

Universitat d'Alacant  
Universidad de Alicante

**Aplicaciones Distribuídas en Internet - Trabajo en grupo**



Josué Perea Martínez  
Luis Alfonso Culiáñez  
Adrián Collados Martínez

# ÍNDICE

<b>¿Qué es Ionic?</b>	<b>3</b>
<b>Características de Ionic</b>	<b>3</b>
<b>Ventajas y desventajas</b>	<b>4</b>
<b>Instalación</b>	<b>5</b>
<b>Ejemplo práctico</b>	<b>7</b>
<b>Bibliografía</b>	<b>10</b>

## ¿Qué es Ionic?

Ionic es uno de los frameworks más utilizados para la creación de aplicaciones móviles híbridas, hoy en día. Es un framework gratuito y, lo más importante, de código abierto lanzado en 2003 por Max Lynch, Ben Sperry y Adam Bradley, que ofrece numerosas bibliotecas de componentes, gestos y herramientas de interfaz de usuario. Todas ellas optimizadas para los diferentes dispositivos móviles, que nos permitirán crear aplicaciones rápidas y altamente interactivas. Esto quiere decir, que desde una misma base de código nos permite crear aplicaciones para entornos Android, IOS y la propia web.

Ionic está basado en tecnologías web (HTML, CSS y JS), además de poderse integrar con otros framework frontend como son Angular, React y Vue.

## Características de Ionic

En este apartado vamos a comentar un poco más en detalle alguna de las características principales que tiene este framework, frente a otros que pueden hacerle competencia.

- Desarrollar y desplegar aplicaciones híbridas, partiendo de un único código podremos generar tanto la aplicación para dispositivos móviles como la aplicación de escritorio. Es decir, nuestro código nos va a servir tanto para Android como para IOS, así como para web.
- Son aplicaciones que ofrecen diseños limpios, sencillos, intuitivos y funcionales.
- Al estar construido sobre tecnologías ya consolidadas como son HTML, CSS y JavaScript, podemos también aprovecharnos de todas las ventajas que estas tecnologías nos ofrecen.
- También podremos incluir framework para la realización de la parte frontend como pueden ser Angular, React o Vue, ya que tenemos una compatibilidad completa con cualquiera de ellos.

## Componentes UI

La facilidad que brinda Ionic para el diseño de la interfaz es una de sus fortalezas, y lo hace a través de sus componentes. Los componentes de Ionic son bloques de construcción de alto nivel que nos ayudan a construir rápidamente la interfaz de usuario de una aplicación. Algunos de sus principales componentes de interfaz de usuario son:

- **Tarjetas (ion-cards).** Uno de los componentes estándar de la interfaz de usuario. Sirve como punto de entrada para obtener información más detallada. Suele estar formado por título, subtítulo y contenido.
- **Listas (ion-lists).** Consta de elementos de varias líneas, que pueden incluir texto, botones, iconos, miniaturas, etc. Las listas de Ionic admiten varias interacciones, como deslizar para mostrar opciones o arrastrar para reordenar o eliminar elementos.
- **Etiquetas (ion-tabs).** A menudo se usan junto con las barras de pestañas (ion-tab-bars) para habilitar la navegación basada en pestañas similar a las aplicaciones nativas.
- **Alertas (ion-alerts).** Sirve para crear un alerta del tipo pop up que aparecerá justo delante de la pantalla que tenga cargada en ese momento el usuario.

## Ventajas y desventajas

Por supuesto tenemos que tener siempre en mente que Ionic Framework es una utilidad destinada a crear aplicaciones móviles, o como última instancia híbridas; y que aunque se pueden realizar otros proyectos con mayor complejidad, el resultado no estará tan optimizado como lo pudieran estar aquellos para los que se ha desarrollado Ionic.

### Ventajas:

- Desde una única fuente podremos llegar a las plataformas que soportan este framework (Android e iOS).
- Es fácil de aprender y utilizar, se basa en tecnologías web HTML, CSS y JavaScript, por lo que los desarrolladores no tienen que aprender nuevas tecnologías.
- Es una herramienta que da soporte a los frameworks habituales como Angular, Vue y React. Además, de incluir numerosos plugins y con vista de ampliar el soporte a otros frameworks.
- Ionic favorece a una mayor productividad y reducción de costes de desarrollo de la aplicación. Desarrollar aplicaciones híbridas con un único código que además, hace que su mantenimiento y escalado sea más sencillo.
- Diseño de interfaces rápido y sencillo, eligiendo elementos UI predeterminados de su librería de componentes en vez de ir codificando uno a uno.

### Desventajas:

- Si la aplicación a desarrollar necesita hacer uso de grandes cantidades de recursos, el rendimiento de la aplicación será menor que en aplicaciones desarrolladas de forma nativa.
- Tiene dependencia con los plugins y debido a que es una herramienta “joven”, será un poco más complicado encontrar módulos compartidos por otros usuarios.
- Debido a esta “juventud” los desarrolladores de Ionic siguen actualizando y cambiando el framework, por lo que los usuarios tienen que actualizarse cada cierto tiempo.

## Instalación

Vamos a ver el proceso de instalación de Ionic y la configuración de algunas herramientas y dependencias necesarias para que funcione.

- Node.js
- Ionic

### Instalación de Node.js

En primer lugar, y para hacer funcionar todo, necesitamos la base donde está construido el Framework. Por dicho motivo necesitamos de NodeJS y de su gestor de paquetes NPM. NodeJS es lo que nos permitirá hacer correr el server desarrollo de Ionic y alguna cosita más para su correcto funcionamiento y NPM nos permitirá instalar los módulos/componentes externos que nos harán falta para construir las apps.

Accederemos a la Web de NodeJS y descargamos el instalador correspondiente a nuestro sistema operativo desde el siguiente [enlace](#).

- En Mac se nos descargará un fichero "pkg" que podremos instalar simplemente haciendo doble clic sobre él.
- En Windows podemos descargar un instalador tipo "msi", el cual se instalará haciendo doble clic sobre el mismo.
- En Linux se nos descargará un fichero comprimido que tendremos que descomprimir, abrir una consola, acceder a la carpeta con el contenido descomprimido, y ejecutar los siguientes comandos para compilar e instalar la librería:
  - ./configure
  - make
  - make install

### Instalación de Ionic



El siguiente paso, una vez instalado Node, debemos instalar la plataforma de Apache Cordova. Será la encargada de gestionar todas las conexiones de nuestros plugins con la parte nativa del sistema:

Para instalarlo vamos a utilizar el gestor de paquetes de Node, npm, que hemos instalado anteriormente.

- `npm install -g cordova`

Esta línea descargará el paquete y lo instalará globalmente en nuestro equipo de desarrollo.

A continuación, procederemos a instalar [Ionic](https://ionicframework.com/) desde su web oficial.

- `npm i -g @ionic/cli`

Para comprobar si se ha instalado correctamente podemos escribir en el terminal el siguiente comando:

- `$ ionic ó $ ionic -v`

Ahora que tenemos instalado Ionic y todo está funcionando correctamente, estamos en disposición de crear nuestra primera aplicación.

Nos colocamos en el directorio que consideremos oportuno para la configuración y escribiremos en la terminal la siguiente línea:

- `ionic start`

En la terminal aparecerá la siguiente pregunta y cuando aceptemos nos abrirá una página en el navegador como se muestra en la siguiente imagen.

- Use the app creation wizard? Yes

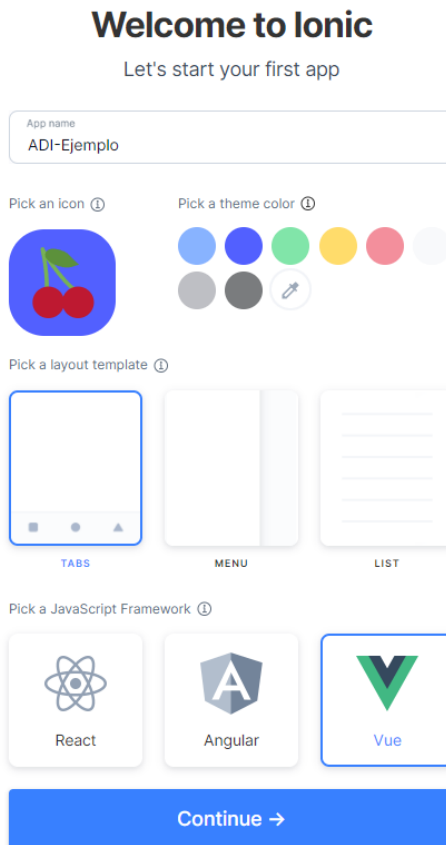


Figura 1

Una vez hemos terminado de configurar el proyecto de Ionic podemos ejecutar la App, la mayor parte del desarrollo de la aplicación se puede gastar directamente en el navegador usando el siguiente comando:

- `ionic serve`

## Ejemplo práctico

Vamos a realizar nuestra primera App con Ionic siguiendo los pasos anteriores. Para ello hemos elegido Vue (como podemos ver en la Figura 1), ya que lo hemos empleado en la asignatura en mayor profundidad. Ejecutaremos antes que nada los siguientes comandos:

```
npm install @capacitor/camera @capacitor/storage @capacitor/filesystem
```

```
$ npm install @ionic/pwa-elements
```

Necesitaremos importar `@ionic/pwa-elements` en el archivo `main.ts`:

```
34 // Above the createApp() line
35 import { defineCustomElements } from '@ionic/pwa-elements/loader';
36
37 // Call the element loader after the platform has been bootstrapped
38 defineCustomElements(window);
```

Ya podemos empezar a programar esta primera App. Se trata de una galería de fotos, en la que podremos conectar con la cámara del dispositivo (pensada para móvil) para subir y guardar las fotos que queramos. Podremos visualizar las ya subidas y borrarlas. Al usar ionic y como hemos visto en las ventajas, este mismo código nos sirve para aplicación web o móvil, independientemente de si es iOS o Android. No la hemos llevado a producción para generar la aplicación nativa para estos sistemas debido a que para iOS necesitamos un sistema MacOS para ello, y en el caso de Android nos bastaría con descargarnos el programa: [Android Studio](#).

Aquí vamos a explicar lo que se realiza en la App. En el archivo `composables/usePhotoGallery.ts` definiremos la función `usePhotoGallery()`, la función que contiene la lógica, que guarda, carga, elimina nuestras fotos. Esta lógica la llamaremos desde el componente `Tab2Page.vue` para realizar las diferentes acciones que el usuario pueda desear. En la vista simplemente mostraremos las fotos que ya hay almacenadas y le proporcionaremos al usuario un botón para acceder a la cámara o a seleccionar una foto que guarde en su dispositivo. Si clickamos en la imagen, tendremos la opción de eliminarla.

Cabe destacar que para que la función de guardar y cargar las fotos funcionen también en móviles hay que importar:

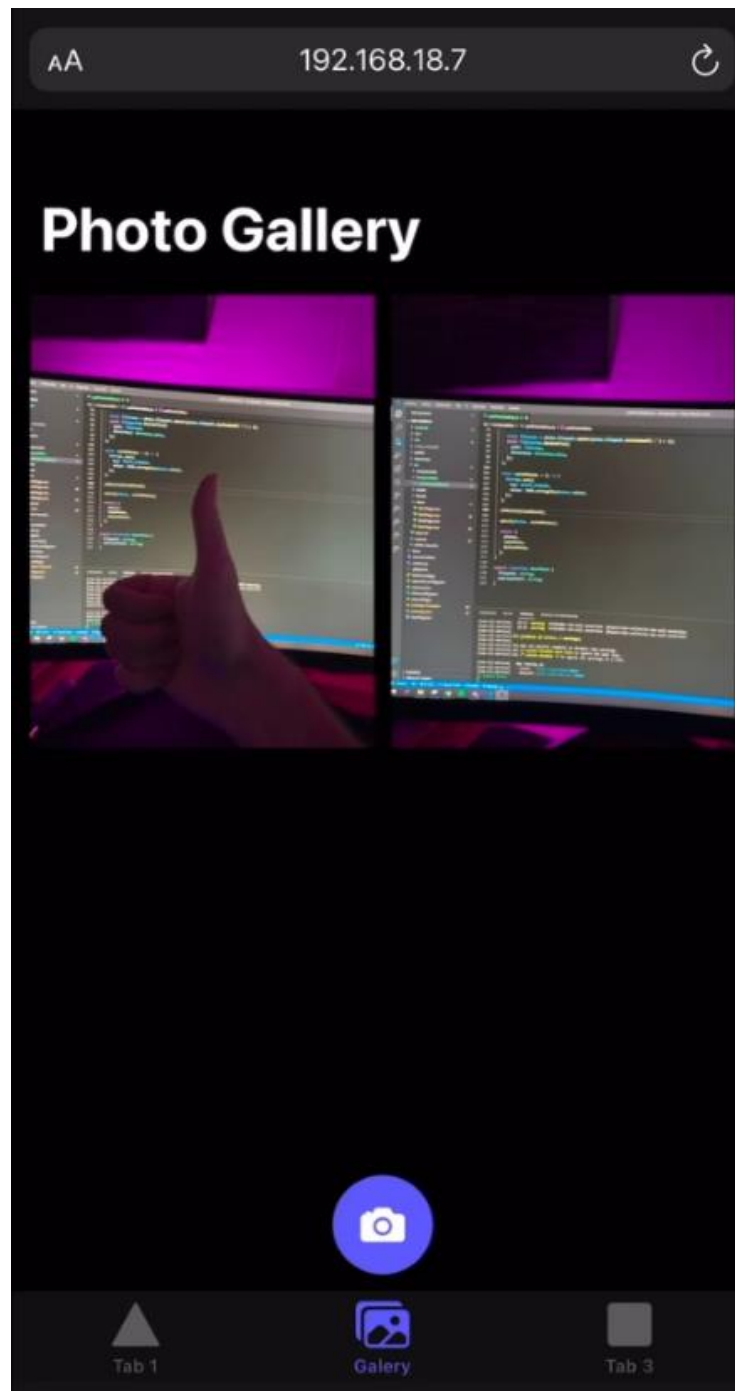
```
import { isPlatform } from '@ionic/vue';

const savePicture = async (
  photo: Photo,
  fileName: string
): Promise<UserPhoto> => {
  let base64Data: string;
  if (isPlatform('hybrid')) {
    const file = await Filesystem.readFile({
      path: photo.path!,
    });
    base64Data = file.data;
  } else {
    const response = await fetch(photo.webPath!);
    const blob = await response.blob();
    base64Data = (await convertBlobToBase64(blob)) as string;
  }
  const savedFile = await Filesystem.writeFile({
    path: fileName,
    data: base64Data,
    directory: Directory.Data,
  });
}
```

Figura 2. Función `savePicture`, que guarda la foto



La comprobación de si estamos en dispositivo móvil o no, la realizamos precisamente con este `isPlatform`, de manera que el parámetro `'hybrid'` detecta el paquete Cordova o Capacitor que hemos instalado previamente.



## Bibliografía

<https://ajgallego.gitbook.io/ionic-v3/chapter1>

Recurso: Ionic-v3 - AJ Gallego

<https://openwebinars.net/blog/ionic-framework-ventajas-desventajas/>

Recurso: Ionic - Esaú Abril Núñez

<https://profile.es/blog/que-es-ionic/>

Recurso: ¿Qué es Ionic? - José María Agüero Atmitim

<https://como-programar.net/blog/como-instalar-ionic/>

Recurso: como-programar.net - Ionic

<https://codigofacilito.com/articulos/installing-ionic4>

Recurso: Petriz Celaya - Ionic - Código facilito

<https://www.programaenlinea.net/ionic-se-usa-sirve/>

Recurso: NGuerrero - Ionic

<https://ionicframework.com/>

Recurso: Official Site Ionic