

Respostas dos Exercícios

Desenvolvimento de Aplicativos Um Guia Prático para Criar Aplicativos com Ionic

Erik leger Dobrychtop



viena

1ª Edição
Santa Cruz do Rio Pardo/SP
Editora Viena
2018

Capítulo 1

1.

Um aplicativo móvel é, em sua essência, um software ou programa basicamente gerado em um computador para desempenhar funções dentro de outros aparelhos. Assim como foi apresentado no livro, os celulares foram tomando cada vez mais o espaço dentro do mercado de eletrônicos e se popularizando e, assim, também ocorreu posteriormente com os aplicativos.

2.

O sistema operacional é aquele que está ligado diretamente ao hardware, causando maior impacto sobre este, controlando e detectando o mesmo.

3.

Android e iOS.

4.

Ionic é um Framework de desenvolvimento de aplicativos para aparelhos móveis que utiliza o HTML5.

5.

Ionic e Titanium.

6.

Facilidade de aprendizado e velocidade de desenvolvimento.

7.

Garante a disponibilidade de memória adequada para os objetos e estruturas de dados de cada programa em execução, em todos os momentos.

8.

Anúncio publicitário como AdMob.

9.

Uma aplicação gratuita com possibilidades de comprar adicionais.

Capítulo 2

1.

É necessário executar o seguinte comando:
npm i -g ionic

2.

É necessário executar o seguinte comando dentro da pasta do aplicativo:
ionic serve

3.

É necessário executar o seguinte comando:
blank e tabs.

4.

É necessário executar o seguinte comando:
ionic start projeto blank

5.

Utilize o comando ionic start projeto blank e modifique o arquivo home.html.

6.

É necessário executar o seguinte comando:
ionic start projeto tabs

7.

Acessar o arquivo home e modificar o arquivo tabs.html localizado no diretório /src/pages/tabs/.

4 • RESPOSTAS DOS EXERCÍCIOS

8.

É necessário executar o seguinte comando:
`# ionic start projeto sidemenu`

9.

Para editar o menu no tema sidemenu, é necessário o arquivo `app.component.ts` localizado no diretório `src/app/`.

10.

É necessário executar o seguinte comando:
`# ionic start projeto super.`

11.

É necessário executar o seguinte comando:
`# ionic start projeto conference.`

12.

É necessário executar o seguinte comando:
`# ionic start projeto tutorial.`

13.

Botões, card e checkbox.

14.

Light e secondary.

15.

Código HTML:
`<button ion-button>Botão</button>`

Capítulo 3

1.

Os atalhos são <F12> ou <Ctrl+Shift+I>.

2.

As extensões são .ts, .html, scss e ts.

3.

Resultado do método:

```
abrirContato(){  
    this.navCtrl.push(ContatoPage);  
}
```

Capítulo 4

1.

Um Alerta é uma caixa de diálogo que apresenta aos usuários informações, ou recolhe informações dos usuários usando entradas de dados.

2.

Toast é uma notificação que pode ser usada para fornecer comentários sobre uma ação ou para exibir uma mensagem do sistema.

3.

Método do alerta:

```
alertaBasico(){  
    let alertaBasico = this.alertCtrl.create({  
        title: 'Aviso',  
        subTitle: 'Mensagem do aviso',  
        50buttons: ['Ok']  
    });
```

4.

Método do toast:

```
toastBasico(){  
  let toastBasico = this.toastCtrl.create({  
    message: 'Usuário ativou um toast',  
    duration: 3000,  
    position: 'top'  
  });  
  toastBasico.present();  
}
```

5.

Bottom, top e middle.

6.

Modal é uma caixa de diálogo exibida sobre a página atual, ela desliza para fora da tela para exibir um design temporário, usado frequentemente para páginas de login ou de cadastro, composição de mensagens e seleção de opções.

7.

Método:

```
abrirModal(){  
  const modal = this.modalCtrl.create('ModalPage');  
  modal.present();  
}
```

8.

Código do input:

```
<ion-input type="tel"></ion-input>
```

9.

Código do input:

```
<ion-input type="password"></ion-input>
```

10.

Código do input:
`<ion-input type="email"></ion-input>`

11.

Código do input:
`<ion-input type="num"></ion-input>`

12.

O comando irá gerar uma página com o nome Home.

13.

Função:

```
function somar1(x, y) {
  return x + y;
}
```

14.

Função:

```
let somar2 = function(x, y) { return x+y; };
```

Capítulo 5

1.

A navegação básica é tratada por meio do componente `<ion-nav>`, que funciona como uma pilha simples em que as novas páginas são adicionadas com `push` e deslocadas, o que significa avançar e recuar no histórico.

2.

É possível utilizar o `@ViewChild` para obter uma instância do componente `Nav`, que é um controlador de navegação.

3.

Um componente Slides é um contêiner de várias seções. Cada seção pode ser deslizada ou arrastada entre elas.

4.

Navega para um slide específico.

5.

Um dado numérico indicando qual index de slides está.

6.

Retornar o índice do slide anterior.

7.

Menu é uma lista de opções ou comandos apresentados ao usuário a partir de um aplicativo.

8.

Método utilizado:
`this.menuCtrl.open();`

9.

Método utilizado:
`this.menuCtrl.close();`

10.

Método utilizado:
`this.menuCtrl.toggle();`

Capítulo 6

1.

É uma função que se executa quando o compilador angular a encontra no DOM, são responsáveis pelo layout HTML, elas moldam ou remodelam a estrutura do DOM, geralmente adicionando, removendo ou manipulando elementos.

2.

É uma tela de interação de desenvolvimento de aplicativos cross-platform, independente da linguagem que determina um documento HTML, como em uma estrutura de árvores onde cada galho é um objeto que representa uma parte do documento.

3.

ngIf, ngFor e ngSwitch.

4.

Código HTML:

```
<div *ngIf="pessoa" >{{pessoa.nome}}</div>
```

5.

O ngFor é uma diretiva estrutural e integrada, usada para mixar coleções como uma matriz e criar um modelo para cada item, que nos permite criar listas e tabelas de apresentação de dados em nossos modelos HTML.

6.

Código HTML:

```
<li *ngFor="let nome of nomes;">
  {{nome}}
</li>
```

7.

O ngSwitch é uma diretiva utilizada quando o usuário tem múltiplas opções de resultado.

8.

Código HTML:

```
<ul [ngSwitch]="diaDaSemana">  
<li *ngSwitchCase="1">Domingo</li>  
<li *ngSwitchCase="2">Segunda-feira</li>  
<li *ngSwitchCase="3">Terça-feira</li>  
<li *ngSwitchDefault>Está dia não é válido</li>  
</ul>
```

9.

Com a diretiva `ngClass` é possível o desenvolvedor adicionar e remover classes css de um elemento html.

10.

Código HTML:

```
<button ion-button [ngStyle]="{'background-color': 'red'}">Botão</button>
```

11.

Solicitar o retorno do arquivo json.

12.

Comando executado no terminal:

```
# ionic generate provider home
```

Capítulo 7

1.

Comando executado no terminal:

```
# ionic cordova platform add android
```

2.

Comando executado no terminal:

```
# ionic cordova emulate android
```

3.

Comando executado no terminal:
`# ionic cordova platform add ios`

4.

Comando executado no terminal:
`# ionic cordova build ios --prod`

5.

Comando executado no terminal:
`# ionic cordova plugin add cordova-plugin-camera`

6.

Método para abrir câmera:

```
abrirCamera() {
  this.camera.getPicture().then((imageData) => {
    let base64Image = 'data:image/jpeg;base64,' + imageData;
  }, (err) => {
  });
}
```

Capítulo 8

1.

É uma união de arquivos, muitas vezes relacionados com registros de dados, como datas, nomes, números, entre outros tipos. Mais especificamente, é um sistema que permite que os dados sejam facilmente acessados, manipulados e atualizados.

2.

Oracle e MySQL.

3.

Recurso disponibilizado pelo Ionic, e este trabalha de uma forma mais tranquila com sistemas de persistências de dados.

4.

Código utilizado para importar:

```
import { Storage } from '@ionic/storage';
```

5.

Utilizada para realizar algo quando a página estiver totalmente aberta e ativa.

6.

Retorna a identificação se o storage está pronto.

7.

Método para adicionar:

```
adicionar(dado: any) {  
  this.storage.ready().then(() => {  
    this.lista.push(dado);  
    this.storage.set(this.tabela, this.lista);  
  })  
}
```

8.

É quando o usuário realiza o movimento que consiste em tocar na tela de um aparelho celular com o dedo, arrastando a tela para baixo e depois soltando-a, como um sinal para o aplicativo para atualizar o conteúdo da tela.

9.

Código HTML:

```
<ion-refresher (ionRefresh)="atualizaFeed($event)">  
<ion-refresher-content></ion-refresher-content>  
</ion-refresher>
```

Método para atualizar:

```
atualizaFeed(refresher) {
  console.log('começou a operação', refresher);
  setTimeout(() => {
    console.log('a operação acabou');
    refresher.complete();
  }, 2000);
}
```

Capítulo 9

1.

CRUD é uma sigla do inglês (Create, Read, Update and Delete), consistindo em quatro operações básicas utilizadas em banco de dados relacionais, ou seja, Criar, Ler, Atualizar e Excluir.

2.

Método para salvar:

```
salvarProduto(){
  this.storageProvider.adicionar(this.produto);
  this.produto = { nome: '', codigo: '', valor: null };
  this.navCtrl.push(IndexPage);
}
```

3.

Método para visualizar:

```
visualizarProduto(produto) {
  this.navCtrl.push(ViewPage, {
    produto: produto
  });
}
```

4.

Método para atualizar:

```
atualizarProduto() {  
  this.storageProvider.atualizar(this.editandoProduto, this.produto);  
  this.navCtrl.push(IndexPage);  
}
```

5.

Método para deletar:

```
deletarProduto(produto:IProduto) {  
  this.storageProvider.deletar(produto);  
}
```

Capítulo 10

1.

É um módulo que permite que o desenvolvedor possa criar suas próprias animações.

2.

É uma animação utilizada para mover o elemento rapidamente, move-se para a esquerda rapidamente.

3.

É uma animação utilizada para mover o elemento rapidamente, move-se para a direita rapidamente.

4.

Nela o usuário tem a sensação de uma breve explosão de luz brilhante, porém é aplicada opacidade ao elemento para torná-lo translúcido.

5.

Nesta animação você terá a impressão de que o elemento está 3D, como se houvesse uma pulsação.

6.

Método HTML:

```
<ion-card class="animated zoomIn">
<ion-card-header>
Titulo 1
</ion-card-header>
<ion-card-content>Texto 1.</ion-card-content>
</ion-card>
<ion-card class="animated flip">
<ion-card-header>
Titulo 1
</ion-card-header>
<ion-card-content>Texto 1.</ion-card-content>
</ion-card>
```

Capítulo 11

1.

Comandos executados no terminal:

```
ionic cordova plugin add cordova-plugin-camera
npm install --save @ionic-native/camera
```

2.

Método para abrir a câmera:

```
abrirCamera() {
this.camera.getPicture().then((imageData) => {
let base64Image = 'data:image/jpeg;base64,' + imageData;
}, (err) => {
});
}
```

3.

Método para abrir a galeria:

```

abrirGaleria(){
  this.camera.getPicture({
    destinationType: this.camera.DestinationType.DATA_URL,
    targetWidth: 1000,
    targetHeight: 1000,
    sourceType: this.camera.PictureSourceType.SAVEDPHOTOALBUM
  }).then((imageData) => {
    this.base64Image = 'data:image/jpeg;base64,' + imageData;
  }, (err) => {
  });
}

```

4.

Streaming é basicamente um modo de transmissão de áudio e vídeo, e é realizado através da rede, para essa situação não é necessário realizar transferência, mas somente acesso online.

5.

Método para mostrar o áudio:

```

mostrarAudio() {
  let options: StreamingAudioOptions = {
    successCallback: () => { console.log('Vídeo finalizado') },
    errorCallback: (e) => { console.log('Erro: ', e) },
    initFullscreen: false
  };
  this.streamingMedia.playAudio('http://soundbible.com/grab.php?id=2196&type=mp3', options);
}

```


6.

Método para mostrar o vídeo:

```

mostrarVideo() {
  let options: StreamingVideoOptions = {
    successCallback: () => { console.log('Vídeo finalizado') },
    errorCallback: (e) => { console.log('Erro: ', e) },
    orientation: 'portrait'
  };
  this.streamingMedia.playVideo('http://www.sample-
  videos.com/video/mp4/360/big_buck_bunny_360p_1mb.mp4', options);
}

```

Capítulo 12

1.

HTTP significa protocolo de transferência HyperText. É um protocolo utilizado para comunicação, utilizado pelos clientes e servidores.

2.

GET e POST.

3.

É aplicado para submeter uma entidade a um recurso específico, as vezes provocando uma alteração no estado do recurso ou solicitando modificações no lado do servidor.

4.

REST é uma abreviação para Representational State Transfer.

5.

Retorno do tipo numérico e retorno do número de status da resposta.

6.

O atributo `data.data` tem um retorno de `any`, podendo ser qualquer tipo, e retornará os dados que estão na resposta.

7.

Retorna os cabeçalhos da resposta.

8.

Código:

```

this.http.get('https://api.hgbrasil.com/weather/',
{format: 'json',woeid: '455822',locale: 'pt'}, {}).then(data => {
  console.log(data.status);
  console.log(data.data);
  console.log(data.headers);
  this.previsao = JSON.parse(data.data);
}).catch(error => {
  console.log(error.status);
  console.log(error.error);
  console.log(error.headers);
});

```

Capítulo 13

1.

A autenticação é um processo no qual o usuário fornece suas credenciais, que serão comparadas em um banco de dados de informações de usuários autorizados, ou em um servidor de autenticação para obter acesso a determinada aplicação.

2.

Método web:

```

loginFacebook(){
  let provider = new firebase.auth.FacebookAuthProvider();
  146firebase.auth().signInWithRedirect(provider).then(() => {
  firebase.auth().getRedirectResult().then((result) => {

```

```

alert(JSON.stringify(result));
}).catch(function(error) {
alert(JSON.stringify(error));
});
})
}

```

3.

Método nativo:

```

loginFacebook() {
this.facebook.login(["email"]).then((loginResponse) => {
148let credencial =
firebase.auth.FacebookAuthProvider.credential(loginResponse.authResponse.
accessToken);
firebase.auth().signInWithCredential(credencial).then((info) => {
alert(JSON.stringify(info));
})
});
}

```

4.

Método:

```

constructor(public navCtrl: NavController, public navParams: NavParams,
private ofAuth:
AngularFireAuth) {
}

```

E no mesmo arquivo, deve ser desenvolvido o método registrar, é fácil verificar que trata-se um método assíncrono, pois possui o termo `async` antes do nome. Observe como ficará o método a seguir:

```

async registrar(usuario: Usuario){
try{
const result = await this.ofAuth.auth.createUserWithEmailAndPassword(usuar
io.email,
usuario.senha);
console.log(result);
}
catch(e){

```

```

console.error(e);
}
}

```

Capítulo 14

1.

Push é uma notificação em forma de mensagem que é enviada diretamente para o dispositivo móvel do usuário, um ponto interessante sobre a mesma é que o desenvolvedor pode selecionar usuários específicos ou então enviar a mensagem para todos.

2.

Comando executado no terminal:

```
# ionic cordova plugin add phonegap-plugin-push
```

3.

Método:

```

constructor(public navCtrl: NavController, private push: Push) {
  this.push.hasPermission()
    .then((res: any) => {
      if (res.isEnabled) {
        alert('Aplicação com permissão');
        160const options: PushOptions = {
          android: {},
          ios: {
            alert: 'true',
            badge: true,
            sound: 'false'
          },
          windows: {},
          browser: {
            pushServiceURL: 'http://push.api.phonegap.com/v1/push'
          }
        };
        const pushObject: PushObject = this.push.init(options);

```

```

pushObject.on('notification').subscribe((notification: any) => {
  alert(notification.message);
});
pushObject.on('registration').subscribe((registration: any) => {
});
pushObject.on('error').subscribe(error => {
});
} else {
  alert('Aplicação sem permissão');
}
});
}

```

Capítulo 15

1.

Depuração é o processo de encontrar defeitos na aplicação.

2.

Para facilitar a identificação e visualização do erro é utilizada a depuração, assim diminuindo as chances uma aplicação ir com erros para a produção.

3.

Comando executado no terminal:

```
# ionic cordova run android --device
```

Capítulo 16

1.

Comando para executar no terminal:

```
# ionic generate page nomePagina
```

2.

Código utilizado para importar o Alert:

```
import { AlertController } from 'ionic-angular';
```

3.

Código utilizado:

```
this.produto = this.navParams.get('produto');
```

4.

Método:

```
funcaoProduto(produto){  
  this.navCtrl.push(ProdutoPage, {  
    produto: produto  
  });  
}
```

Capítulo 17

1.

APK significa Android Package, o que é uma forma de compactação para a plataforma Android.

2.

Comando executado no terminal:

```
# ionic cordova build ios --prod
```

3.

```
app.module.ts
```

4.

Comando executado no terminal:

```
# ionic cordova build android --prod --release
```