

TUTORIAL IONIC

PRELIMINARES

CRIANDO O APP

Utilizando o PowerShell ou outro terminal de comandos, siga os passos:

1. Cria diretório pai

```
> md lab01
```

2. Entra no diretório pai

```
> cd .\lab01\
```

3. Utiliza o Ionic-cli (implementação em NodeJS) para criar um app baseado num dos templates. Após este passo, aparecerá um menu com opções de templates (utilizaremos o SideMenu)

```
> ionic start agenda-app
```

4. Vai para a pasta do App.

```
> cd agenda-app
```

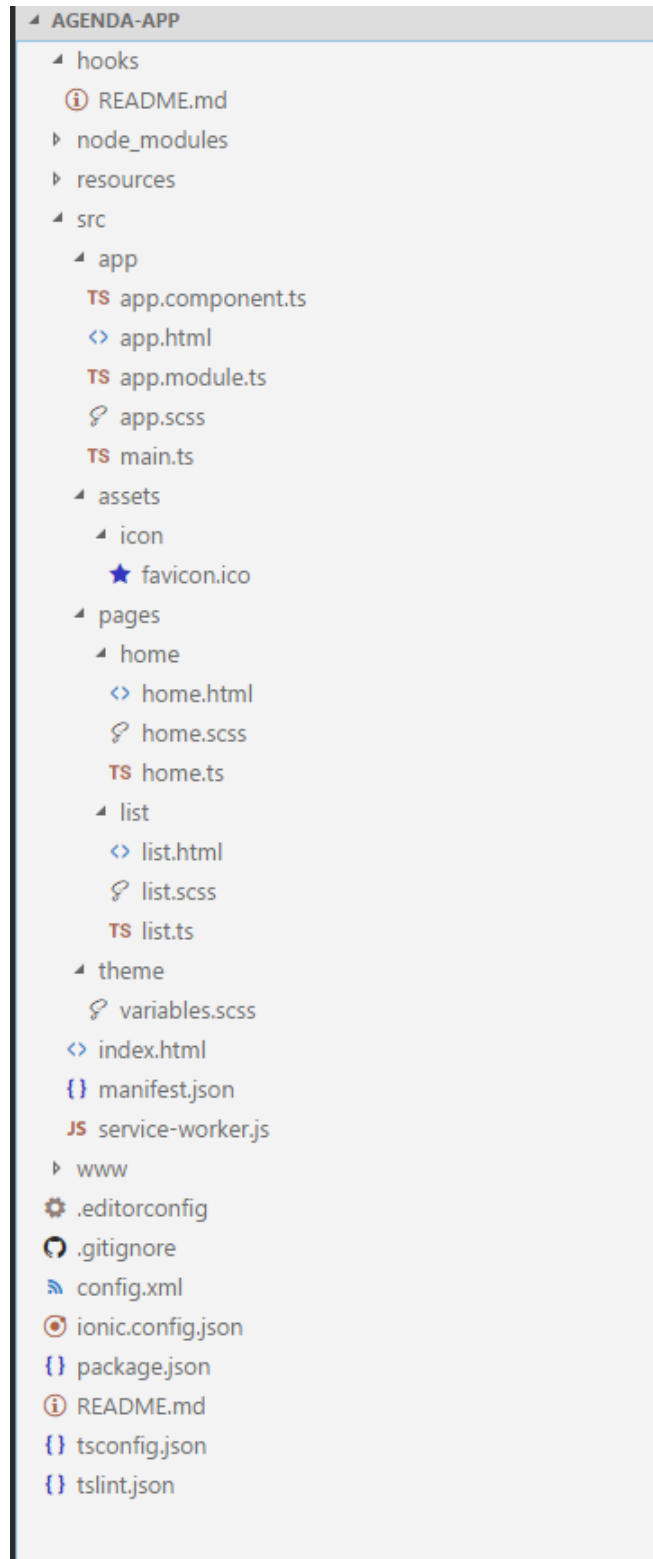
5. Roda o app no Chrome

```
> ionic serve
```

ENTENDENDO A ESTRUTURA DO APP

Do Visual Studio Code ou outra ferramenta de desenvolvimento

1. Abrir a pasta `..\lab01\agenda-app`



/HOOKS

Ganchos (Hooks) Cordova são scripts especiais que podem ser adicionados por desenvolvedores de aplicativos e plugins ou mesmo pelo seu próprio sistema de compilação para personalizar comandos cordova. Os ganchos Cordova permitem que você realize atividades especiais em torno de comandos cordova. Por exemplo, você pode ter uma ferramenta personalizada que verifica a formatação do código em seu arquivo javascript. E, você gostaria de executar esta ferramenta antes de cada compilação. Nesse caso, você poderia usar um gancho 'before_build' e instruir o run-time do cordova a executar a ferramenta personalizada a ser invocada antes de cada compilação.

Fonte: <https://cordova.apache.org/docs/en/latest/guide/appdev/hooks/>

/NODE_MODULES

Pasta onde serão armazenados todos os pacotes NodeJS necessários para "compilar" e executar a aplicação. Normalmente esta pasta é gerenciada pelo NPM, utilitário gerenciador de pacotes do NodeJS.

/RESOURCES

Pasta que armazena os recursos (imagens, ícones, etc.) específicos de cada plataforma. Nesta pasta ficam, por exemplo os Splashes de cada plataforma, com suas diversas dimensões.

/SRC

Pasta do código fonte do projeto

/SRC/APP

Pasta que contém o fonte do app. Este é o módulo principal da aplicação, que vai ser o startup e configurações iniciais.

/SRC/ASSETS

Recursos da aplicação que independem da plataforma. Por exemplo uma imagem de logo que aparecerá na página inicial.

/SRC/THEME

Arquivos css e scss. Aqui podem ser adicionados arquivos para modificação de css e scss. Aqui, por exemplo podem ser colocadas cores e comportamentos específicos de componentes.

/SRC/*

Outras subpastas podem ser adicionadas à pasta src. Por exemplo, provedores, serviços e modelos de dados.

/WWW

Pasta onde é feito o "Deploy" da aplicação. Aqui os arquivos são codificados para serem executados diretamente num navegador.

ARQUIVOS ESPECIAIS E IMPORTANTES

/SRC/INDEX.HTML

Arquivo que será inicializado no browser do Cordova dentro do app ou desktop

/SRC/MANIFEST.JSON

Arquivo de manifesto do Cordova. Usado para declarar componentes externos ou que necessitem de permissão especial para rodar, como, por exemplo os Plugins.

/CONFIG.XML

Arquivo de configurações do Cordova, define onde localizar os elementos como os Splashes e ícones.

/IONIC.CONFIG.JSON

Configurações do Ionic, como, por exemplo o nome do App.

ENTENDENDO A ESTRUTURA DAS PÁGINAS

Cada página é composta de 3 arquivos:

HTML -> Estrutura física visual da página

TS -> Arquivo Typescript (pode ser js (JavaScript) também) que contém o chamado code behind, ou o código que será vinculado à página, contendo as regras e escopo específicos da mesma.

SCSS -> Arquivo para personalização visual da página especificamente

MODULE.TS -> Arquivo opcional de módulo (padrão de projeto), com o objetivo de especificar objetos a serem injetados na classe da página, além de conter declarações e exportação de classes para outros módulos.

ENTENDENDO O APP

A pasta `src/app` tem o código de inicialização do App, contendo o `app.html` com o layout principal da aplicação, `app.component.ts` seu código fonte, `app.module.ts` contendo as classes a serem injetadas, `app.scss` com a personalização visual do app e o `main.ts`, sendo este último o arquivo bootstrapper ou ponto de entrada inicial da aplicação.

Toda a lógica do Menu Principal está neste componente.

CONSTRUINDO O APLICATIVO

PERSONALIZANDO A PÁGINA HOME

Criar uma pasta dentro de `src/assets` chamada `image`

Copiar a imagem `"gif_mobilus.gif"` em

<https://drive.google.com/open?id=0B2Lh6LBs1BqkcElwUERNZ20xQ3c> para a pasta `src/assets/image`

Alterar a página `src/pages/home/home.html`, trocando o título e editando o conteúdo, conforme imagem abaixo:

```
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Agenda</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding text-center>
  <h3>Minha Agenda</h3>
  
</ion-content>

<ion-footer text-center>
  Powered by Mobilus Tecnologia
</ion-footer>
```

CRIANDO UMA NOVA PÁGINA

Abrir o console na raiz do projeto, digitar:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
PS C:\henko\lab01\agenda-app> ionic generate page contatos
```

O comando criará uma nova página vazia.

CHAMANDO A NOVA PÁGINA NO MENU

Modifique o arquivo `src/app/app.component.ts`, para inserir a chamada à nova página no Menu, conforme imagem abaixo:

```
import { Component, ViewChild } from '@angular/core';
import { Nav, Platform } from 'ionic-angular';
import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { ContatosPage } from '../pages/contatos/contatos';

@Component({
  templateUrl: 'app.html'
})
export class MyApp {
  @ViewChild(Nav) nav: Nav;

  rootPage: any = HomePage;

  pages: Array<{title: string, component: any}>;

  constructor(public platform: Platform, public statusBar: StatusBar, public splashScreen: SplashScreen) {
    this.initializeApp();

    // used for an example of ngFor and navigation
    this.pages = [
      { title: 'Home', component: HomePage },
      { title: 'List', component: ListPage },
      { title: 'Contatos', component: ContatosPage }
    ];
  }

  initializeApp() {
    this.platform.ready().then(() => {
      // Okay, so the platform is ready and our plugins are available.
      // Here you can do any higher level native things you might need.
      this.statusBar.styleDefault();
      this.splashScreen.hide();
    });
  }

  openPage(page) {
    // Reset the content nav to have just this page
    // we wouldn't want the back button to show in this scenario
    this.nav.setRoot(page.component);
  }
}
```

E o arquivo `src/app/app.module.ts` para inserir o componente que será injetado quando o objeto for solicitado.

```

import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { ContatosPage } from '../pages/contatos/contatos';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler}
  ]
})
export class AppModule {}

```

PERSONALIZANDO A PÁGINA DE CONTATOS

Altere a página do contato para colocar uma barra de busca, conforme imagem:

```

<!--
  Generated template for the ContatosPage page.

  See http://ionicframework.com/docs/components/#navigation for more info on
  Ionic pages and navigation.
-->
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Contatos</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <ion-searchbar placeholder="Nome do contato"></ion-searchbar>
</ion-content>

```

CRIANDO UM PROVIDER DE CONTATOS

Providers são classes que prestam serviços aos componentes Angular/Ionic. Para criar um novo provider, digite na janela do terminal:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
PS C:\henko\lab01\agenda-app> ionic generate provider contatos
```

O comando irá criar a pasta src/providers/contatos, o arquivo src/providers/contatos/contatos.ts e modificar o arquivo src/app/app.module.ts, para incluir uma referência ao objeto como provider, conforme imagem abaixo:

```
import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { ContatosPage } from '../pages/contatos/contatos';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { ContatosProvider } from '../providers/contatos/contatos';

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler},
    ContatosProvider
  ]
})
export class AppModule {}
```

Altere o arquivo src/providers/contatos/contatos.ts conforme imagem abaixo:


```

import { Injectable } from '@angular/core';
import 'rxjs/add/operator/map';

/**
 * Generated class for the ContatosProvider provider.
 *
 * See https://angular.io/docs/ts/latest/guide/dependency-injection.html
 * for more info on providers and Angular DI.
 */
@Injectable()
export class ContatosProvider {
  pessoas: Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> = [];

  constructor() {
    console.log('Hello ContatosProvider Provider');
  }

  addContato(nome, endereco, cep, lat, long, email) {
    this.pessoas.push({nome, endereco, cep, lat, long, email});
  }

  getContatos(): Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> {
    return this.pessoas;
  }
}

```

Altere o arquivo `/src/pages/contatos/contatos.ts`, para fazer referência ao provider, conforme imagem abaixo:

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../../providers/contatos/contatos';

/**
 * Generated class for the ContatosPage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */
@Component({
  selector: 'page-contatos',
  templateUrl: 'contatos.html',
})
export class ContatosPage {
  contatos: Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> = [];

  constructor(public navCtrl: NavController, public navParams: NavParams, private contatosProvider: ContatosProvider) {
    this.contatos = contatosProvider.getContatos();
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatosPage');
  }
}

```

RETORNANDO UM ITEM SE A LISTA ESTIVER VAZIA

Caso a lista esteja vazia, apresenta um item para teste e posicionamento. Em um cenário real, esta implementação não seria desta forma.

Altere o arquivo `src/providers/contatos/contatos.ts`, conforme imagem:

```

import { Injectable } from '@angular/core';
import 'rxjs/add/operator/map';

/**
 * Generated class for the ContatosProvider provider.
 *
 * See https://angular.io/docs/ts/latest/guide/dependency-injection.html
 * for more info on providers and Angular DI.
 */
@Injectable()
export class ContatosProvider {
  pessoas: Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> = [];

  constructor() {
    this.pessoas.push({
      nome: 'Alexandre Montanha', endereco: 'Av. Raja Gabáglia, 4343 - Santa Lucia, Belo Horizonte - MG', cep: '30350-577',
      lat: -19.9679803, long: -43.9540716, email: 'alexmontanha@mobilus.com.br'})
  }

  addContato(nome, endereco, cep, lat, long, email) {
    this.pessoas.push({nome, endereco, cep, lat, long, email});
  }

  getContatos(): Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> {
    return this.pessoas;
  }
}

```

BOTÃO PARA ADICIONAR NOVO CONTATO

Alterar o arquivo `src/pages/contatos/contatos.html` conforme imagem abaixo:

```

<!--
  Generated template for the ContatosPage page.

  See http://ionicframework.com/docs/components/#navigation for more info on
  Ionic pages and navigation.
-->

<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu"></ion-icon>
    </button>
    <ion-title>Contatos</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <ion-searchbar placeholder="Nome do contato"></ion-searchbar>
  <ion-list>
    <ion-item *ngFor="let contato of contatos">
      {{ contato.nome }}
    </ion-item>
  </ion-list>
</ion-content>

<ion-footer>
  <button ion-button color="light" (click)="addContato()">
    <ion-icon name="add-circle"></ion-icon>
  </button>
</ion-footer>

```

Alterar o arquivo `src/pages/contatos/contatos.ts` para fazer a chamada do botão de novo contato, conforme imagem abaixo:

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../providers/contatos/contatos';

/**
 * Generated class for the ContatosPage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */

@Component({
  selector: 'page-contatos',
  templateUrl: 'contatos.html',
})
export class ContatosPage {
  contatos: Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> = [];

  constructor(public navCtrl: NavController, public navParams: NavParams, private contatosProvider: ContatosProvider) {
    this.contatos = contatosProvider.getContatos();
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatosPage');
  }

  addContato() {
    alert('Aqui vai adicionar contato');
  }
}

```

CRIANDO UMA PÁGINA DE EDIÇÃO/ADIÇÃO DE CONTATOS

Criar uma página para editar e adicionar detalhes do contato.

No terminal, digite o seguinte comando:

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

PS C:\henko\lab01\agenda-app> ionic generate page contato-detalle

```

CHAMANDO A PÁGINA DETALHE

Altere o arquivo src/app/app.module.ts para incluir a nova página, conforme imagem abaixo:

```

import { BrowserModule } from '@angular/platform-browser';
import { ErrorHandler, NgModule } from '@angular/core';
import { IonicApp, IonicErrorHandler, IonicModule } from 'ionic-angular';

import { MyApp } from './app.component';
import { HomePage } from '../pages/home/home';
import { ListPage } from '../pages/list/list';
import { ContatosPage } from '../pages/contatos/contatos';

import { StatusBar } from '@ionic-native/status-bar';
import { SplashScreen } from '@ionic-native/splash-screen';
import { ContatosProvider } from '../providers/contatos/contatos';
import { ContatoDetalhePage } from '../pages/contato-detalhe/contato-detalhe';

@NgModule({
  declarations: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage,
    ContatoDetalhePage
  ],
  imports: [
    BrowserModule,
    IonicModule.forRoot(MyApp),
  ],
  bootstrap: [IonicApp],
  entryComponents: [
    MyApp,
    HomePage,
    ListPage,
    ContatosPage,
    ContatoDetalhePage
  ],
  providers: [
    StatusBar,
    SplashScreen,
    {provide: ErrorHandler, useClass: IonicErrorHandler},
    ContatosProvider
  ]
})
export class AppModule {}

```

Alterar o arquivo src/pages/contatos/contatos.ts para contemplar a chamada à nova página

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../providers/contatos/contatos';
import { ContatoDetalhePage } from '../pages/contato-detalhe/contato-detalhe';

/**
 * Generated class for the ContatosPage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */

@Component({
  selector: 'page-contatos',
  templateUrl: 'contatos.html',
})
export class ContatosPage {
  contatos: Array<{nome: string, endereco: string, cep: string, lat: number, long: number, email: string}> = [];

  constructor(public navCtrl: NavController, public navParams: NavParams, private contatosProvider: ContatosProvider) {
    this.contatos = contatosProvider.getContatos();
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatosPage');
  }

  addContato() {
    this.navCtrl.push(ContatoDetalhePage);
  }
}

```

PERSONALIZANDO A PÁGINA DE DETALHE DE CONTATOS

A página detalhe deve ser preparada para incluir os dados de um contato, chamando, inclusive o provider de contatos.

Altere o arquivo src/pages/contato-detalhe/contato-detalhe.html, como a imagem abaixo:

```

<!--
  Generated template for the ContatoDetalhePage page.

  See http://ionicframework.com/docs/components/#navigation for more info on
  Ionic pages and navigation.
-->
<ion-header>
  <ion-navbar>
    <ion-title>contato-detalhe</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <form>
    <ion-item>
      <ion-label>e-mail</ion-label>
      <ion-input type="text" [(ngModel)]="contato.email" name="email"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>Nome</ion-label>
      <ion-input type="text" [(ngModel)]="contato.nome" name="nome"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>CEP</ion-label>
      <ion-input type="text" [(ngModel)]="contato.cep" name="cep"></ion-input>
    </ion-item>
    <ion-item>
      <ion-label>Endereço</ion-label>
      <ion-input type="text" [(ngModel)]="contato.endereco" name="endereco"></ion-input>
    </ion-item>
    <button ion-button color="secondary" block (click)="confirmaContato()">Confirma</button>
    <button ion-button color="danger" block (click)="cancelaContato()">Cancela</button>
  </form>
</ion-content>

```

Inserir na página `src/pages/contato-detalhe/contato-detalhe.ts` os códigos para tratamento das tarefas de confirmação e cancelamento do cadastro. Conforme imagem abaixo:

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../../providers/contatos/contatos';

/**
 * Generated class for the ContatoDetalhePage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */

@Component({
  selector: 'page-contato-detalhe',
  templateUrl: 'contato-detalhe.html',
})
export class ContatoDetalhePage {
  contato = { nome: '', endereco: '', cep: '', lat: 0, long: 0, email: '' };

  constructor(public navCtrl: NavController, public navParams: NavParams, private contatosProvider: ContatosProvider) {
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatoDetalhePage');
  }

  confirmaContato() {
    const c = this.contato;
    this.contatosProvider.addContato(c.nome, c.endereco, c.cep, c.lat, c.long, c.email);
    this.navCtrl.pop();
  }

  cancelaContato() {
    this.navCtrl.pop();
  }
}

```

EDITANDO CONTATOS

Acrescentar suporte à uma chamada para editar os contatos.

Alterando o arquivo `src/pages/contatos/contatos/html`, conforme imagem abaixo:

```

<!--
Generated template for the ContatosPage page.

See http://ionicframework.com/docs/components/#navigation for more info on
Ionic pages and navigation.
-->
<ion-header>
  <ion-navbar>
    <button ion-button menuToggle>
      <ion-icon name="menu">/ion-icon</ion-icon>
    </button>
    <ion-title>Contatos</ion-title>
  </ion-navbar>
</ion-header>

<ion-content padding>
  <ion-searchbar placeholder="Nome do contato">/ion-searchbar</ion-searchbar>
  <ion-list>
    <ion-item *ngFor="let contato of contatos" (click)="editContato(contato)">
      {{ contato.nome }}
    </ion-item>
  </ion-list>
</ion-content>

<ion-footer>
  <button ion-button color="light" (click)="addContato()">
    <ion-icon name="add-circle">/ion-icon</ion-icon>
  </button>
</ion-footer>

```

Alterar o arquivo, `src/pages/contato-detalle/contato-detalle.ts`, conforme se segue:

```

import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../../providers/contatos/contatos';

/**
 * Generated class for the ContatoDetalhePage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */

@Component({
  selector: 'page-contato-detalle',
  templateUrl: 'contato-detalle.html',
})
export class ContatoDetalhePage {
  contato = { nome: '', endereco: '', cep: '', lat: 0, long: 0, email: '' };
  cttparam: any = null;

  constructor(public navCtrl: NavController, public navParams: NavParams, private contatosProvider: ContatosProvider) {
    this.cttparam = navParams.get('cttParam');
  }

  if (this.cttparam != null) {
    this.contato.nome = this.cttparam.nome;
    this.contato.endereco = this.cttparam.endereco;
    this.contato.cep = this.cttparam.cep;
    this.contato.lat = this.cttparam.lat;
    this.contato.long = this.cttparam.long;
    this.contato.email = this.cttparam.email;
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatoDetalhePage');
  }

  confirmaContato() {
    const c = this.contato;

    if (this.cttparam != null) {
      //edita o contato
    } else {
      this.contatosProvider.addContato(c.nome, c.endereco, c.cep, c.lat, c.long, c.email);
    }

    this.navCtrl.pop();
  }

  cancelaContato() {
    this.navCtrl.pop();
  }
}

```

E o arquivo `src/pages/contatos/contatos.ts`, para fazer a chamada de adicionar contatos, conforme imagem abaixo:

```
import { Component } from '@angular/core';
import { NavController, NavParams } from 'ionic-angular';
import { ContatosProvider } from '../../providers/contatos/contatos';
import { ContatoDetalhePage } from '../../pages/contato-detalhe/contato-detalhe';

/**
 * Generated class for the ContatosPage page.
 *
 * See http://ionicframework.com/docs/components/#navigation for more info
 * on Ionic pages and navigation.
 */

@Component({
  selector: 'page-contatos',
  templateUrl: 'contatos.html',
})
export class ContatosPage {
  contatos: Array<(nome: string, endereco: string, cep: string, lat: number, long: number, email: string)> = [];

  constructor(public navCtrl: NavController, public NavParams: NavParams, private contatosProvider: ContatosProvider) {
    this.contatos = contatosProvider.getContatos();
  }

  ionViewDidLoad() {
    console.log('ionViewDidLoad ContatosPage');
  }

  addContato() {
    this.navCtrl.push(ContatoDetalhePage);
  }

  editContato(contato) {
    this.navCtrl.push(ContatoDetalhePage, {cttParam: contato});
  }
}
```

