

# Ionic Framework

---

SENAI - Jaguariúna - 3DES - 2021/1

Reenye Lima

Rafael Alves

Wellington Martins

# Aula Passada

● Necessário chave para utilização da API;

● Instalação do plugin de geolocation;

● No arquivo app.module.ts adicionar a o provider “Geolocation”

● Maps estático, adiciona no html o marcador img com src apontando para a API do Google;



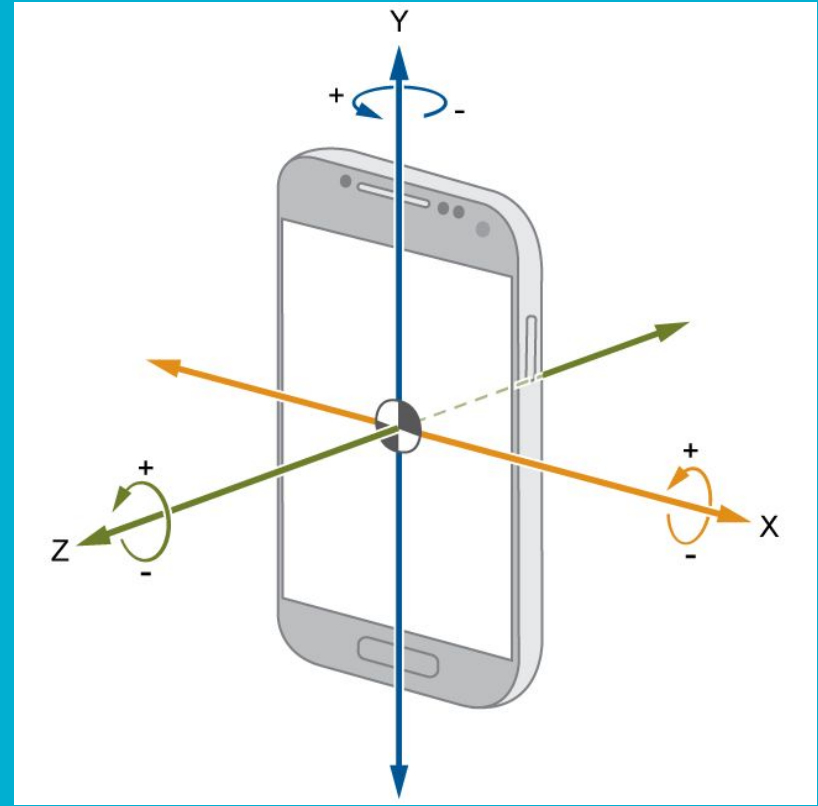
# Aula Passada

---

- Typescript, precisa adicionar script no header;
  - Então criar o objeto: `const map = new google.maps.Map(document.getElementById("map") as HTMLElement,...`
  - Marcador: `const marker = new google.maps.Marker({ position: senai, map: map,...`
  - Para rota consultar o slide anterior
- **Vou passar na sala para consultar a atividade da semana passada.**

# Sensores

- Quais sensores vocês conhecem nos celulares de vocês?
- O que é o acelerômetro?
- O que é o giroscópio?



# Sensores

## — • O que é o acelerômetro?

Controla o movimento baseado nos eixos (x,y,z). Aplicativos de monitores fitness utilizam esse sensor para registrar seus passos.

## • O que é o giroscópio?

Detecta se você girou o aparelho no seu próprio eixo, e se está apontando para cima ou para baixo. Aplicativo de realidade aumentada como Pokémon Go utiliza esse tipo de sensor. (MOSTRAR SIMULADOR Android Studio com sua diferença)



# Criação Projeto

---

Para criação do projeto entre com os comandos no terminal

```
ionic start giroscopio-acelerometro blank
```

```
cd giroscopio-acelerometro
```

# Giroscópio Ionic

---

Para trabalhar com giroscópio no projeto, deve ser adicionado os recursos no projeto como comando:

```
ionic cordova plugin add cordova-plugin-gyroscope (CORDOVA)
```

OU

```
npm i cordova-plugin-gyroscope (CAPACITOR)
```

```
npm install --save @ionic-native/gyroscope
```

```
npx cap sync
```

Documentação do [Giroscópio](#)

# Giroscópio Ionic

---

Deve ser adicionado o giroscópio no arquivo [app.module.ts](#)

```
import { Gyroscope, GyroscopeOrientation, GyroscopeOptions } from  
'@ionic-native/gyroscope/ngx';
```



# Giroscópio Ionic

---

No mesmo arquivo [app.module.ts](#), adicionar o giroscópio no provider

```
providers: [  
  StatusBar,  
  SplashScreen,  
  { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },  
  Gyroscope  
],
```

# Giroscópio Ionic

---

Existem dois métodos para trabalhar com giroscópio, primeiro método `getCurrent()`

- Obtém os dados do giroscópio
- Tem que passar os parâmetros options
- Esses parâmetros são utilizados para configurar o sensor
  - Poder ser setado, por exemplo, a frequência em que será feita a leitura dos dados no sensor



# Giroscópio Ionic

---

No arquivo home.page.ts adicionar:

```
import { Gyroscope,
GyroscopeOrientation,
GyroscopeOptions } from
 '@ionic-native/gyroscope/ngx';
```

```
this.gyroscope.getCurrent()
  .then((orientation: GyroscopeOrientation) => {

console.log(orientation.x,orientation.y,orientation.z,orientation.timestamp);
    this.xOrient=orientation.x;
    this.yOrient=orientation.y;
    this.zOrient=orientation.z;
    this.timestamp=orientation.timestamp;

  })
  .catch()
```

# Giroscópio Ionic

---

O outro método para trabalhar com giroscópio: watch()

- Acompanha as mudanças dos valores do giroscópio:

```
let options: GyroscopeOptions = {  
  frequency: 10  
};
```

```
this.gyroscope.watch(options)  
  .subscribe((orientation: GyroscopeOrientation) => {  
    console.log(orientation.x, orientation.y, orientation.z,  
orientation.timestamp);  
    this.xOrient=orientation.x;  
    this.yOrient=orientation.y;  
    this.zOrient=orientation.z;  
    this.timestamp=orientation.timestamp;  
  });
```

# Giroscópio Ionic

---

- Para simular nosso aplicativo:

ionic capacitor add android

ionic build

ionic capacitor build android

ionic capacitor update android

- Para simulação:
  - Android Studio
    - Configurar [Android Studio](#)
    - Funciona apenas com um celular físico
  - Para qualquer nova alteração no projeto é necessário executar o build do android novamente
  - Para ler as saídas do dispositivo: chrome://inspect

# Acelerômetro Ionic

---

Para trabalhar com acelerômetro, é necessário adicionar o plugin device motion

```
ionic cordova plugin add cordova-plugin-device-motion (cordova)
```

ou

```
npm i cordova-plugin-device-motion (capacitor)
```

```
npm install --save @ionic-native/device-motion
```

```
npx cap sync
```

Para trabalhar com acelerômetro, é necessário adicionar o plugin device motion

# Acelerômetro Ionic

---

Deve ser adicionado device-motion em [app.module.ts](#) como no provider:

```
import { DeviceMotion, DeviceMotionAccelerationData } from '@ionic-native/device-motion';
```

```
providers: [  
  StatusBar,  
  SplashScreen,  
  { provide: RouteReuseStrategy, useClass: IonicRouteStrategy },  
  Gyroscope,  
  DeviceMotion  
],
```

# Acelerômetro Ionic

---

Parecido com o Giroscópio, para obter informações do acelerômetro

```
this.deviceMotion.getCurrentAcceleration().then(  
  (acceleration: DeviceMotionAccelerationData) =>  
    console.log(acceleration),  
  
  // (error: any) => console.log(error)  
  
);
```



# Acelerômetro Ionic

---

Para obter informações do acelerômetro de maneira recorrente:

```
var subscription = this.deviceMotion.watchAcceleration().  
subscribe((acceleration: DeviceMotionAccelerationData) => {  
  console.log(acceleration);  
  this.accX=acceleration.x;  
  this.accY=acceleration.y;  
  this.accZ=acceleration.z;  
});
```

# Fonte

---

<https://simpleactivity435203168.wordpress.com/2018/06/28/gyroscope-and-accelerometer-with-ionic-3/>

Códigos disponível no Git:

<https://github.com/rafaelalvesmartins/ionic>

# Atividades

- 1) Faça um projeto no ionic utilizando blanks. Faça a impressão na tela dos dados dos 3 eixos do acelerômetro e giroscópio com uma data frequência determinado pelo usuário. Exemplo de tela:

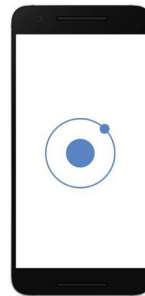
```
<ion-list>
<ion-item>
  <ion-label>Frequência:</ion-label>
  <ion-input type="text"
[(ngModel)]="frequencia"></ion-input>
</ion-item>
<div padding>
  <ion-button (click)="setFrequencia()">Set</ion-button>
</div>
</ion-list>
```

The mockup shows a mobile application interface with two sections. The top section is titled 'Giroscópio' and displays 'X: -1', 'Y: -1', and 'Z: -1'. Below this is a text input field labeled 'Frequência:' and a 'Set' button. The bottom section is titled 'Acelerômetro' and also displays 'X: -1', 'Y: -1', and 'Z: -1'. It has a similar text input field labeled 'Frequência:' and a 'Set' button.

# Próxima Aula

---

- Câmera no ionic



IONIC - white Screen Of Death