



ionic

Aplicaciones Móviles Híbridas. Nivel I

Noviembre, 2018



Objetivos del nivel

- Conocer las tecnologías que se utilizan al crear aplicaciones híbridas.
- Aprender a crear aplicaciones con Ionic.
- Ejecutar y simular aplicaciones móviles.

Prerrequisitos del nivel

- Angular Nivel I

Acerca de este manual

Este manual pertenece al Centro de Asesoramiento y Desarrollo Informático C.A. (CADIF1). Para obtener más información sobre este u otros cursos visite nuestra sitio Web www.cadif1.com, escribanos a la dirección de correo cadi@cadif1.com o visítenos en nuestra sede ubicada en la Av. Pedro León Torres con calle 59, Centro Comercial Sotavento, piso 2 oficina 27, Barquisimeto estado Lara, Venezuela. Tlf. 0251-7179247, 0251-4410268.

Las marcas mencionadas en este manual son propiedad de sus respectivos dueños. Copyright 2018. Todos los derechos reservados.

ACADEMIA DE SOFTWARE



Contenido del nivel

Capítulo 1. Introducción a Ionic

- 1.1.- Ionic.
- 1.2.- Instalación.
- 1.3.- Crear un Proyecto.
- 1.4.- Ejecutar la Aplicación.

Capítulo 2. Estructura Del Proyecto

- 2.1.- Contenido Del Proyecto.
- 2.2.- Componente Principal.
- 2.3.- Páginas.

Capítulo 3. Estructura de Las Vistas

- 3.1.- Ion-content.
- 3.2.- Ion-header e Ion-footer.
- 3.3.- Ion-navbar.

Capítulo 4. Componentes de Ionic. Parte 1

- 4.1.- Iconos.
- 4.2.- Botones.
- 4.3.- Tool Bar.

Capítulo 5. Componentes de Ionic. Parte 2

- 5.1.- Cards.
- 5.2.- Listas.
- 5.3.- Slides.

Capítulo 6. Contenido Dinámico

- 6.1.- Binding de Una Vía.
- 6.2.- Ngfor.
- 6.3.- Ngif.

Capítulo 7. Tabs

- 7.1.- Componente Tabs.
- 7.2.- Crear Páginas.
- 7.3.- Cambiar Página Inicial.

Capítulo 8. Interactuando Con el Usuario

- 8.1.- Programar Eventos.
- 8.2.- Toats.
- 8.3.- Alerts.

Capítulo 9. Prompt

- 9.1.- Mostrar Prompt.
- 9.2.- Obtener Los Datos.
- 9.3.- Confirms.

Capítulo 10. Inputs

- 10.1.- Tipos de Inputs.
- 10.2.- Ngmodel.

Capítulo 11. Navegación. Parte 1

- 11.1.- Modelo de Navegación.
- 11.2.- Push y Pop.
- 11.3.- Ventanas Modales.

Capítulo 12. Navegación. Parte 2

- 12.1.- Paso de Parámetros.
- 12.2.- Recepción de Parámetros.

Capítulo 13. Providers

- 13.1.- Concepto.
- 13.2.- Creación de Providers.
- 13.3.- Uso de Providers.

Capítulo 14. HttpClient

14.1.- Get.

14.2.- Post.



Capítulo 1. INTRODUCCIÓN A IONIC

1.1.- Ionic

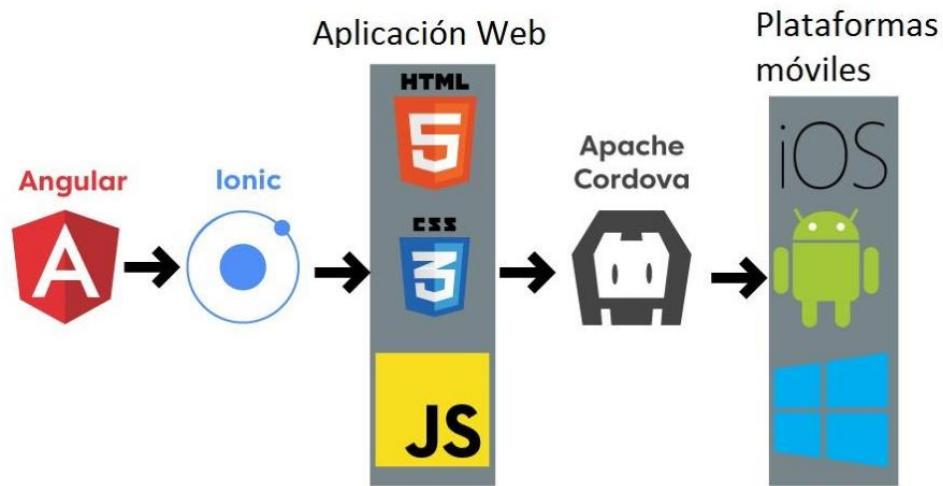
Cuando se trata de crear aplicaciones que van a ejecutarse en un dispositivo móvil existen varias alternativas:

- aplicación Web: una aplicación web se crea con tecnología estándares (HTML, JS y CSS) que son compatibles con cualquier navegador. Se asume que todos los sistemas operativos para dispositivos móviles tienen un navegador, el usuario puede abrir la URL de la aplicación y ejecutarla. Las aplicaciones web pueden ejecutarse en cualquier sistema operativo, se usa la misma aplicación para todos los sistemas, no se instalan, pero no pueden acceder a ciertas características del hardware y generalmente su desempeño depende de la velocidad de la red.
- aplicación nativa: se crean con tecnologías particulares de cada sistema operativo. Se compilan y se instalan. Se debe crear una versión para cada sistema operativo, pueden acceder a todas las características del hardware, su desempeño generalmente es mejor y la experiencia de usuario generalmente también es más agradable.

Son 2 extremos totalmente opuestos.

Las aplicaciones híbridas son una alternativa que ofrece un punto intermedio entre las 2 anteriores, porque permiten al programador crear una sola versión de la aplicación utilizando tecnologías estándares de la web, y a la vez crear una aplicación nativa particular para cada sistema operativo, logrando así un mejor desempeño y mejorando la experiencia del usuario final.

Ionic es un framework para crear tanto aplicaciones híbridas como aplicaciones nativas para Android, IOs y Windows. La primera versión fue liberada en 2013 usando AngularJs. A partir de la versión 3 usa TypeScript y Angular, es por tal razón que tener conocimiento en ambas tecnologías facilita el aprendizaje de la plataforma. Existen 2 versiones, la versión gratuita y la versión Pro. La página oficial del framework es <https://ionicframework.com/>.



1.2.- Instalación

Ionic es un módulo de Node.js, por lo tanto, para instalarlo es pre requisito tener instalado este software. Para verificar si está instalado se ejecuta el comando: node -v.

```
C:\Windows\system32\cmd.exe
E:\Dropbox\cadi\cursos\cursolonionic>npm install -g ionic
C:\Users\jose\AppData\Roaming\npm\ionic -> C:\Users\jose\AppData\Roaming\npm\node_modules\ionic\bin\ionic
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\ionic\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ ionic@3.20.0
added 260 packages from 165 contributors in 155.396s
E:\Dropbox\cadi\cursos\cursolonionic>
```

Luego de instalado, se puede verificar las versiones de las herramientas instaladas usando el comando:

ionic info:

```
E:\Dropbox\cadi\cursos\cursolonicejemplo>ionic info
cli packages: C:\Users\jose\AppData\Roaming\npm\node_modules>
  @ionic/cli-utils  : 1.19.2
  ionic (Ionic CLI) : 3.20.0

local packages:
  @ionic/app-scripts : 3.1.11
  Ionic Framework    : ionic-angular 3.9.2

System:
  Node      : v8.11.0
  npm       : 6.2.0
  OS        : Windows 7

Misc:
  backend   : pro

E:\Dropbox\cadi\cursos\cursolonicejemplo>
```

1.3.- Crear un Proyecto

Para facilitar el trabajo inicial al crear un nuevo proyecto Ionic incluye una herramienta muy útil: las plantillas. Una plantilla es proyecto base, que contiene además de la estructura general del proyecto, un conjunto de vistas (páginas) con un diseño inicial, que sirve de punto de partida del desarrollo de la aplicación. Para crear la primera aplicación se ejecuta el comando:

ionic start "nombre" "plantilla"

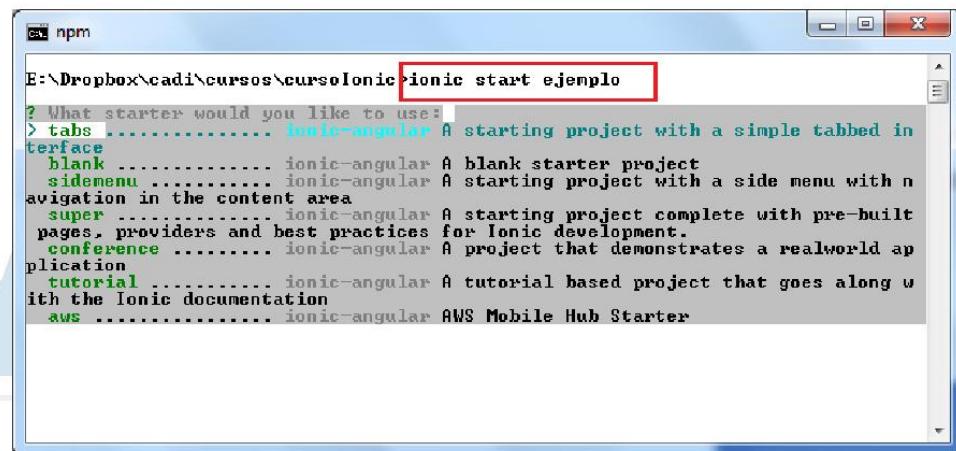
Donde el "nombre" será el nombre de la aplicación que se creará (se creará una carpeta con ese nombre) y "plantilla" es la plantilla base que se usará para crear la aplicación. Por ejemplo:

```
E:\Dropbox\cadi\cursos\cursolonic>ionic start ejemplo blank
[U] Creating directory .\ejemplo - done!
[U] Downloading and extracting blank starter - done!
```

Hay varios tipos de plantillas que por defecto trae incorporado Ionic, entre ellas están:

- tabs : una páginas con 3 tabs.
- sidemenu: una página con un menú a la izquierda
- blank: una página en blanco
- super: un proyecto con un asistente de inicio y con 14 diseños de páginas listos para usar.
- tutorial: un proyecto de inicio guiado.

Si no se especifica una plantilla, Ionic CLI pregunta cuál plantilla se usará:



The screenshot shows a Windows command prompt window titled 'npm'. The command entered is 'ionic start ejemplo'. A red box highlights the command line. Below it, a list of available starters is displayed:

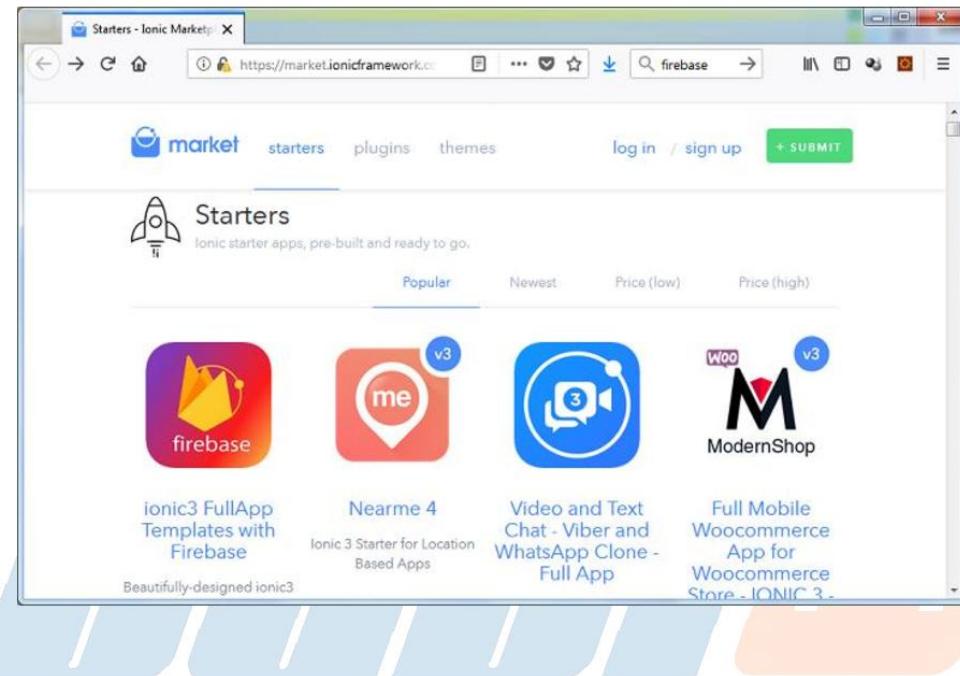
```
? What starter would you like to use?
> tabs ..... ionic-angular A starting project with a simple tabbed interface
  blank ..... ionic-angular A blank starter project
  sidemenu ..... ionic-angular A starting project with a side menu with navigation in the content area
  super ..... ionic-angular A starting project complete with pre-built pages, providers and best practices for Ionic development.
  conference ..... ionic-angular A project that demonstrates a realworld application
  tutorial ..... ionic-angular A tutorial based project that goes along with the Ionic documentation
  aws ..... ionic-angular AWS Mobile Hub Starter
```

Al crear la aplicación, se instalarán en la carpeta del proyecto todas las dependencias necesarias para que funcione la aplicación. Esto puede tardar unos minutos mientras se descargan los archivos en la carpeta node_modules. Para evitar que se descarguen las dependencias se agrega al comando de creación del proyecto el parámetro --no-deps. El comando completo es:

```
ionic start "ejemplo" "blank" --no-deps
```

También se pregunta si desea usar compatibilidad con Cordova. Por el momento se puede responder "No" porque esto se necesitará para la creación de la aplicación distribuible en los dispositivos móviles. En tiempo de desarrollo, se puede omitir porque se puede agregar posteriormente al momento de necesitarlo.

Adicionalmente, existe un mercado de plantillas listas para usar que se pueden descargar del sitio <https://market.ionicframework.com/starters/>, algunas gratuitas y otras pagas. Se pueden descargar y usar.

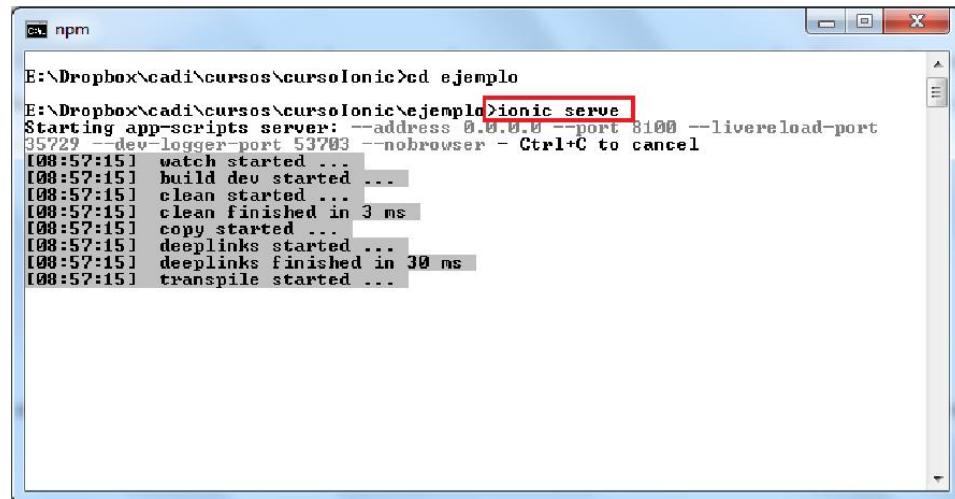


1.4.- Ejecutar la Aplicación

ACADEMIA DE SOFTWARE

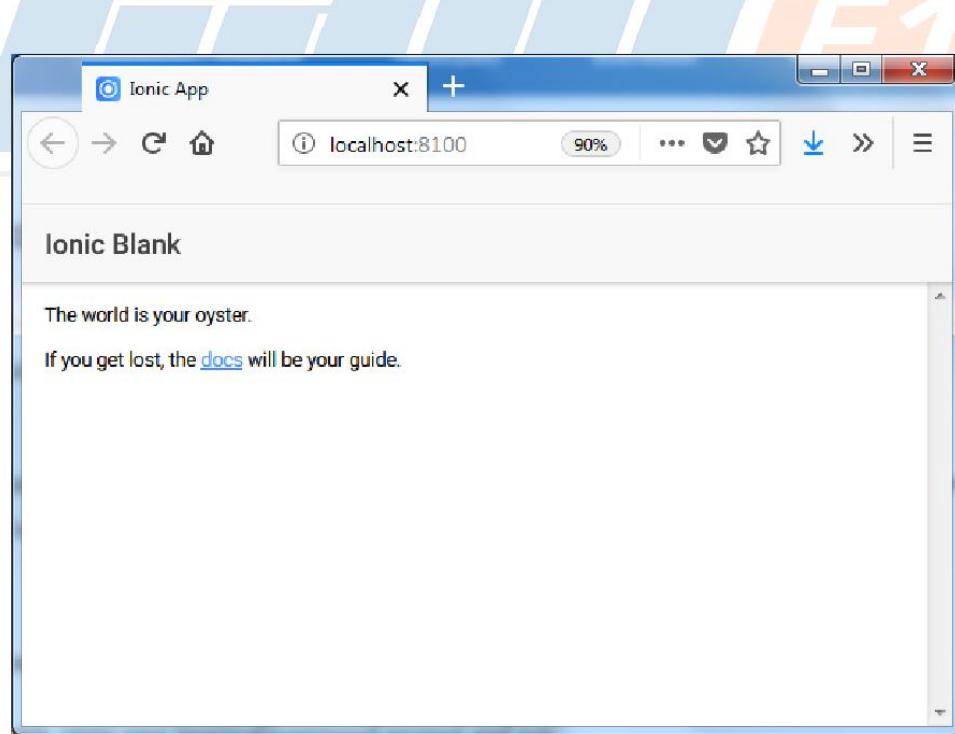
Luego de creada la aplicación, hay que moverse a la carpeta nueva y ejecutar el comando:

```
ionic serve
```

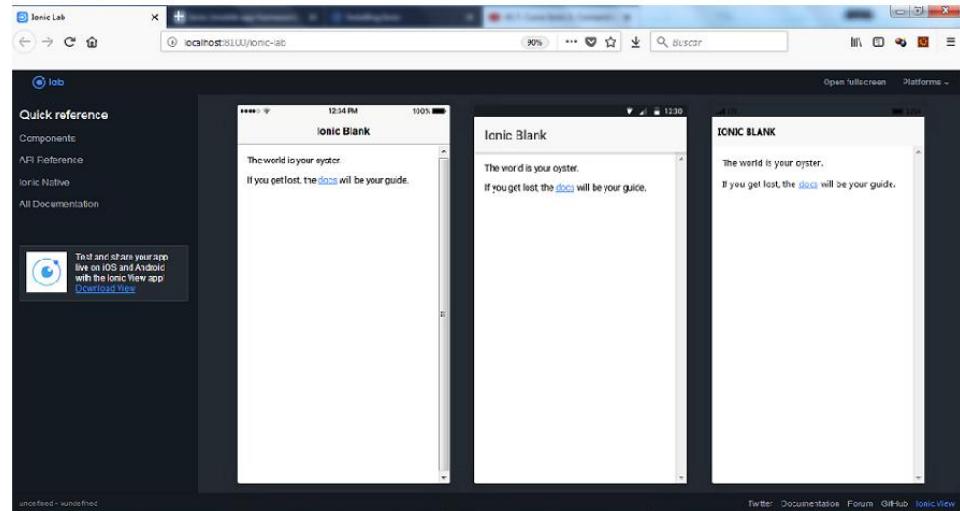


```
E:\Dropbox\cadi\cursos\cursolonic\ejemplo>ionic serve
Starting app-scripts server: --address 0.0.0.0 --port 8100 --livereload-port
35729 --dev-logger-port 53703 --nobrowser - Ctrl+C to cancel
[08:57:15] watch started ...
[08:57:15] build dev started ...
[08:57:15] clean started ...
[08:57:15] clean finished in 3 ms
[08:57:15] copy started ...
[08:57:15] deeplinks started ...
[08:57:15] deeplinks finished in 30 ms
[08:57:15] transpile started ...
```

Este comando iniciará un servidor web local para probar la aplicación en un navegador. El navegador debería abrirse automáticamente accediendo a la url <http://localhost:8100>



Otra forma de ejecutar la aplicación es visualizándola en un marco que simula al dispositivo móvil. El comando es: ionic serve --lab



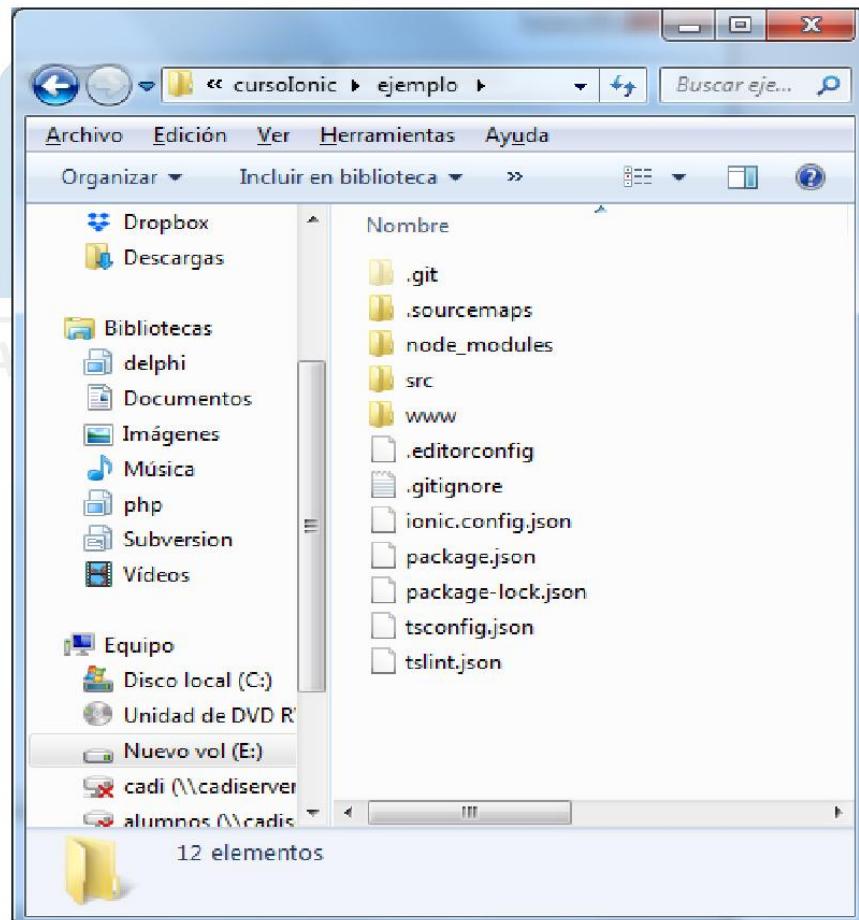
En ocasiones hace falta instalar el paquete @ionic/lab para poder ejecutar en este modo. Se usa el comando npm i @ionic/lab -g. También se puede visualizar la aplicación en un dispositivo móvil real, instalando la aplicación "IonicDevApp". Esta aplicación, siempre y cuando el dispositivo esté conectado en la misma red wifi de la computadora donde se está programando, detecta automáticamente si se está ejecutando una aplicación, para luego mostrarla en una lista donde el usuario puede seleccionarla y así probar su ejecución en el dispositivo.

Capítulo 2. ESTRUCTURA DEL PROYECTO

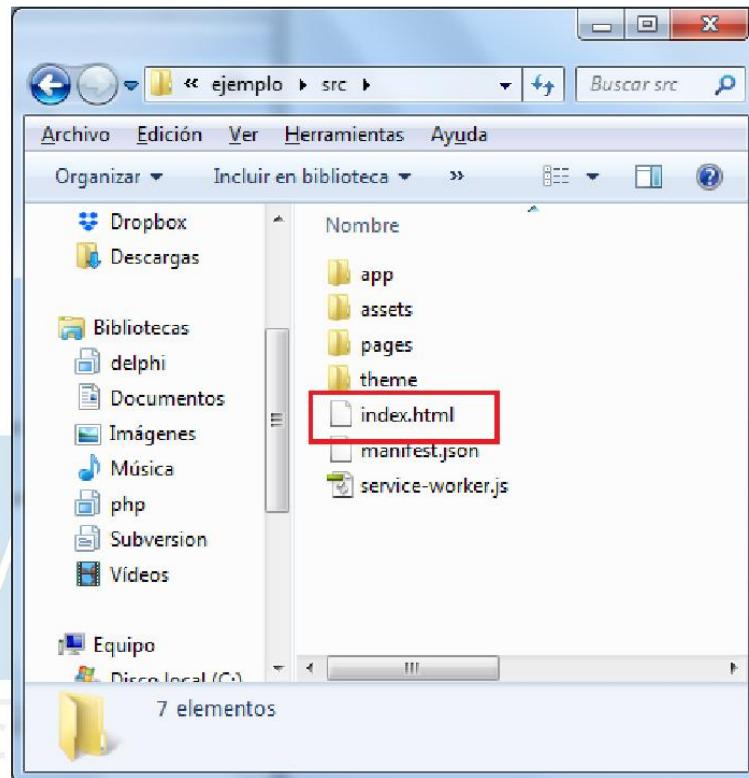
2.1.- Contenido Del Proyecto

Ionic usa Angular, por lo tanto, la estructura de un proyecto es la misma. En la carpeta del proyecto se crean varios archivos y carpetas. Algunos archivos son de configuración de git, de npm y de TypeScript. Se crean 3 carpetas:

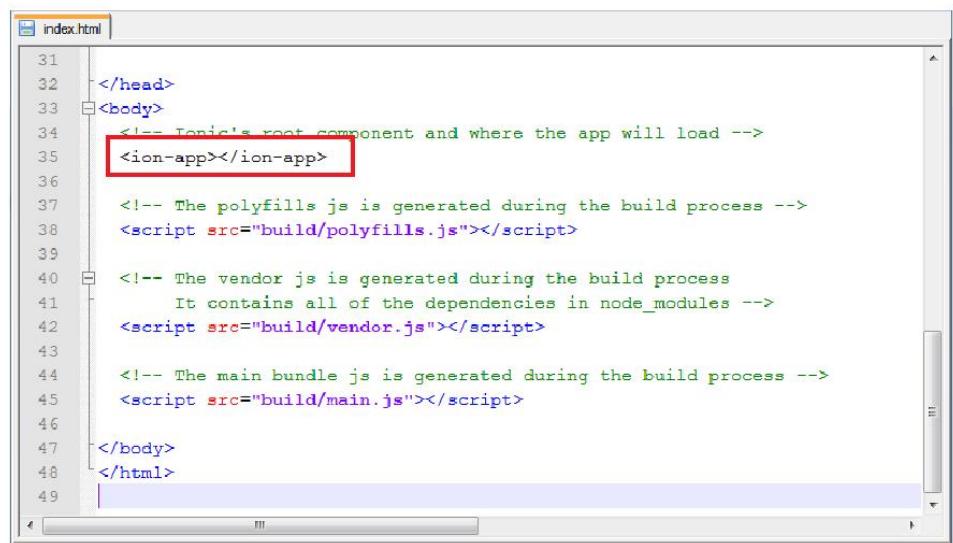
- node_modules: módulos de node (dependencias) necesarios para compilar el proyecto
- src: se encuentra el código fuente del proyecto.
- www: el proyecto compilado y listo para ejecutarse en un navegador



En la carpeta "src" está el archivo "index.html", que es el punto de entrada de la aplicación. En este archivo hay una referencia a la etiqueta: <ion-app>, que indica que ahí se cargará la aplicación.



El contenido del archivo index es el siguiente:



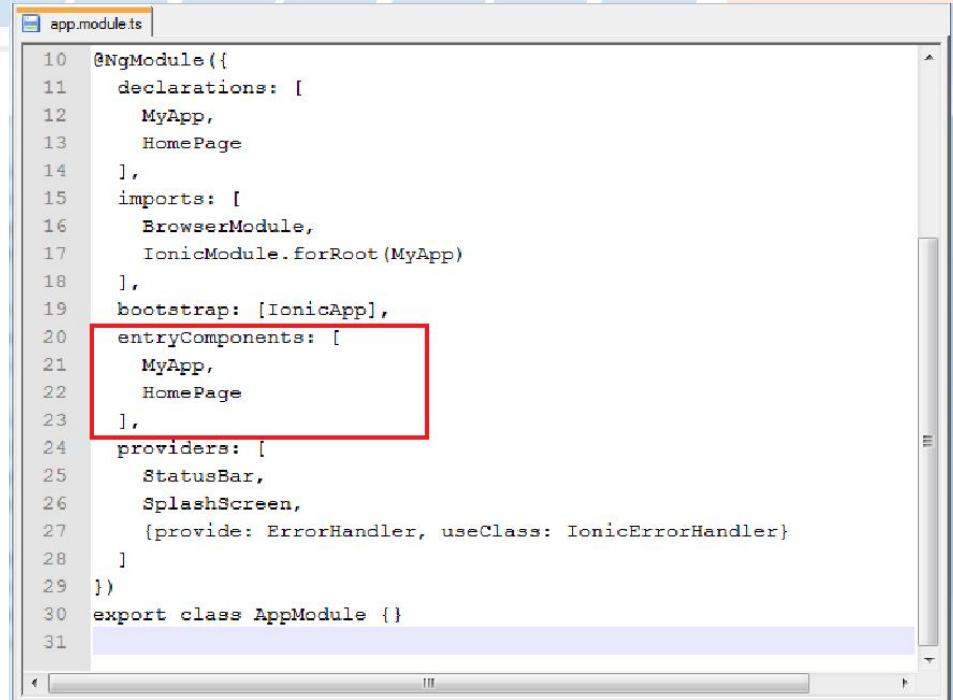
```

31  </head>
32  <body>
33  <!-- Ionic's root component and where the app will load -->
34  <ion-app></ion-app>
35
36  <!-- The polyfills js is generated during the build process -->
37  <script src="build/polyfills.js"></script>
38
39  <!-- The vendor js is generated during the build process
40      It contains all of the dependencies in node_modules -->
41  <script src="build/vendor.js"></script>
42
43  <!-- The main bundle js is generated during the build process -->
44  <script src="build/main.js"></script>
45
46  </body>
47  </html>
48
49

```

2.2.- Componente Principal

La base de las aplicaciones Angular es el uso de módulos y componentes. Ionic crea por defecto un módulo donde se hace referencia a los elementos del proyecto:

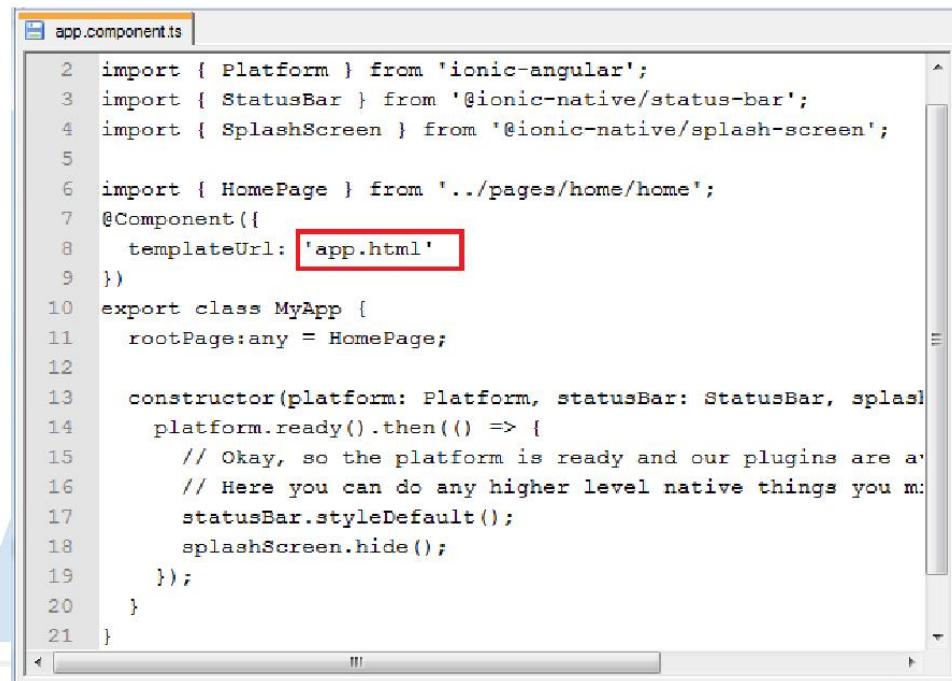


```

10  @NgModule({
11    declarations: [
12      MyApp,
13      HomePage
14    ],
15    imports: [
16      BrowserModule,
17      IonicModule.forRoot(MyApp)
18    ],
19    bootstrap: [IonicApp],
20    entryComponents: [
21      MyApp,
22      HomePage
23    ],
24    providers: [
25      StatusBar,
26      SplashScreen,
27      {provide: ErrorHandler, useClass: IonicErrorHandler}
28    ]
29  })
30 export class AppModule {}
31

```

Un componente permite dividir la aplicación en partes reusables. Esta conformado por un archivo TypeScript y un archivo HTML (típicamente). Al crear una aplicación, Ionic crea un componente principal, que será el que se ejecutará al iniciar la aplicación.



```
app.component.ts
1 import { Platform } from 'ionic-angular';
2 import { StatusBar } from '@ionic-native/status-bar';
3 import { SplashScreen } from '@ionic-native/splash-screen';
4
5
6 import { HomePage } from '../pages/home/home';
7 @Component({
8   templateUrl: 'app.html'
9 })
10 export class MyApp {
11   rootPage:any = HomePage;
12
13   constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
14     platform.ready().then(() => {
15       // Okay, so the platform is ready and our plugins are available.
16       // Here you can do any higher level native things you need to.
17       statusBar.styleDefault();
18       splashScreen.hide();
19     });
20   }
21 }
```

2.3.- Páginas

Una aplicación móvil esta compuesta por varias vistas o interfaces, que se muestran al usuario en la medida que éste interactúa con la aplicación. Ionic maneja cada interfaz con el nombre de "pages" (páginas en español). Una página es un componente de Angular. Al crear un proyecto en blanco, se crea una página (la principal). Más adelante se verá como agregar nuevas páginas al proyecto.

```

app.component.ts
1 import { Platform } from 'ionic-angular';
2 import { StatusBar } from '@ionic-native/statusBar';
3 import { SplashScreen } from '@ionic-native/splash-screen';
4
5 import { HomePage } from '../pages/home/home';
6
7 @Component({
8   templateUrl: 'app.html'
9 })
10 export class MyApp {
11   rootPage:any = HomePage;
12
13   constructor(platform: Platform, statusBar: StatusBar, splashScreen: SplashScreen) {
14     platform.ready().then(() => {
15       // Okay, so the platform is ready and our plugins are available.
16       // Here you can do any higher level native things you might need.
17       statusBar.styleDefault();
18       splashScreen.hide();
19     });
20   }
21 }

```

```

app.html
1 <ion-nav [root]="rootPage"></ion-nav>
2
3
4
5
6
7
8
9
10
11
12
13

```

```

home.ts
1 import { Component } from '@angular/core';
2 import { NavController } from 'ionic-angular';
3
4 @Component({
5   selector: 'page-home',
6   templateUrl: 'home.html'
7 })
8 export class HomePage {
9
10   constructor(public navCtrl: NavController) {
11
12   }
13 }

```

The diagram shows three code editors. The first editor on the left contains the `app.component.ts` file. The second editor in the middle contains the `app.html` file. The third editor on the right contains the `home.ts` file. Red arrows point from the `rootPage` assignment in `app.component.ts` to the `<ion-nav [root]` tag in `app.html`, and from the `templateUrl` assignment in `home.ts` to the `templateUrl` declaration in `app.html`.

En la definición de la clase, se hace referencia al archivo HTML que contiene la vista:

```

home.ts
1 import { Component } from '@angular/core';
2 import { NavController } from 'ionic-angular';
3
4 @Component({
5   selector: 'page-home',
6   templateUrl: 'home.html'
7 })
8 export class HomePage {
9
10   constructor(public navCtrl: NavController) {
11
12   }
13 }

```

```

home.html
1 <ion-header>
2   <ion-navbar>
3     <ion-title>
4       Mi Primer App
5     </ion-title>
6   </ion-navbar>
7 </ion-header>
8
9 <ion-content padding>
10   The world is your oyster.
11   <p>
12     Hola Mundo.
13   </p>
14 </ion-content>

```

The diagram shows two code editors. The left editor contains the `home.ts` file. The right editor contains the `home.html` file. A red arrow points from the `templateUrl` assignment in `home.ts` to the `templateUrl` declaration in `home.html`.

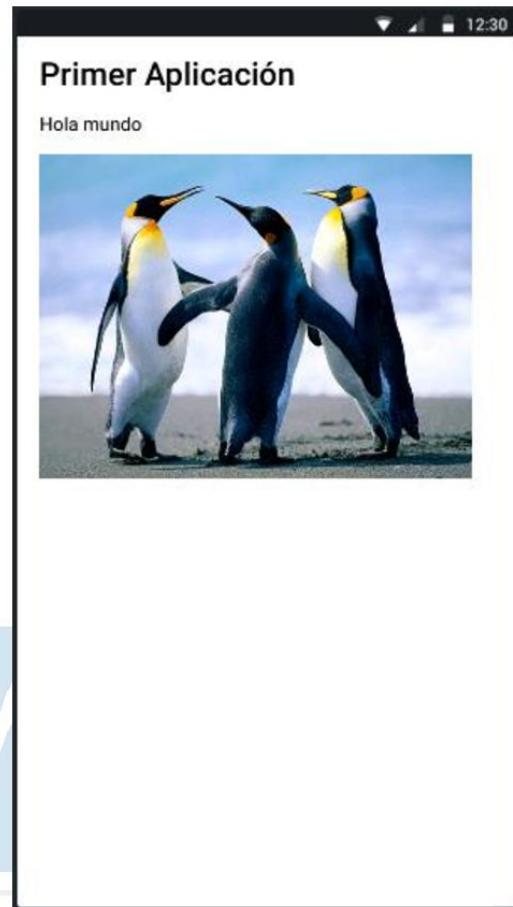
Capítulo 3. ESTRUCTURA DE LAS VISTAS

3.1.- ion-content

Al ser Ionic una Framework para crear aplicaciones móviles usando tecnologías Web (HTML, JavaScript y CSS) es natural esperar que las interfaces se construyan usando etiquetas HTML. Ionic incorpora un conjunto de etiquetas de uso especial, de éstas la más básica, la etiqueta "ion-content". Esta etiqueta se utiliza para especificar el contenido de la interfaz. Todo lo que se desee mostrar en la interfaz, salvo algunas particulares, debe estar entre estas etiquetas. Por ejemplo:

```
home.html x
1  <ion-content padding>
2    <h2>
3      | Primer Aplicación
4    </h2>
5    <p>
6      | Hola mundo
7    </p>
8    
9
10   </ion-content>
```

Si los elementos agregados adentro del contenido necesitan "scroll", se agregará automáticamente. Sólo puede haber una etiqueta "ion-content" en una interfaz. En el ejemplo anterior se muestra como agregar en la interfaz algunos elementos HTML estándar, tales como títulos (h1), párrafos (p) e imágenes (img). El resultado de este ejemplo se muestra en la siguiente imagen:



3.2.- Ion-header e ion-footer

Ion-header es una etiqueta contenedora, que muestra información en el encabezado de la interfaz. Puede contener toolbars (barras de herramientas) y navbars (barra de navegación).

```
home.html x  
1 <ion-header>  
2   <ion-title>  
3     | Curso de Ionic  
4   </ion-title>  
5 </ion-header>  
6  
7 <ion-content padding>  
8   <h2>  
9     | Primer Aplicación  
10   </h2>  
11   <p>  
12     | Hola mundo  
13   </p>  
14     
15 </ion-content>
```

El resultado del ejemplo anterior es el siguiente:



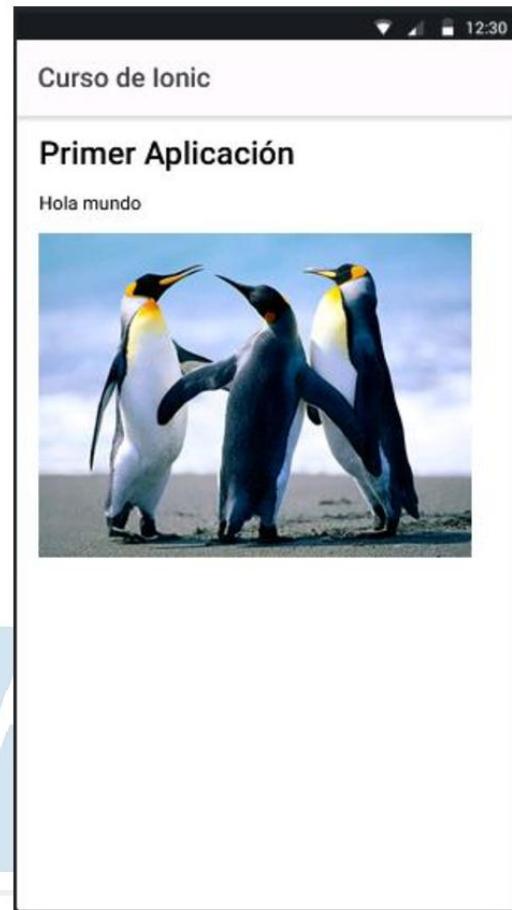
Asimismo se usa la etiqueta ion-footer, que muestra información el pie de la interfaz

3.3.- ion-navbar

El Navbar actúa como una barra de herramientas para la navegación, que incluye el botón "Atras" (Back) que se agrega automáticamente durante la navegación. Un Navbar puede contener un título (ion-title), cualquier cantidad de botones, una barra de búsqueda, entre otros. El siguiente es un ejemplo del uso del Navbar con un título:

```
home.html x
1  <ion-header>
2  |  <ion-navbar>
3  |  |  <ion-title>
4  |  |  |  Curso de Ionic
5  |  |  </ion-title>
6  |  |  </ion-navbar>
7  |  </ion-header>
8
9  <ion-content padding>
10 |  <h2>
11 |  |  Primer Aplicación
12 |  </h2>
13 |  <p>
14 |  |  Hola mundo
15 |  </p>
16 |  
17 </ion-content>
```

El resultado del ejemplo anterior es el que se muestra en la siguiente imagen. Se puede notar que no hay mucha diferencia con el ejemplo anterior donde no había Navbar:



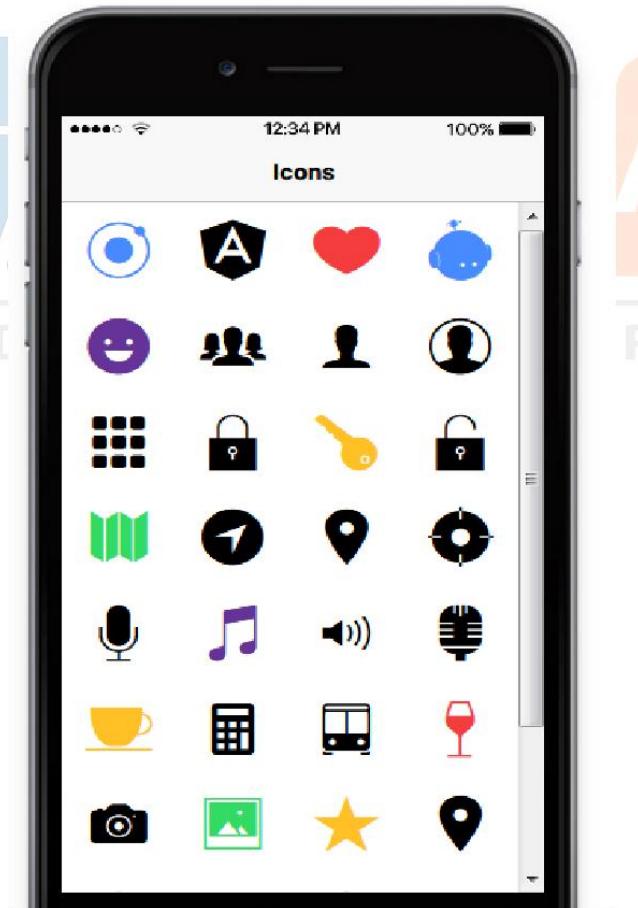
Capítulo 4. COMPONENTES DE IONIC. PARTE 1

4.1.- Iconos

Ionic trae más de 700 íconos incorporados. Su uso es muy simple, solamente utilizando las etiquetas "ion-icon". La propiedad principal es "name", donde se especifica el ícono que se desea mostrar. Por ejemplo:

```
<ion-icon name="heart"></ion-icon>
```

En la url <https://ionicframework.com/docs/ionicons/> se puede encontrar todos los íconos que vienen con el framework.



Algunos ejemplos:

Name	iOS	iOS-Outline	Material Design
add	+	+	+
add-circle	⊕	⊕	⊕
alarm	⏰	⌚	⌚
albums	📁	📁	📁
alert	❗	❗	❗
american-football	🏈	🏈	🏈

4.2.- Botones

Los botones son una forma esencial de interactuar con una aplicación y navegar entre sus interfaces. Los botones consisten de un texto y/o un ícono.

Por razones de accesibilidad, en Ionic los botones utilizan la etiqueta estándar `<button>`, pero es mejorado agregándole la propiedad "ion-button".



Los botones pueden tener alguna de las siguientes propiedades:

- color: indica el color de fondo del botón. Si no se especifica se pintará azul.
- outline: indica que el botón tiene color de fondo.
- clear: indica que el botón no tendrá color de fondo (es excluyente con la anterior).
- block: indica que el botón ocupará el ancho del bloque donde se encuentra.
- full: indica que el ancho del botón ocupará todo el ancho de la pantalla.
- round: indica que tendrá botones redondeados.
- small o large: indica el tamaño del botón.

```
<!-- Colors -->
<button ion-button>Default</button>

<button ion-button color="secondary">Secondary</button>

<button ion-button color="danger">Danger</button>

<button ion-button color="light">Light</button>

<button ion-button color="dark">Dark</button>

<!-- Shapes -->
<button ion-button full>Full Button</button>

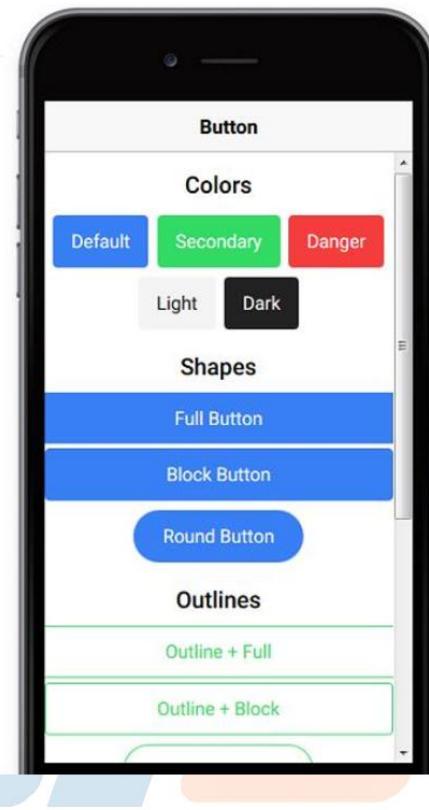
<button ion-button block>Block Button</button>

<button ion-button round>Round Button</button>

<!-- Outline -->
<button ion-button full outline>Outline + Full</button>

<button ion-button block outline>Outline + Block</button>

<button ion-button round outline>Outline + Round</button>
```

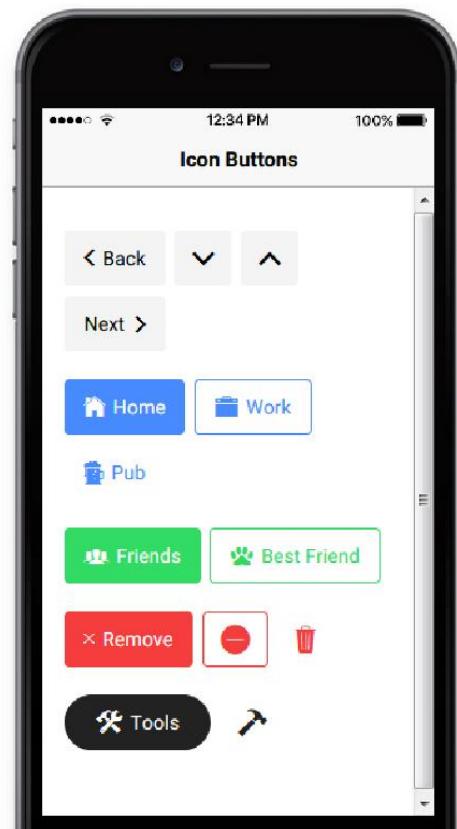


Los botones pueden tener íconos incorporados, por ejemplo:

```
<!-- Float the icon left -->
<button ion-button icon-start>
  <ion-icon name="home"></ion-icon>
  Left Icon
</button>

<!-- Float the icon right -->
<button ion-button icon-end>
  Right Icon
  <ion-icon name="home"></ion-icon>
</button>

<!-- Only icon (no text) -->
<button ion-button icon-only>
  <ion-icon name="home"></ion-icon>
</button>
```



4.3.- Tool Bar

Un toolbar (barra de herramientas) es una barra genérica que puede ser usado como header, sub header, footer o sub footer. Ion-toolbar está basada en flexbox, por lo tanto, no importa cuántas toolbar tenga la página, estas se mostrarán y el ion-contents se ajustará correctamente.

```
<ion-header>
  <ion-toolbar>
    <ion-title>Toolbar</ion-title>
  </ion-toolbar>
</ion-header>

<ion-content></ion-content>

<ion-footer>
  <ion-toolbar>
    <ion-title>Footer</ion-title>
  </ion-toolbar>
</ion-footer>
```



ACADEMIA DE SOFTWARE

Se pueden agregar botones a los toolbars, tanto del header como del footer. Antes, debe agregarse un componente "ion-buttons", que agrupa a uno o varios botones y puede dársele el atributo "start" o "end" para ubicar los botones que contiene al inicio o al final de la barra. Por ejemplo:

```
<ion-header>
  <ion-toolbar>
    <ion-buttons start>
      <button ion-button icon-only color="royal">
        <ion-icon name="search"></ion-icon>
      </button>
    </ion-buttons>
    <ion-title>Send To...</ion-title>
    <ion-buttons end>
      <button ion-button icon-only color="royal">
        <ion-icon name="person-add"></ion-icon>
      </button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```

