

## Plataformas Ionic 2

Ionic 2 dispone de varias plataformas para generar la compilación, Android, IOS (solo desde Mac), Blackberry, Windows,... para ver el listado de plataformas, usar el comando *ionic platform list*. Cada plataforma suele tener sus dependencias, por ejemplo, para poder añadir (platform add nombre\_plataforma)/construir (ionic build nombre\_plataforma) la plataforma de Android, previamente hay que tener instalado el SDK de Android.

## Navegación en Ionic 2, componente NavController

Usando este componente en cada página se disponen para usar las características de este controlador, que a diferencia del routing de angular 2, en Ionic 2 tiene su propio sistema, copiando la funcionalidad de IOS, basado en capas o también llamada “navegación natural”.

- *En cada selección, se hace push de la vista siguiente*
  - En la práctica del restaurante, al hacer clic sobre uno del listado se llama a la función `itemTapped` que hace un `this.navCtrl.push` con parámetros llamando a `ItemDetailsPage`. **Para poder hacer push de una página esta debe ser importada en el componente que la llama:**

```
itemTapped(event, restaurante) {  
    this.navCtrl.push(ItemDetailsPage, {  
        item: restaurante  
    });  
}
```

- *Si en algún momento, seleccionas el botón atrás, se hace un pop de la vista actual.*

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

Declaración del uso del componente:

```
import { NavController } from 'ionic-angular';  
  
class MyComponent {  
    constructor(public navCtrl: NavController) {  
  
    }  
}
```

## LifeCycles

Un aspecto muy importante a conocer de Ionic 2 son los ciclos de vida de una página, el caso más claro es cuando queremos recargar la página cada vez que se muestre por algún tipo de necesidad, como reflejar cambios que afectan a la página que se han hecho en otras páginas/secciones de la app. Por ejemplo, si eliminamos un restaurante, se debe recargar de nuevo el listado cuando se muestre de nuevo, entonces, en el componente del listado, que es al que le afecta, se debe declarar el método *ionViewWillEnter*.

```
ionViewWillEnter () {  
    console.log('Pagina principal activa');  
    this.getRestaurantes();  
}
```

[En la página oficial comenta a fondo todos los tipos.](#)

## Ionic 2, acceder a cámara

Toda la info oficial de Ionic [sobre el acceso a la cámara](#).

```
ionic platform add android

ionic plugin add cordova-plugin-camera

ionic state save
```

Después de este último paso, hay que verificar que ionic ha añadido el permiso en el archivo AndroidManifest.xml, es vital para que la App pueda acceder al recurso.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Llamada a cámara desde un component.

Vista

```
<ion-content padding>
<button ion-button block (click)="getPicture()">Toma una foto</button>
  <img [src]="image" *ngIf="image" />
</ion-content>
```

Component

```
import { Component } from '@angular/core';
import { Camera } from 'ionic-native';

@Component({
  templateUrl: 'copy-hello-ionic.html'
})
export class CopyHelloIonicPage {
  image: string;
  constructor() {}
  getPicture(){
    let options = {
      destinationType: Camera.DestinationType.DATA_URL,
      targetWidth: 1000,
      targetHeight: 1000,
      quality: 100
    }
    Camera.getPicture( options )
      .then(imageData => {
        this.image = `data:image/jpeg;base64,${imageData}`;
      })
      .catch(error =>{
        alert( error );
      });
  }
}
```

Obtener imagen desde la galería

Para obtener la imagen desde la galería de fotos del dispositivo, basta con añadir el parámetro `sourceType` : `Camera.PictureSourceType.PHOTOLIBRARY` en las options.