

# Ionic Framework

(baseado na InfoQ)

# Roteiro

- Apresentando o Ionic Framework
- Apps Híbridos
- Apache Cordova
- AngularJS
- Ionic Framework
  - Iniciando um projeto Ionic
  - Estrutura de um projeto Ionic
  - Elementos UI
  - A
  - API Ionic e AngularJS

# Ionic Framework

- É uma ferramenta para desenvolvimentos de aplicações híbridas usando HTML5



Criado em 2013 por Bem Sperry (CEO) e  
Max Lynch (CTO)



“É inteiramente, 100% open-source. Sob licença MIT, significa que qualquer um pode usá-lo, com propósito comercial ou pessoal – isso não importa para nós”

Bem Sperry

# Características

- Componente de UI ricos e otimizados para mobile



- Componente de UI ricos e otimizados para mobile
- Performance



- Componente de UI ricos e otimizados para mobile
- Performance
- AngularJS



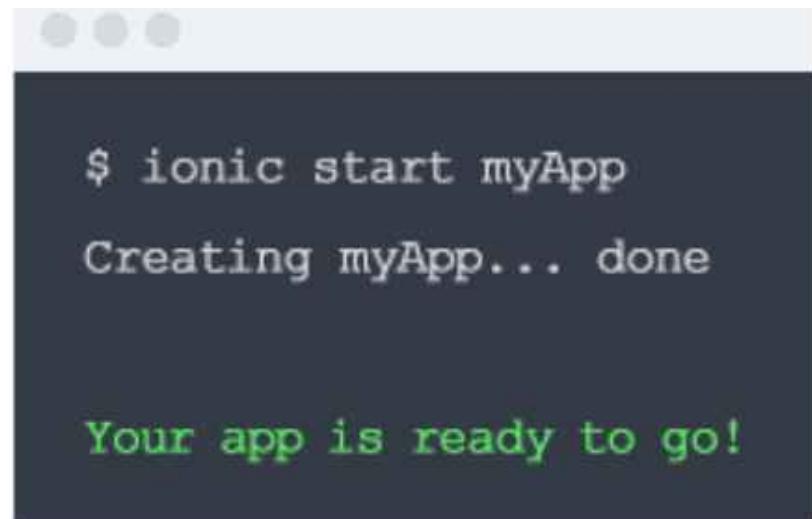
- Componente de UI ricos e otimizados para mobile
- Performance
- AngularJS
- Apache Cordova



- Componente de UI ricos e otimizados para mobile
- Performance
- AngularJS
- Apache Cordova
- Construído com Sass



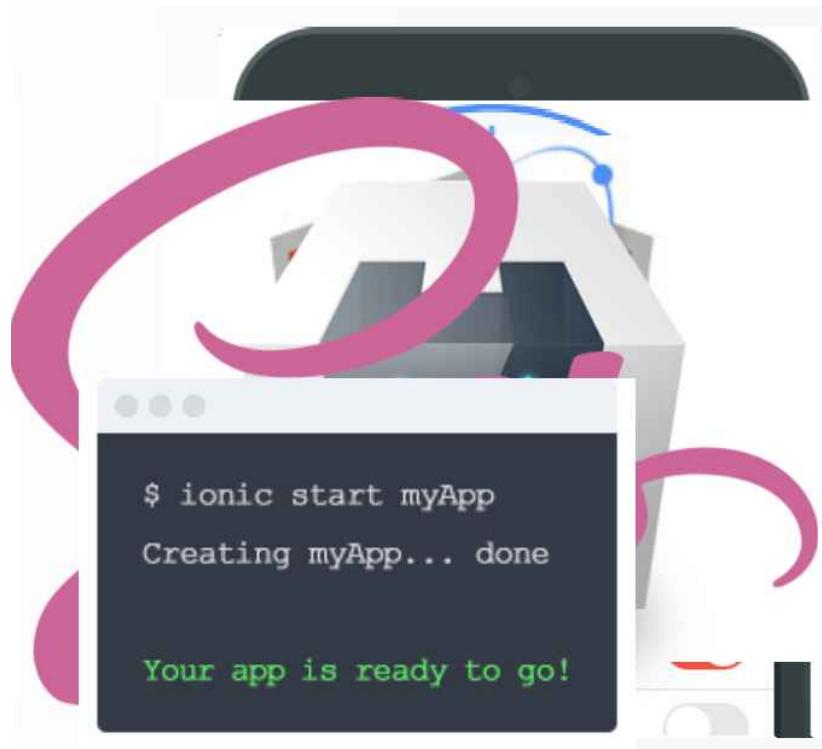
- Componente de UI ricos e otimizados para mobile
- Performance
- AngularJS
- Apache Cordova
- Construído com Sass
- Um poderoso CLI (Command line interface)



```
$ ionic start myApp
Creating myApp... done
Your app is ready to go!
```

A screenshot of a terminal window with a dark background and light-colored text. At the top, there are three small circular icons. The text inside the terminal shows the command '\$ ionic start myApp' followed by the message 'Creating myApp... done'. Below this, the text 'Your app is ready to go!' is displayed in green.

- Componente de UI ricos e otimizados para mobile
- Performance
- AngularJS
- Apache Cordova
- Construído com Sass
- Um poderoso CLI (Command line interface)
- Documentação rica e comunidade ativa



# Apps Híbridos



# Tipos de aplicações móveis

- Aplicações Nativas
- Web Apps
- Aplicações Híbridas



- **Aplicações Híbridas** são aplicações móveis desenvolvidas com linguagem web (HTML, CSS e Javascript), que podem acessar recursos nativos dos dispositivos.

Apache Cordova



Phonegap



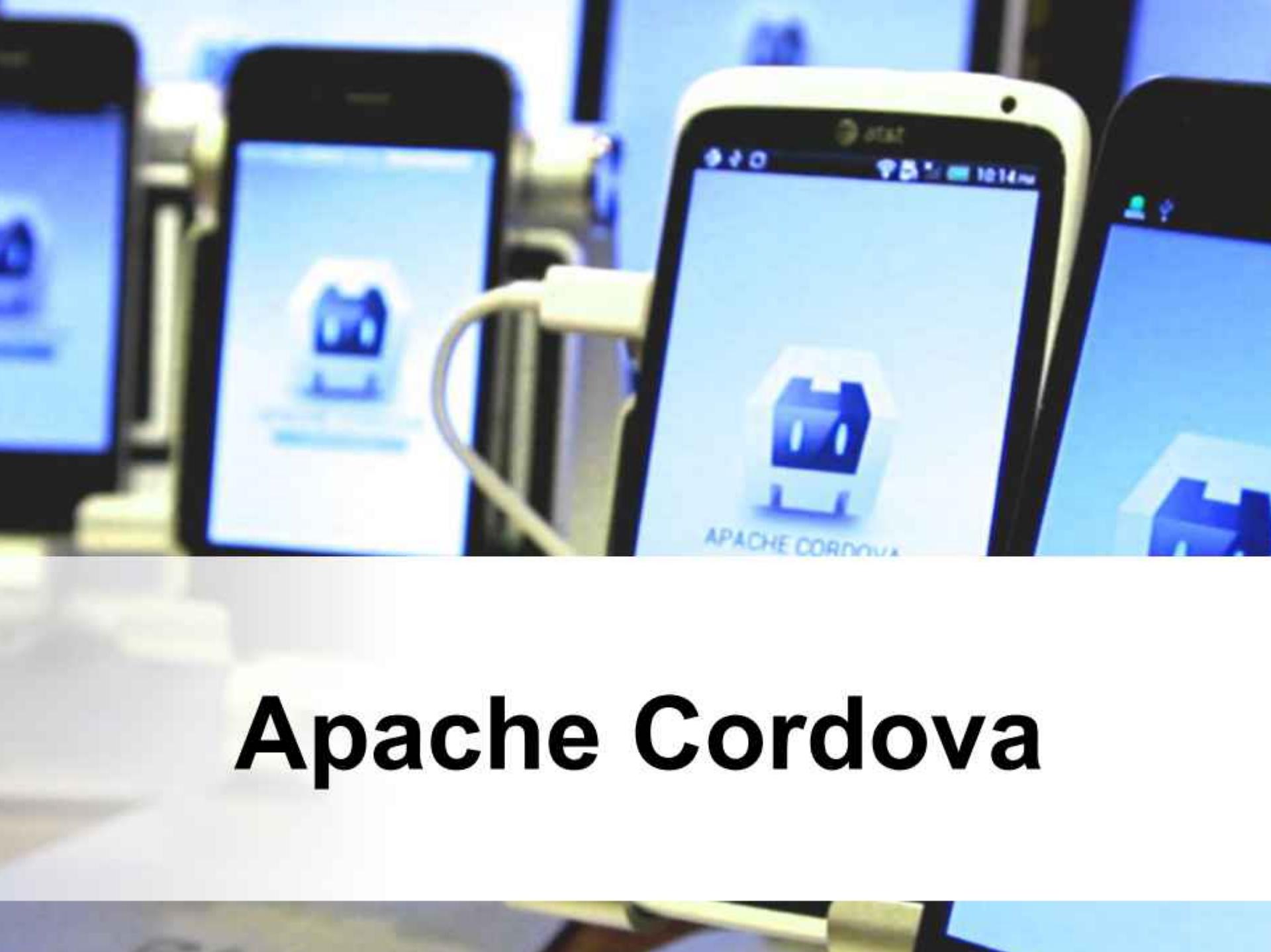
Sencha Touch



AppGyver Steroids



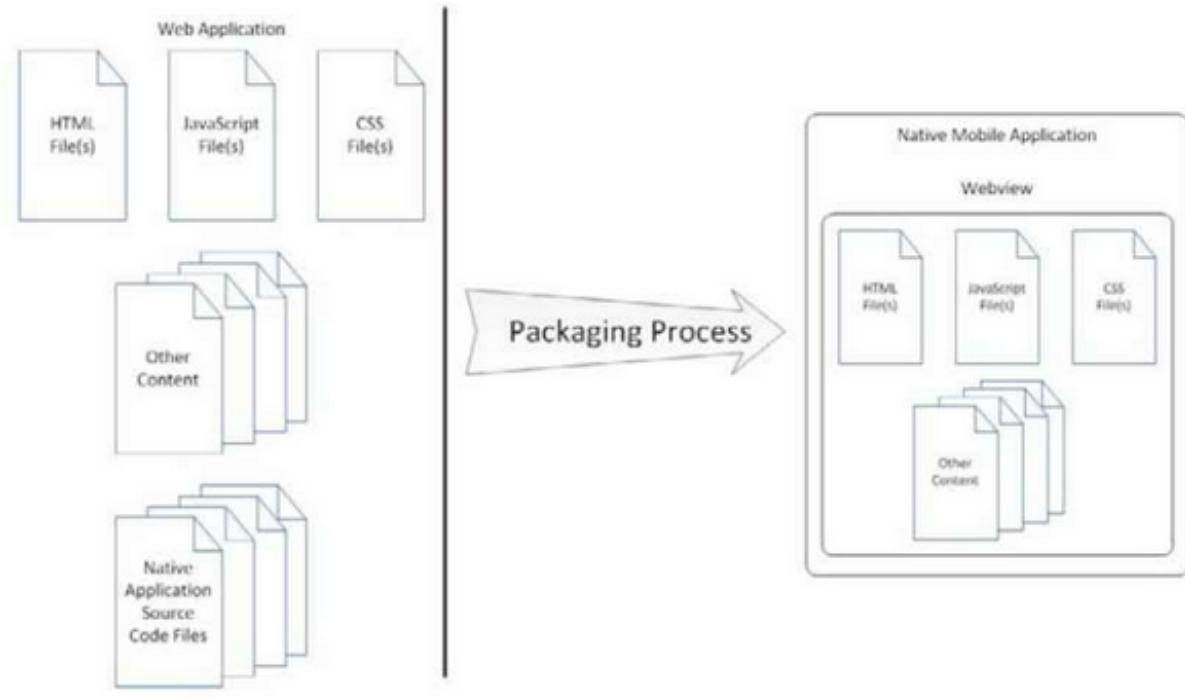
# Apache Cordova



- É um framework gratuito e de código aberto para a construção de aplicações móveis multiplataforma usando HTML5.
- É constituído por um conjunto de ferramentas para:
  - Gerenciar o processo de criação de aplicações;
  - Gerenciamento de plug-ins;
  - Build de código nativo (usando SDK's nativos);
  - E teste das aplicações usando simuladores e emuladores.

# Uma aplicação Cordova consiste em:

- Uma aplicação web;
- Um container nativo que renderiza a página web (web view)
- Uma suite de API's para a aplicação web acessar recursos nativos.



# A API do Apache Cordova

- Accelerometer
- Camera
- Capture
- Compass
- Connection
- Contacts
- Device
- Events
- File
- Geolocation
- Globalization
- InAppBrowser
- MediaNotofication
- Splashscreen
- Storage

- Ex: Câmera

```
navigator.camera.getPicture(onSuccess, onFail, { quality: 50,
    destinationType: Camera.DestinationType.FILE_URI });

function onSuccess(imageURI) {
    var image = document.getElementById('myImage');
    image.src = imageURI;
}

function onFail(message) {
    alert('Failed because: ' + message);
}
```

```
keydown.internet', for  
21) s.internet.close()  
  
  - function(data, klass){  
    internet.content').append(data);  
    'loading').remove();  
    'body').children().  
  
lose  
c).unbind('keydown', funct  
internet).on('con  
false  
  
nternet' + fun  
init(), s.intern
```

# AngularJS

- Um framework MVC Javascript para aplicações web dinâmicas;
- Desenvolvido e mantido pelo Google;
- Estende o HTML (diretivas);
- Manipulação mínima de DOM;
- Usa abordagem declarativa para expressar lógica nas views;



## Module

```
<html ng-app="demoApp">  
  <div ng-controller="demoController">  
    ....  
  </div>  
</html>
```

### View (DOM)

```
<div id="name">
```

\$scope

### Controller

(AngularJS)

```
var demoApp = angular.module('dempApp', []);  
demoApp.controller('demoController',  
function($scope){....});
```

### Config

(AngularJS)

```
var demoApp = angular.module('dempApp', []);  
demoApp.config(function($routeProvider){....});
```

### Directives (extends DOM)

ng-app, ng-model, ng-bind, ng-controller, etc

### Filter

```
<tr ng-repeat="person in  
persons filter:searchText">
```

### Factory

(AngularJS)

```
var demoApp = angular.module('demoApp', []);  
demoApp.factory('demoService', function()  
{  
  var newServiceInstance;  
  //factory function body that constructs demoServiceInstance  
  return demoServiceInstance;  
});
```

### Routes

(AngularJS)

### Service

(AngularJS)

### Provider

(AngularJS)

### Value

(AngularJS)

# Conceitos e Recursos

- Data Binding
- Templates Client-Side
- Separation of concerns
- Dependency Injection
- Deep Linking
- Form Validation
- Server Communication

# Data Binding e MVC

- Data Binding
  - Utiliza expressões ou um atributo ng-model para recuperar ou receber dados.
- Arquitetura baseada em MVC
  - a visão pode ser expressa por templates
  - o controlador possui um escopo que é definido por uma visão que é sua subordinada
  - o controller é responsável por relacionar-se com o modelo para adicionar os dados no seu escopo

# Data Binding

```
1. <!doctype html>
2. <html ng-app>
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular.min.js"></script>
5.   </head>
6.   <body>
7.     <div>
8.       <label>Name:</label>
9.       <input type="text" ng-model="yourName" placeholder="Enter a name here">
10.      <hr>
11.      <h1>Hello {{yourName}}!</h1>
12.    </div>
13.  </body>
14. </html>
```

# Controllers

```
1. <!doctype html>
2. <html ng-app="todoApp">
3.   <head>
4.     <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular.min.js"></script>
5.     <script src="todo.js"></script>
6.     <link rel="stylesheet" href="todo.css">
7.   </head>
8.   <body>
9.     <h2>Todo</h2>
10.    <div ng-controller="TodoController">
11.      <span>{{remaining()}} of {{todos.length}} remaining</span>
12.      [ <a href="" ng-click="archive()">archive</a> ]
13.      <ul class="unstyled">
14.        <li ng-repeat="todo in todos">
15.          <input type="checkbox" ng-model="todo.done">
16.          <span class="done-{{todo.done}}">{{todo.text}}</span>
17.        </li>
18.      </ul>
19.      <form ng-submit="addTodo()">
20.        <input type="text" ng-model="todoText" size="30"
placeholder="add new todo here">
```

```
1.  <!doctype html>
2.  <html ng-app="todoApp">
3.    <head>
4.      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular.min.js"></script>
5.      <script src="todo.js"></script>
6.      <link rel="stylesheet" href="todo.css">
7.    </head>
8.    <body>
9.      <h2>Todo</h2>
10.     <div ng-controller="TodoController">
11.       <span>{{remaining()}} of {{todos.length}} remaining</span>
12.       [ <a href="" ng-click="archive()">archive</a> ]
13.       <ul class="unstyled">
14.         <li ng-repeat="todo in todos">
15.           <input type="checkbox" ng-model="todo.done">
16.           <span class="done-{{todo.done}}">{{todo.text}}</span>
17.         </li>
18.       </ul>
19.       <form ng-submit="addTodo()">
20.         <input type="text" ng-model="todoText" size="30"
21.               placeholder="add new todo here">
22.         <input class="btn-primary" type="submit" value="add">
23.       </form>
24.     </div>
25.   </body>
26. </html>
```

```
1. angular.module('todoApp', [])
2.   .controller('TodoController', ['$scope', function($scope) {
3.     $scope.todos = [
4.       {text:'learn angular', done:true},
5.       {text:'build an angular app', done:false}];
6.
7.     $scope.addTodo = function() {
8.       $scope.todos.push({text:$scope.todoText, done:false});
9.       $scope.todoText = '';
10.    };
11.
12.    $scope.remaining = function() {
13.      var count = 0;
14.      angular.forEach($scope.todos, function(todo) {
15.        count += todo.done ? 0 : 1;
16.      });
17.      return count;
18.    };
19.
20.    $scope.archive = function() {
21.      var oldTodos = $scope.todos;
22.      $scope.todos = [];
23.      angular.forEach(oldTodos, function(todo) {
24.        if (!todo.done) $scope.todos.push(todo);
25.      });
26.    };
27.  }]);

```

# Template e Deep-Linking #1/2

- Templates
  - são escritos em HTML e podem contém elementos e atributos específicos do AngularJS.
  - Estes são os tipos de elementos angulares e atributos que você pode usar:
    - **directive** - um atributo ou elemento que amplia um elemento DOM existente ou representa um componente DOM reutilizável.
    - **markup** - A dupla notação de chave `{}{}` para vincular expressões de dados aos elementos do AngularJS.
    - **filter** – formata dados para visualização.
    - **form control** - valida a entrada do usuário

# Template e Deep-Linking #2/2

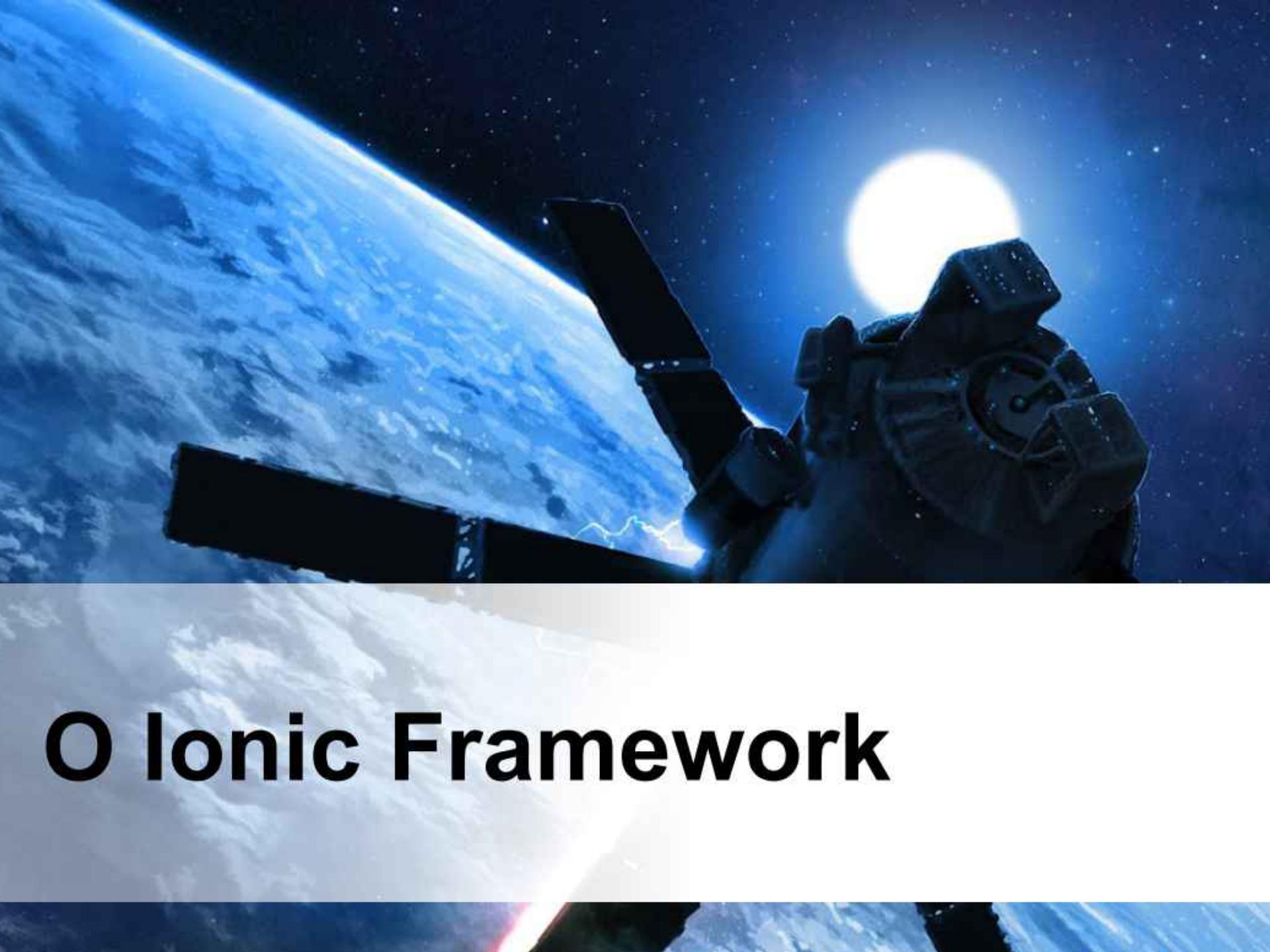
- Deep-Linking
  - Consiste na ideia de usar roteamentos de forma hierárquica para alcançar um determinado recurso
  - Exemplo:
    - **/pedido/1**: acessa o pedido nº 1;
    - **/pedido/1/vendedor**: apresenta o nome do vendedor que realizou o pedido nº 1;
    - **/pedido/1/vendedor/comissao**: apresenta o valor da comissão do vendedor que realizou o pedido nº 1.

```
1.  <!doctype html>
2.  <html ng-app="project">
3.    <head>
4.      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular.min.js"></script>
5.      <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular-resource.min.js">
6.        </script>
7.        <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.0/
angular-route.min.js">
8.          </script>
9.          <script src="https://cdn.firebaseio.com/js/client/1.0.18.firebaseio.js"><scr
ipt>
10.         <script src="https://cdn.firebaseio.com/libs/angularfire/0.8.0/
angularfire.min.js"></script>
11.         <link rel="stylesheet" href="bootstrap.css">
12.         <script src="project.js"></script>
13.     </head>
14.     <body>
15.       <h2>JavaScript Projects</h2>
16.       <div ng-view></div>
17.     </body>
18.   </html>
```

```
1. angular.module('project', ['ngRoute', 'firebase'])
2.
3.   .value('fbURL', 'https://angularjs-projects.firebaseio.com/')
4.
5.   .factory('Projects', function($firebase, fbURL) {
6.     return $firebase(new Firebase(fbURL)).$asArray();
7.   })
8.
9.   .config(function($routeProvider) {
10.     $routeProvider
11.       .when('/', {
12.         controller:'ListCtrl',
13.         templateUrl:'list.html'
14.       })
15.       .when('/edit/:projectId', {
16.         controller:'EditCtrl',
17.         templateUrl:'detail.html'
18.       })
19.       .when('/new', {
20.         controller:'CreateCtrl',
21.         templateUrl:'detail.html'
22.       })
23.       .otherwise({
24.         redirectTo:'/'
25.       });
26.   })

```

# O Ionic Framework



# Iniciando

- **Opção 1**

Baixe o último release oficial do Ionic

<http://ionicframework.com/docs/overview/#download>

- **Opção 2**

Utilizar o utilitário de linha de comando

```
$ npm install -g ionic  
$ ionic start myproject
```

# Iniciando

- Você pode iniciar um projeto a partir de um dos templates prontos:

```
2
3 #template pode ser
4 # tabs (default) | sidemenu | blank
5 ionic start myapp [template]
6
7
```

# Iniciando

Depois de iniciado um projeto você pode:

- Configurar o projeto para usar Sass:
  - `ionic setup sass`
- Desenvolver no Browser com o live reload:
  - `ionic server`
- Adicionar uma plataforma (iOS ou Android):
  - `ionic platform add <android | ios>`
- Dar o build na sua app:
  - `ionic build <android | ios>`
- Emular sua app:
  - `ionic emulate <android | ios>`
- Executar sua app em um dispositivo:
  - `ionic run <android | ios>`

# Iniciando

- Um projeto Ionic iniciado por CLI é um projeto cordova

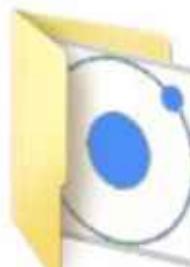
```
$ cordova platform add android [ios]  
$ cordova plugin add org.apache.cordova.inappbrowser
```

# Estrutura do projeto

myapp > platforms > android > assets > www



css



img



js



lib



plugins



templates



cordova.js



cordova\_plugins.j  
s



index.html

# Estrutura do projeto

android ➔ assets ➔ www ➔ lib ➔ ionic ➔ js



angular



angular-ui



ionic.bundle.js



ionic.bundle.min.  
js



ionic.js



ionic.min.js



ionic-angular.js



ionic-angular.mi  
n.js

# Estrutura do projeto

- O framework é oferecido em todo o seu potencial sendo carregado como módulo Angular:

```
angular.module('todo', ['ionic'])
```

- Que utiliza os seguintes módulos Angular
  - angular-resource
  - angular-animate
  - angular-sanitize
  - angular-ui-router

# Estrutura do projeto

- O Ionic define uma extensa quantidade de diretivas Angular próprias:

```
<body>
  <ion-side-menus>
    <ion-side-menu-content>
    </ion-side-menu-content>
    <ion-side-menu side="left">
    </ion-side-menu>
  </ion-side-menus>
</body>
```

# Usando o Ionic e o Cordova

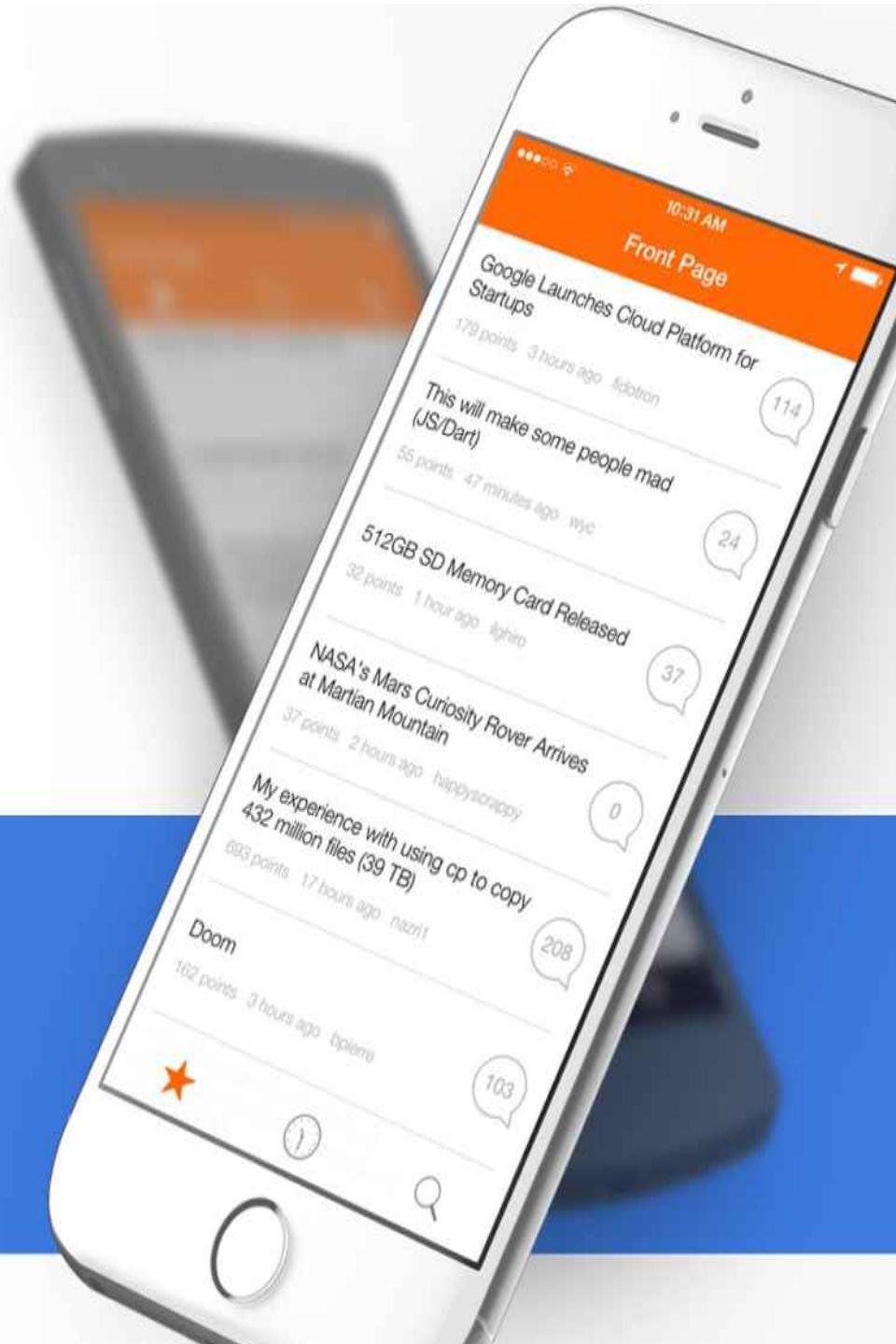
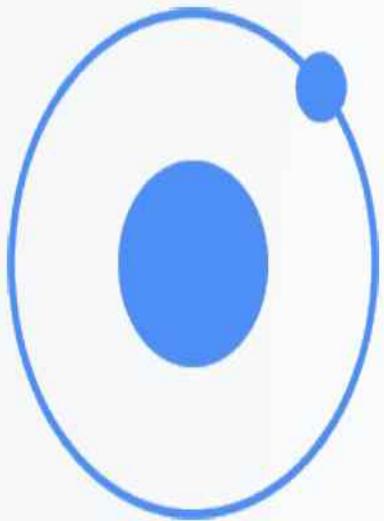
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Todo</title>
    <meta name="viewport" content="initial-scale=1, maximum-scale=1, user-scalable=no, width=device-width">

    <link href="lib/ionic/css/ionic.css" rel="stylesheet">

    <script src="lib/ionic/js/ionic.bundle.js"></script>

    <!-- Needed for Cordova/PhoneGap (will be a 404 during development) -->
    <script src="cordova.js"></script>
  </head>
  <body>
    </body>
</html>
```

# Elementos de UI



# CSS

- Header
- Content
- Footer
- Buttons
- List
- Cards
- Forms
- Toggle
- Checkbox
- Radio
- buttons
- Range
- Select
- Tabs
- Grid
- Utility
- Colors
- Icons
- Padding
- Animation

## bar-light

```
<div class="bar bar-header bar-light">  
  <h1 class="title">bar-light</h1>  
</div>
```

## bar-stable

```
<div class="bar bar-header bar-stable">  
  <h1 class="title">bar-stable</h1>  
</div>
```

## bar-positive

```
<div class="bar bar-header bar-positive">  
  <h1 class="title">bar-positive</h1>  
</div>
```

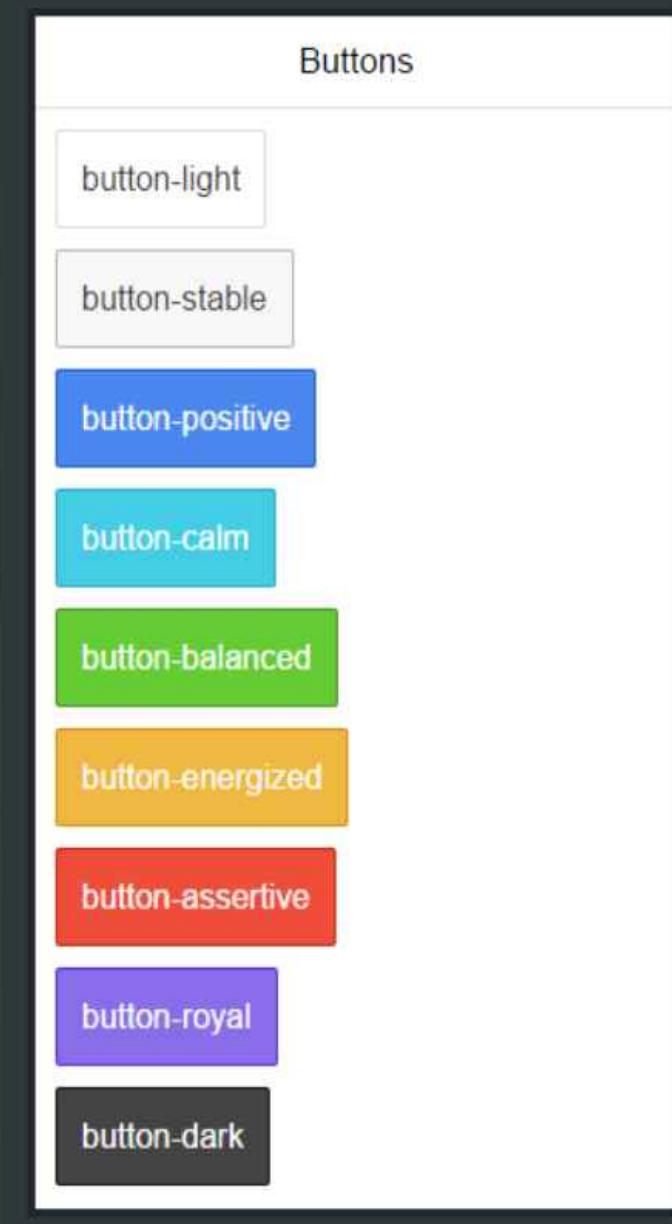
## bar-calm

```
<div class="bar bar-header bar-calm">  
  <h1 class="title">bar-calm</h1>  
</div>
```

Header

Sub Header

```
<button class="button">  
  Default  
</button>  
  
<button class="button button-light">  
  button-light  
</button>  
  
<button class="button button-stable">  
  button-stable  
</button>  
  
<button class="button button-positive">  
  button-positive  
</button>  
  
<button class="button button-calm">  
  button-calm  
</button>  
  
<button class="button button-balanced">  
  button-balanced  
</button>  
  
<button class="button button-energized">  
  button-energized  
</button>  
  
<button class="button button-assertive">  
  button-assertive  
</button>  
  
<button class="button button-royal">  
  button-royal  
</button>  
  
<button class="button button-dark">  
  button-dark  
</button>
```



```
<button class="button button-block button-positive">  
  Block Button  
</button>
```

## Block Buttons

button-light

button-stable

button-positive

button-calm

button-balanced

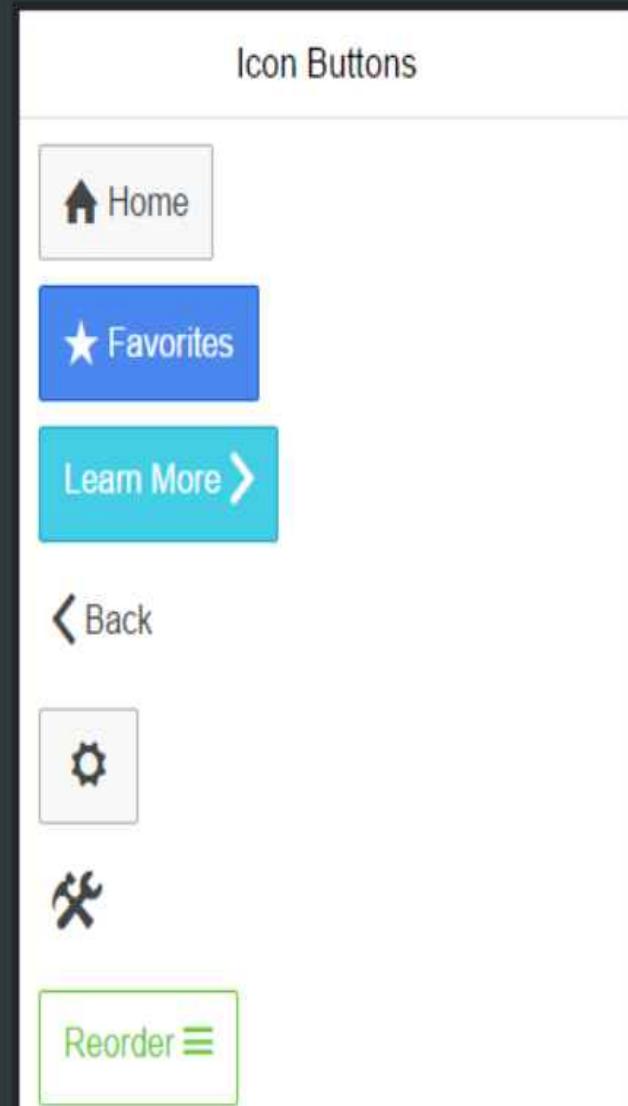
button-energized

button-assertive

button-royal

button-dark

```
<button class="button">  
  <i class="icon ion-loading-c"></i> Loading...  
</button>  
  
<button class="button icon-left ion-home">Home</button>  
  
<button class="button icon-left ion-star button-positive">Favorites</button>  
  
<a class="button icon-right ion-chevron-right button-calm">Learn More</a>  
  
<a class="button icon-left ion-chevron-left button-clear button-dark">Back</a>  
  
<button class="button icon ion-gear-a"></button>  
  
<a class="button button-icon icon ion-settings"></a>  
  
<a class="button button-outline icon-right ion-navicon button-balanced">Reorder</a>
```



```
<div class="list">  
  
  <a class="item item-avatar" href="#">  
      
    <h2>Venkman</h2>  
    <p>Back off, man. I'm a scientist.</p>  
  </a>  
  
  ...  
  
</div>
```

- ### Item Avatars
-  Venkman  
Back off, man. I'm a scientist.
  -  Egon  
We're gonna go full stream.
  -  Ray  
Ugly little spud, isn't he?
  -  Winston  
That's a big Twinkie.
  -  Tully  
Okay, who brought the dog?
  -  Dana  
I am The Gatekeeper!

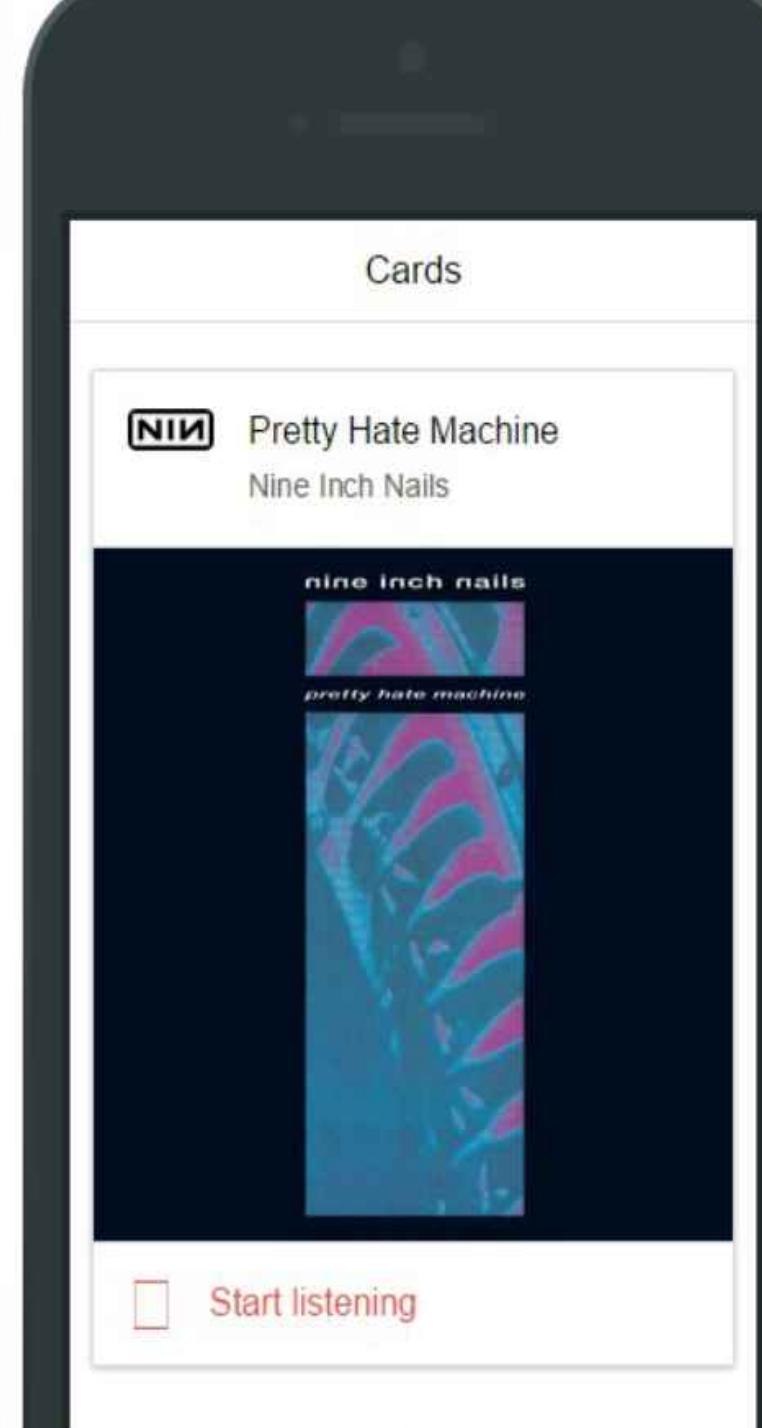
```
<div class="list card">

<div class="item item-avatar">
  
  <h2>Pretty Hate Machine</h2>
  <p>Nine Inch Nails</p>
</div>

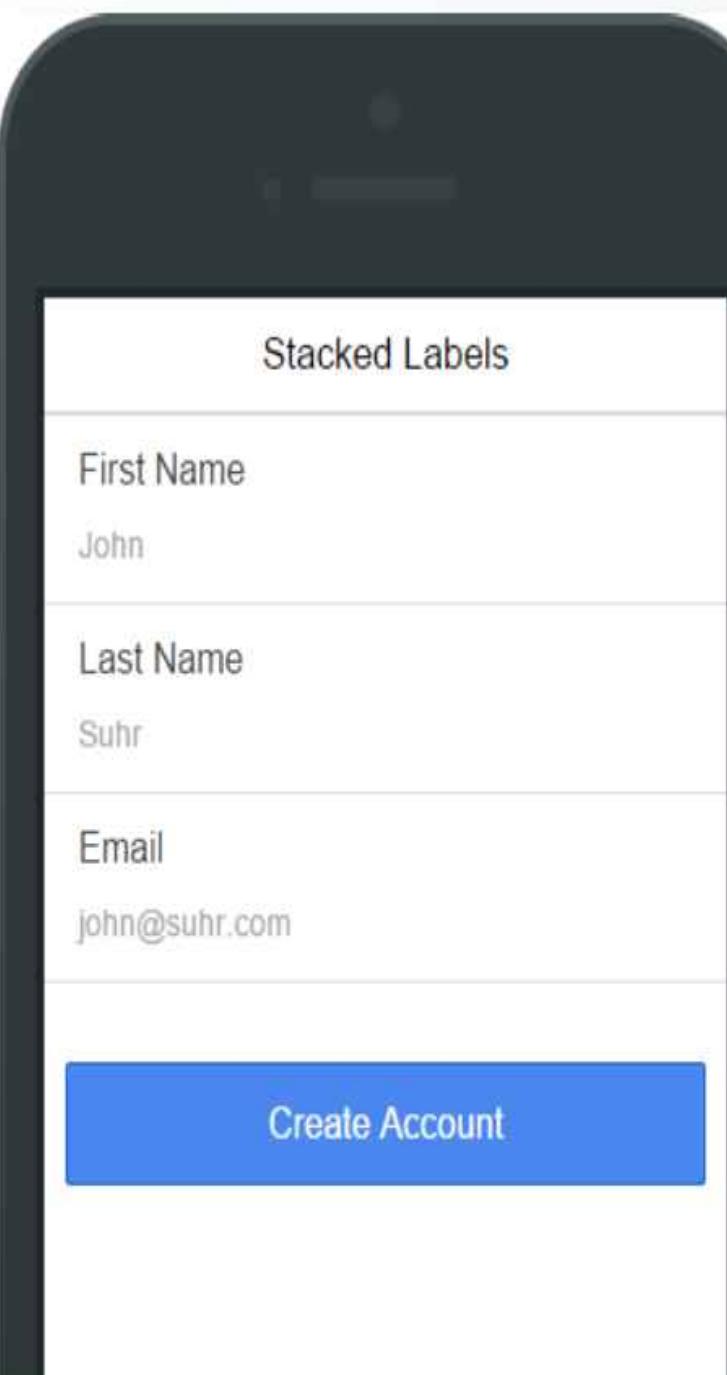
<div class="item item-image">
  
</div>

<a class="item item-icon-left assertive" href="#">
  <i class="icon ion-music-note"></i>
  Start listening
</a>

</div>
```

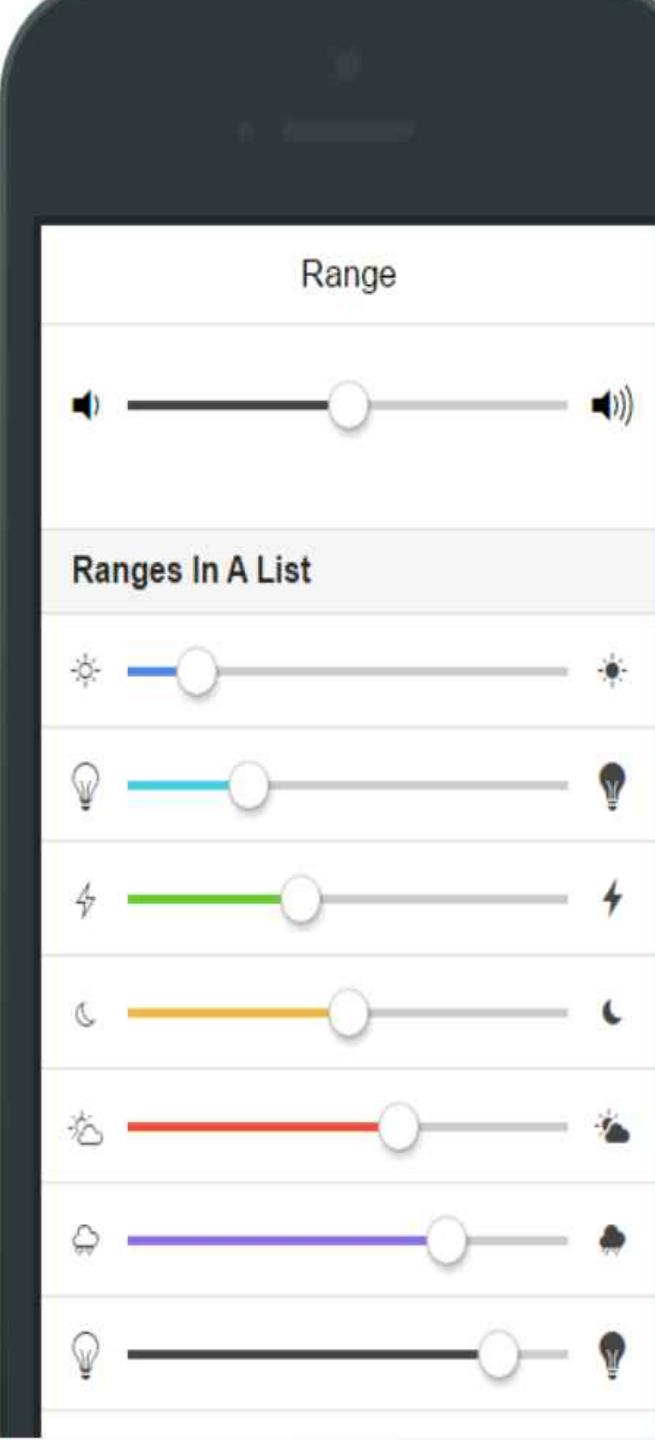


```
<div class="list">
  <label class="item item-input item-stacked-label">
    <span class="input-label">First Name</span>
    <input type="text" placeholder="John">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Last Name</span>
    <input type="text" placeholder="Suhr">
  </label>
  <label class="item item-input item-stacked-label">
    <span class="input-label">Email</span>
    <input type="text" placeholder="john@suhr.com">
  </label>
</div>
```



```
<div class="range">
  <i class="icon ion-volume-low"></i>
  <input type="range" name="volume">
  <i class="icon ion-volume-high"></i>
</div>

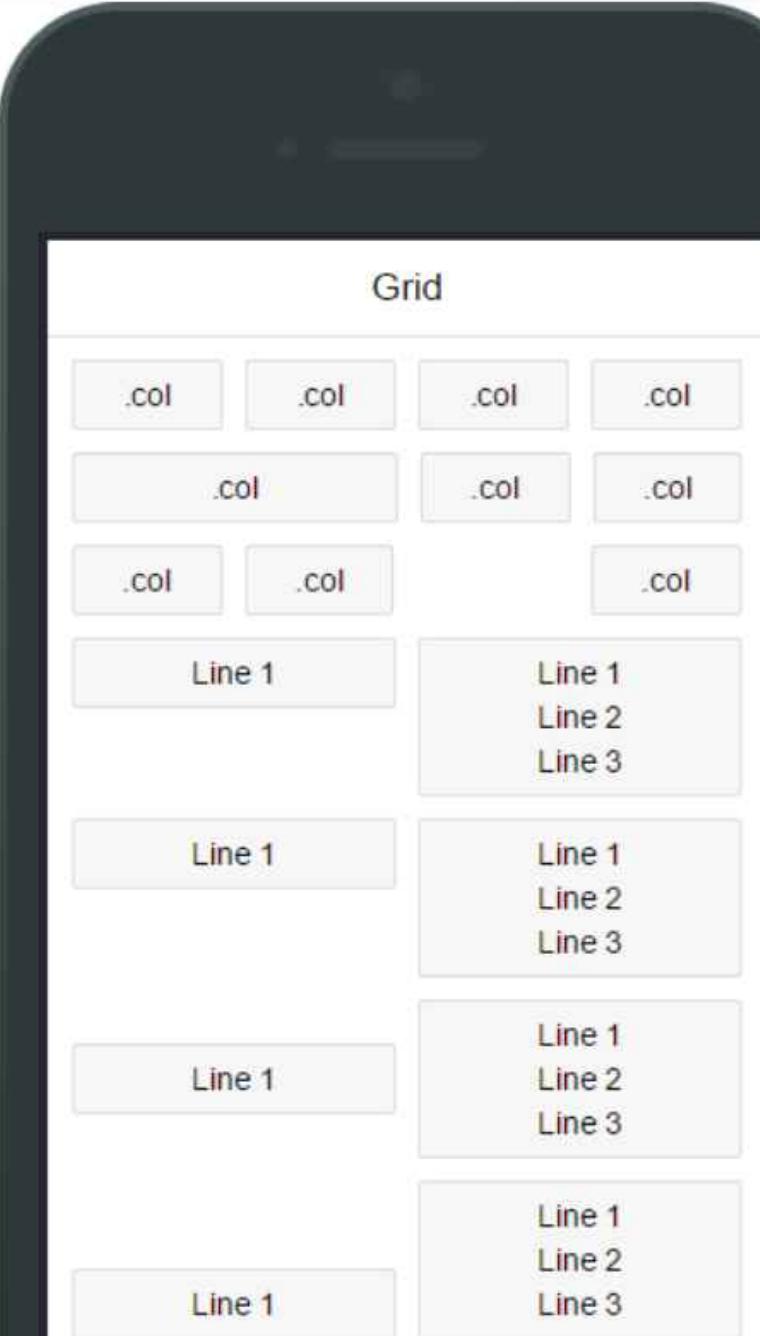
<div class="list">
  <div class="item range range-positive">
    <i class="icon ion-ios7-sunny-outline"></i>
    <input type="range" name="volume" min="0" max="100" value="33">
    <i class="icon ion-ios7-sunny"></i>
  </div>
</div>
```



```
<div class="row">
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
  <div class="col">.col</div>
</div>
```

#### Offset Column Percentage Classnames

.col-offset-10	10%
.col-offset-20	20%
.col-offset-25	25%
.col-offset-33	33.3333%
.col-offset-50	50%
.col-offset-67	66.6666%
.col-offset-75	75%
.col-offset-80	80%
.col-offset-90	90%



## Utility Colors

light

stable

positive

calm

balanced

energized

assertive

royal

dark



⟳

↑

→

↓

←

▲

▶

▼

◀

↑

→

↓

↑

➡

⬅

↔

☒

☒

☒

☒

☒

☒

☒

☒

☰

☰

☰

☰

☰

✓

✓

✓

✗

✗

✗

+

+

+

-

-

–

i

i

?

?

⌚

\*

!

# Animations

- Fade-in
- Nav-title-slide-ios7
- No-animation
- Reverse
- Slide-in-lift
- Slide-in-right
- Slide-in-up
- Slide-left-right-ios7
- Slide-left-right
- Slide-out-left
- Slide-out-right
- Slide-right-left-ios7
- Slide-right-left



# A API Ionic e AngularJS

# Javascript

- Action Sheet
- Backdrop
- Content
- Events
- Form inputs
- Gesture
- Headers/footers
- Keyboard
- List
- Loadig
- Modal
- Navigation
- Platform
- Popover
- Popup
- Scroll
- Side menus
- Slide box
- Tabs
- Tap & click
- Utility

# Eventos

- On-hold
- On-tap
- On-touch
- On-release
- On-drag
- On-drag-up
- On-drag-right
- On-drag-down
- On-drag-left
- On-swipe
- On-swipe-up
- Up-swipe-right
- On-swipe-
- On-swipe-left

```
<button on-hold="onHold()" class="button">  
Test</button>
```

# Navegação

- ion-nav-view
- ion-view
- ion-nav-bar
- ion-nav-button
- ion-nav-back-button
- nav-clear
- \$ionicNavBarDelegate

```
6 <ion-navbar class="bar-stable nav-title-slide-ios7">
7   <ion-nav-back-button class="button-icon icon ion-ios7-
8     -arrow-back">
9     Back
10    </ion-nav-back-button>
11  </ion-navbar>
12
13 <ion-tabs class="tabs-icon-only tabs-positive">
14
15   <!-- Home Tab -->
16   <ion-tab title="Home" icon="icon ion-home" href="#/home">
17     <ion-nav-view name="index"></ion-nav-view>
18   </ion-tab>
19
20   <!-- About Tab -->
21   <ion-tab title="About" icon="icon ion-heart" href="#/about">
22     <ion-nav-view name="about"></ion-nav-view>
23   </ion-tab>
24
25   Angular UI-Router
26   <!-- Contact Tab -->
27   <ion-tab title="Contact" icon="icon ion-gear-b" href="#/tab/account">
28     <ion-nav-view name="contact"></ion-nav-view>
29   </ion-tab>
30
31 </ion-tabs>
```

HTML JS Result Edit

← Also Factual

111,111,111 x 111,111,111 =  
12,345,678,987,654,321

1 in every 4 Americans has appeared on T.V.

11% of the world is left-handed.

1 in 8 Americans has worked at a McDonalds restaurant.

\$283,200 is the absolute highest amount of money you can win on Jeopardy.

101 Dalmatians, Peter Pan, Lady and the Tramp, and Mulan are the only Disney cartoons where both parents are present and don't die throughout the movie.

 Home     Scientific Facts

 Home     About     Contact

# Ionic Framework

Passo a Passo para Instalação  
(Linux)

# Requisitos

- Node.js
- Cordova
- Ionic
- Ionic-Material
- Android SDK
  - precisa ser adicionado ao path com a variável de ambiente ANDROID\_HOME
- Java JDK
  - precisa ser adicionado ao path com a variável de ambiente JAVA\_HOME

# Instalação do Node.js

- Realizar download do npm no link abaixo
  - <https://nodejs.org/download>
- Instalar o node.js ou descompactar, dependendo do SO
- Testar instalação

```
$ npm version
```

# Instalação do Cordova

- Digite o seguinte comando:

```
$ npm install -g cordova
```

- Dentro do diretório de trabalho crie um projeto cordova chamado **MyFirstApp** dentro de um diretório chamado **app**:

```
$ cordova create app ifpb.firstapp MyFirstAp  
$ cd app  
$ cordova platforms add android
```

# Instalação do Ionic Framework

- Digite o seguinte comando:

```
npm install -g cordova ionic
```

- Dentro do diretório de trabalho crie um projeto ionic chamado **MyApp** baseado no template **blank**:

```
$ ionic start MyApp blank  
$ cd MyApp  
$ ionic platform add android  
$ ionic build android  
$ ionic emulate android
```

# Instalação do Ionic Material

- Digite o seguinte comando:

```
$ npm install -g cordova ionic bower  
$ cd MyApp  
$ bower install ionic-material
```

- Adicione Ionic Material no projeto **MyApp**:

```
<link href="lib/ionic/css/ionic.css" rel="stylesheet">  
<link href="lib/ionic-material/ionic.material.min.css" rel="stylesheet">  
<link href="css/style.css" rel="stylesheet">  
  
<script src="lib/ionic/js/ionic.bundle.js"></script>  
<script src="lib/ionic-material/ionic.material.min.js"></script>
```

```
var app = angular.module('YOUR_APP_NAME', ['ionic', 'ionic-material']);
```

# Ionic Framework

Primeira Aplicação

# Passos de Criação

- Criar uma aplicação Ionic

```
$ ionic start webapp-storage-in-memory blank
```

- Adicionar suporte a plataforma android

```
$ cd webapp-storage-in-memory
$ ionic platform add android
```

# Desenvolvimento de Telas

- Criar os templates correspondentes as seguintes telas:

Listagem de Alunos		
Matrícula	Nome	CPF
09020911	Carlos Henrique	876872234
09090211	Antônio Carlos	09889789
09030511	Maurício Azevedo	12109809

**Formulário de Cadastro**

---

Nome do Aluno

CPF do Aluno

Matrícula do Aluno

# Plano de Implementação (app.config + app.controller)

- Criar um controlador para a tela 1:

```
14 //configurar o deep-linking
15 app.config(function($stateProvider, $urlRouterProvider){
16   $stateProvider.state('home', {
17     url: '/home',
18     templateUrl: 'tpl/table.html',
19     controller: 'homeController'
20   })
21   $urlRouterProvider.otherwise('/home')
22 })
23 //criar o controle para a home
24 app.controller('homeController', ['$scope', '$state', 'studentModel']
25   $scope.students = studentModel.listAll()
26   $scope.append = function(){
27     $state.go('form')
28   }
29 })
```

# Plano de Implementação (template table.html)

- Criar o template da tela 1:

```
1 <ion-pane>
2   <ion-header-bar class="bar-stable">
3     <h1 class="title">WebApp Without Persistence</h1>
4     <div class="buttons">
5       <button class="button" ng-click='append()'>"+</button>
6     </div>
7   </ion-header-bar>
8   <ion-content>
9     <table style="width: 100%">
10    <thead style="background: #ccc">
11      <tr class="row">
12        <td class="col col-20">Matrícula</td>
13        <td class="col col-50">Nome</td>
14        <td class="col">CPF</td>
15      </tr>
16    </thead>
17    <tbody>
18      <tr class="row" ng-repeat="student in students">
19        <td class="col col-20">{{student.code}}</td>
20        <td class="col col-50">{{student.name}}</td>
21        <td class="col">{{student.cpf}}</td>
22      </tr>
23    </tbody>
24  </table>
25 </ion-content>
26 </ion-pane>
```

# Plano de Implementação (app.config + app.controller)

- Criar um controlador para a tela 2:

```
21 $stateProvider.state('form', {  
22   url: '/form',  
23   templateUrl: 'tpl/form.html',  
24   controller: 'formController'  
25 })  
26 $urlRouterProvider.otherwise('/home')  
27 })  
28 //criar o controle para a home  
29 app.controller('homeController', ['$scope', '$state', 'studentModel'  
30   $scope.students = studentModel.listAll()  
31   $scope.append = function(){  
32     $state.go('form')  
33   }  
34 }])  
35 //criar o controle para a form  
36 app.controller('formController', ['$scope', '$state', 'studentModel'  
37   //inicializa o modelo  
38   $scope.student = {}  
39   $scope.save = function(){  
40     studentModel.append($scope.student, function(){  
41       $state.go('home')  
42     })  
43   }  
44 }])
```

# Plano de Implementação

## (template form.html)

- Criar o template da tela 2:

```
1 <ion-pane>
2   <ion-header-bar class="bar-stable">
3     <h1 class="title">Formulário de Cadastro</h1>
4   </ion-header-bar>
5   <ion-content>
6     <div class="list">
7       <label class="item item-input item-stacked-label">
8         <span class="input-label">Nome do Aluno</span>
9         <input type="text" ng-model="student.name" placeholder="seu nome" />
10      </label>
11      <label class="item item-input item-stacked-label">
12        <span class="input-label">CPF</span>
13        <input type="number" ng-model="student.cpf" placeholder="apenas números" />
14      </label>
15      <label class="item item-input item-stacked-label">
16        <span class="input-label">Matrícula</span>
17        <input type="number" ng-model="student.code" placeholder="apenas números" />
18      </label>
19      <div class="row">
20        <div class="col">
21          <button class="button button-block button-assertive">cancelar</button>
22        </div>
23        <div class="col">
24          <button class="button button-block button-positive" ng-click="submitForm()>salvar</button>
25        </div>
26      </div>
27    </div>
28  </ion-content>
```

# Plano de Implementação (modelo)

- Criar o modelo de dados:

```
45 //criar o modelo de estudante
46 app.factory('studentModel', function($state){
47   var students = []
48   return {
49     append: function(s, callback){
50       console.info(JSON.stringify(s))
51       students.push(s)
52       console.info(JSON.stringify(students))
53       callback()
54     },
55     listAll: function(){
56       return students
57     }
58   }
59 })
60
```

# Persistência de Dados

- Para trabalhar com persistência de dados usando SQLite no Ionic é necessário delegar este serviço para o Cordova.
- Necessário incluir o plugin do Cordova
- Necessário incluir o módulo do Cordova para AngularJS

# Incluindo Plugin do Cordova

- No diretório da aplicação digite o seguinte comando:

```
$ cordova plugin add \
https://github.com/brodysoft/Cordova-SQLitePlugin.git
```

# Incluindo módulo do Cordova para AngularJS

- Faça o download do arquivo no link abaixo:

```
https://github.com/driftyco/ng-cordova/archive/master.zip
```

- Descompacte e copie todo o conteúdo da pasta **dist** para o diretório **lib/ng-cordova** da sua aplicação.
- Carregue o arquivo JS no seu index.html

```
19      <script src="cordova.js"></script>
20      <script src="lib/ng-cordova/ng-cordova.js"></script>
21
```

# Carregando o módulo na App

- Carregue o módulo ngCordova na app e crie sua base de dados:

```
1 //criar a aplicação
2 var app = angular.module('webapp', ['ionic', 'ngCordova'])
3 //inicializar a execução
4 app.run(function($ionicPlatform, $cordovaSQLite) {
5   $ionicPlatform.ready(function() {
6     if(window.cordova && window.cordova.plugins.Keyboard) {
7       cordova.plugins.Keyboard.hideKeyboardAccessoryBar(true);
8     }
9     if(window.StatusBar) {
10       StatusBar.styleDefault();
11     }
12   //criando o banco de dados na primeira vez se não existir
13   db = $cordovaSQLite.openDB("my.db");
14   $cordovaSQLite.execute(db,
15     "CREATE TABLE IF NOT EXISTS " +
16     "student (" +
17       "code integer primary key, " +
18       "name text, " +
19       "cpf text)"
20   );
21 });
22 })
```

# Carregando o módulo na App

- Modifique o código do modelo para recuperar e salvar os dados no BD:

```
68 app.factory('studentModel', function($state, $cordovaSQLite){  
69   //  
70   var SQL_INSERT = "INSERT INTO student(code, name, cpf) VALUES (?, ?, ?)"  
71   var SQL_SELECT = "SELECT code, name, cpf FROM student"  
72   //  
73   return {  
74     append: function(s, callback){  
75       $cordovaSQLite.execute(db, SQL_INSERT, [s.code, s.name, s.cpf]).then(function(res){  
76         callback()  
77       }, function (err) {  
78         console.error(err);  
79       })  
80     },  
81     listAll: function(callback){  
82       $cordovaSQLite.execute(db, SQL_SELECT, []).then(function(res) {  
83         if(res.rows.length > 0) {  
84           result = []  
85           for(var i = 0; i < res.rows.length; i++){  
86             result.push(res.rows.item(i))  
87           }  
88           callback(result)  
89         } else {  
90           console.log("No results found");  
91         }  
92       }, function (err) {  
93         console.error(err);  
94       });  
95     }  
96   }  
97 })
```