

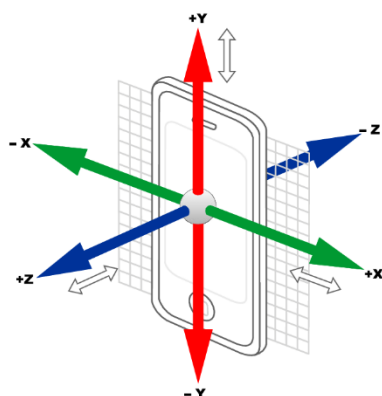
## Contextualização

Um acelerômetro é um dispositivo eletrônico projetado para medir a aceleração de um objeto em relação a um sistema de coordenadas. Amplamente integrado a smartphones, ele é usado para detectar movimentos, inclinações, quedas e mudanças de posição. Sua versatilidade possibilita diversas aplicações, como rotação automática da tela, controle em jogos interativos e detecção de movimentos.

O acelerômetro mede a aceleração ao longo de três eixos principais (Figura 01), que formam um sistema de coordenadas tridimensional:

- Eixo X: Horizontal (sentido frente-trás\*).
- Eixo Y: Vertical (sentido cima-baixo).
- Eixo Z: Horizontal (sentido esquerda-direita).

Por exemplo, ao mover um smartphone, o acelerômetro registra a aceleração em cada direção (X, Y, Z) e gera valores correspondentes. Além disso, ele é capaz de detectar a inclinação do dispositivo em relação ao solo. A aceleração causada pela gravidade (aproximadamente  $9,8 \text{ m/s}^2$ ) também é registrada, permitindo identificar se o smartphone está na posição vertical, deitado ou inclinado.



Orientação dos eixos tridimensionais em um smartphone

Na imagem, observa-se a orientação tridimensional dos eixos utilizada em smartphones. O **eixo X** (em verde) representa o movimento lateral, no sentido esquerda-direita do dispositivo. O **eixo Y** (em vermelho) indica o movimento vertical, de baixo para cima. Já o **eixo Z** (em azul) corresponde ao movimento frontal, apontando da parte interna do aparelho para fora da tela. Essa convenção é padronizada por frameworks e bibliotecas de sensores, garantindo que diferentes dispositivos interpretem os movimentos de forma consistente.

Figura 01 - Acelerômetro em smartphone

## Criar o projeto

- ionic start ExemploAcelerometro
  - Angular
  - blank
  - NgModules

## Abrir o projeto

- cd ExemploAcelerometro
- code .

## Executar os comandos na pasta do projeto

- ionic capacitor add android
- npm install @capacitor/motion
- npm install @capacitor/screen-orientation
- npm install chart.js

## Ajustar as configurações do projeto

- Alterar o arquivo [AndroidManifest.xml](#)
  - [android/app/src/main/AndroidManifest.xml](#)

```

EXPLORADOR
EXEMPLOACELEROMETRO
  .vscode
  android
  app
  build
  src
  androidTest
  main
    java
    res
    AndroidManifest.xml
AndroidManifest.xml U X
android > app > src > main > AndroidManifest.xml
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android">
4    <application
27      <provider
32        <meta-data
34          android:resource="@xml/tile_paths" ></meta-data>
35      </provider>
36    </application>
37
38    <!-- Permissions -->
39    <uses-permission android:name="android.permission.INTERNET" />
40    <uses-permission android:name="android.permission.BODY_SENSORS" />
41  </manifest>
    
```

- Código
 

```
<uses-permission android:name="android.permission.BODY_SENSORS" />
```

- os comandos abaixo devem ser executados na raiz da pasta do projeto:
  - ionic **g page** pages/inicio
  - ionic **g page** pages/sensor
  - ionic **g page** pages/grafico

### Ajustar a estrutura do projeto

- excluir** a pasta "home" e seus arquivos
- alterar o código do arquivo [app-routing.module.ts](#)
  - definir a rota "inicio" como padrão
  - excluir a rota "home"
  - código do arquivo [app-routing.module.ts](#) após as modificações é ilustrado abaixo:

```

EXPLORADOR
EXEMPLOACELEROMETRO
  .angular
  .vscode
  android
  node_modules
  src
    app
      home
      pages
        grafico
        inicio
        TS inicio-routing.module.ts
        TS inicio.module.ts
        inicio.page.html
        inicio.page.scss
        TS inicio.page.spec.ts
        TS inicio.page.ts
      sensor
    TS app-routing.module.ts
    app.component.html
    app.component.scss
    TS app.component.spec.ts
    TS app.component.ts
    TS app.module.ts
  assets
  environments
  theme

src > app > TS app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {
6     path: '',
7     redirectTo: 'inicio',
8     pathMatch: 'full'
9   },
10  {
11    path: 'inicio',
12    loadChildren: () => import('./pages/inicio/inicio.module').then( m => m.InicioPageModule)
13  },
14  {
15    path: 'grafico',
16    loadChildren: () => import('./pages/grafico/grafico.module').then( m => m.GraficoPageModule)
17  },
18  {
19    path: 'sensor',
20    loadChildren: () => import('./pages/sensor/sensor.module').then( m => m.SensorPageModule)
21  },
22 ];
23
24 @NgModule({
25   imports: [
26     RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
27   ],
28   exports: [RouterModule]
29 })
30 export class AppRoutingModule { }
  
```

### Implementar o código da [página inicio](#)

- o código do arquivo [inicio.page.ts](#) após as modificações é ilustrado abaixo:

```

EXPLORADOR
EXEMPLOACELEROMETRO
  .angular
  .vscode
  android
  node_modules
  src
    app
      home
      pages
        grafico
        inicio
        TS inicio-routing.module.ts
        TS inicio.module.ts
        inicio.page.html
        inicio.page.scss
        TS inicio.page.spec.ts
        TS inicio.page.ts
      sensor
    TS app-routing.module.ts
    app.component.html
    app.component.scss
    TS app.component.spec.ts
    TS app.component.ts
    TS app.module.ts
  assets
  environments
  theme

src > app > pages > inicio > TS inicio.page.ts > ...
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-inicio',
5   templateUrl: './inicio.page.html',
6   styleUrls: ['./inicio.page.scss'],
7 })
8 export class InicioPage implements OnInit {
9
10   public appPages = [
11     { title: 'Sensor', url: '/sensor', icon: 'speedometer', color: "dark" },
12     { title: 'Gráfico', url: '/grafico', icon: 'bar-chart', color: "dark" }
13   ];
14
15   constructor() { }
16
17   ngOnInit() {
18   }
19
20 }
  
```

- o código do arquivo [inicio.page.html](#) após as modificações é ilustrado abaixo:

```

1 <ion-header>
2   <ion-toolbar color="dark">
3     <ion-title>Exemplo Acelerômetro </ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <ion-list>
9     <ion-item *ngFor="let page of appPages" [routerLink]="[page.url]" detail>
10      <ion-icon slot="start" [md]="page.icon + '-sharp'" slort="start" [color]="page.color"></ion-icon>
11      <ion-label>{{ page.title }}</ion-label>
12    </ion-item>
13  </ion-list>
14 </ion-content>

```

## Implementar o código da [página sensor](#)

- o código do arquivo [sensor.page.ts](#) após as modificações é ilustrado abaixo:

```

1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { Motion } from '@capacitor/motion';
3 import { ScreenOrientation } from '@capacitor/screen-orientation';
4
5 @Component({
6   selector: 'app-sensor',
7   templateUrl: './sensor.page.html',
8   styleUrls: ['./sensor.page.scss'],
9 })
10 export class SensorPage implements OnInit, OnDestroy{
11
12   x: number = 0;
13   y: number = 0;
14   z: number = 0;
15   gravidade: number = 1; //com gravidade
16
17   constructor() {
18   }
19
20   ngOnInit() {
21     ScreenOrientation.lock({ orientation: 'portrait' });
22
23     Motion.addListener('accel', (event) => {
24       if (this.gravidade == 0) {
25         this.x = event.acceleration.x || 0;
26         this.y = event.acceleration.y || 0;
27         this.z = event.acceleration.z || 0;
28       } else {
29         this.x = event.accelerationIncludingGravity.x || 0;
30         this.y = event.accelerationIncludingGravity.y || 0;
31         this.z = event.accelerationIncludingGravity.z || 0;
32       }
33     });
34   }
35
36   ngOnDestroy(){
37     ScreenOrientation.unlock();
38     Motion.removeAllListeners();
39   }
40 }

```

- Com o uso do `@capacitor/screen-orientation`, é possível bloquear a orientação da tela para o modo *portrait* (retrato) em aplicativos desenvolvidos com Ionic.
- O método `ngOnDestroy`, disponível no framework Angular, é utilizado em componentes ou diretivas para executar ações de limpeza antes que o Angular os destrua.
- event.acceleration**: Para medir movimentos ou forças dinâmicas sem o ruído causado pela gravidade. Por exemplo:
  - Medir a velocidade com que alguém sacode o telefone.
  - Detectar mudanças bruscas no movimento.
- event.accelerationIncludingGravity**: Para determinar a orientação do dispositivo ou medir forças onde a gravidade é relevante. Por exemplo:
  - Detectar se o telefone está inclinado ou de cabeça para baixo.
  - Criar interfaces baseadas na inclinação do dispositivo, como jogos.

- o código do arquivo [sensor.page.html](#) após as modificações é ilustrado abaixo:

```

1 <ion-header>
2   <ion-toolbar color="dark">
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/inicio"></ion-back-button>
5     </ion-buttons>
6     <ion-title>Sensor</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11   <ion-grid>
12     <ion-item>
13       <ion-label class="titulo">Eixos</ion-label>
14     </ion-item>
15     <ion-item>
16       <ion-label position="fixed" class="eixoX">X</ion-label>
17       <ion-label class="eixoX">{{ x }}</ion-label>
18     </ion-item>
19     <ion-item>
20       <ion-label position="fixed" class="eixoY">Y</ion-label>
21       <ion-label class="eixoY">{{ y }}</ion-label>
22     </ion-item>
23     <ion-item>
24       <ion-label position="fixed" class="eixoZ">Z</ion-label>
25       <ion-label class="eixoZ">{{ z }}</ion-label>
26     </ion-item>
27   </ion-grid>
28
29   <ion-grid>
30     <ion-item>
31       <ion-label class="titulo">Efeito da gravidade</ion-label>
32     </ion-item>
33     <ion-item>
34       <ion-segment [(ngModel)]="gravidade" color="dark">
35         <ion-segment-button [value]="0">
36           <ion-label>Não</ion-label>
37         </ion-segment-button>
38         <ion-segment-button [value]="1">
39           <ion-label>Sim</ion-label>
40         </ion-segment-button>
41       </ion-segment>
42     </ion-item>
43   </ion-grid>
44 </ion-content>

```

- o código do arquivo [sensor.page.scss](#) após as modificações é ilustrado abaixo:

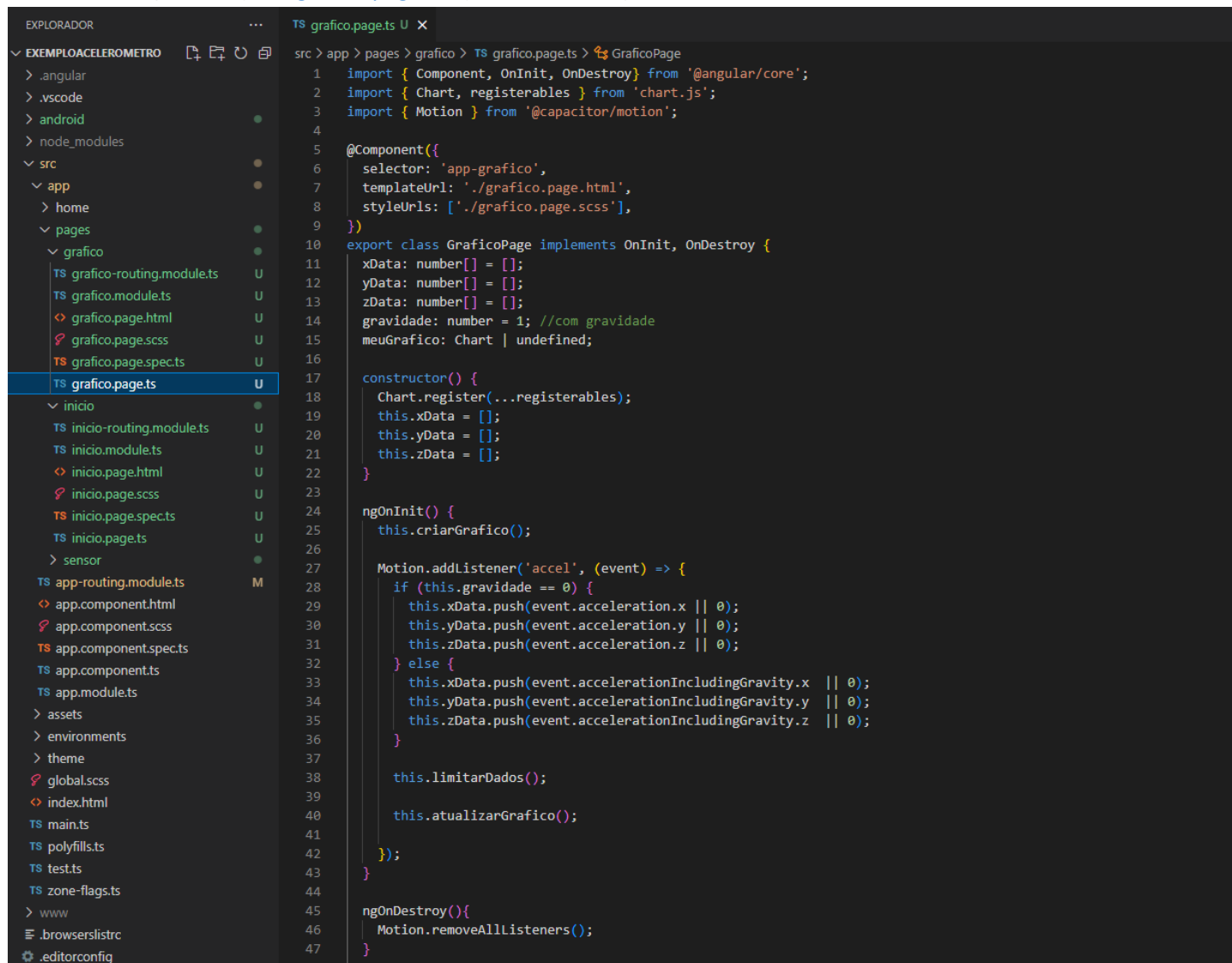
```

1 .eixoX{
2   font-size: x-large;
3   color: red;
4 }
5
6 .eixoY{
7   font-size: x-large ;
8   color: green;
9 }
10
11 .eixoZ{
12   font-size: x-large ;
13   color: blue;
14 }
15
16 .titulo{
17   font-size: x-large;
18   margin-top: 20px;
19 }

```

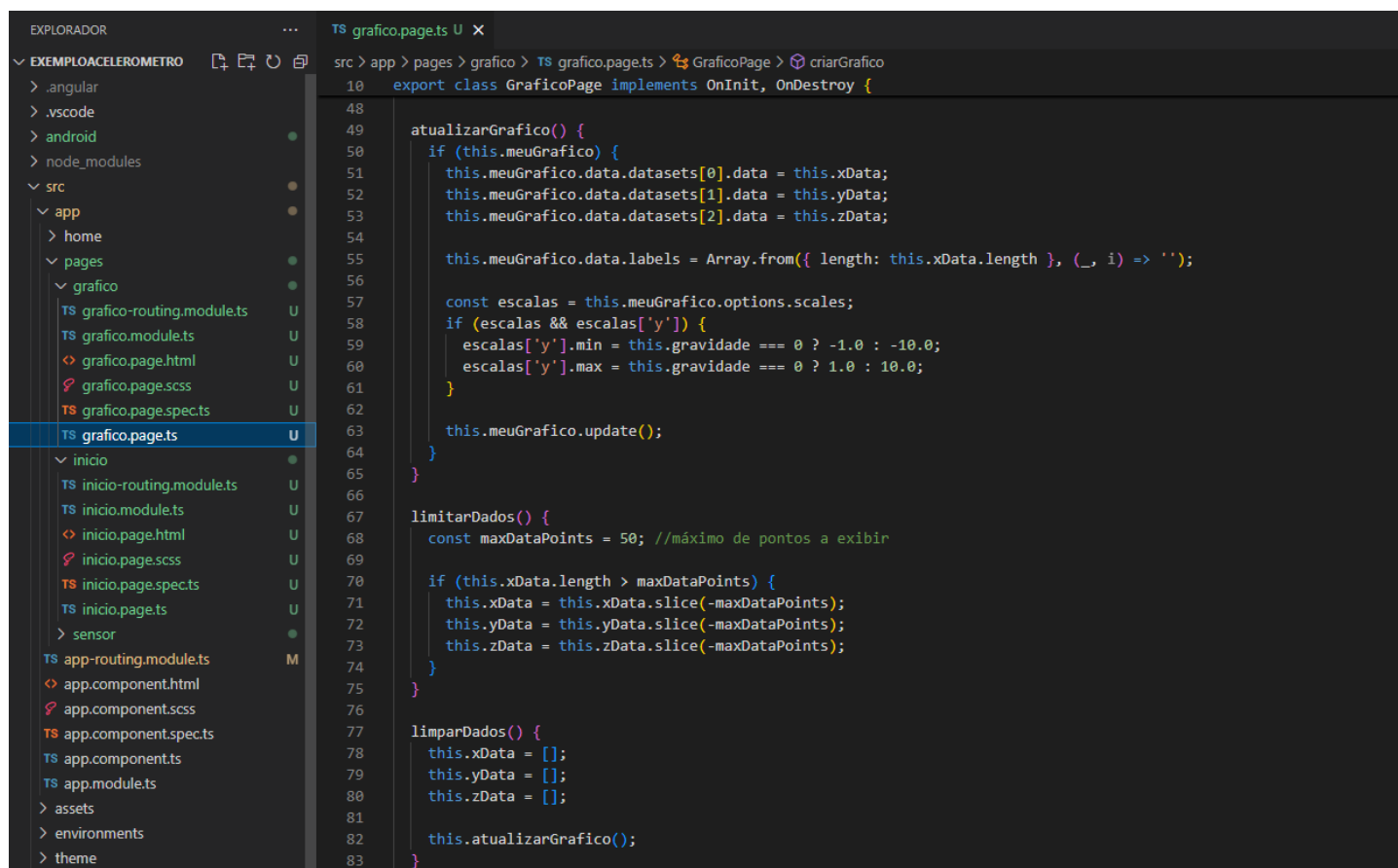
## Implementar o código da [página gráfico](#)

- o código do arquivo [grafico.page.ts](#) após as modificações é ilustrado abaixo:



```
EXPLORADOR  TS grafico.page.ts U X
EXEMPLOACELEROMETRO
  .angular
  .vscode
  android
  node_modules
  src
    app
      home
      pages
        grafico
          TS grafico-routing.module.ts U
          TS grafico.module.ts U
          grafico.page.html U
          grafico.page.scss U
          TS grafico.page.spect.ts U
          TS grafico.page.ts U
        inicio
          TS inicio-routing.module.ts U
          TS inicio.module.ts U
          inicio.page.html U
          inicio.page.scss U
          TS inicio.page.spect.ts U
          TS inicio.page.ts U
        sensor
      TS app-routing.module.ts M
      app.component.html U
      app.component.scss U
      TS app.component.spect.ts U
      TS app.component.ts U
      TS app.module.ts U
    assets
    environments
    theme
    global.scss
    index.html
    TS main.ts
    TS polyfills.ts
    TS test.ts
    TS zone-flags.ts
  www
  .browserlistrc
  .editorconfig

src > app > pages > grafico > TS grafico.page.ts > GraficoPage
1 import { Component, OnInit, OnDestroy } from '@angular/core';
2 import { Chart, registerables } from 'chart.js';
3 import { Motion } from '@capacitor/motion';
4
5 @Component({
6   selector: 'app-grafico',
7   templateUrl: './grafico.page.html',
8   styleUrls: ['./grafico.page.scss'],
9 })
10 export class GraficoPage implements OnInit, OnDestroy {
11   xData: number[] = [];
12   yData: number[] = [];
13   zData: number[] = [];
14   gravidade: number = 1; //com gravidade
15   meuGrafico: Chart | undefined;
16
17   constructor() {
18     Chart.register(...registerables);
19     this.xData = [];
20     this.yData = [];
21     this.zData = [];
22   }
23
24   ngOnInit() {
25     this.criarGrafico();
26
27     Motion.addListener('accel', (event) => {
28       if (this.gravidade == 0) {
29         this.xData.push(event.acceleration.x || 0);
30         this.yData.push(event.acceleration.y || 0);
31         this.zData.push(event.acceleration.z || 0);
32       } else {
33         this.xData.push(event.accelerationIncludingGravity.x || 0);
34         this.yData.push(event.accelerationIncludingGravity.y || 0);
35         this.zData.push(event.accelerationIncludingGravity.z || 0);
36       }
37
38       this.limparDados();
39
40       this.atualizarGrafico();
41     });
42   }
43
44   ngOnDestroy() {
45     Motion.removeAllListeners();
46   }
47
48   criarGrafico() {
49     export class GraficoPage implements OnInit, OnDestroy {
50       atualizarGrafico() {
51         if (this.meuGrafico) {
52           this.meuGrafico.data.datasets[0].data = this.xData;
53           this.meuGrafico.data.datasets[1].data = this.yData;
54           this.meuGrafico.data.datasets[2].data = this.zData;
55
56           this.meuGrafico.data.labels = Array.from({ length: this.xData.length }, (_, i) => '');
57
58           const escalas = this.meuGrafico.options.scales;
59           if (escalas && escalas['y']) {
60             escalas['y'].min = this.gravidade == 0 ? -1.0 : -10.0;
61             escalas['y'].max = this.gravidade == 0 ? 1.0 : 10.0;
62           }
63
64           this.meuGrafico.update();
65         }
66       }
67
68       limparDados() {
69         const maxDataPoints = 50; //máximo de pontos a exibir
70
71         if (this.xData.length > maxDataPoints) {
72           this.xData = this.xData.slice(-maxDataPoints);
73           this.yData = this.yData.slice(-maxDataPoints);
74           this.zData = this.zData.slice(-maxDataPoints);
75         }
76       }
77
78       limparDados() {
79         this.xData = [];
80         this.yData = [];
81         this.zData = [];
82
83         this.atualizarGrafico();
84       }
85     }
86   }
87 }
```



```
EXPLORADOR  TS grafico.page.ts U X
EXEMPLOACELEROMETRO
  .angular
  .vscode
  android
  node_modules
  src
    app
      home
      pages
        grafico
          TS grafico-routing.module.ts U
          TS grafico.module.ts U
          grafico.page.html U
          grafico.page.scss U
          TS grafico.page.spect.ts U
          TS grafico.page.ts U
        inicio
          TS inicio-routing.module.ts U
          TS inicio.module.ts U
          inicio.page.html U
          inicio.page.scss U
          TS inicio.page.spect.ts U
          TS inicio.page.ts U
        sensor
      TS app-routing.module.ts M
      app.component.html U
      app.component.scss U
      TS app.component.spect.ts U
      TS app.component.ts U
      TS app.module.ts U
    assets
    environments
    theme
    global.scss
    index.html
    TS main.ts
    TS polyfills.ts
    TS test.ts
    TS zone-flags.ts
  www
  .browserlistrc
  .editorconfig

src > app > pages > grafico > TS grafico.page.ts > GraficoPage > criarGrafico
10 export class GraficoPage implements OnInit, OnDestroy {
48
49   atualizarGrafico() {
50     if (this.meuGrafico) {
51       this.meuGrafico.data.datasets[0].data = this.xData;
52       this.meuGrafico.data.datasets[1].data = this.yData;
53       this.meuGrafico.data.datasets[2].data = this.zData;
54
55       this.meuGrafico.data.labels = Array.from({ length: this.xData.length }, (_, i) => '');
56
57       const escalas = this.meuGrafico.options.scales;
58       if (escalas && escalas['y']) {
59         escalas['y'].min = this.gravidade == 0 ? -1.0 : -10.0;
60         escalas['y'].max = this.gravidade == 0 ? 1.0 : 10.0;
61       }
62
63       this.meuGrafico.update();
64     }
65   }
66
67   limparDados() {
68     const maxDataPoints = 50; //máximo de pontos a exibir
69
70     if (this.xData.length > maxDataPoints) {
71       this.xData = this.xData.slice(-maxDataPoints);
72       this.yData = this.yData.slice(-maxDataPoints);
73       this.zData = this.zData.slice(-maxDataPoints);
74     }
75   }
76
77   limparDados() {
78     this.xData = [];
79     this.yData = [];
80     this.zData = [];
81
82     this.atualizarGrafico();
83   }
84 }
```

EXPLORADOR

...

TS grafico.page.ts U X

EXEMPLOACELEROMET...

< > ↺ ↻

> .angular

> .vscode

> android

> node\_modules

> src

> app

> home

> pages

> grafico

TS grafico-routing.module.ts U

TS grafico.module.ts U

< > grafico.page.html U

< > grafico.page.scss U

TS grafico.page.spect.ts U

TS grafico.page.ts U

> inicio

TS inicio-routing.module.ts U

TS inicio.module.ts U

< > inicio.page.html U

< > inicio.page.scss U

TS inicio.page.spect.ts U

TS inicio.page.ts U

> sensor

TS app-routing.module.ts M

< > app.component.html

< > app.component.scss

TS app.component.spect.ts

TS app.component.ts

TS app.module.ts

> assets

> environments

> theme

< > global.scss

< > index.html

TS main.ts

TS polyfills.ts

TS test.ts

TS zone-flags.ts

> www

.browserslistrc

.editorconfig

.eslintrc.json

.gitignore

angular.json

capacitor.config.ts

ionic.config.json

karma.conf.js

package-lock.json M

package.json M

tsconfig.app.json

tsconfig.json

src > app > pages > grafico > TS grafico.page.ts > ...

10 export class GraficoPage implements OnInit, OnDestroy {

83 }

84

85 criarGrafico() {

86 const ctx = document.getElementById('meuGrafico') as HTMLCanvasElement;

87

88 // Configuração do gráfico

89 this.meuGrafico = new Chart(ctx, {

90 type: 'line',

91 data: {

92

93 datasets: [

94 {

95 label: 'X',

96 data: this.xData,

97 borderColor: 'rgb(255,0,0)',

98 borderWidth: 1,

99 pointRadius: 0,

100 fill: false,

101 },

102 {

103 label: 'Y',

104 data: this.yData, //Y

105 borderColor: 'rgb(0,255,0)',

106 borderWidth: 1,

107 pointRadius: 0,

108 fill: false,

109 },

110 {

111 label: 'Z',

112 data: this.zData,

113 borderColor: 'rgb(0,0,255)',

114 borderWidth: 1,

115 pointRadius: 0,

116 fill: false,

117 },

118 ]

119 },

120 options: {

121 responsive: true,

122 plugins: {

123 legend: {

124 display: false,

125 },

126 },

127 scales: {

128 x: {

129 display: false,

130 },

131 y: {

132 beginAtZero: true,

133 min: this.gravidade === 0 ? -1.0 : -10.0,

134 max: this.gravidade === 0 ? 1.0 : 10.0,

135 },

136 },

137 },

138 });

139 }

140 }

- o código do arquivo [grafico.page.html](#) após as modificações é ilustrado abaixo:

```

1 <ion-header>
2   <ion-toolbar color="dark">
3     <ion-buttons slot="start">
4       <ion-back-button defaultHref="/inicio"></ion-back-button>
5     </ion-buttons>
6     <ion-title>Gráfico</ion-title>
7   </ion-toolbar>
8 </ion-header>
9
10 <ion-content>
11   <ion-grid>
12     <div class="areaGráfico">
13       <canvas id="meuGráfico"></canvas>
14     </div>
15   </ion-grid>
16
17   <ion-grid>
18     <ion-item>
19       <ion-label class="titulo">Efeito da gravidade</ion-label>
20     </ion-item>
21     <ion-item>
22       <ion-segment [(ngModel)]="gravidade" color="dark">
23         <ion-segment-button [value]="0">
24           <ion-label>Não</ion-label>
25         </ion-segment-button>
26         <ion-segment-button [value]="1">
27           <ion-label>Sim</ion-label>
28         </ion-segment-button>
29       </ion-segment>
30     </ion-item>
31     <ion-button type="button" expand="block" color="dark" (click)="limparDados()">Limpar</ion-button>
32   </ion-grid>
33 </ion-content>

```

- o código do arquivo [grafico.page.scss](#) após as modificações é ilustrado abaixo:

```

1 canvas {
2   width: 100%;
3   height: 300px;
4 }
5

```

## Testar o projeto no celular

- Conecte o celular ao computador utilizando um cabo USB. Em seguida, execute os comandos na pasta do projeto, selecione o dispositivo desejado e aguarde a conclusão da instalação do aplicativo.
  - ionic build
  - ionic capacitor run android

```

D:\2024\SIHS\ExemploAcelerometro>ionic capacitor run android
? Which device would you like to target? (Use arrow keys)
> Xiaomi 22101320G (8d930668)
   Pixel 2 API 30 (emulator) (Pixel_2_API_30)

```

- A figura abaixo ilustra o teste do aplicativo no celular.

