



Aplicaciones Mobile con AngularJS y Ionic Framework

Marcos Reynoso

@mfourky

marcosareynoso@gmail.com

27 de marzo de 2014

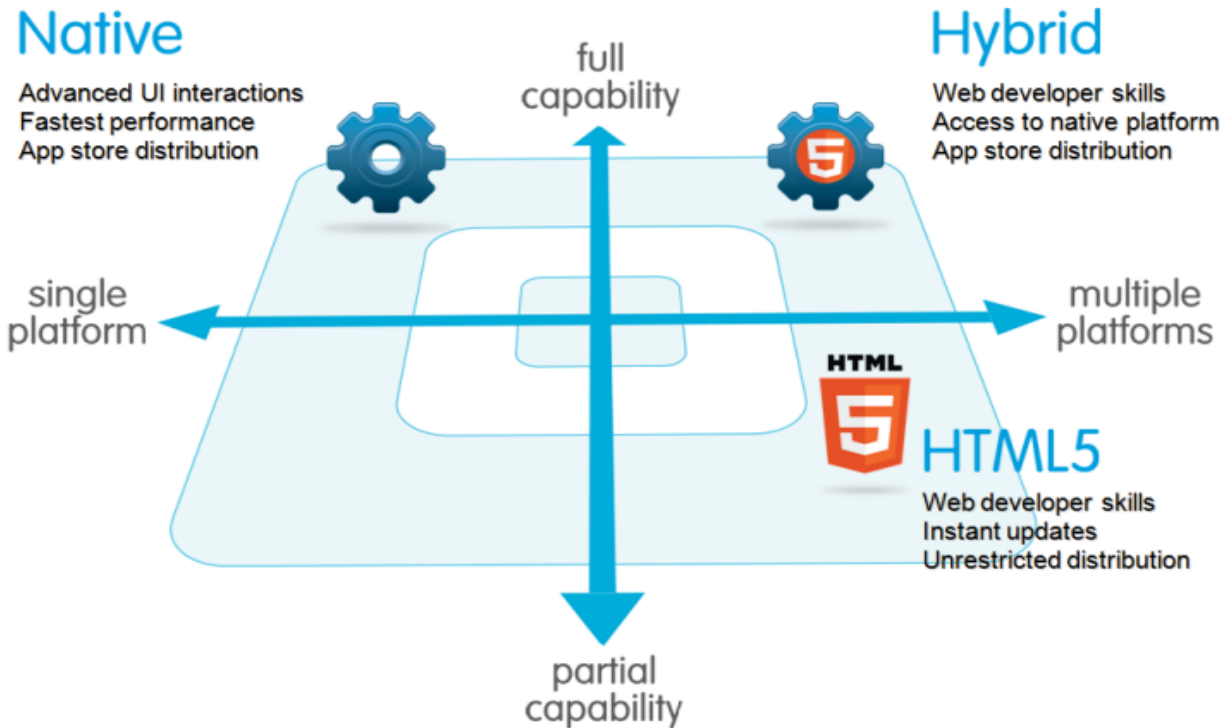


Agenda

- Introducción a las aplicaciones mobile Híbridas con Phonegap / Apache Cordova
- Usos anteriores: JQuery Mobile Angular Adapter, angular-jqm
- Introducción a Ionic Framework
- Aplicación de Ejemplo



Aplicaciones Mobile Híbridas



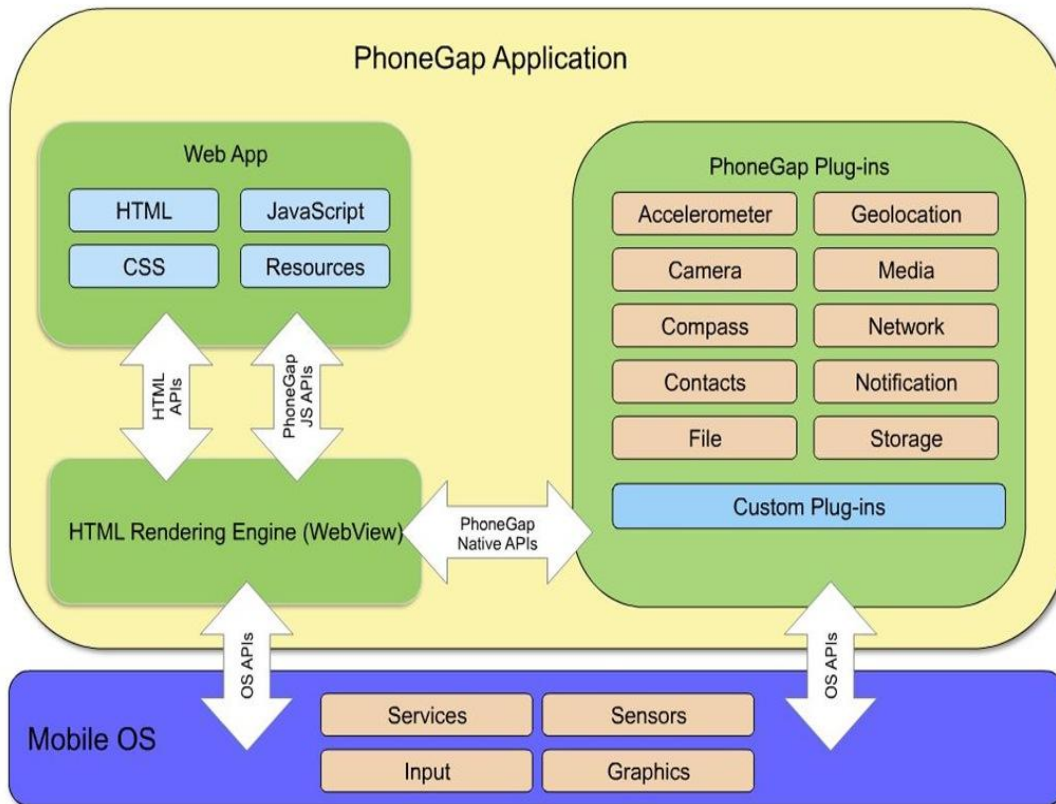


Phonegap / Apache Cordova

- Phonegap es un framework para el desarrollo de aplicaciones mobile híbridas.
- Desarrollado por Nitobi y comprado posteriormente por Adobe Systems.
- En 2011 Adobe dona el proyecto a la fundación Apache, tomando el nombre de Apache Cordova.
- Actualmente Adobe mantiene en paralelo a Phonegap como la versión más comercial.



Arquitectura de Phonegap





Phonegap / Cordova CLI

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

```
$ cordova create ~/KewlApp KewlApp
```

```
$ cd ~/KewlApp
```

```
$ cordova platform add ios android
```

```
$ cordova plugin add http://example.org/Kewlio-1.2.3.tar.gz
```

```
$ cordova build
```

```
$ cordova emulate ios (requiere ios-sim)
```

```
$ cordova serve
```



Frameworks antes de Ionic Framework

- JQuery Mobile Angular Adapter
- Angular-JQM
- Topcoat(sólo UI)



Ionic Framework

Es un framework Open Source, para el desarrollo de aplicaciones mobile, que utilizan tecnologías web y se integra completamente con AngularJS.

Fue desarrollado por la empresa Driftyco y el primer release es de Noviembre de 2013 y actualmente está en la versión 1.0.0 Beta.





Ventajas de Ionic Framework

- Animaciones Aceleradas por Hardware.
- Mínima manipulación del DOM
- No usa JQuery.
- Fácil adaptación del diseño
- Soporte de Phonegap integrado.
- Los componentes UI son directivas de AngularJS.
- Ruteo con AngularUI



Desarrollo de Ionic Framework

- AngularJS
- CSS con Sass
- Gulp como Builder
- Bower
- Karma
- E2E Test con Protractor



Desarrollo Ionic Framework

`npm install && npm install -g gulp protractor`

`gulp o gulp build` to build

`gulp build --release` to build with minification & strip debugs

`gulp watch` to watch and rebuild on change

`gulp karma` to test one-time

`gulp karma-watch` to test and re-run on source change

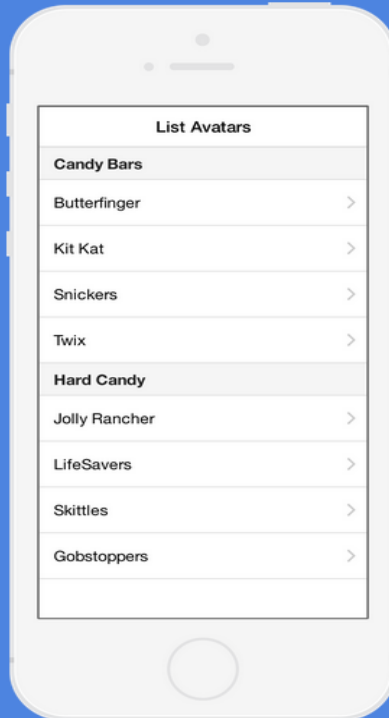
`gulp protractor` to test e2e tests locally

`gulp cloudtest` to run e2e tests in the cloud



Componentes

Lists



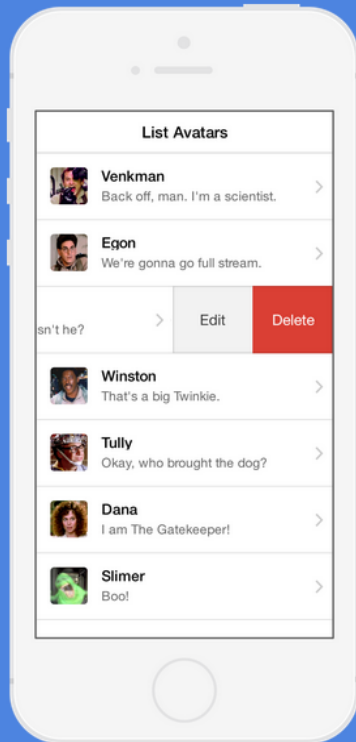
```
<div class="list">
  <div class="item item-divider">
    Candy Bars
  </div>
  <a class="item" href="#">
    Butterfinger
  </a>
  <a class="item" href="#">
    Kit Kat
  </a>
  ...
</div>
```



Componentes

Complex Lists

- AngularJS Directive
- Buttons exposed by swiping
- Reorder
- Delete



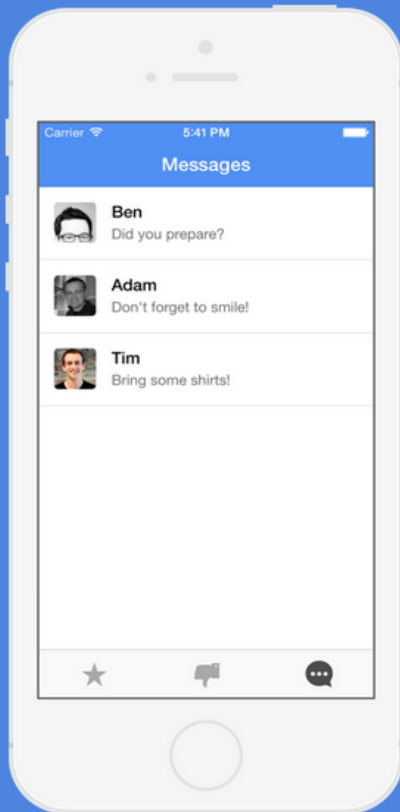
```
<ion-list>
  <ion-item ng-repeat="item in items" option-button
s="buttons" class="item-thumbnail-left">
    
    <h2>{{ item.name }}</h2>
    <p>{{ item.quote }}</p>
  </ion-item>
</ion-list>
```



Componentes

Tabs

- Nested views
- Each tab has its own nav history



```
<ion-tabs tabs-type="tabs-icon-only">

  <ion-tab title="Home" icon="ion-star">
    <ion-nav-view></ion-nav-view>
  </ion-tab>

  <ion-tab title="Feedback" icon="ion-thumbsdown">
    <ion-nav-view></ion-nav-view>
  </ion-tab>

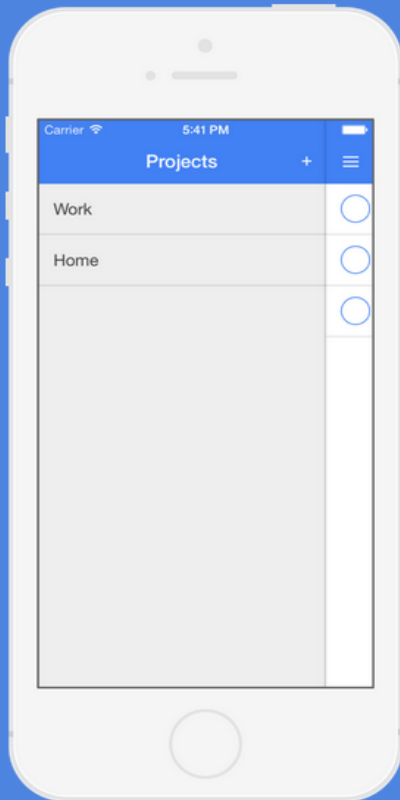
  <ion-tab title="Messages" icon="ion-chatbubble-working">
    <ion-nav-view></ion-nav-view>
  </ion-tab>

</ion-tabs>
```



Componentes

Side Menu



```
<ion-side-menus>

  <ion-pane ion-side-menu-content="">
    <ion-nav-bar type="bar-positive" back-button-type="button-icon" back-button-icon="ion-ios7-arrow-back"></ion-nav-bar>
    <ion-nav-view></ion-nav-view>
  </ion-pane>

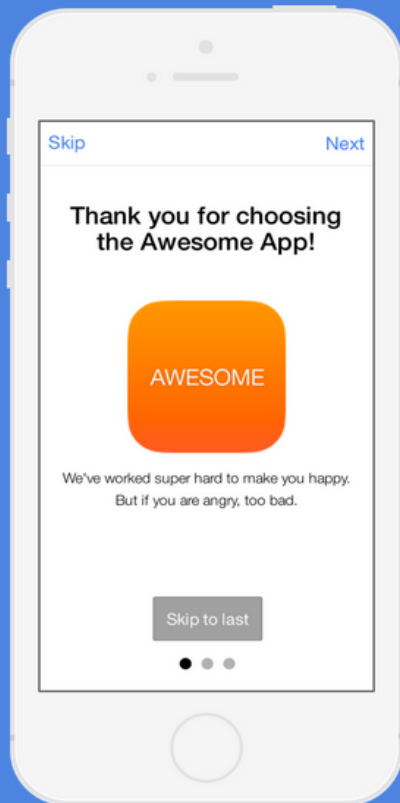
  <ion-side-menu side="left">
    <header class="bar bar-header bar-positive">
      <div class="title">Projects</div>
    </header>
    <ion-content has-header="true">
      <div class="list">
        <a href="#/work" class="item">Work</a>
        <a href="#/home" class="item">Home</a>
      </div>
    </ion-content>
  </ion-side-menu>

</ion-side-menus>
```



Componentes

Slide Box

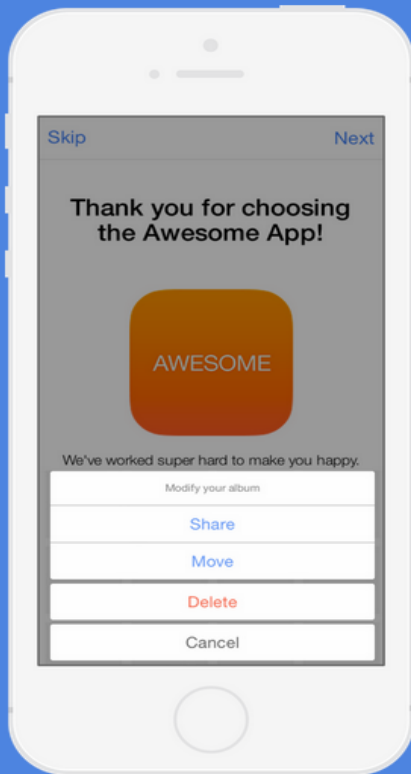


```
<ion-slide-box on-slide-changed="slideChanged(index)">
  <ion-slide>
    Slide 1
  </ion-slide>
  <ion-slide>
    Slide 2
  </ion-slide>
  <ion-slide>
    Slide 3
  </ion-slide>
</ion-slide-box>
```




Componentes

Action Sheet

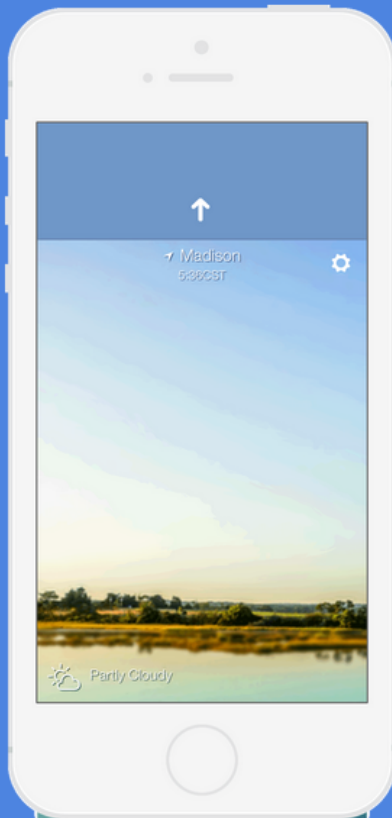


```
$ionicActionSheet.show({  
  titleText: 'Modify your album',  
  buttons: [  
    { text: 'Share' },  
    { text: 'Move' },  
  ],  
  destructiveText: 'Delete',  
  cancelText: 'Cancel',  
  buttonClicked: function(index) {  
    console.log('BUTTON CLICKED', index);  
    return true;  
  }  
});
```



Componentes

Pull to Refresh



```
<ion-content>
  <ion-refresher on-refresh="refreshData()"></ion-refresher>
  <!-- content -->
</ion-content>
```

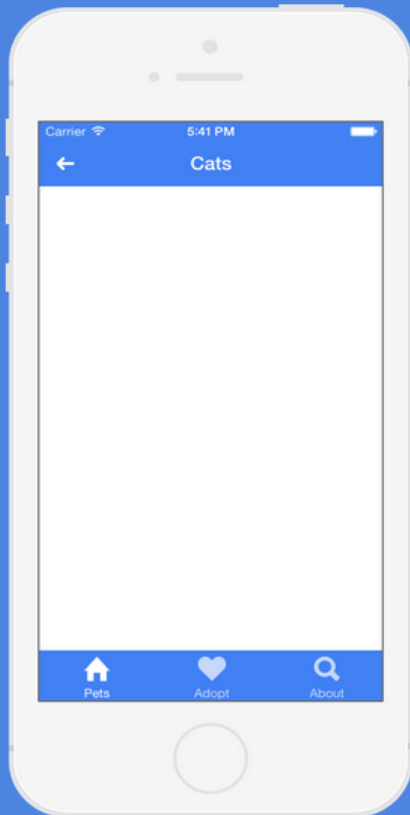


Componentes

Navigation

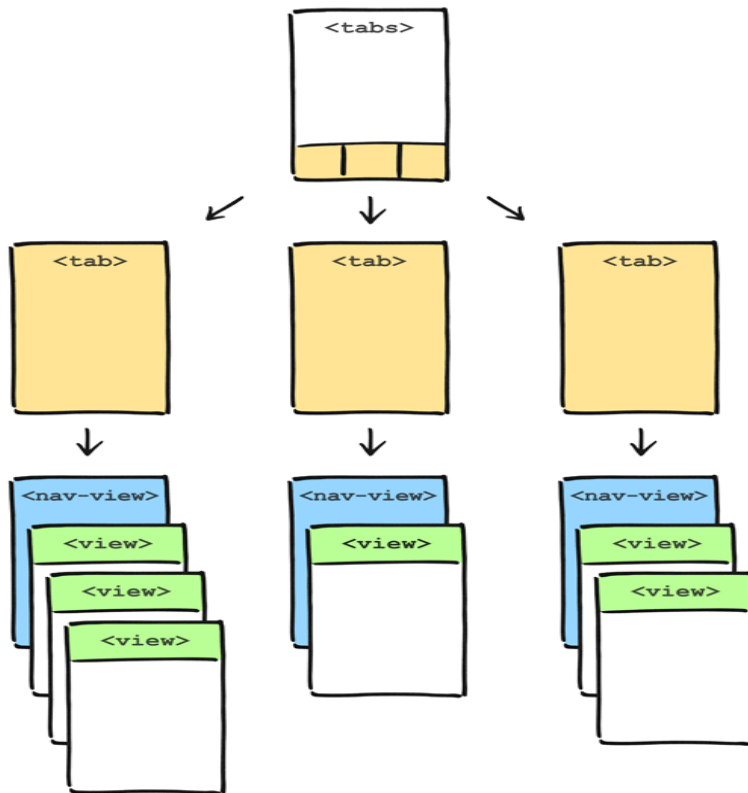
- Shows back button when possible
- Each tab has its own history stack
- Works with Android's back button

```
<ion-nav-bar back-button-icon="ion-chevron-left" back-button-type="button-icon"></ion-nav-bar>  
<ion-nav-view animation="slide-left-right"></ion-nav-view>
```





Angular-UI Router





Ionic CLI

\$ sudo npm install -g ionic

\$ ionic start myApp

\$ ionic platform add ios android

\$ ionic build ios

\$ ionic emulate ios

\$ ionic run ios



Links

- <http://ionicframework.com>
- <http://angularjs.org/>
- <http://phonegap.com>
- <http://cordova.apache.org>
- <https://github.com/angular-ui/ui-router>
- <http://gulpjs.com/>
- <http://sass-lang.com/>
- <https://github.com/angular/protractor>