

The Ionic Framework logo is a stylized '@' symbol composed of concentric circles and a small dot, rendered in a light blue color. It is centered in the background of the slide.

IONIC FRAMEWORK 2

Formador: Vladimir Bataller

Requisitos:

- HTML y CSS
- Javascript.
- Angular 2.

Características

Ionic es un framework para crea aplicaciones móviles híbridas (basdas en html, css y Javascript) . Ionic 2 es una completa redefinición de Ionic basada en Angular 2.

- Agrega automáticamente estilos diferentes para cada sistema operativo móvil, para, de ese modo, asemejarse al estilo de las aplicaciones nativas de cada sistema.
- Está basado en Apache Cordova y Angular 2.
- Se desarrolla en TypeScript, Sass y HTML5.
- Tiene múltiples componentes que permiten crear cualquier tipo de app móvil.

Historia

Crear una nueva aplicación

Para generar la estructura de una nueva aplicación se emplea la herramienta **@angular/cli (Common Line Interface)** cuya url es la siguiente: <https://cli.angular.io>, aunque se puede instalar mediante npm.

Para empezar a trabajar con Ionic, se debe ejecutar los siguientes comandos:

```
npm -g install ionic cordova  
ionic start --v2 nombreProyecto plantilla  
ionic serve -l
```

Esto abrirá el navegador con la siguiente url: **http://localhost:8100**

Si se emplea la siguiente url **http://localhost:8100/ionic-lab** se puede ver la misma app en varios S.O.

plantilla puede ser: *blank*, *tabs* o *sidemenu* (adicionalmente hay otra plantilla llamada *tutorial*)

Al generar un proyecto, se crea una jerarquía de carpetas. En la ruta **src/app** es donde está el código fuente de la aplicación.

Para editar el código, se puede emplear cualquier editor, pero se recomienda uno de los siguientes:

- Visual Studio Code
- Atom

Navegación

El objeto NavController de Ionic permite navegar a otra página (push), volver a la página anterior (pop) o reestablecer la navegación a una nueva pagina (setRoot).

El **NavController** se puede recibir inyectado en el constructor del componentes de una página.

```
constructor(public navCtrl:NavController, public navParams:NavParams){  
    this.navParams.get("num"); //Lectura del atributo num enviado desde la página de origen.  
}  
botonPulsado(event, item) {  
    //El segundo argumento permite pasar datos a la página de destino de la navegación  
    this.navCtrl.push(DestinoComponentPage, datos);  
}
```

Nota: Nav y Tab herendan de NavController.

Para volver a la página de la cual se provenía, se ejecutará:

```
this.navCtrl.pop();
```

De todas formas si la página tiene <ion-navbar> automáticamente mostrará botón de volver atrás (que automáticamente llama a pop() cuando es pulsado).

app.html

La plantilla ligada al componente de la aplicación define el menú de la aplicación (ion-menu) y el contenedor (ion-nav) donde se mostrarán las páginas de la aplicación.

```
<ion-menu [content]="content">
  <ion-header>
    <ion-toolbar>
      <ion-title>Pages</ion-title>
    </ion-toolbar>
  </ion-header>
  <ion-content>
    <ion-list>
      <button ion-item *ngFor="let p of pages" (click)="openPage(p)">{{p.title}}</button>
    </ion-list>
  </ion-content>
</ion-menu>
```

```
<ion-nav [root]="rootPage" #content swipeBackEnabled="false"></ion-nav>
```

El elemento ion-nav es el que tendrá asociado el NavController.

Plugin camera

Ionic aporta unos plugins que permiten el acceso a Cordova desde una perspectiva más orientada a objetos, promesas y basada en TypeScript.

Instalar el plugin de cordova para camara:

```
ionic plugin add cordova-plugin-camera
```

En la clase en la que se va a usar, se debe importar:

```
import {Camera} from "ionic-native";
```

Ejemplo de código para capturar una imagen y mostrarla en un img asociado a la variable this.foto:

```
var opciones = {quality:50,sourceType: Camera.PictureSourceType.CAMERA,  
                destinationType: Camera.DestinationType.FILE_URI, targetWidth:400, targetHeight:400,  
                correctOrientation:true};  
  
let msg = this.alertCtrl.create({title:"Camara no disponible", buttons:["ok"]});  
  
Camera.getPicture(opciones).then((urilmagen)=> this.foto=urilmagen) .catch(()=> msg.present());
```