

# Chart js

Prof. Me. Hélio Esperidião



# Chart.js

# O que é?

- Chart.js é uma biblioteca JavaScript gratuita e de código aberto para visualização de dados, que oferece suporte a oito tipos de gráficos: barra, linha, área, pizza, bolha, radar, polar e dispersão.
- Material:
  - <https://www.chartjs.org/>
  - <https://github.com/chartjs/Chart.js>

# Exemplo estático

Exemplos sem  
acesso a banco de  
dados ou apis.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Grafico</title>
  <style>
    #divGrafico01 {
      width: 800px;
    }
  </style>
</head>
```

```
<body>  
  <div id="divGrafico01">  
    <canvas id="graficoBarros"></canvas>  
  </div>  
</body>
```

```
<script>
```

```
const ctx = document.getElementById('graficoBarros');
```

```
new Chart(ctx, {
```

```
  type: 'bar',
```

```
  data: {
```

```
    labels: ['Jan', 'Fev', 'Mar', 'Abr', 'mai', 'jun']
```

```
    datasets: [{
```

```
      label: 'Vendas de maçãs',
```

```
      data: [12, 19, 3, 5, 2, 13],
```

```
      borderWidth: 1
```

```
    ]
```

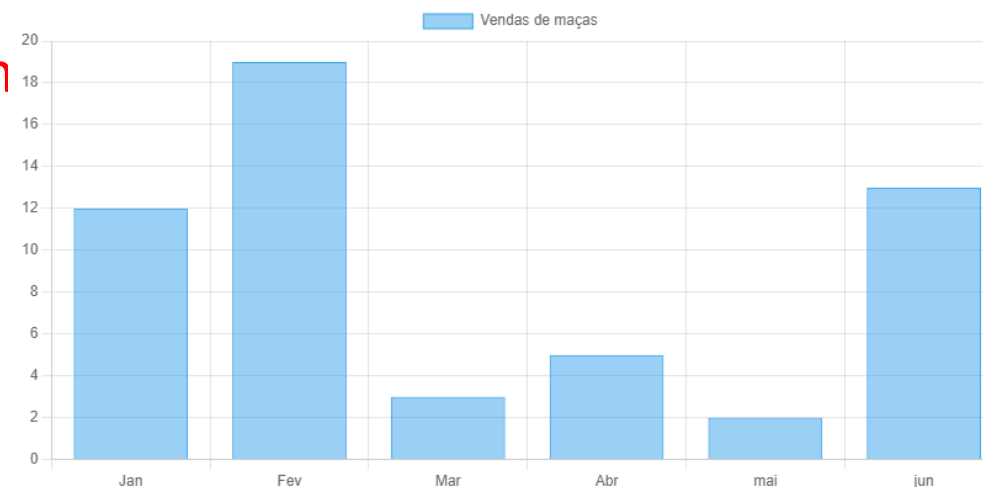
```
  },
```

```
  options: {}
```

```
}
```

```
});
```

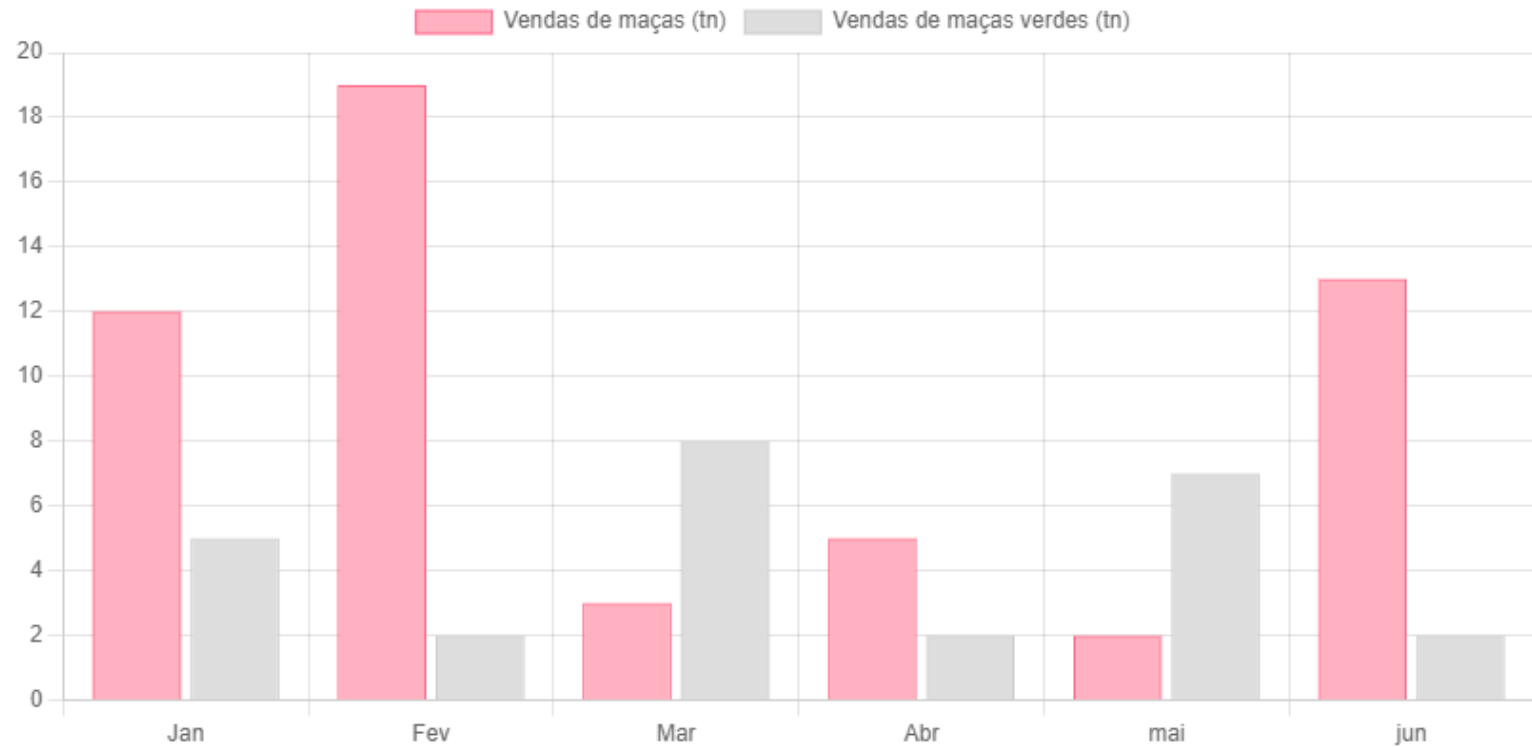
```
</script>
```



```

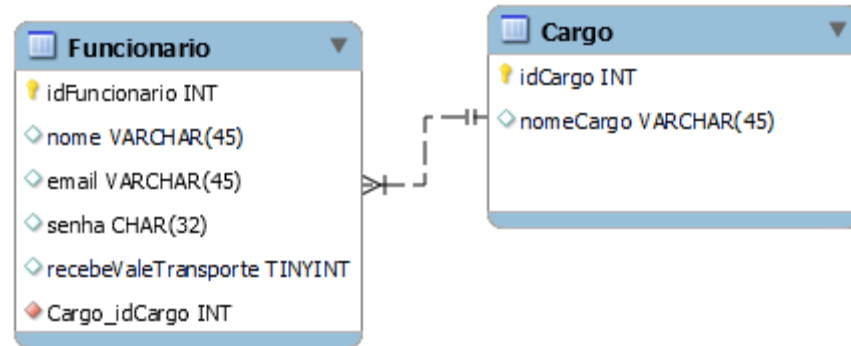
<script>
const ctx = document.getElementById('graficoBarros');
new Chart(ctx, {
  type: 'bar',
  data: {
    labels: ['Jan', 'Fev', 'Mar', 'Abr', 'mai', 'jun'],
    datasets: [
      {
        label: 'Vendas de maçãs (tn)',
        data: [12, 19, 3, 5, 2, 13],
        borderWidth: 1,
        borderColor: '#FF6384',
        backgroundColor: '#FFB1C1',
      },
      {
        label: 'Vendas de maçãs verdes (tn)',
        data: [5, 2, 8, 2, 7, 2],
        borderWidth: 1,
        borderColor: '#CCC',
        backgroundColor: '#DDD',
      }
    ]
  },
  options: { }
});
</script>

```



# Exemplo dinâmico

- O exemplo a seguir recupera um json com dados via fetch e monta um gráfico com os dados.
- No gráfico será apresentada a quantidade de funcionários em cada cargo e a quantidade de funcionários que recebe vale transporte em cada cargo.



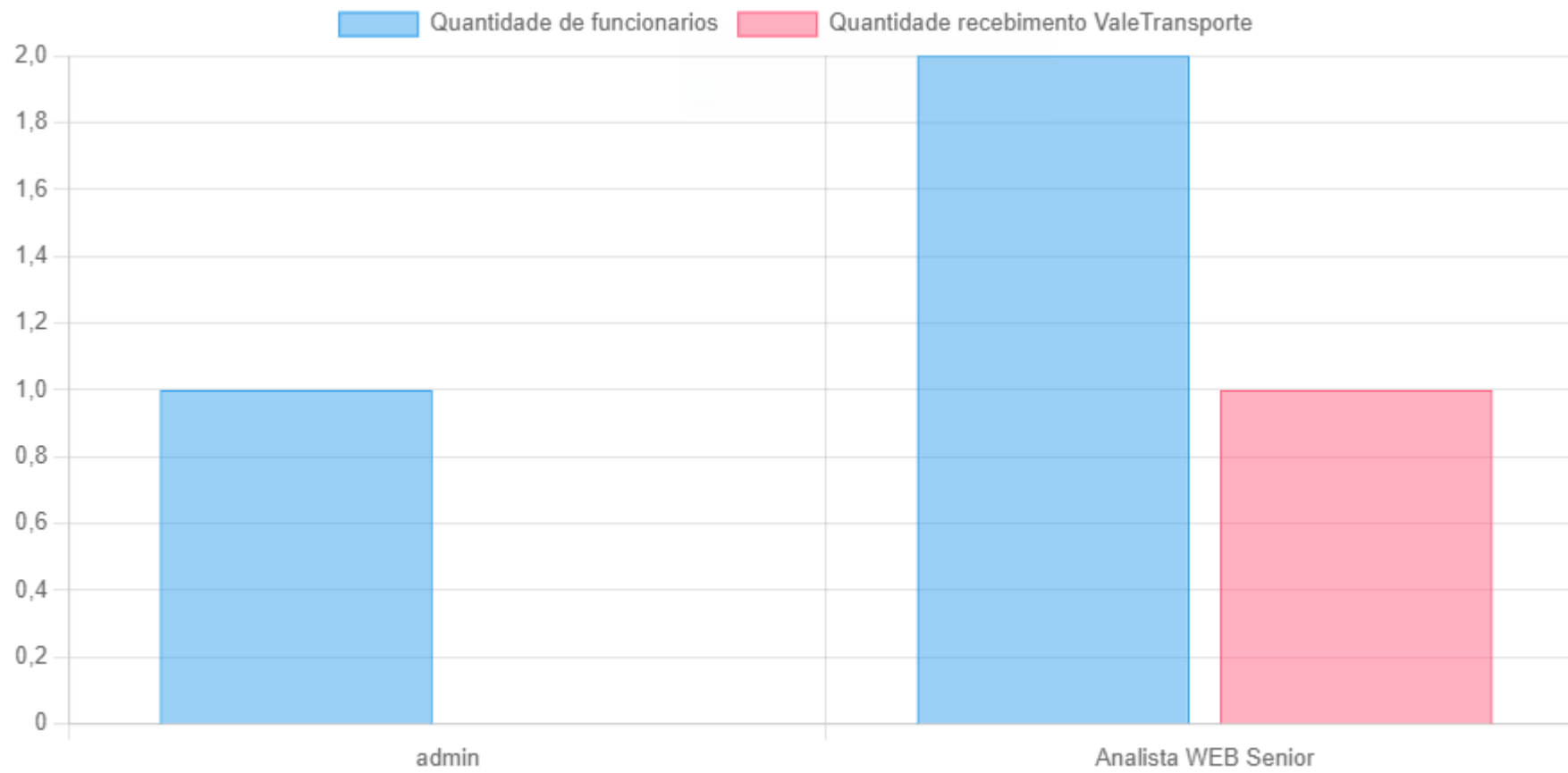


- Os dados podem ser obtidos por meio de uma API php, javascript ou typescript.
- Os dados obtidos são derivados da instrução sql abaixo:
- **SELECT nomeCargo, count(nomecargo) AS qtdFuncionarios, sum(recebeValeTransporte) as valeTransporte FROM funcionario JOIN cargo ON idCargo= Cargo\_idCargo GROUP BY cargo\_idCargo ORDER BY nomeCargo;**

# Formato de json obtido.

```
{
  "status": true,
  "msg": "executado com sucesso",
  "codigo": "201",
  "dados": [
    {
      "nomeCargo": "admin",
      "qtdFuncionarios": 1,
      "valeTransporte": "0"
    },
    {
      "nomeCargo": "Analista WEB Senior",
      "qtdFuncionarios": 2,
      "valeTransporte": "1"
    }
  ]
}
```

# RResultado



# HTML

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Grafico</title>
  <style>
    #divGrafico01 {

      width: 800px;
    }
  </style>
</head>
```

# HTML

```
<body>  
  funcionarios/graficos  
  <div id="divGrafico01">  
    <canvas id="graficoBarros"></canvas>  
  </div>  
</body>  
<script src="/js/chart.js"></script>  
<script src="/js/graficos.js"></script>  
  
</html>
```

# graficos.js

```
fetch_graficos_get();
```

```
let eixoX = [];
```

```
let eixoY_qtdFuncionarios=[]
```

```
let eixoY_qtdValeTransporte=[]
```

```
function fetch_graficos_get() {  
  const URI = "/funcionarios/graficos/";  
  const operacaoAssincrona = fetch(URI, {  
    method: "GET",  
    headers: {  
      'Accept': 'application/json',  
      'Content-Type': 'application/json',  
      //"Authorization": 'Bearer <' + TOKEN + '>',  
    }  
  });  
  operacaoAssincrona.then((response) => {  
    return response.text();  
  }).then((jsonResposta) => {  
    console.log("RECEBIDO:", jsonResposta);  
    const objetoJson = JSON.parse(jsonResposta);  
    montarVetores(objetoJson);  
    gerarGrafico();  
  });  
  operacaoAssincrona.catch((error) => {  
    console.error("Error:", error);  
  });  
}
```

fetch\_graficos\_get();

# montarVetores(objetoJson)

```
function montarVetores(objetoJson){  
    let i=0;  
    for (let objeto of objetoJson.dados) {  
        eixoX[i] = objeto.nomeCargo;  
        eixoY_qtdFuncionarios[i] = objeto.qtdFuncionarios;  
        eixoY_qtdValeTransporte[i] = objeto.valeTransporte;  
        i++;  
    }  
}
```



```
function gerarGrafico() {  
  const ctx = document.getElementById('graficoBarros');  
  new Chart(ctx, {  
    type: 'bar',  
    data: {  
      labels: eixoX,  
      datasets: [  
        {  
          label: 'Quantidade de funcionarios',  
          data: eixoY_qtdFuncionarios,  
          borderWidth: 1,  
        },  
        {  
          label: 'Quantidade recebimento ValeTransporte',  
          data: eixoY_qtdValeTransporte,  
          borderWidth: 1,  
        },  
      ],  
    }, options: {}  
  });  
}
```