

Introdução ao Desenvolvimento Móvel com Ionic

Rafael Guterres Jeffman

Jornada Acadêmica PUC/FACIN

Julho de 2019

Desenvolvimento para Dispositivos Móveis

- ▶ Desenvolvimento Nativo
 - iOS
 - Android
- ▶ Desenvolvimento Híbrido
 - Ionic
 - Xamarin
 - PhoneGap



Desenvolvimento Nativo



- ▶ Acesso a toda a API do dispositivo móvel.
- ▶ Tira vantagem de todos os componentes do dispositivo.
- ▶ Requer uma base de código para cada plataforma.
- ▶ Limita a base de usuários.

Desenvolvimento Híbrido

- ▶ Uma única base de código.
- ▶ Não limita a base de usuários.
- ▶ Utiliza ferramentas conhecidas.
- ▶ Pode não disponibilizar todos os recursos dos dispositivos.



Ionic



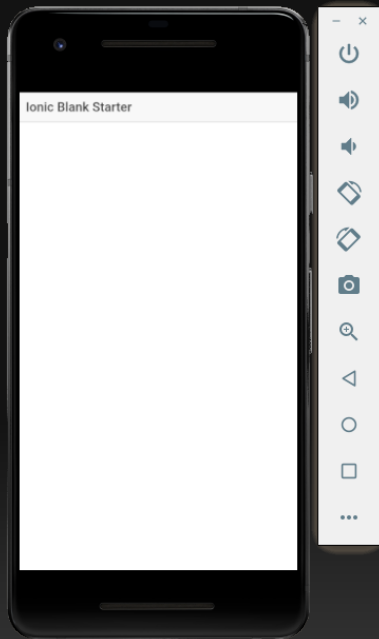
- ▶ Desenvolvimento híbrido baseado em HTML5.
- ▶ Utiliza Cordova ou Capacitor para criar a aplicação nativa.
- ▶ Open Source (MIT)
- ▶ <https://ionicframework.com>

Instalação do Ionic

- ▶ Ionic utiliza diversas ferramentas, a maioria rodando sob Node.js.
- ▶ Utiliza o Apache Cordova para gerar a aplicação nativa.
- ▶ Utiliza os ambientes nativos para gerar as aplicações.
- ▶ Uso de dispositivos ou emuladores para testar as aplicações.
- ▶ <https://ionicframework.com/docs/installation/cli>

Configuração do Ambiente de Desenvolvimento

- ▶ JAVA_HOME
- ▶ ANDROID_SDK_ROOT
- ▶ E os diretórios tools/bin, tools e emulator devem estar no PATH



Ionic Framework

- ▶ *UI Toolkit*
- ▶ *Framework agnostic*
- ▶ Baseado em ferramentas conhecidas (HTML, CSS e JavaScript)
- ▶ Integra facilmente com Angular e React (Vue?)
- ▶ Multi-plataforma
- ▶ Baseado em padrões da Web

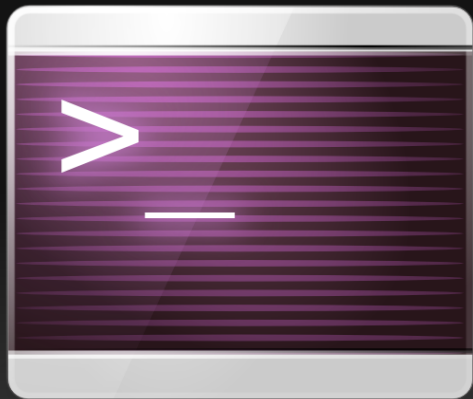


Principais conceitos

- ▶ Componentes UI
- ▶ Continuidade de Plataforma

Aplicações Ionic tem aparência nativa.
- ▶ Navegação não-linear
- ▶ Acesso a componentes nativos (com Cordova ou Capacitor)
- ▶ Temas

Ferramentas Básicas



- ▶ Git
- ▶ Node.js
- ▶ Editor de código (Visual Studio Code, Atom, ...)
- ▶ CLI Terminal

Criando um novo projeto.

- ▶ `ionic start --list`
- ▶ `ionic start PhotoGallery blank`
 - Perguntará qual framework utilizar (`--type`)
 - Criará uma estrutura de diretórios para o projeto.
 - Atualizará as dependências do Node.js.
- ▶ `cd PhotoGallery`

Estrutura do Projeto

- ▶ e2e - Testes automatizados: "unit" e "end-to-end"
- ▶ *.json - Arquivos de configuração do projeto.
- ▶ .gitignore - configuração para adicionar o projeto ao Git.
- ▶ src - *onde a mágica acontece.*

```
contador
├── angular.json
├── e2e
├── ionic.config.json
├── node_modules
├── package.json
├── package-lock.json
├── src
├── tsconfig.json
└── tslint.json
```

Estrutura do Projeto - **src**

- ▶ index.html - **app-root**
- ▶ theme - tema da interface
- ▶ assets - imagens, audios, etc.
- ▶ app - a aplicação...

```
contador/src
├── app
├── assets
├── environments
├── global.scss
├── index.html
├── karma.conf.js
├── main.ts
├── polyfills.ts
├── test.ts
├── theme
├── tsconfig.app.json
├── tsconfig.spec.json
├── tslint.json
└── zone-flags.ts
```

Estrutura do Projeto - **src/app**

- ▶ Arquivos da aplicação.
- ▶ Diretórios das páginas da aplicação.
- ▶ Diretórios de serviços.
- ▶ Diretórios dos outros componentes.

```
contador/src/app/
```

```
|— app.component.html  
|— app.component.spec.ts  
|— app.component.ts  
|— app.module.ts  
|— app-routing.module.ts  
|— app.scss  
|— home
```

Página da aplicação

- ▶ *.html - template da página.
- ▶ *.scss - Definições Sass para a página.
- ▶ page.page.ts - código TypeScript da página.

```
contador/src/app/home/  
├── home.module.ts  
├── home.page.html  
├── home.page.scss  
├── home.page.spec.ts  
└── home.page.ts
```


TypeScript

Classes

- ▶ Declaração: **class** *NomeDaClasse* {...}
- ▶ Campos: [**public**] *nomeDoCampo* : *tipo*
- ▶ Tipos: **string**, **number**, **boolean**, **any**
- ▶ O uso de **this** para acessar um campo não é opcional!

Construtores

- ▶ **constructor** (*parametros*)
- ▶ Parâmetros de construtor viram **campos**, logo, podem ser *public* ou *private*.
- ▶ Exemplo:

```
class Person {  
    constructor(public name: string, public age: number)  
}
```

Métodos

- ▶ *nomeDoMetodo* (*lista de parâmetros*) [*: tipo*] { ... }
- ▶ Arrow Functions: (***param: type***) => { *code* }

Arrays e Tuplas

- ▶ Array: *identificador: type[]*
pares: number[] = [0, 2, 4, 6, 8]
- ▶ Tupla: *identificador: [type, type, ...]*
software: [string, number] = ["Ionic", 4]

O Contador de Cliques

Contador de Cliques!

Clicked 0 times.



```

<ion-header>
  <ion-toolbar>
    <ion-title>
      Contador de Cliques!
    </ion-title>
  </ion-toolbar>
</ion-header>

<ion-content>
  <div style="display:block; width:100%; text-align:center">
    <h1>Clicked {{this.counter}} times.</h1>
    <ion-button (click)="increaseCounter()">
      <ion-icon slot="icon-only" name="cafe"></ion-icon>
    </ion-button>
  </div>
</ion-content>

```

```

import { Component } from '@angular/core';

@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage {
  public counter: number = 0;

  constructor() {}

  increaseCounter() {
    this.counter += 1;
    console.log("counter == " + this.counter)
  }
}

```


Lista de Contatos

Lista de Contatos

Luana

555-1234

555-3241

Rafael

555-3421

Maya

555-4321

Services

- ▶ *Data Services* são uma boa forma de separar a lógica de obtenção dos dados da apresentação.
- ▶ Você cria um serviço com o comando `ionic generate service <nome>`

```
import { Injectable } from '@angular/core';

@Injectable({
  providedIn: 'root'
})
export class ContatosService {
  public contatos: Contato[] = [
    new Contato("Luana", ["555-1234", "555-3241"]),
    new Contato("Rafael", ["555-3421"]),
    new Contato("Maya", ["555-4321"])
  ];
  constructor() { }
}

class Contato {
  constructor(public name:string, public phones:string[]) {}
}
```


Ônibus POA

Objetivo

- ▶ Consumir a API pública de transportes da cidade de Porto Alegre
- ▶ Localização de Paradas
- ▶ Linhas de Ônibus
- ▶ Paradas das linhas de Ônibus
- ▶ Paradas dentro de uma região a partir de coordenadas GPS

Isso se o Júnior mantivesse
a API funcionando e
atualizada...

Obrigado!

Links

- ▶ Links úteis
- ▶ <https://slides.tchelixux.org>
- ▶ <https://github.com/rafasgj/ionic-crash-course.git>
- ▶ <https://certificados.tchelixux.org>