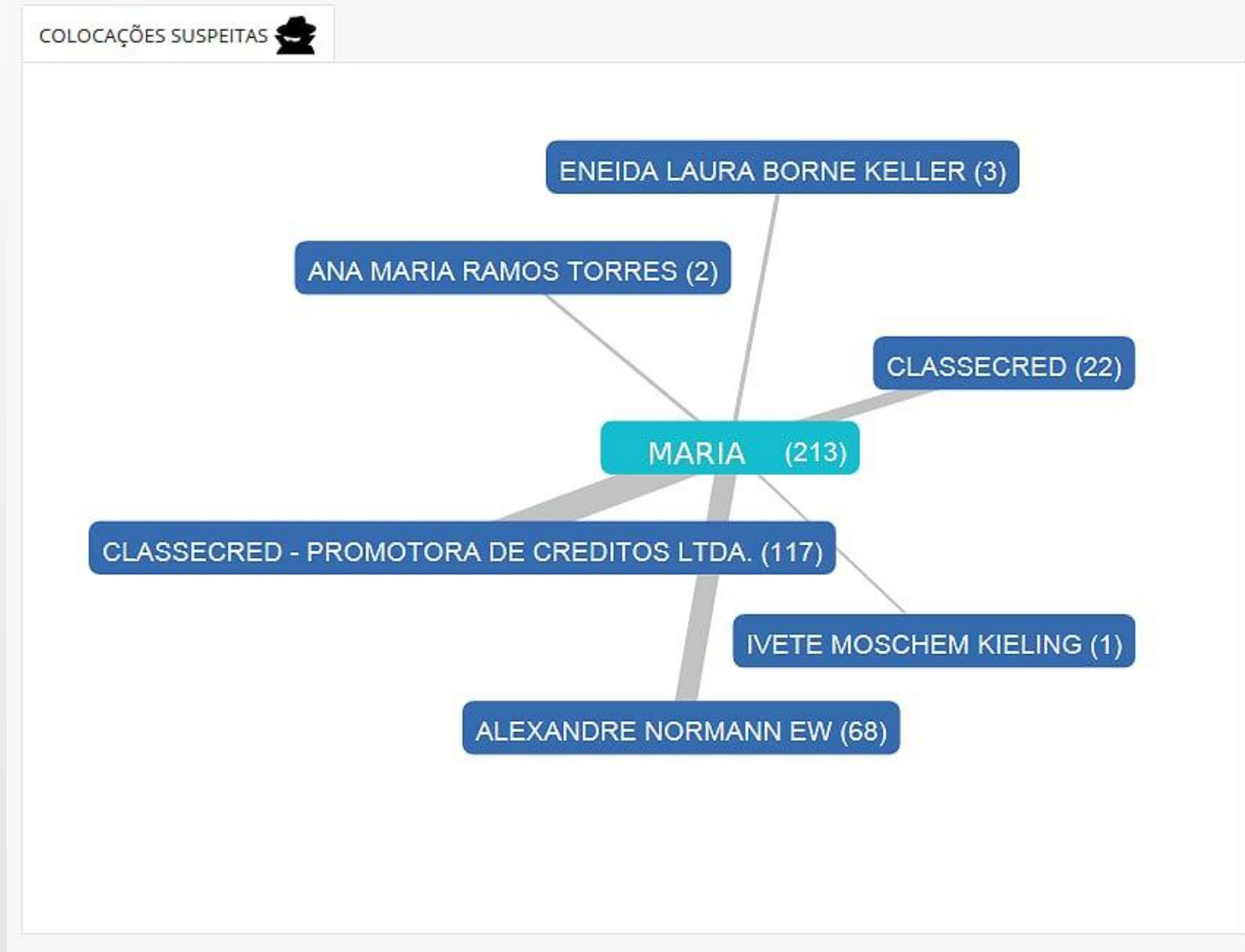
EXTENSÕES QLIKVIEW

- ▶ Tanto no Qlikview quanto no QlikSense, é possível criar extensões
- ▶ Contornar problemas como centralização do painel em ambientes produtivos
- ▶ Ampliar opções de visualizações
- ▶ Segue um tutorial de como adicionar extensões ao Qlikview, como as 3 mencionadas nos slides seguintes:

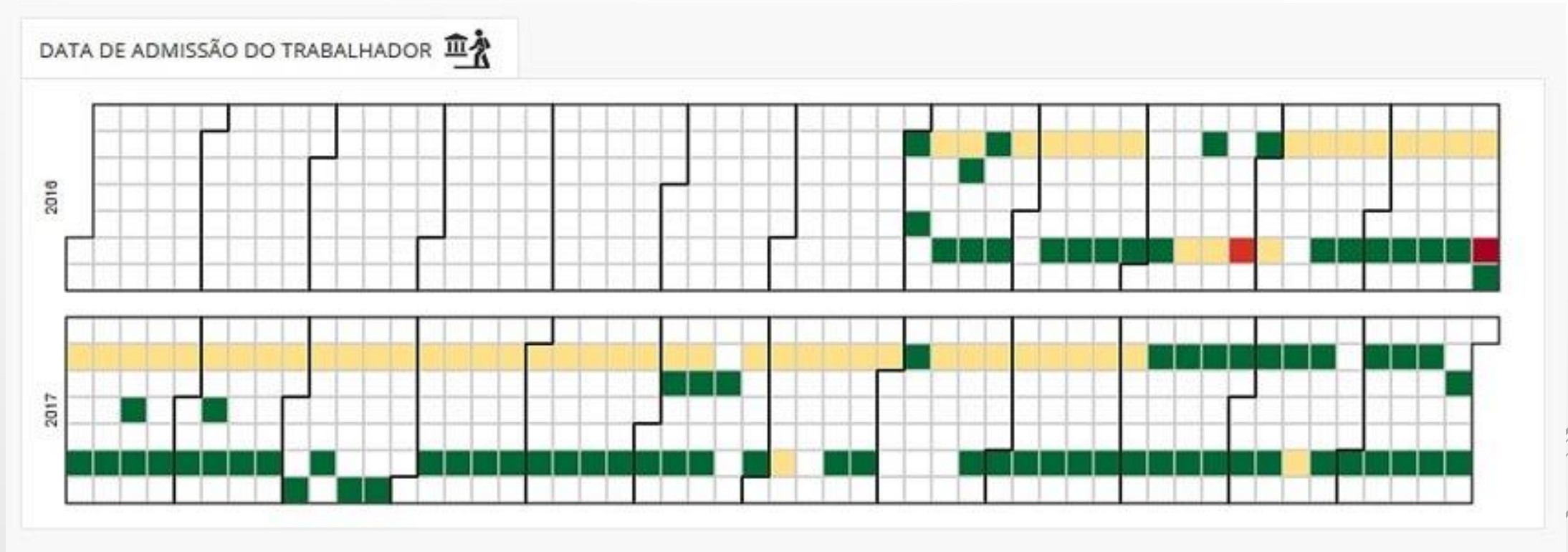
<https://goo.gl/rMRH6Y>

EXTENSÕES QLIKVIEW

- ▶ Extensão usando um componente D3 Force-Directed Graph, baseado em:
 - ▶ <https://bl.ocks.org/mbostock/4062045>
 - ▶ Extensão em:
https://github.com/Dienert/d3_forceweightgraph
- ▶ Mesma base para a aplicação do Facebook
- ▶ A espessura da aresta corresponde ao valor da conexão



EXTENSÕES QLIKVIEW

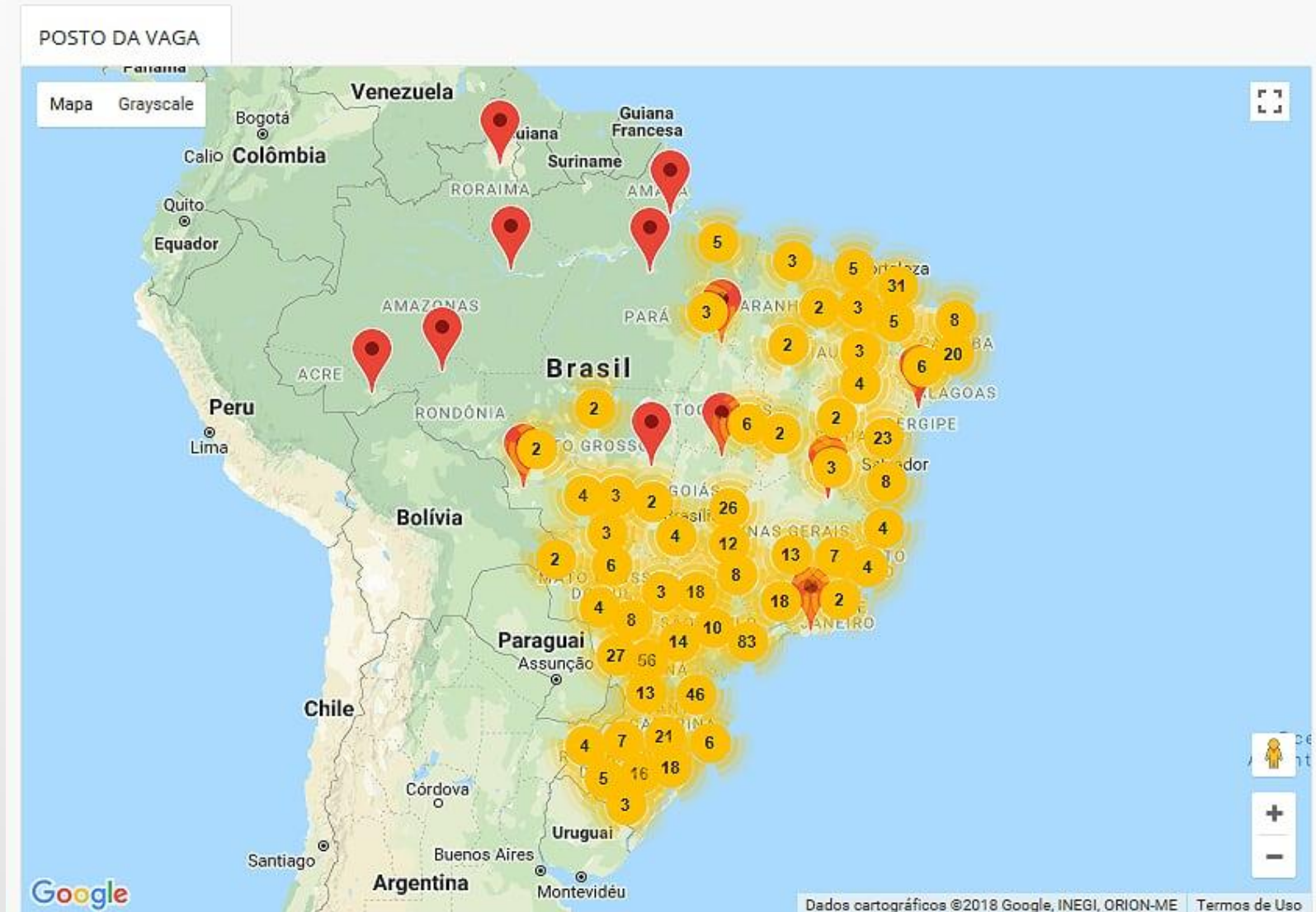


Extensão usando um componente D3 de Calendário baseado em: <https://bl.ocks.org/mbostock/4063318>

Extensão em: https://github.com/Dienert/d3_calendar

EXTENSÕES QLIKVIEW

- ▶ Extensão usando API do Google
- ▶ API: <https://developers.google.com/maps/documentation/javascript/marker-clustering?authuser=0>
- ▶ Extensão em: <https://github.com/mindspank/GoogleMaps-Cluster-QV11>



VISUALIZAÇÃO DA CAPTURA DO FACEBOOK

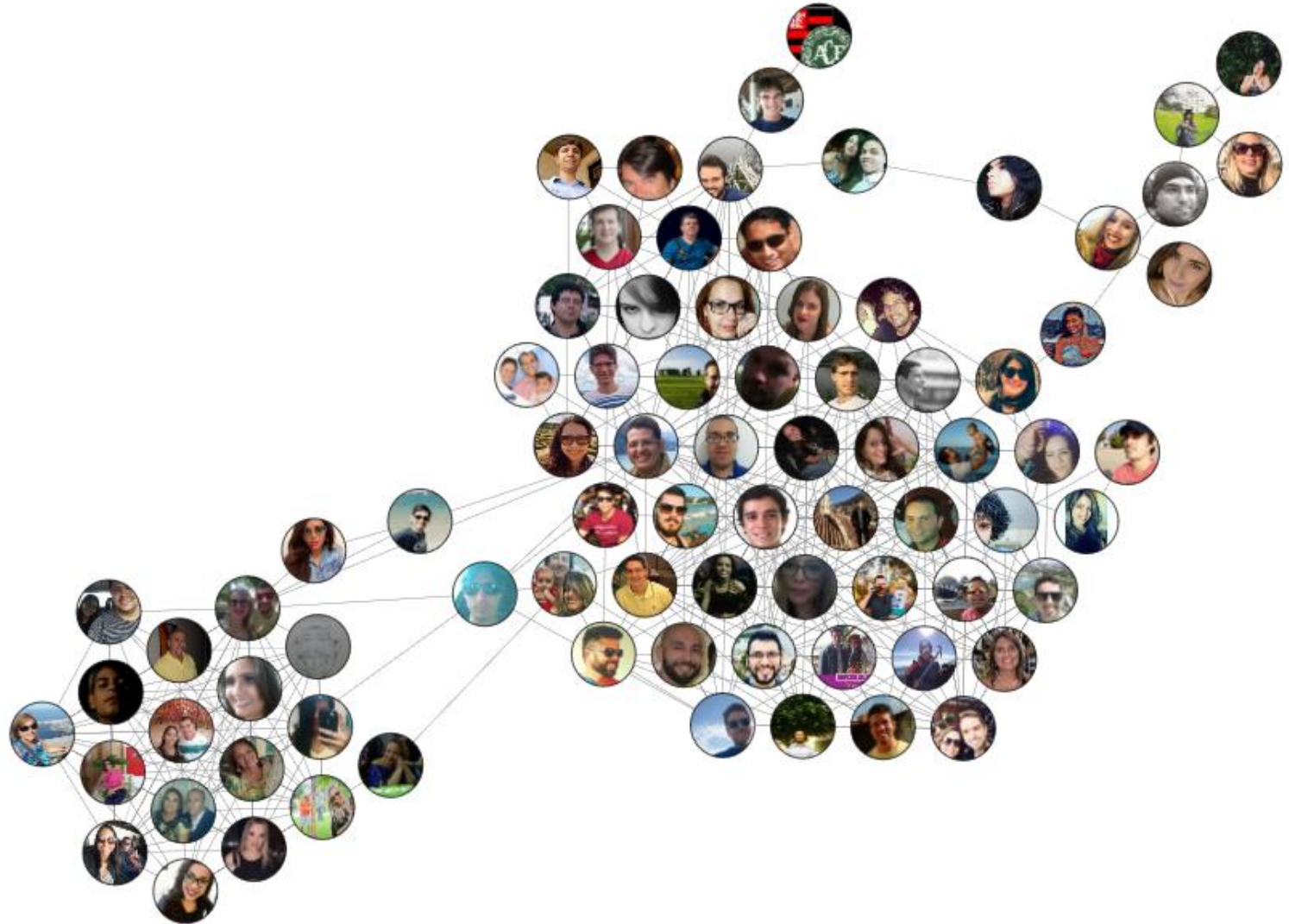
Funcionalidades

- ▶ Destaca conexões ao passar o mouse em um nó
- ▶ Permite buscar um amigo pelo nome numa lista ordenada
- ▶ Ao dar dois cliques abre o perfil do amigo
- ▶ Permite Zoom
- ▶ Permite deslocar ao clicar, segurar e arrastar
- ▶ Disponível em:

<https://github.com/Dienert/facebook-python>

Inspirado na extensão do Chrome Lost Circles - Social Network:

<https://chrome.google.com/webstore/detail/lost-circles-social-netwo/ehpmfdlcppenimpibdifodjgfafkjhl>



VISUALIZAÇÃO DA CAPTURA DO FACEBOOK

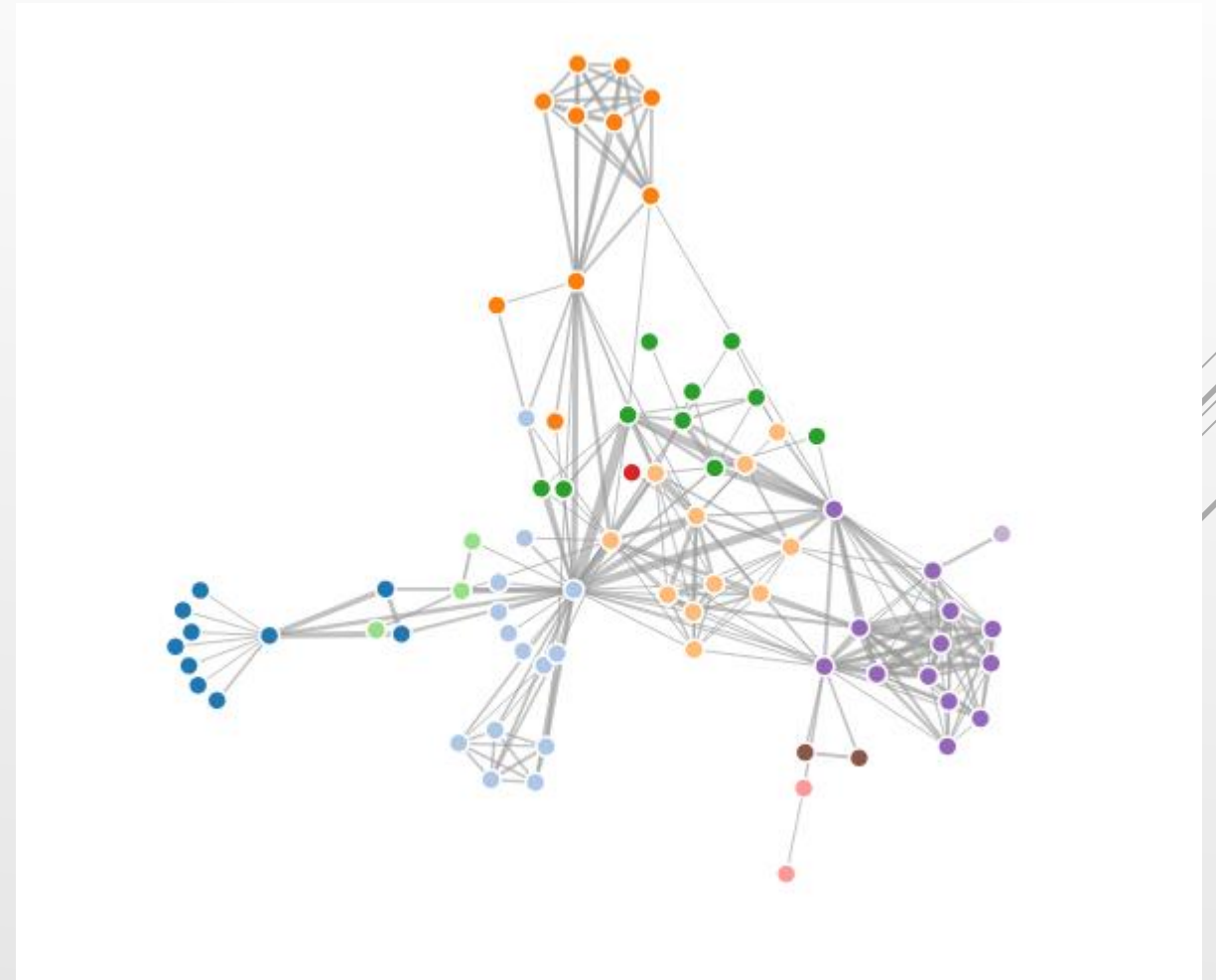
► Aplicações Base

Force-Directed Graph

- <https://bl.ocks.org/mbostock/4062045>
- Usa um JSON com array de nós e elos
- Permite clicar em um nó e puxá-lo, ele arrasta os conectados

Outros exemplos de funcionalidades neste gráfico:

- <http://www.coppelia.io/2014/07/an-a-to-z-of-extra-features-for-the-d3-force-layout/>



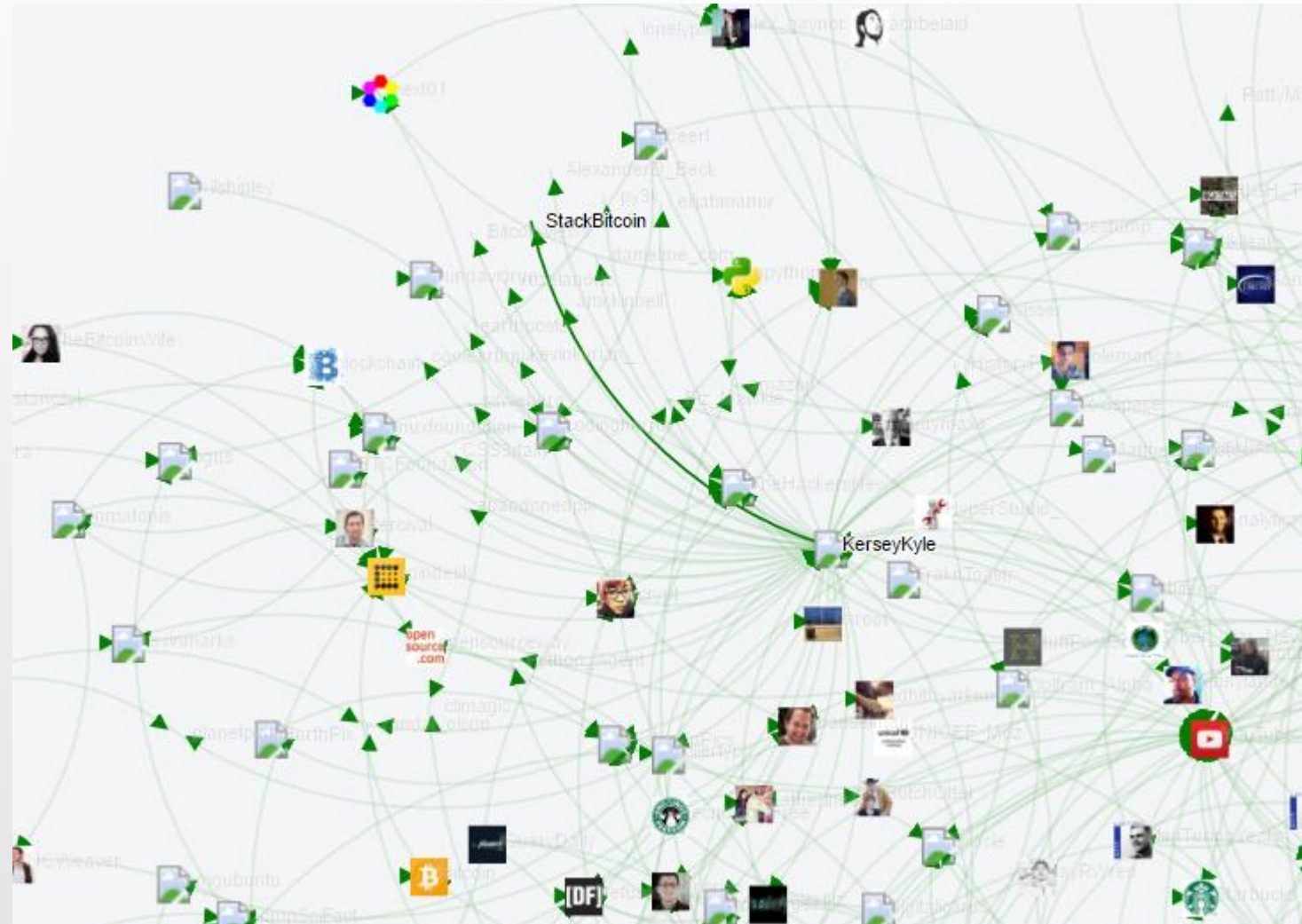
VISUALIZAÇÃO DA CAPTURA DO FACEBOOK

► Aplicações Base

Destacar arestas com o mouse

- <http://jsfiddle.net/kerseykyle/89YcH/>

- Possui Zoom
- Pode arrastar a estrutura com o mouse
- Ao colocar o mouse sobre um nó, suas conexões ficam destacadas e as outras transparentes

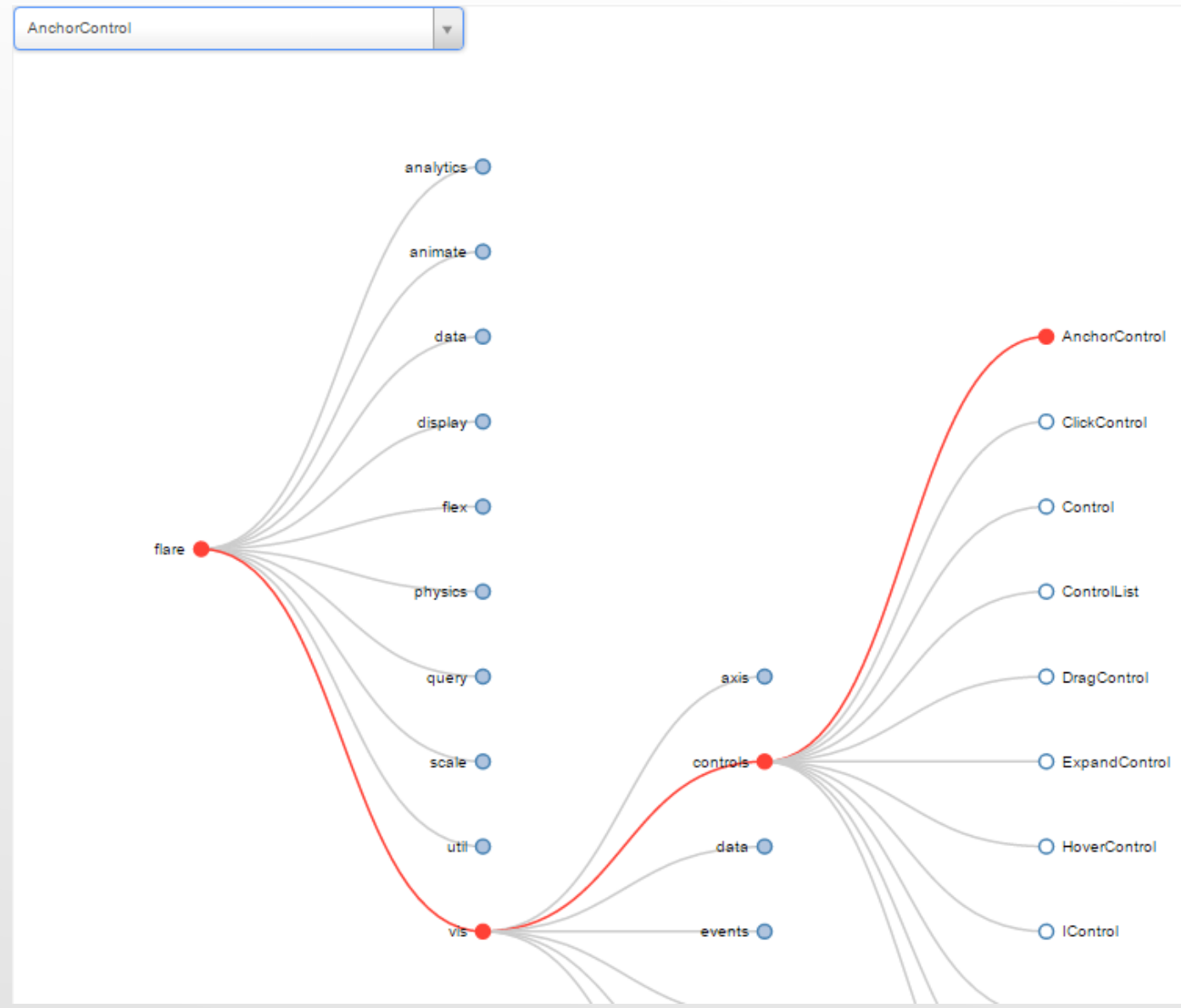


VISUALIZAÇÃO DE DADOS

► Aplicações Base

CollapsibleTree Search

- <http://bl.ocks.org/PBrockman/0f22818096428b12ea23>
- Ao escolher qualquer nó, na comboBox mostra-se o caminho até chegar a esse nó

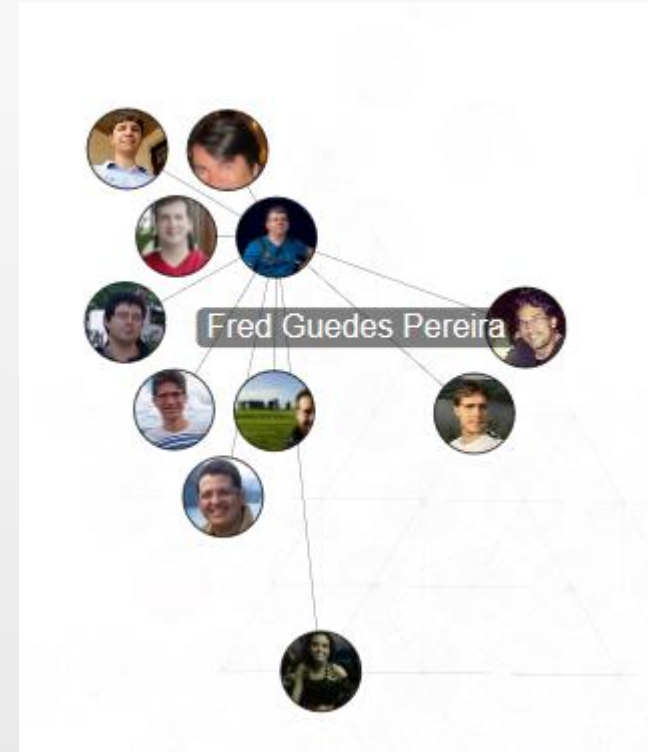


VISUALIZAÇÃO DA CAPTURA DO FACEBOOK

► Resultado

Destacar arestas com o mouse

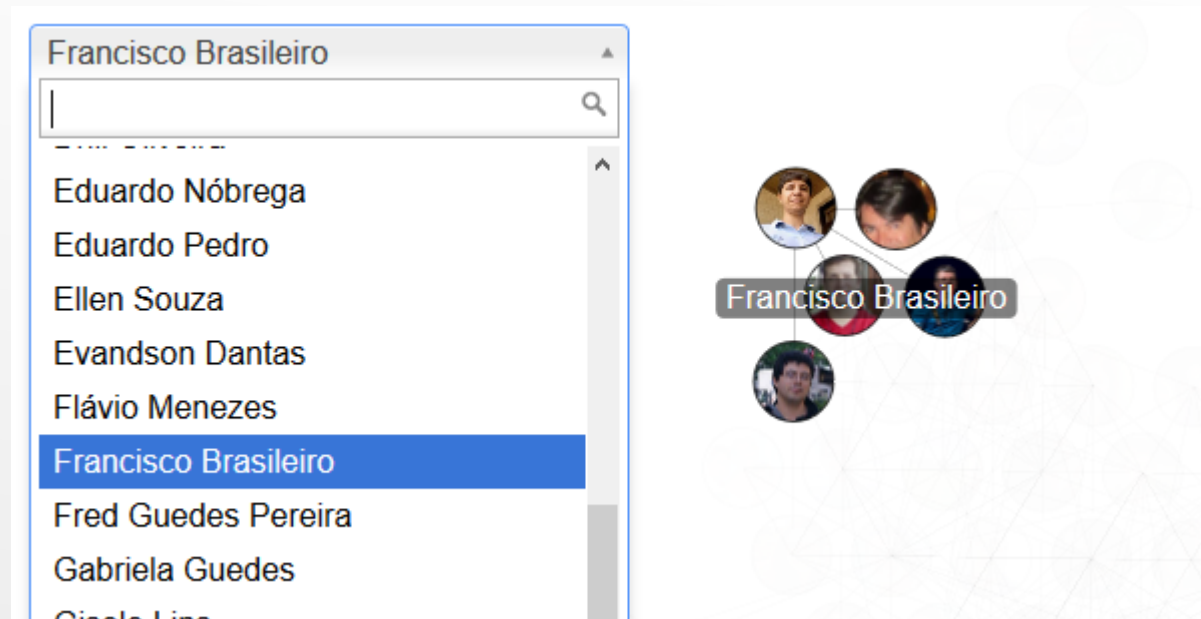
- Ao colocar o mouse sobre alguém



VISUALIZAÇÃO DA CAPTURA DO FACEBOOK

► Resultado

Ao procurar o amigo na lista, ele e suas conexões são destacadas e o restante fica transparente



OUTROS EXEMPLOS DE VISUALIZAÇÕES

Vencedores de um concurso de visualização de dados de 2019

Storytelling sobre Fake News no Twitter nas Eleições dos EUA-2016

O que existe nas profundezas do mar?

Complexidade econômica global

25 Melhores Visualizações de dados de 2019

Uma grande lista de visualizações de 2019

Lista interativa de Visualizações em D3

Visualizações para séries temporais

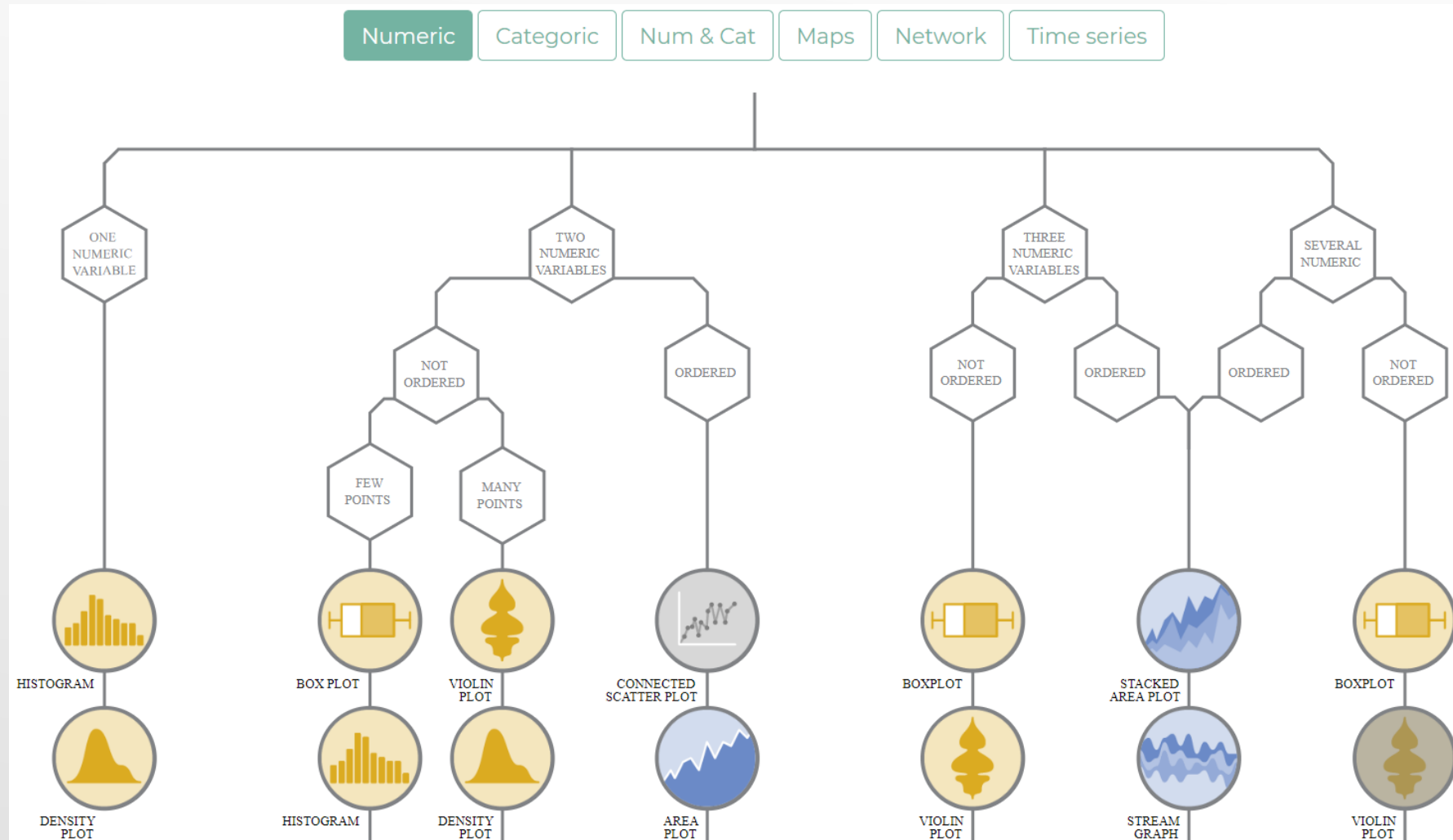
Galeria Bokeh

Explore do Observablehq.com

1 Exemplo de Visualização encontrada no explore do Observable

COMO ESCOLHER UMA VISUALIZAÇÃO?

Um bom começo: <https://www.data-to-viz.com/>



ONDE SE INSPIRAR?

Shirley Wu twitter github

Nadieh Bremer twitter github (jurado de concursos de #dataviz)

Mike Bostock twitter github observablehq (criador do D3)

Jeremy Ashkenas twitter github observablehq (trabalha no Observablehq.com)

Seguir as hashtag no LinkedIn e Instagram: #dataviz
#datavisualization #d3 #d3js #observablehq #datascience
#scrapy

VISUALIZAÇÃO DE DADOS



Data-Driven Documents (d3js.org) github.com/d3/d3

- ▶ Visualização de dados com padrões Web:
HTML5, SVG, Canvas, CSS, JavaScript
- ▶ Combina visualizações poderosas e técnicas de interação com uma abordagem orientada a dados (data-driven) para manipulação da DOM
- ▶ DOM = Document object Model (Convenção para HTML, XHTML e XML)
- ▶ Apresenta benefícios como
 - ▶ Maior expressividade
 - ▶ Compatibilidade com ferramentas e tecnologias existentes como CSS e Debuggers
- ▶ Criador: Mike Bostock [slides do próprio para um workshop](#)

VISUALIZAÇÃO DE DADOS



Data-Driven Documents (D3.js)

- ▶ **SVG** (Scalable Vector Graphics) → é uma linguagem XML que descreve de forma vetorial gráficos e desenhos bidimensionais de forma estática, dinâmica ou animada e que pode ser embarcada no HTML
- ▶ Vetorial → não perde nem um pouco de qualidade quando é feito zoom ou resize
- ▶ Permite animações, efeitos e provê uma expressividade muito maior do que HTML
- ▶ Especificação do SVG → <https://www.w3.org/TR/SVG/>
- ▶ MDN → <https://developer.mozilla.org/en-US/docs/Web/SVG>

VISUALIZAÇÃO DE DADOS



Data-Driven Documents (D3.js)

- ▶ **CSS** (Cascading Style Sheets) → definição da apresentação (fontes, cores, margens, opacidade, etc)
- ▶ D3 não oferece benefícios para o CSS como para o HTML e o SVG, exceto para os estilos locais (dentro das tags)
- ▶ Mas continua sendo usado da mesma maneira

VISUALIZAÇÃO DE DADOS



Data-Driven Documents (D3.js)

- ▶ Como o D3 é um framework **JavaScript**, essa a é linguagem de programação usada
- ▶ É uma linguagem leve, interpretada e tem funções de primeira-classe
 - ▶ Significa que funções podem ser passadas e retornadas como parâmetro
- ▶ Baseada em protótipos → onde um objeto serve de base para outros
- ▶ Multiparadigma → orientação a objetos, imperativa e declarativa (por exemplo funcional)
- ▶ Linguagem de scripts dinâmica

VISUALIZAÇÃO DE DADOS



Data-Driven Documents (D3.js)

Mas CALMA, não precisa saber tudo de cada tecnologia!

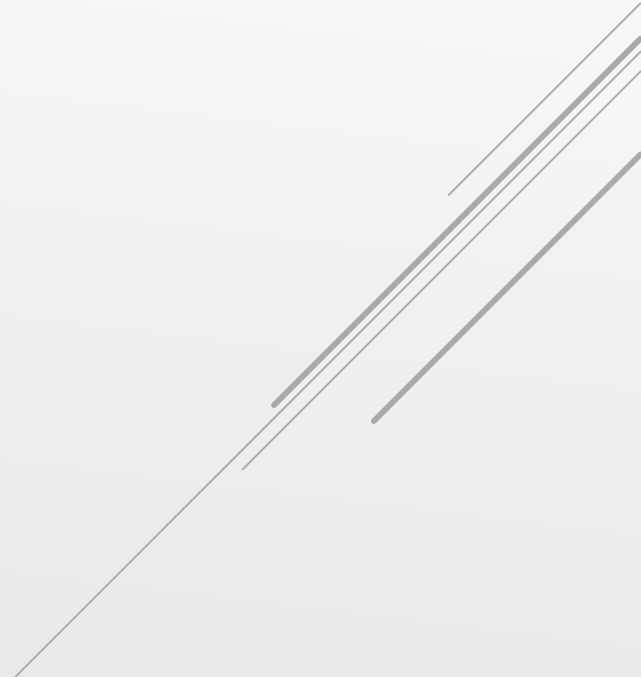


VISUALIZAÇÃO DE DADOS



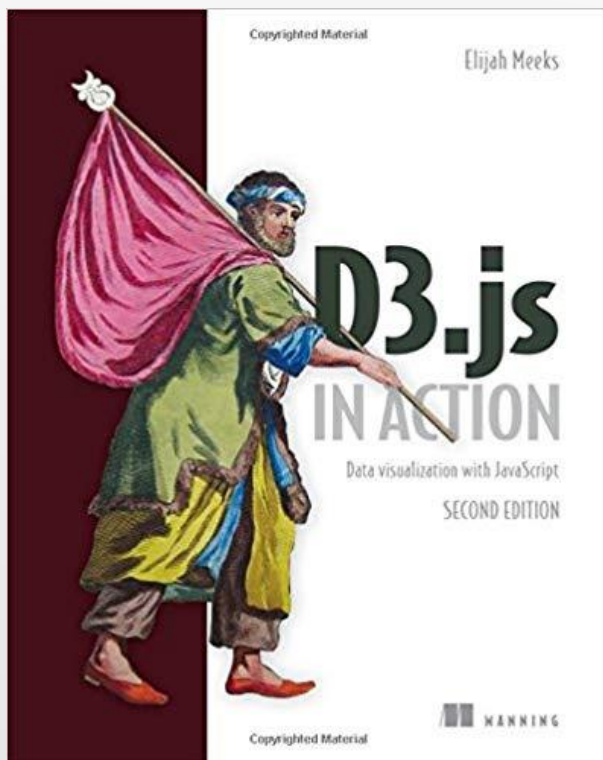
Data-Driven Documents (D3.js)

Vamos começar devagar

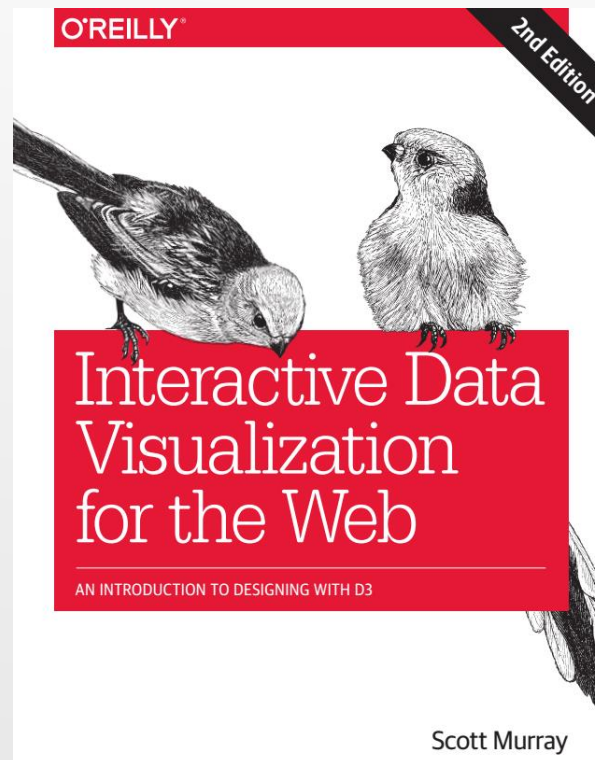


D3.JS

► Referências



[Link](#)



[Link](#)



[Link](#)

D3.JS

► Esta aula se baseia neste livro:

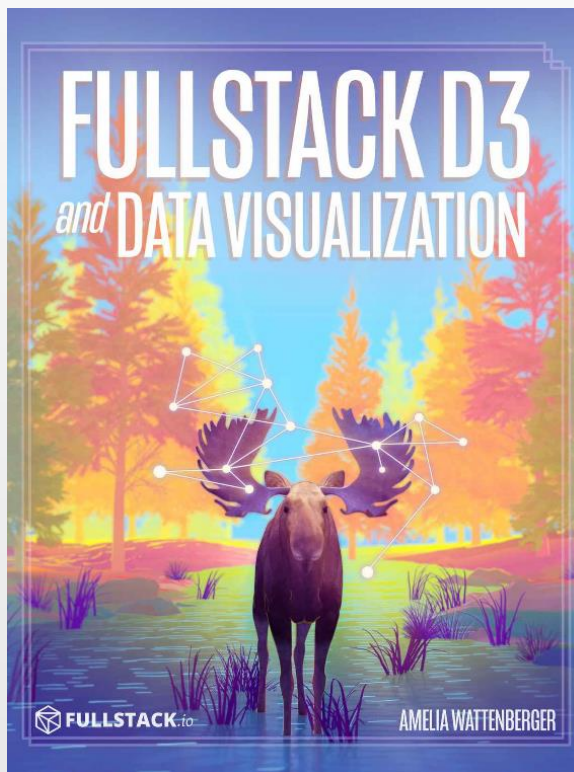
► D3.V5

► Direto ao ponto

► Muitos Exemplos Concretos

► Última versão:

10/12/2019



[Link](#)

Autora:

Amelia Wattenberger

[@Wattenberger](#)

Editor:

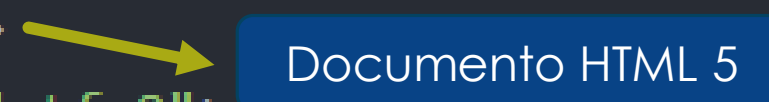
Nate Murray

[@fullstackio](#)

SVG E HTML5

Código em: <http://bit.do/fpJRs>

```
1  <!DOCTYPE html>
2  <meta charset="utf-8">
3  <html>
4  <body>
5      <svg width="800" height="800">
6          <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7      </svg>
8  </body>
9  </html>
```

A yellow arrow points from the text "<!DOCTYPE html>" on line 1 to a blue rectangular box containing the text "Documento HTML 5".

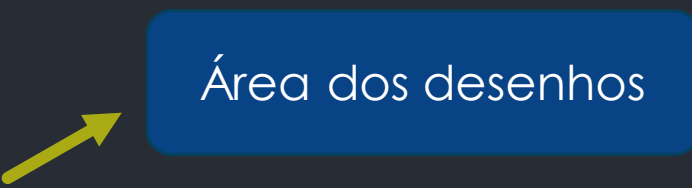
SVG E HTML5

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <body>
5   <svg width="800" height="800">
6     <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7   </svg>
8 </body>
9 </html>
```

Codificação de Caracteres

SVG E HTML5

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <body>
5     <svg width="800" height="800">
6         <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7     </svg>
8 </body>
9 </html>
```



Área dos desenhos

A yellow arrow points from the text 'Área dos desenhos' in a blue box to the opening <svg> tag on line 5 of the code.

SVG E HTML5

```
1  <!DOCTYPE html>
2  <meta charset="utf-8">
3  <html>
4  <body>
5      <svg width="800" height="800">
6          <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7      </svg>
8  </body>
9  </html>
```

Coordenadas (x,y)
do ponto 1

SVG E HTML5

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <body>
5   <svg width="800" height="800">
6     <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7   </svg>
8 </body>
9 </html>
```

Coordenadas (x,y)
do ponto 2

SVG E HTML5

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <body>
5   <svg width="800" height="800">
6     <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7   </svg>
8 </body>
9 </html>
```

Cor do contorno

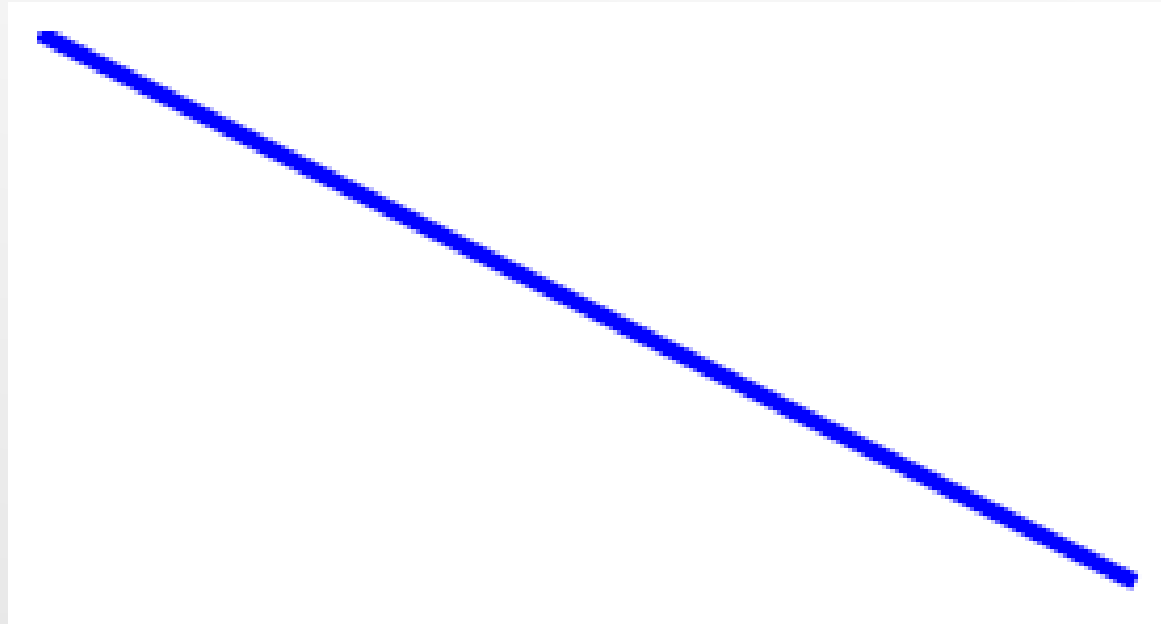
SVG E HTML5

```
1 <!DOCTYPE html>
2 <meta charset="utf-8">
3 <html>
4 <body>
5   <svg width="800" height="800">
6     <line x1="0" y1="0" x2="200" y2="100" stroke="blue" stroke-width="3"/>
7   </svg>
8 </body>
9 </html>
```

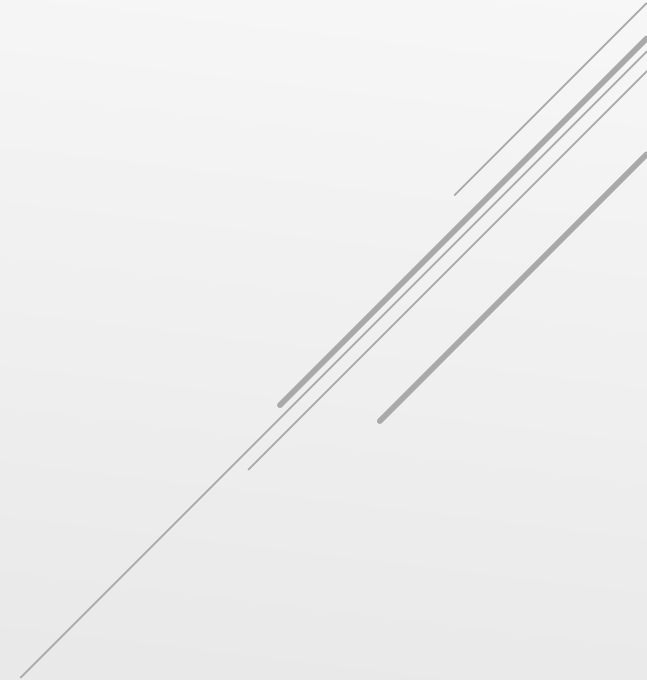
↑
Largura do contorno

SVG E HTML5

$(x1,y1)$



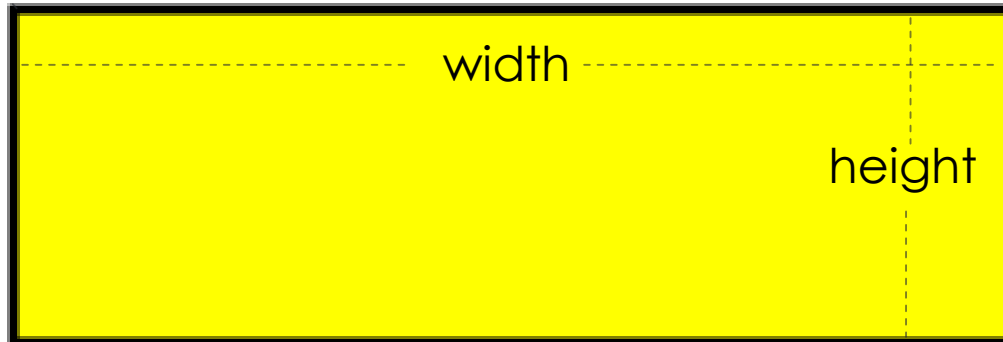
$(x2,y2)$



SVG E HTML5

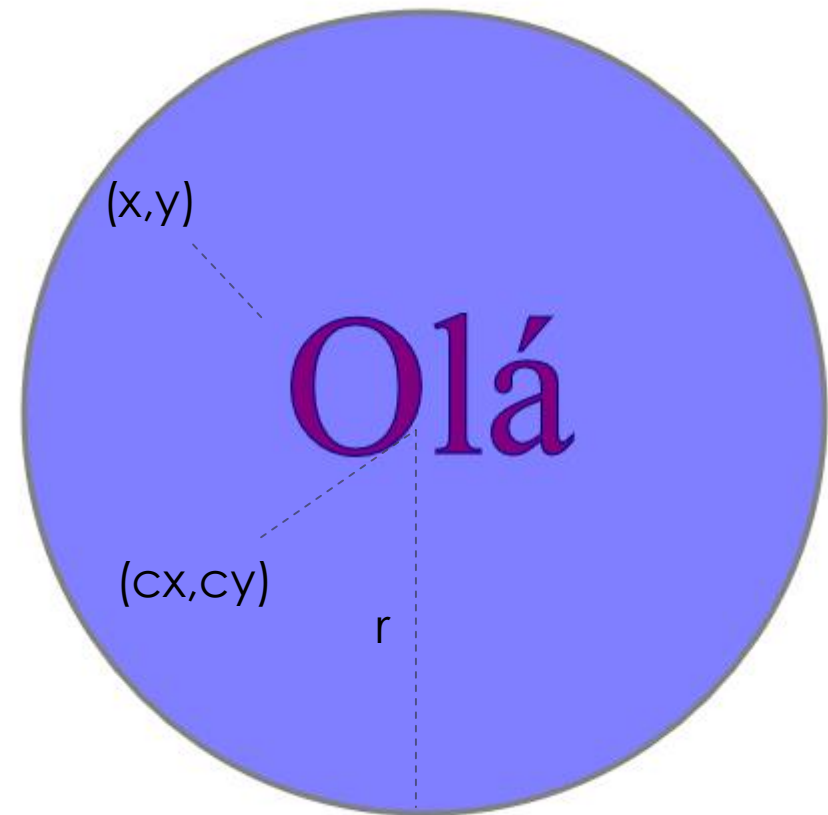
```
<rect width="300" height="100" x="200" y="100" fill="yellow" stroke="black" stroke-width="3"/>
```

(x,y)



SVG E HTML5

```
<svg width="800" height="800">
  <text x="230" y="270" fill="red"
        stroke="black"
        font-size="100">
    Olá
  </text>
  <circle r="200" cx="300" cy="250"
          fill="blue" opacity="0.5"
          stroke="black" stroke-width="3"/>
</svg>
```

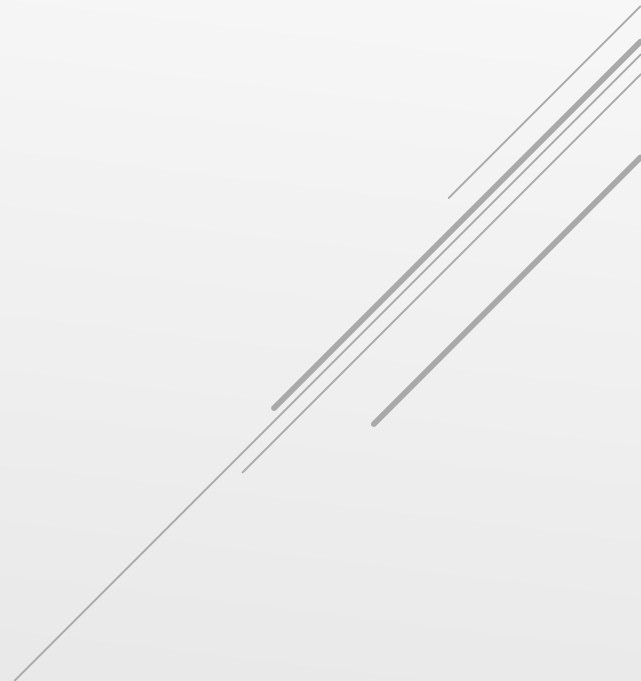


SVG E HTML5

CSS3 – Transformação: Rotação & Transição

```
<rect width="300" height="100" x="200" y="100" fill="yellow" stroke="black" stroke-width="3"/>
```

```
<style>
rect {
  fill: red;
}
rect:hover {
  -ms-transform: rotate(360deg); /* IE 9 */
  -webkit-transform: rotate(360deg); /* Safari */
  transform: rotate(360deg);
  fill: blue;
  transform-origin: 350px 150px;
  transition: transform 2s, fill 2s;
}
</style>
```



SVG E HTML5

CSS3 – Transformação:Translação

```
circle:hover {  
    -ms-transform: translate(230px, 0); /* IE 9 */  
    -webkit-transform: translate(230px, 0); /* Safari */  
    transform: translate(230px, 0);  
    fill-opacity: 0.5;  
    transition: transform 2s, fill-opacity 2s;  
}
```

```
<svg width="800" height="800">  
    <text font-size="60" x="335" y="250">Buuu!! </text>  
    <circle r="200" cx="300" cy="250" fill="blue" stroke="black" stroke-width="3"/>  
</svg>
```


D3.JS – 1º EXEMPLO

- ▶ Acessar <https://darksky.net/dev/register>
- ▶ Se registrar e copiar a Secret Key Api
- ▶ Entrar no Google Maps a Latitude e Longitude que desejar
 - ▶ Clicar com o botão direito no lugar escolhido e depois em “O que há aqui?”



D3.JS – 1º EXEMPLO

▶ Instalar o Node.js: <https://nodejs.org/en/>

▶ Na linha de comando executar:

```
npm install -g live-server
```

▶ Clonar do git:

<https://github.com/Dienert/explore-data-visualization-with-D3>

▶ Depois rodar o seguinte comando na pasta “examples”:

```
live-server
```

▶ A documentação da API está em:

<https://darksy.net/dev/docs#data-point>

D3.JS – FUNÇÕES ASSÍNCRONAS E PROMISES

► Async e await:

<https://observablehq.com/@dienert/introducao-a-promises>

```
1  async function drawLineChart() {  
2    |    const dataset = await d3.json('./scripts/my_weather_data.json');  
3    |  }
```

D3.JS – 1º EXEMPLO

Funções de Acesso

- ▶ Facilidade com mudanças
- ▶ Documentação
- ▶ Enquadramento/Tipo do gráfico

```
5 // Acessando dados
6 const yAccessor = d => d.temperatureMax;
7 const dateParser = d3.timeParse("%Y-%m-%d")
8 const xAccessor = d => dateParser(d.date);
9 console.log(xAccessor(dataset[0]))
```

D3.JS – 1º EXEMPLO

Desenhando

- ▶ Wrapper: contém todos os elementos SVG do gráfico.
- ▶ Bounds (limite interno): contém todos os elementos de dados, nesse caso a linha.

```
15 let dimensions = {  
16     width: window.innerWidth * 0.9,  
17     height: 400,  
18     margin: {  
19         top: 15,  
20         right: 15,  
21         bottom: 40,  
22         left: 60,  
23     },  
24 }
```

```
25 dimensions.boundedWidth = dimensions.width - dimensions.margin.left - dimensions.margin.right  
26 dimensions.boundedHeight = dimensions.height - dimensions.margin.top - dimensions.margin.bottom
```

D3.JS – 1º EXEMPLO

Desenhando

- ▶ `d3.select(css_selector)`:
retorna um selector do
elemento = `const wrapper`

(Equivale ao `response.css()`
do Scrapy)

`#wrapper`: id
`.wrapper`: class

Lembrando

- ▶ `selector.append(tag)`:

cria um elemento na seleção e retorna um novo selector = `const svg`


```
26 // Desenhando
27 const wrapper = d3.select("#wrapper")
28 console.log(wrapper)
29 const svg = wrapper.append("svg")
30 console.log(svg)
31     svg.attr("width", dimensions.width)
32     svg.attr("height", dimensions.height)
```


D3.JS – 1º EXEMPLO

► *selector.attr*(atributo, valor):

Adiciona ou substitui um atributo na seleção

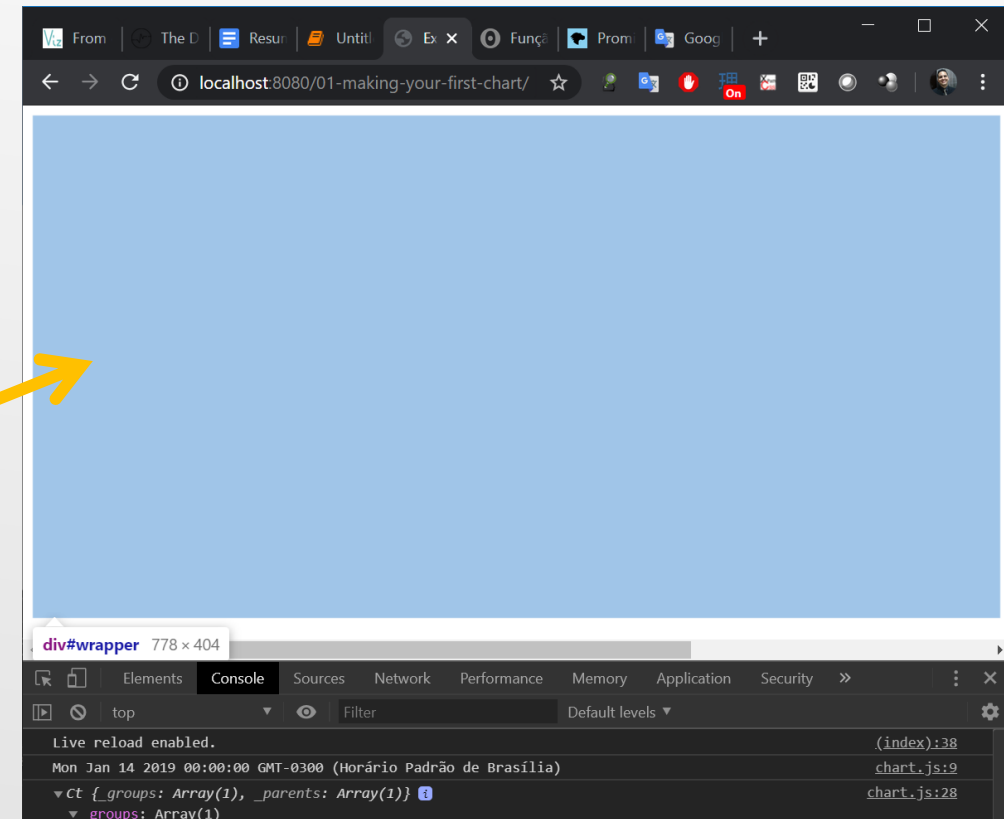
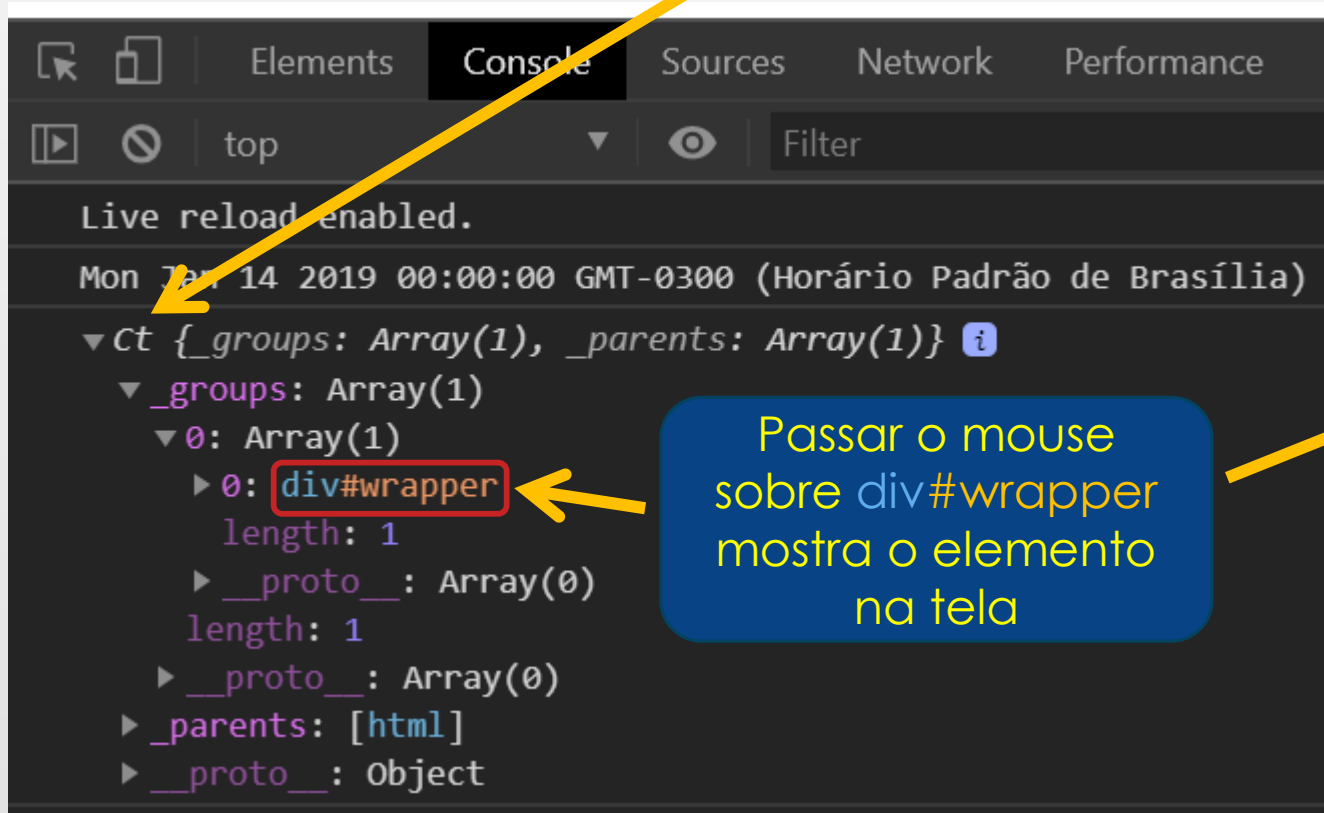
```
26 // Desenhando
27 const wrapper = d3.select("#wrapper")
28 console.log(wrapper)
29 const svg = wrapper.append("svg")
30 console.log(svg)
31 svg.attr("width", dimensions.width)
32 svg.attr("height", dimensions.height)
```



D3.JS – 1º EXEMPLO

Olhando no console F12

```
26 // Desenhando
27 const wrapper = d3.select("#wrapper")
28 console.log(wrapper)
29 const svg = wrapper.append("svg")
30 console.log(svg)
31 svg.attr("width", dimensions.width)
32 svg.attr("height", dimensions.height)
```



D3.JS – 1º EXEMPLO

Podemos Reescrever:

Com:


```
26 // Desenhando
27 const wrapper = d3.select("#wrapper")
28 console.log(wrapper)
29 const svg = wrapper.append("svg")
30 console.log(svg)
31     svg.attr("width", dimensions.width)
32     svg.attr("height", dimensions.height)
```

```
29 // Desenhando
30 const svg = d3.select("#wrapper")
31     .append("svg")
32     .attr("width", dimensions.width)
33     .attr("height", dimensions.height)
```

- ➔ Qualquer método que selecione ou crie um novo objeto, retorna essa nova seleção
- ➔ Qualquer método que manipule a seleção atual retorna a mesma seleção

D3.JS – 1º EXEMPLO

```
35 // Podemos desenhar o gráfico dentro de um elemento "g" e movê-lo todo de uma vez usando a propriedade de
    transformação do CSS
36 const bounds = svg.append("g") // Pense no "g" como o equivalente a uma "div" dentro do SVG
37     .style("transform", `translate(${dimensions.margin.left}px,
38                               ${dimensions.margin.top}px)`)
```



- Interpolação de Strings crase (``) em vez de aspas simples (') ou aspas duplas("")
- \${variavel} é interpretado e concatenado (unido) ao restante do texto.

D3.JS – 1º EXEMPLO

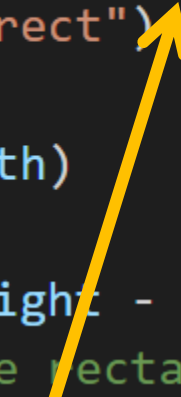
```
40 // Criando a escala
41 const yScale = d3.scaleLinear()
42   .domain(d3.extent(dataset, yAccessor)) // d3.extent retorna um array com a temperatura mínima e máxima
43   .range([dimensions.boundedHeight, 0]) // faixa com o menor e maior número de pixels a mostrar
```



- Criamos a escala do Eixo y, que é linear entre a temperatura mínima e a máxima para 0 e a altura em pixels da dimensão determinada (`boundedHeight`)

D3.JS – 1º EXEMPLO

```
45 // Vamos visualizar a média das temperaturas em um retângulo
46 const meanTemperaturePlacement = yScale(d3.mean(dataset, yAccessor))
47 const meanTemperatures = bounds.append("rect")
48   .attr("x", 0)
49   .attr("width", dimensions.boundedWidth)
50   .attr("y", meanTemperaturePlacement)
51   .attr("height", dimensions.boundedHeight - meanTemperaturePlacement)
52   .attr("fill", "#e0f3f3") // Make the rectangle frosty to indicate freezing
```



- Calculamos a temperatura média
- Plotamos um retângulo para mostrar quando a temperatura está abaixo da média

D3.JS – 1º EXEMPLO

```
54 // Usa uma escala de tempo, pois estamos lidando do datas
55 const xScale = d3.scaleTime()
56     .domain(d3.extent(dataset, xAccessor))
57     .range([0, dimensions.boundedWidth])
```

- Definimos a escala para o eixo X, da mesma forma que Y, para as datas máximas e mínimas, contudo dessa vez não usamos a escala linear, usamos uma escala de tempo, também em relação à quantidade de pixels.

D3.JS – 1º EXEMPLO

```
bounds.append("path").attr("d", "M 0 0 L 100 0 L 100 100 L 50 100 L 50 50 L 0 50 Z")
```



O atributo d do elemento svg <path> aceita alguns comandos que podem ser com letra maiúscula (se dado um valor absoluto) ou minúscula (se dado um valor relativo):

- ▶ M move para um ponto (seguido dos valores x e y)
- ▶ L desenha uma linha até um ponto (seguido dos valores x e y)
- ▶ Z desenha uma linha para o primeiro ponto"

D3.JS – 1º EXEMPLO

```
73 // Cria um gerador que converte pontos de dados em uma string para o atributo d do path
74 const lineGenerator = d3.line()
75 // Transforma o ponto para o a função de acesso apropriada e a escala para pegar o valor do
76 // espaço em pixels
77     .x(d => xScale(xAccessor(d)))
78     .y(d => yScale(yAccessor(d)))
```

O método x recebe os valores do x em pixels, assim como o método y

```
80 const line = bounds.append("path")
81 // Alimenta a função geradora de linha como o nosso dataset
82 .attr("d", lineGenerator(dataset))
83 // Elementos SVG por padrão têm o preenchimento preto (fill) e sem traço (stroke);
84 // o que resulta em um forma preenchida, a menos que adicionamos algum estilo
85 .attr("fill", "none")
86 .attr("stroke", "#af9358")
87 .attr("stroke-width", 2)
```

D3.JS – 1º EXEMPLO

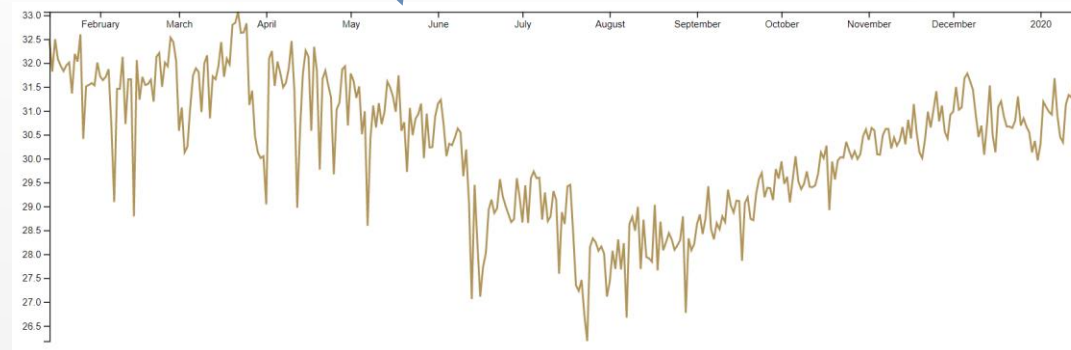
Mas o eixo X ficou em cima, mesmo colocando axisBottom

Desenhando o eixo Y

```
89 | const yAxisGenerator = d3.axisLeft()  
90 |   .scale(yScale)  
91 | const yAxis = bounds.append("g")  
92 |   .call(yAxisGenerator)
```

Desenhando o eixo X

```
94 | const xAxisGenerator = d3.axisBottom()  
95 |   .scale(xScale)  
96 | const xAxis = bounds.append("g")  
97 |   .call(xAxisGenerator)
```



É uma boa ideia criar um g para conter os eixos, por 3 razões:

1. Manter a DOM organizada, para debugging ou exportação
2. Se quisermos remover ou atualizar os eixos
3. Ao modificar todo o eixo como um todo

D3.JS – 1º EXEMPLO

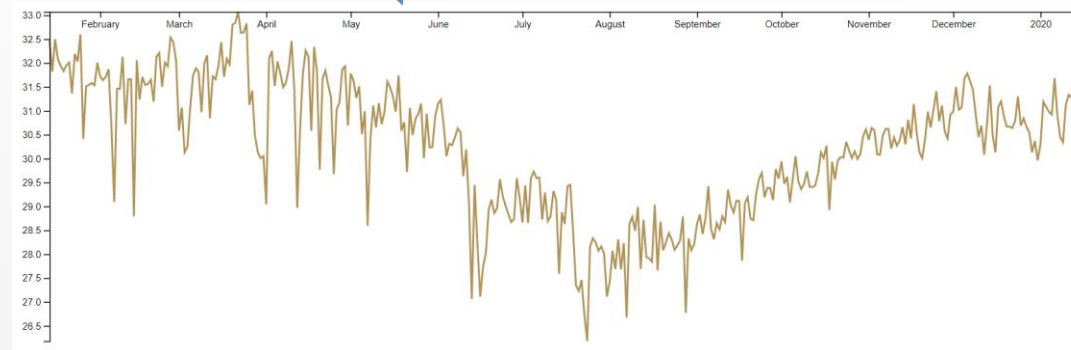
Mas o eixo X ficou em cima, mesmo colocando axisBottom

Desenhando o eixo Y

```
89 | const yAxisGenerator = d3.axisLeft()  
90 |   .scale(yScale)  
91 | const yAxis = bounds.append("g")  
92 |   .call(yAxisGenerator)
```

Desenhando o eixo X

```
94 | const xAxisGenerator = d3.axisBottom()  
95 |   .scale(xScale)  
96 | const xAxis = bounds.append("g")  
97 |   .call(xAxisGenerator)
```



É uma boa ideia criar um g para 2ºC conter os eixos, por 3 razões:

1. Manter a DOM organizada, para debugging ou exportação
2. Se quisermos remover ou atualizar os eixos
3. Ao modificar todo o eixo como um todo

OBSERVABLE

- ▶ [Observablehq.com](https://observablehq.com)
- ▶ Coleção de introduções ao Observable:
<https://observablehq.com/collection/@observablehq/introduction>
- ▶ Comece pela seguinte:
▶ <https://observablehq.com/@dienert/five-minute-introduction>

ABSTRAÇÕES SOBRE D3

► React

- Biblioteca Javascript declarativa, eficiente e flexível para construir interfaces.
- Executa sobre o Node.js

► Configuração:


- Instalar Node.js: <https://nodejs.org/en/>
- Criar a aplicação react:

```
npm install -g create-react-app  
create-react-app my-app
```

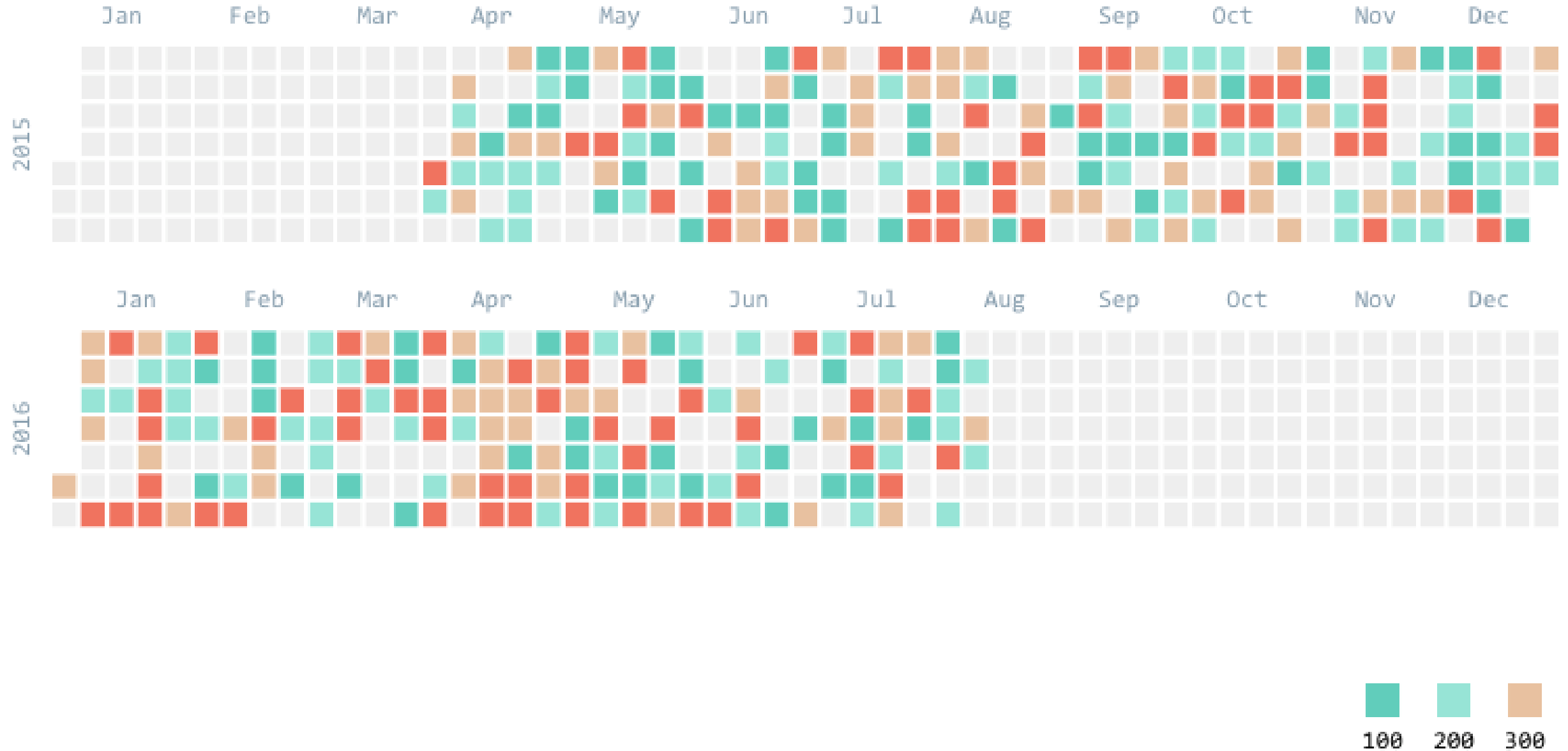
- Instalar Yarn:

<https://yarnpkg.com/lang/en/docs/install/#windows-stable>

ABSTRAÇÕES SOBRE D3

- ▶ Utilizar um framework que provê componentes que abstraem a implementação em D3
 - ▶ Existem vários, um exemplo:
 - ▶ `yarn add @nivo/calendar`
 - ▶ <http://nivo.rocks/components>
 - ▶ <http://nivo.rocks/storybook/>
- 

REPORT ON THE YEAR



OBRIGADO!

