

Gráficos Comerciais na WEB com Chart.js

Compilação feita com base nas referências por:

Prof. José Cintra

<http://www.josecintra.com/blog>

Download dos exemplos disponíveis no [GitHub](#)

Apresentação

O objetivo deste minicurso é fornecer conceitos básicos sobre a utilização da biblioteca Javascript Chart.js para renderização de gráficos comerciais na WEB através do elemento HTML5 Canvas.



Introdução

Os gráficos, ditos comerciais, são representações visuais utilizadas para exibir dados, demonstrar padrões, tendências e ainda, comparar informações qualitativas e quantitativas num determinado espaço de tempo.

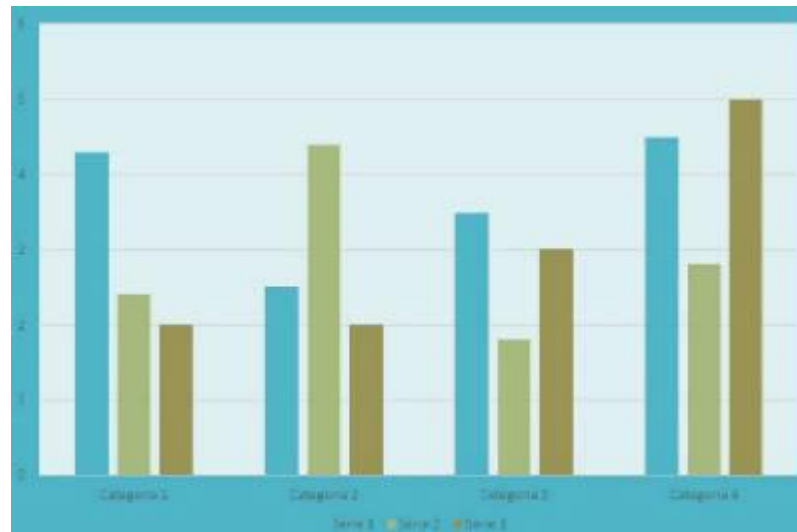
São ferramentas utilizadas em diversas áreas de estudo pois tornam a interpretação e/ou análise das informações mais rápida e objetiva.

Nas páginas a seguir relacionaremos alguns tipos de gráficos mais conhecidos...

Tipos de Gráficos

Gráfico de Barras

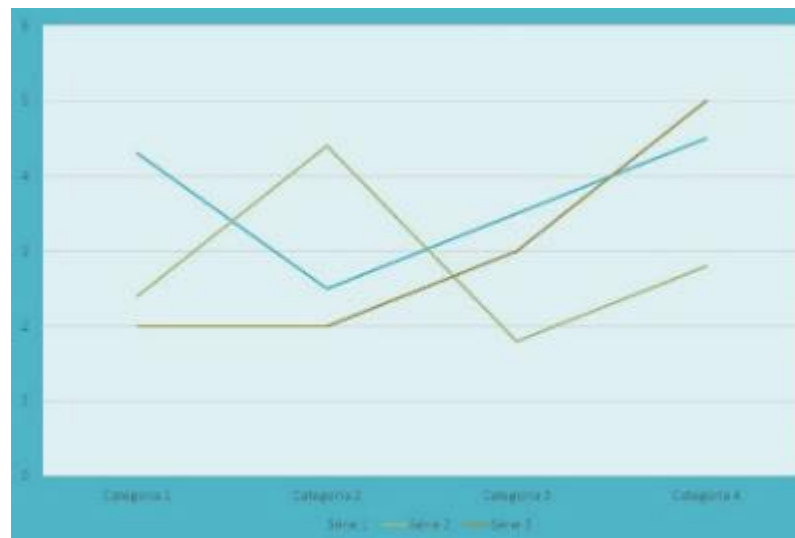
Também conhecido como “Gráfico de Colunas”, eles são usados para comparar quantidades ou mesmo demonstrar valores pontuais de determinado período



Tipos de Gráficos

Gráficos de Linha

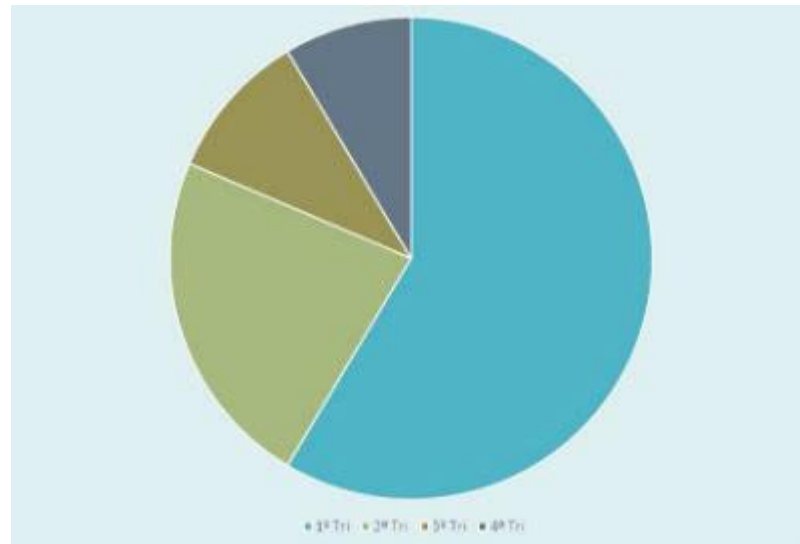
Também chamado de “Gráfico de Segmento”, ele é usado para apresentar valores (sequências numéricas) em determinado espaço de tempo. Ou seja, mostra as evoluções ou diminuições de algum fenômeno.



Tipos de Gráficos

Gráfico Pizza

Também chamado de “Gráfico de Setores”, esse modelo recebe esse nome pois tem a forma de uma pizza, ou seja, é circular. Eles são utilizados para reunir valores a partir de um todo, segundo o conceito de proporcionalidade.

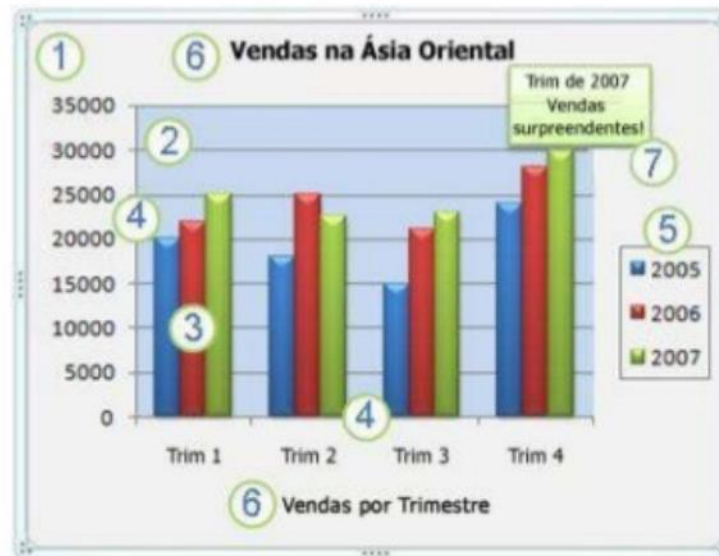


Elementos dos gráficos

Na figura abaixo, destacamos as principais partes de um gráfico.

De acordo com esse exemplo, podemos depreender que a empresa X costuma vender mais no último trimestre de cada ano, com totais de 30.000 para 2007, cerca de 27.000 em 2006 e abaixo de 25000 em 2005. Sendo que, em 2007, houve um pico de vendas.

1. Área do Gráfico
2. Área de Plotagem
3. Ponto de Dados e Série de Dados
4. Eixo Horizontal e Vertical
5. Legenda
6. Título do Gráfico
7. Rótulo de Dados



Fonte : www.michelfabiano.blogspot.com

A biblioteca Chart.js

Chart.js é uma biblioteca Javascript simples e poderosa cujo objetivo é permitir a visualização de dados na web através de gráficos.

Suas principais características são:

- Open Source;
- 8 tipo de gráficos
- Baseada em HTML5 Canvas
- Responsiva
- Permite inserir animações
- E muito mais! Veja aqui mais detalhes sobre a biblioteca:

<https://www.chartjs.org/>

Chart.js Instalação

Veja como preparar o ambiente de desenvolvimento para o uso da biblioteca:

1) Você pode referenciar o JS pelo CDN ou fazer o download.

GITHUB: <https://github.com/chartjs/Chart.js/releases/latest>

CDN: <https://cdnjs.com/libraries/Chart.js>

2) Definir um elemento HTML5 Canvas na página

3) Opcionalmente, preparar estilos CSS de responsividade

4) Definir o conjunto de dados para ser exibido

5) Criar um script JS para programar o gráfico

Chart.js - Exemplo de gráfico

Nos slides a seguir, vamos mostrar um exemplo da criação de um gráfico de barras, através da biblioteca Chart.js, para o seguinte cenário:

A exibição das temperaturas médias mensais das salas do datacenter de uma empresa no primeiro semestre do ano de 2018. A temperaturas médias mensais colhidas nos dois prédios foram as seguintes:

	Jan	Fev	Mar	Abr	Mai	Jun
Prédio 1	30	25	28	31	26	23
Prédio 2	25	30	26	29	31	20

Chart.js - Exemplo de gráfico

Veja abaixo como nosso gráfico será exibido pelo Chart.js:

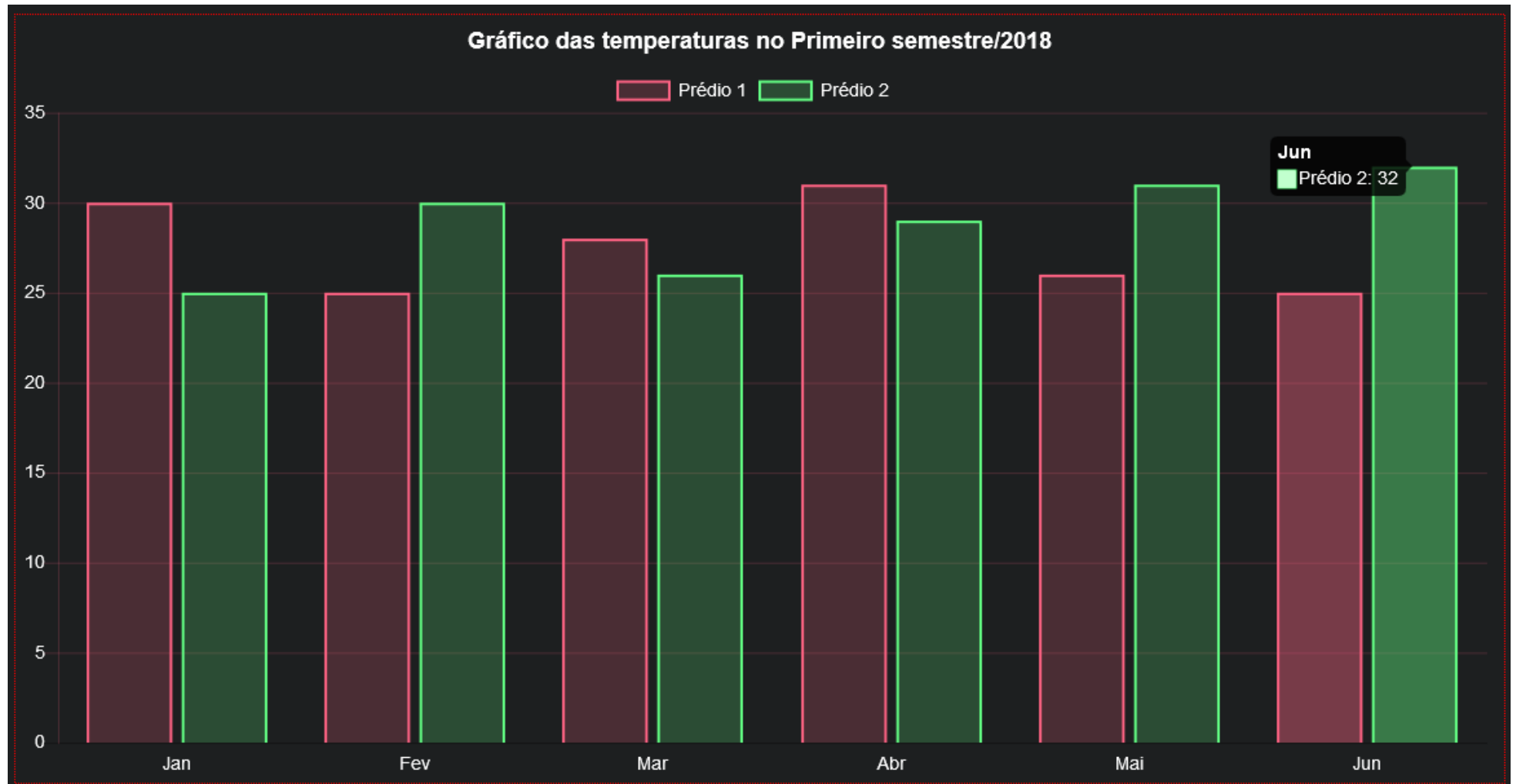


Chart.js – HTML

Na criação do arquivo ‘index.html’, destacamos os seguintes pontos:

1. No head, apontamos para uma página de estilos ‘style.css’, onde está definida a classe ‘chart-container’;
2. Usamos o CDN para referenciar a biblioteca;
3. Apontamos para o nosso arquivo JS que irá definir o nosso gráfico.

```
<html>
  <head>
    <title>Exemplo de gráfico com Chart.js</title>
    <meta name='viewport' content='width=device-width,initial-scale=1'>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <div class="chart-container">
      <canvas id="chart"></canvas>
    </div>
    <script
      src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.7.2/Chart.bundle.min.js">
    </script>
    <script src="chart.js"></script>
  </body>
</html>
```

Chart.js – CSS

Na criação do arquivo ‘style.css’, destacamos os seguintes pontos:

1. A classe ‘chart-container’ define o posicionamento do canvas para garantir a responsividade
2. Os demais estilos definem as cores, bordas e espaçamentos.

```
body {  
  background: #1D1F20;  
  padding: 16px;  
}  
  
canvas {  
  border: 1px dotted red;  
}  
  
.chart-container {  
  position: relative;  
  margin: auto;  
  height: 80vh;  
  width: 80vw;  
}
```

Chart.js – JS – Definindo os Dados

Inicialmente, definimos os dados a serem exibidos. Veja a explicação a seguir

```
var data = {
  labels: ["Jan", "Fev", "Mar", "Abr", "Mai", "Jun"],
  datasets: [{
    label: "Prédio 1",
    backgroundColor: "rgba(255,99,132,0.2)",
    borderColor: "rgba(255,99,132,1)",
    borderWidth: 2,
    hoverBackgroundColor: "rgba(255,99,132,0.4)",
    hoverBorderColor: "rgba(255,99,132,1)",
    data: [30, 25, 28, 31, 26, 23]
  },
  {
    label: "Prédio 2",
    backgroundColor: "rgba(99,255,132,0.2)",
    borderColor: "rgba(99,255,132,1)",
    borderWidth: 2,
    hoverBackgroundColor: "rgba(99,255,132,0.4)",
    hoverBorderColor: "rgba(99,255,132,1)",
    data: [25, 30, 26, 29, 31, 20]
  }
  ]
};
```

Chart.js – JS – Definindo os Dados

A variável **data** contém todas as informações necessárias sobre os dados que serão exibidos no gráfico, incluindo:

- **labels:** identificam as séries de dados. No nosso caso, os nomes dos meses;
- **Datasets:** identificam os dados propriamente ditos. No nosso caso, são as temperaturas para os prédios 1 e 2. Incluindo as legendas para cada um e as cores dos elementos

Chart.js – JS – Definindo as Opções

A seguir, definimos as opções gerais de configuração do gráfico. Veja a explicação a seguir

```
var options = {
  title: {
    display: true,
    text: 'Gráfico das temperaturas no Primeiro semestre/2018',
    fontSize: 18
  },
  maintainAspectRatio: false,
  scales: {
    yAxes: [{
      gridLines: {
        display: true,
        color: "rgba(255,99,132,0.2)"
      },
      ticks: { beginAtZero: true }
    }],
    xAxes: [{
      gridLines: {display: false}
    }]
  }
};
```


Chart.js – JS – Definindo as Opções

Entre as opções definidas pra o gráfico, destacamos:

- **title:** define o título do gráfico e sua aparência;
- **scales:** nesse caso são definidos como serão exibidos os eixos do gráfico. No caso dos gráficos de barra, a opção `'beginAtZero'` definida como `'true'`, força as barras iniciarem a apresentação a partir da ordenada 0 (zero). Nesse Caso, optamos por não exibir as `gridLines` verticais.

Chart.js – JS – Exibindo

Para finalizar, definimos algumas opções gerais das fontes de letras e renderizamos o gráfico com os dados definidos:

```
Chart.defaults.global.defaultFontColor = 'white';  
Chart.defaults.global.defaultFontSize = 14;  
Chart.Bar('chart', {  
  options: options,  
  data: data  
});
```

Chart.js – Populando com Ajax

Para exibir um gráfico com dados vindo do Banco de Dados ou arquivo JSON, crie uma função Ajax para exibir o gráfico:

```
function ajax_chart(chart, url, data) {  
  var data = data || {};  
  $.getJSON(url, data).done(function(response) {  
    chart.data.labels = response.labels;  
    chart.data.datasets[0].data = response.data.quantity;  
    chart.update();  
  });  
}
```

Exemplo de arquivo JSON criado por uma aplicação no servidor:

```
{  
  "labels": ["Jan", "Feb", "Mar", "Apr", "May", "Jun"],  
  "data": {"quantity": [30, 25, 28, 31, 26, 23]}  
}
```

Para gerar o JSON, você pode usar sua linguagem preferida. Em breve disponibilizarei um exemplo completo em PHP.

Conclusão

Os exemplos fornecidos devem lhe trazer algumas ideias sobre o que você pode fazer com a biblioteca Chart.js.

É claro que existem muitas outras opções de customização, mas os tópicos apresentados aqui são suficientes para a construção de seus primeiros gráficos.

Consulte as referências finais para aprofundar-se no assunto.

Abraços e até a próxima

<fim> Obrigado !!! </fim>

Referências

Web

- [Site Oficial do ChartJS](#)
- [Blog de Tobias Bjerrome Ahlin](#)
- [GIST de Ali Mas'ud](#)