

Enlace proyecto <https://github.com/TatanSan/app-galeria-fotos-Ionic>

## Codificación para el uso de la cámara

Instalación de librería propia de Ionic,

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\PROGRAMACION\Ionic\appGaleriaFotos> npm install @ionic/pwa-elements
```

Posteriormente realiza la instalación de los módulos correspondientes a la librería.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  node + v  [ ] [ ]

[ng] node_modules_ionic_core_dist_esm_ion-chip_entry_js.js      -      8.07 kB
[ng] src_app_tab2_tab2_module_ts.js                             tab2-tab2-module  7.53 kB
[ng] src_app_tab3_tab3_module_ts.js                             tab3-tab3-module  7.53 kB
[ng] src_app_tab1_tab1_module_ts.js                             tab1-tab1-module  7.53 kB
[ng] node_modules_ionic_core_dist_esm_index-020f5464_js.js     index-020f5464-js  6.84 kB
[ng] node_modules_ionic_core_dist_esm_ion-ripple-effect_entry_js.js -      6.32 kB
[ng] node_modules_ionic_core_dist_esm_ion-avatar_3_entry_js.js  -      6.04 kB
[ng] node_modules_ionic_core_dist_esm_ion-img_entry_js.js       -      4.77 kB
[ng] node_modules_ionic_core_dist_esm_ion-backdrop_entry_js.js  -      3.39 kB
[ng] node_modules_ionic_core_components_md_transition_js.js     md-transition-js  2.98 kB
[ng] node_modules_ionic_core_dist_esm_status-tap-b41ece3c_js.js status-tap-b41ece3c-js 2.89 kB
[ng] node_modules_ionic_core_dist_esm_ion-text_entry_js.js      -      1.94 kB
[ng]
```

Definimos el ámbito de la aplicación y se hace el llamado a la librería pwa-elements en el Main.

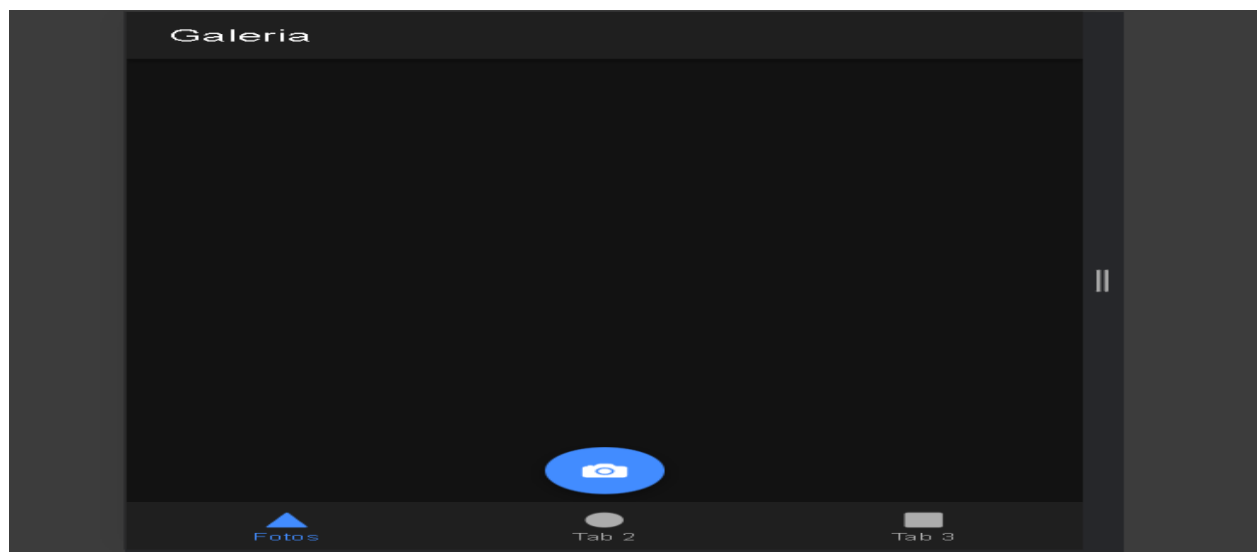
```
TS main.ts
src > TS main.ts > ...
1  import { enableProdMode } from '@angular/core';
2  import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
3  import { defineCustomElements } from '@ionic/pwa-elements/loader'
4
5  import { AppModule } from './app/app.module';
6  import { environment } from './environments/environment';
7
8  if (environment.production) {
9    enableProdMode();
10 }
11
12 platformBrowserDynamic().bootstrapModule(AppModule)
13   .catch(err => console.log(err));
14
15 defineCustomElements(window)
```

## Configuración de header y botón “camera”

```
src > app > tab1 > tab1.page.html > ion-content
Go to component
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>
4       Galeria <!--renombramos el boton-->
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content [fullscreen]="true">
10  <ion-fab vertical="bottom" horizontal="center" slot="fixed">
11    <!--configuramos la posicion del boton, ubicacion y posicion sobre cualquier objeto-->
12    <ion-fab-button>
13      <ion-icon name="camera"></ion-icon><!--propiedad para llamar el icono llamado "camera"-->
14    </ion-fab-button>
15  </ion-fab>
16 </ion-content>
17
```

## Cambio de etiqueta del icono “camera”

```
<ion-tab-bar slot="bottom">
  <ion-tab-button tab="tab1" href="/tabs/tab1">
    <ion-icon aria-hidden="true" name="triangle"></ion-icon>
    <ion-label>Fotos</ion-label>
  </ion-tab-button>
</ion-tab-bar>
```



Se hace la instalación de los plugins con capacitor camera/storage/filesystem

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\PROGRAMACION\Ionic\appFotoGalery> npm install @capacitor/camera @capacitor/storage @capacitor/filesystem
npm ERR! code ERESOLVE
npm ERR! ERESOLVE unable to resolve dependency tree
npm ERR!
npm ERR! While resolving: appFotoGalery@0.0.1
npm ERR! Found: @capacitor/core@5.7.1
npm ERR! node_modules/@capacitor/core
npm ERR!   @capacitor/core@"5.7.1" from the root project
npm ERR!
npm ERR! Could not resolve dependency:
npm ERR! peer @capacitor/core@"^3.0.0" from @capacitor/storage@1.2.5
npm ERR! node_modules/@capacitor/storage
npm ERR!   @capacitor/storage@"*" from the root project
```

Sin embargo, si se presenta conflicto puede deberse a conflicto entre dependencias, lo cual indica que tiene errores con las versiones instaladas y las usadas actualmente, para lo que se debe forzar la instalación del comando con `--force` o con `--legacy-peer-deps`, para validar su instalacion se puede acceder a los modulos y validar la carpeta de @capacitor, allí nos deberá crear las subcarpetas solicitadas.

```
@capacitor
├── app
├── camera
├── cli
├── core
├── filesystem
├── haptics
├── keyboard
├── status-bar
├── storage
├── @colors
├── @discoveryjs
├── @esbuild
├── @eslint
├── @eslint-community
├── @fastify
├── @humanwhocodes
├── @ionic
├── @isaacs
├── @istanbuljs
├── @jest
├── @jridgewell
├── @leichtgewicht
├── @ljharb
└── @ngtools

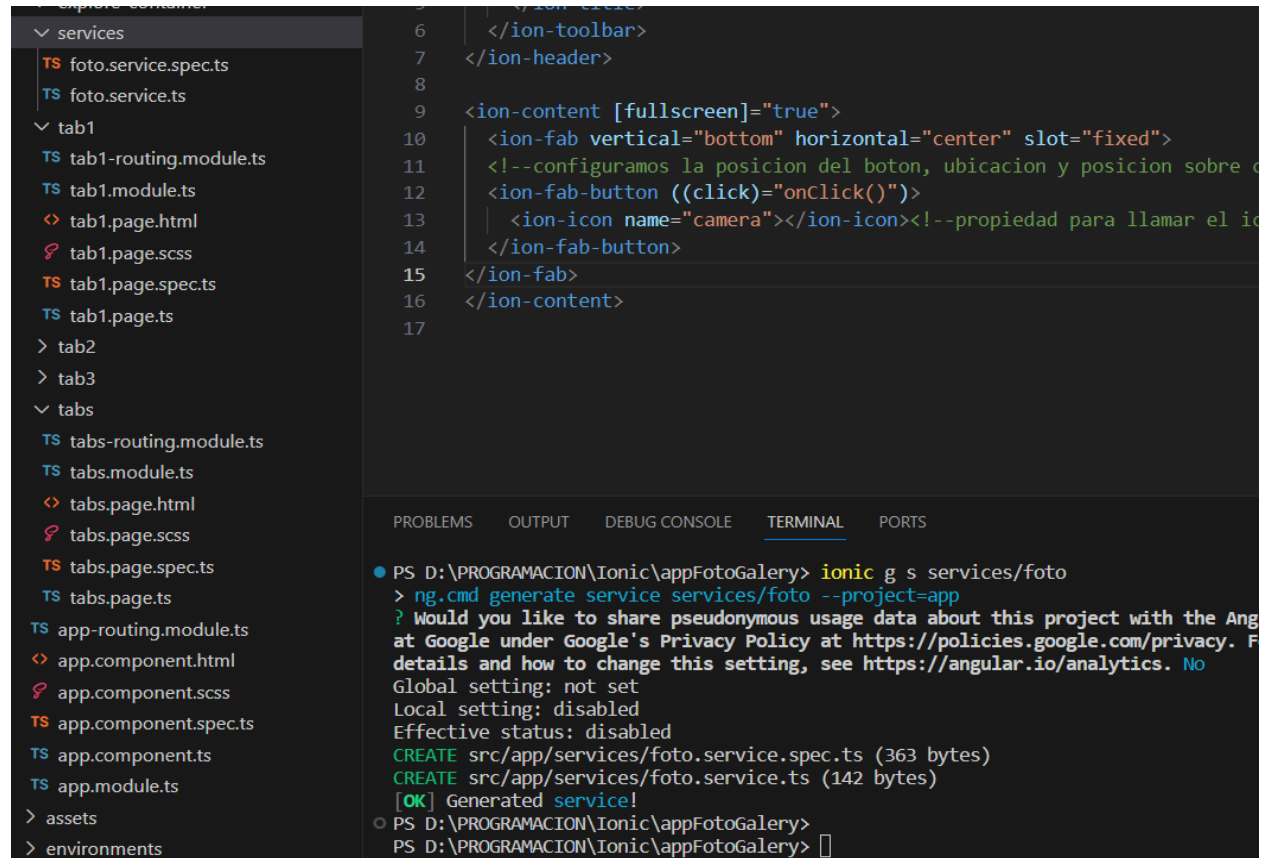
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
npm WARN node_modules/@capacitor/core
npm WARN peer @capacitor/core@"^3.0.0" from @capacitor/storage@1.2.5
npm WARN node_modules/@capacitor/storage
npm WARN @capacitor/storage@"*" from the root project
npm WARN deprecated @capacitor/storage@1.2.5: package has been renamed

added 3 packages, and audited 1246 packages in 5s

188 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS D:\PROGRAMACION\Ionic\appFotoGalery>
```

Creación de carpeta “services” para el servicio “fotos”



The screenshot shows an IDE with a file explorer on the left, a code editor in the center, and a terminal at the bottom.

**File Explorer (Left):**

- services
  - foto.service.spec.ts
  - foto.service.ts
- tab1
  - tab1-routing.module.ts
  - tab1.module.ts
  - tab1.page.html
  - tab1.page.scss
  - tab1.page.spec.ts
  - tab1.page.ts
- tab2
- tab3
- tabs
  - tabs-routing.module.ts
  - tabs.module.ts
  - tabs.page.html
  - tabs.page.scss
  - tabs.page.spec.ts
  - tabs.page.ts
- app-routing.module.ts
- app.component.html
- app.component.scss
- app.component.spec.ts
- app.component.ts
- app.module.ts
- assets
- environments

**Code Editor (Center):**

```
5 | </ion-title>
6 | </ion-toolbar>
7 | </ion-header>
8 |
9 | <ion-content [fullscreen]="true">
10 |   <ion-fab vertical="bottom" horizontal="center" slot="fixed">
11 |     <!-- configuramos la posicion del boton, ubicacion y posicion sobre el contenido -->
12 |     <ion-fab-button ((click)="onClick()">
13 |       <ion-icon name="camera"></ion-icon><!-- propiedad para llamar el icono -->
14 |     </ion-fab-button>
15 |   </ion-fab>
16 | </ion-content>
17 |
```

**Terminal (Bottom):**

```
PS D:\PROGRAMACION\Ionic\appFotoGalery> ionic g s services/foto
> ng.cmd generate service services/foto --project=app
? Would you like to share pseudonymous usage data about this project with the Angular team at Google under Google's Privacy Policy at https://policies.google.com/privacy. For details and how to change this setting, see https://angular.io/analytics. No
Global setting: not set
Local setting: disabled
Effective status: disabled
CREATE src/app/services/foto.service.spec.ts (363 bytes)
CREATE src/app/services/foto.service.ts (142 bytes)
[OK] Generated service!
PS D:\PROGRAMACION\Ionic\appFotoGalery>
```

Creación del método addNewToGallery() desde la clase foto.servicer.ts

```
<> tab1.page.html TS tab1.page.ts TS foto.service.ts
src > app > services > TS foto.service.ts > FotoService > addNewToGallery

2 //importamos los plugins que se crearon de los servicios y llamado a los archiv
3 import{Camera, CameraResultType, CameraSource, Photo} from '@capacitor/camera'
4 import{ Filesystem, Directory} from '@capacitor/filesystem'
5 import{ Storage} from '@capacitor/storage'
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class FotoService {
11
12   constructor() { }
13   //llamdo a funcion publica para ser llamada desde afuera asincronica al descono
14
15   public async addNewToGallery(){
16
17   }
```

Para que hacer el llamado a la función desde la clase tab1.page.html para el evento “click”

```
10 <!--configuramos la posicion del boton, ubicacion y posicion sobre cualquier objeto-->
11 <ion-fab-button (click)="addNewToGallery()"> (component) IonFabButton
12   <ion-icon name="camera"></ion-icon><!--propiedad para llamar el icono llamado "camera"-->
13 </ion-fab-button>
```

Téngase en cuenta que se debe declarar el método, las ayudas del editor pueden indicar si este no se realizó automáticamente.

Los métodos se declaran dentro de la misma clase en la subclase tab1.page.ts

```
<> tab1.page.html TS tab1.page.ts TS foto.service.ts
src > app > tab1 > TS tab1.page.ts > ...

3 @Component({
4   selector: 'app-tab1',
5   templateUrl: 'tab1.page.html',
6   styleUrls: ['tab1.page.scss']
7 })
8 export class Tab1Page {
9   addNewToGallery() {
10     throw new Error('Method not implemented.');
```

Pasamos los parámetros al método que ya está en la librería, se le indica lo que se requiere.

```
    constructor() { }  
    //llamdo a funcion publica para ser llamada desde afuera asincrono  
  
    public async addNewToGallery(){  
        //proceso para tomar la foto  
        const fotoCapturada = await Camera.getPhoto({  
            resultType: CameraResultType.Uri,  
            source: CameraSource.Camera,  
            quality: 100  
        })  
    }
```

Declaramos una variable publica en el typescrip en la parte funcional llamada “FotoService”, servicio que anteriormente creamos.

```
src > app > tab1 > TS tab1.page.ts > Tab1Page > constructor  
1   import { FotoService } from '../services/foto.service';  
2   import { Component } from '@angular/core';  
3  
4   @Component({  
5       selector: 'app-tab1',  
6       templateUrl: 'tab1.page.html',  
7       styleUrls: ['tab1.page.scss']  
8   })  
9   export class Tab1Page {  
10      addNewToGallery() {  
11          throw new Error('Method not implemented.');12      }  
13      constructor(public FotoService: FotoService) {}  
14  
15  }  
16
```

Se crea dentro de la clase una función para llamar la función no al servicio

```

2  import { Component } from '@angular/core';
3
4  @Component({
5    selector: 'app-tab1',
6    templateUrl: 'tab1.page.html',
7    styleUrls: ['tab1.page.scss']
8  })
9  export class Tab1Page {
10     addNewToGallery() {
11       throw new Error('Method not implemented.');

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS


```

Por lo tanto, se hace el reemplazo del evento “click” del archivo tab1.html y reemplazamos la función llamada para no hacer referencia directa al servicio

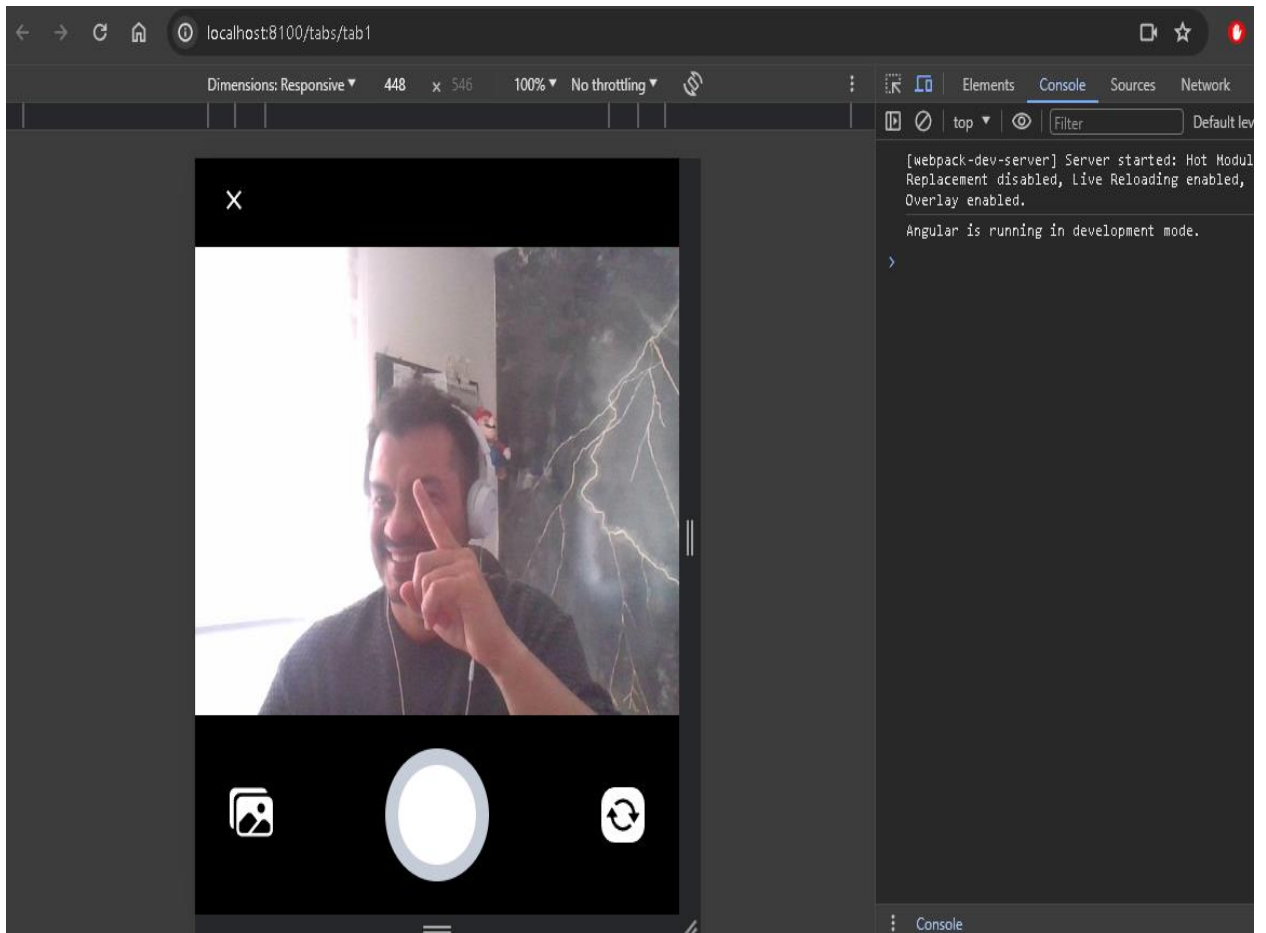
```

<ion-tab vertical="bottom"
<!--configuramos la posic (method) Tab1Page.addPhotoToGallery()
<ion-fab-button (click)="addPhotoToGallery()">
  <ion-icon name="camera"></ion-icon><!--propiedad para llamar
</ion-fab-button>
</ion-fab>
</ion-content>

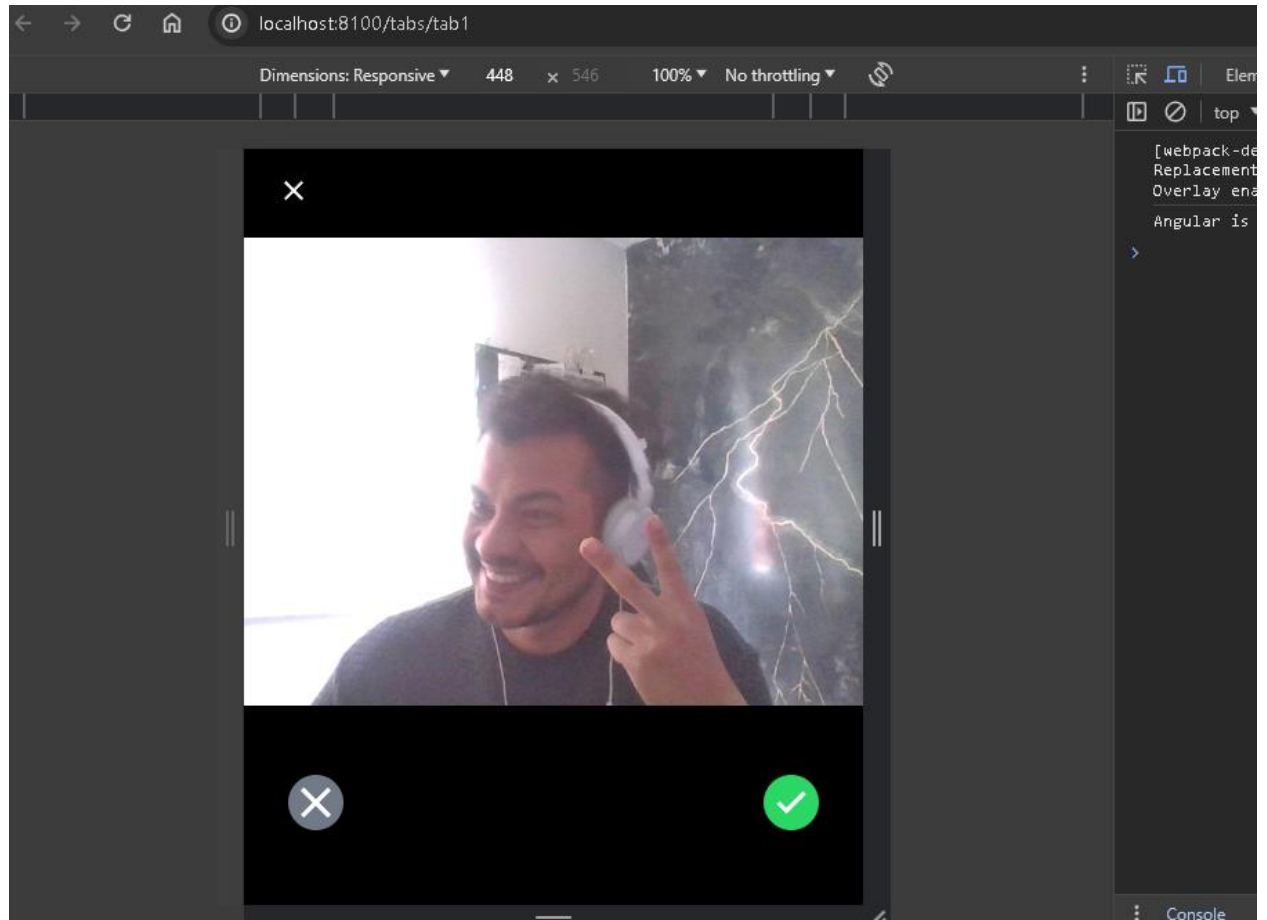
```

De esta manera validamos la funcionalidad y primera prueba de la cámara, téngase en cuenta que se debe activar la cámara frontal ya que la laptop no tiene cámara

trasera. Ejemplo de funcionalidad del código anterior:







### Interfaz para almacenamiento de fotos en el directorio:

Creamos dentro de la carpeta “app” una subcarpeta “models” y un archivo “foto.interface.ts” el cual se exportara la interface foto.

Esto permitirá almacenar la foto en el sistema de archivos tanto en web con el código “webViewPath” y para Android como en iOS con el código “filePath”.

```
TS foto.service.ts TS foto.inteface.ts TS tab1.page.ts
src > app > models > TS foto.inteface.ts > Foto
1  export interface Foto{
2      filepath: string;
3      webViewPath: string;
4  }
```

La importamos para usar en el servicio

```
TS foto.service.ts • TS tab1.page.ts <> tab1.page.html
rc > app > services > TS foto.service.ts > ...
3 import{Camera, CameraResultType, CameraSource, Photo}
4 import{ Filesystem, Directory} from '@capacitor/filesystem'
5 import{ Storage} from '@capacitor/storage'
6 import{Foto} from '../models/foto.interface'
7
8 @Injectable({
9   providedIn: 'root'
```

Creamos un arreglo para el almacenamiento de fotos, esto se hara de manera temporal para posteriormente almacenarlo en el sistema de archivos, en un principio el arreglo será vacío.

```
TS foto.service.ts 1 • TS foto.interface.ts TS tab1.page.ts • <
rc > app > services > TS foto.service.ts > FotoService > addNewTo
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class FotoService {
12   //Arreglo temporal para almacenar fotos
13   public fotos: Foto [] = [];
14
```

Inicializamos el arreglo para que almacene la foto al inicio “unshift”

```
//proceso para tomar la foto
const fotoCapturada = await Camera.getPhoto({
  resultType: CameraResultType.Uri,
  source: CameraSource.Camera,
  quality: 100
})

this.fotos.unshift({
  filepath: "foto_",
  webViewPath: fotoCapturada.webPath
})
```

En caso de presentar error al declarar el webViewPath: fotoCapturada.webPath que indique que esta tratando de asignar un valor que podría ser string o undefined a una propiedad que se espera que sea solo de tipo string. Esto puede ocurrir si la propiedad webViewPath en la interfaz Foto está definida como string | undefined.

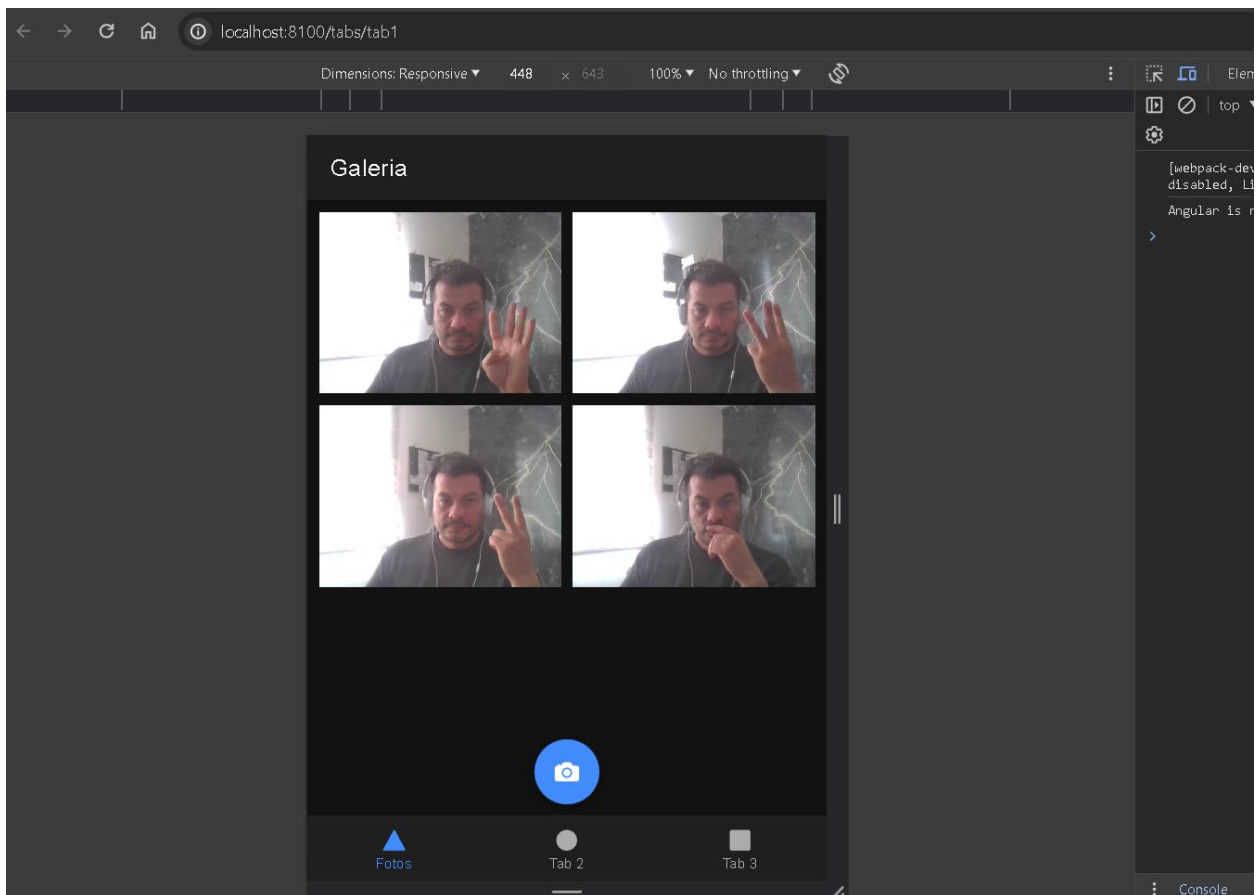
Dado que se declaro correctamente de tipo string en la interfaz puede intentar solucionar con el operador ( ?? ), que proporciona un valor por defecto en caso de que sea “undefined”, de la siguiente manera.

```
7      this.fotos.unshift({
8        filepath: "foto_", //nombre con que se almacena la foto capturada
9        webViewPath: fotoCapturada.webViewPath ?? '//en caso de presentar error usar el operador ?? ' '
10      })
11    }
12  }
13 }
```

Dentro del archivo .html del tab1, crearemos la grilla que almacene las fotos en el arreglo y haga el recorrido del “array”, en el que se manejara un índice “\*ngFor” “\*ngFor=”let foto of FotoService.fotos; index as position”>” Y mostraremos la foto” llamando a la propiedad que es donde esta almacenada la foto “<ion-img [src]=”foto.webviewPath”></ion-img>”

```
9  <ion-content [fullscreen]="true">
10    <ion-grid><!--creacion de grilla para almacenar fotos-->
11      <ion-row><!--definimos el tamaño de las columnas y su posicion-->
12        <ion-col size="6"
13          *ngFor="let foto of FotoService.fotos; index as position">
14          <ion-img [src]="foto.webviewPath"></ion-img>
15        </ion-col>
16      </ion-row>
17    </ion-grid>
```

## Ejemplo de funcionalidad del código anterior:



Guardaremos las capturas en el Array, para ello creamos al final una función de tipo público, además una constante que permita almacenar la foto capturada y la grabe en el mismo arreglo.

En caso de presentar error al generar el método, por temas de versión en desuso se debe instanciar si el IDE no lo hace automáticamente.

```
public async savePicture(cameraPhoto: CameraPhoto){  
  
}
```

```
TS foto.service.ts 1 X TS foto.inteface.ts TS tab1.page.ts tab1.page.html  
src > app > services > TS foto.service.ts > FotoService > addNewToGallery  
1 import { Injectable } from '@angular/core';  
2 //importamos los plugins que se crearon de los servicios y llamado a los archivos a utilizar  
3 import{Camera, CameraPhoto, CameraResultType, CameraSource, Photo} from '@capacitor/camera'
```

Debemos desusar el método anterior, puede ser comentándolo como forma de reciclar código o posteriores pruebas o eliminándolo.

```
})

/*this.fotos.unshift({
  filepath: "foto_",//nombre con que se almacena la foto capturada
  webviewPath: fotoCapturada.webPath ?? ''//en caso de presentar error usar el operador ?? ''
})
*/
```

Al final creamos la función que permita convertir la imagen del formato blob a formato base64 que es la forma en que se almacena este tipo de archivo de esta app.

```
public async readAsBase64(cameraPhoto: CameraPhoto){
  //convertir de blob a base64
  const response = await fetch(cameraPhoto.webPath ?? '')
  const blob = await response.blob()
```

Dentro de la función “savePicture”, creamos una constante que convierta la foto a formato base64 y haga el llamado a la función y la retorne, luego una constante que escriba la foto tomada en el directorio y asigne un formato (nombre) a la foto.

```
29      filepath: "foto_",//nombre con que se almacena la foto capturada
30      webviewPath: fotoCapturada.webPath ?? ''//en caso de presentar error usar el operador ?? ''
31    })
32    */
33
34    //constante que almacena la foto capturada y la graba
35    const savedImageFile = await this.savePicture(fotoCapturada)
36    this.fotos.unshift(savedImageFile)
37  }
38
39  public async savePicture(cameraPhoto: CameraPhoto){
40
41    //convertir la imagen a formato base64 que es la forma en que se almacena este tipo de archivo de esta app
42    const base64Data = await this.readAsBase64(cameraPhoto)
43    //escribir la foto en el directorio
44    const fileName = new Date().getTime() + '.jpeg';//formato de la foto(nombre de la foto)
45    const saveFile = await Filesystem.writeFile({
46      path: fileName,
47      data: base64Data,
48      directory: Directory.Data
49    })
50
51    return{
52      filepath: fileName,
53      webviewPath: cameraPhoto.webPath ?? ''
54    }
55  }
56
57  public async readAsBase64(cameraPhoto: CameraPhoto){
58    //convertir de blob a base64
```

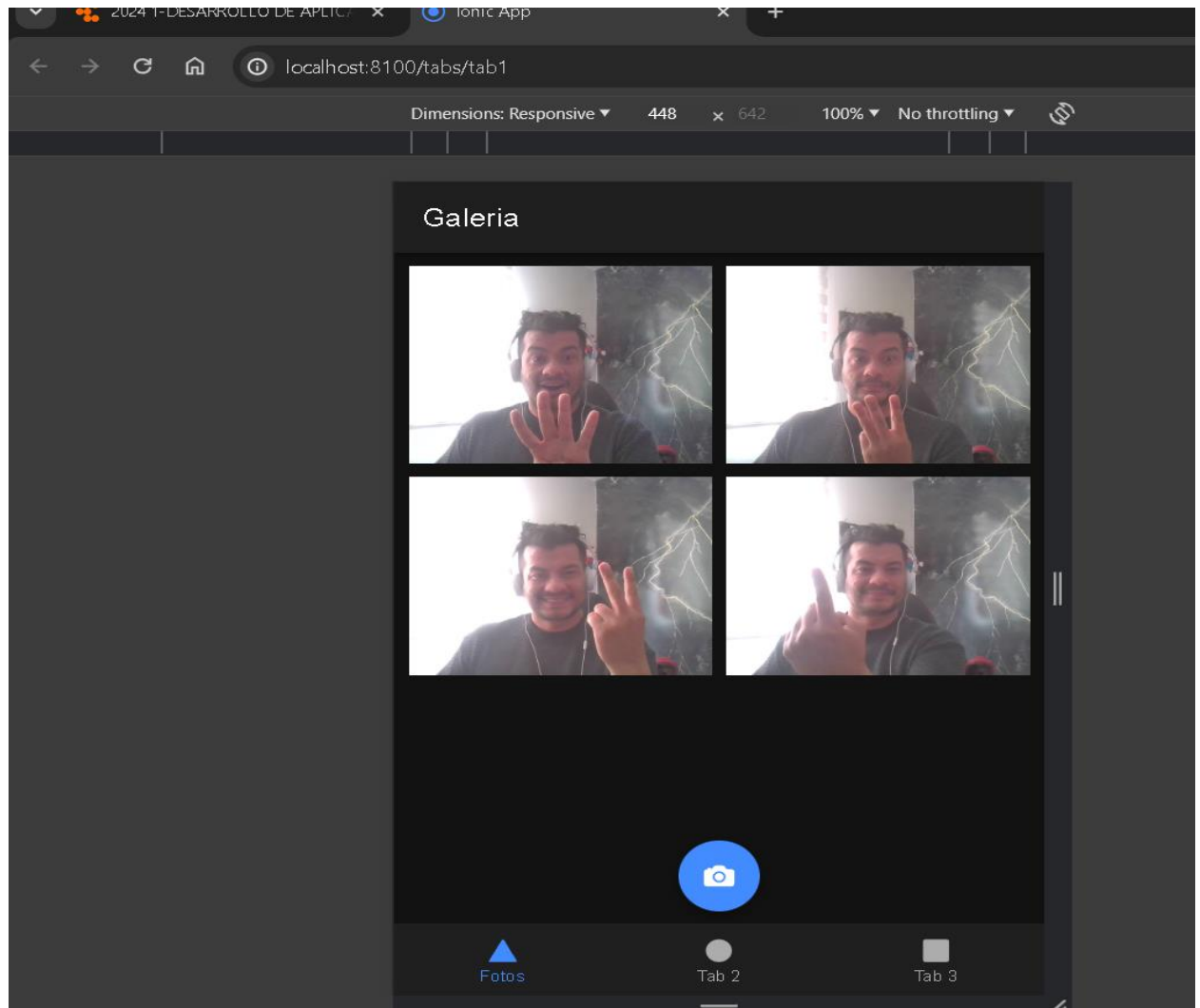
Si la propiedad “data” nos genera un error, creamos al final una función que nos permita hacer “promesas”, que son rutinas que van a devolver algo

```
//funcion flecha que use promesas para un posible retorno de un valor
convertBlobToBase64 =(blob: Blob) => new Promise((resolve, reject) => {
  const reader = new FileReader
  reader.onerror = reject
  reader.onload = () => {
    resolve(reader.result)
  }
  reader.readAsDataURL(blob)
})
}
```

Y retornamos el valor de la función “readAsBase64” para eliminar el error que se presenta en “data” convirtiéndolo a string ya que es una cadena.

```
52     webViewPath: cameraPhoto.webPath ?? ''
53   }
54 }
55
56 public async readAsBase64(cameraPhoto: CameraPhoto){
57     //convertir de blob a base64
58     const response = await fetch(cameraPhoto.webPath ?? '')
59     const blob = await response.blob()
60
61     return await this.convertBlobToBase64(blob) as string
62 }
63 //funcion flecha que use promesas para un posible retorno de un valor
64 convertBlobToBase64 =(blob: Blob) => new Promise((resolve, reject) => {
65     const reader = new FileReader
66     reader.onerror = reject
67     reader.onload = () => {
68         resolve(reader.result)
69     }
70     reader.readAsDataURL(blob)
71 })
72 }
```

### **Ejemplo de que se guarda en el Array:**



**Tener en cuenta que como se graba en un arreglo esto se reiniciara al momento de iniciar la app nuevamente, por lo tanto, es necesario que quede guardada en el sistema de archivos que es donde si quedara guardada hasta que sea borrada manualmente.**

### Cargue de fotos desde el directorio creado:

Declarar una variable privada de tipo string para acceder a las fotos, se le asigna un nombre o “clave” para poder recuperar las imágenes.

```
0    })
1    export class FotoService {
2      //Arreglo temporal para almacenar fotos
3      public fotos: Foto [] = [];
4      private PHOTO_STORAGE: string = "fotos"
5
6      constructor() { }
7      //llamdo a funcion publica para ser llamada
8
9      public async addNewToGallery(){
```

Creamos una función “loadsaved” que permita recuperar las fotos del almacenamiento de “cache”. Creamos una constante que cargue las fotos almacenadas en el directorio con `this.fotos = JSON.parse(listaFotos.value || "[]")`

#### 1. Forma más corta es tratar el valor nulo como una cadena vacía

```
71      reader.readAsDataURL(blob)
72    })
73
74    public async loadSaved(){
75      //Recuperacion de las fotos en caché
76      const listaFotos = await Storage.get({key: this.PHOTO_STORAGE })
77      this.fotos = JSON.parse(listaFotos.value || "[]")
```



## 2. Forma condicional antes de parsear formato a JSON.

Nos aseguramos de que “listaFotos.value” no sea “null” antes de intentar parsearlo como JSON. Para eso verificamos con una condicional.

```
71     reader.readAsDataURL(blob)
72   })
73
74   public async loadSaved(){
75     // Recuperacion de las fotos en caché
76     const listaFotos = await Storage.get({ key: this.PHOTO_STORAGE });
77
78     // Verificar si listaFotos.value no es null
79     if (listaFotos.value !== null) {
80       // Parsear listaFotos.value como JSON y asignarlo a this.fotos
81       this.fotos = JSON.parse(listaFotos.value);
82     } else {
83       // Si listaFotos.value es null, asignar un array vacío a this.fotos
84       this.fotos = [];
85     }
86   }
87 }
```

Así nos aseguramos de que “this.fotos” siempre sea asignado correctamente, ya sea con los datos obtenidos de listaFotos.value o con un Array vacío en caso de que “listaFotos.value” sea null.

Ahora mostramos las fotos leídas en formato base64 con un “For”, y leemos cada foto almacenada en el sistema de archivos. Para eso leemos con “readFile”, obteniendo los parámetros como “path” y “directory”.

```
77     this.fotos = JSON.parse(listaFotos.value || "[]")
78
79     //Desplegar fotos leídas en formato base64
80     for(let foto of this.fotos){
81
82       //Leer cada foto almacenada en el sistema de archivos
83       const readFile = await Filesystem.readFile({
84         path: foto.filepath,
85         directory: Directory.Data
86       })
87     }
88
89   }
90 }
```

Con el código anterior se cargarán las fotos en ambiente Android o IOS, por lo que se requiere el cargue en ambientes web, para lo que se procede a codificar el cargue de las imágenes en web y funcione en los tres ambientes.

Para eso usaremos la siguiente línea de código

```
foto.webviewPath = `data:image/jpeg;base64,${readFile.data}`
```

OJO, se debe tener presente que el uso de los símbolos es primordial, ya que se va a indicar que cargue un formato, no debe confundirse las comillas (“), ni el apostrofe (‘), el símbolo a usar es comilla simple invertida ( ` ` ), para eso puede usar el comando Alt + 96 o con el teclado AltGr + }.

```
37
38      //solo para plataforma Web: cargar las fotos en base64
39      foto.webviewPath = `data:image/jpeg;base64,${readFile.data}`
40    }
41
```

Cuando se agregue la imagen a la galería se deberá guardar en el Storage, para ello seteamos la información en formato JSON para la web.

```
this.fotos.unshift(savedImageFile)

//seteamos la informacion en formato JSON para la web
Storage.set({
  key: this.PHOTO_STORAGE,
  value: JSON.stringify(this.fotos)
})
}

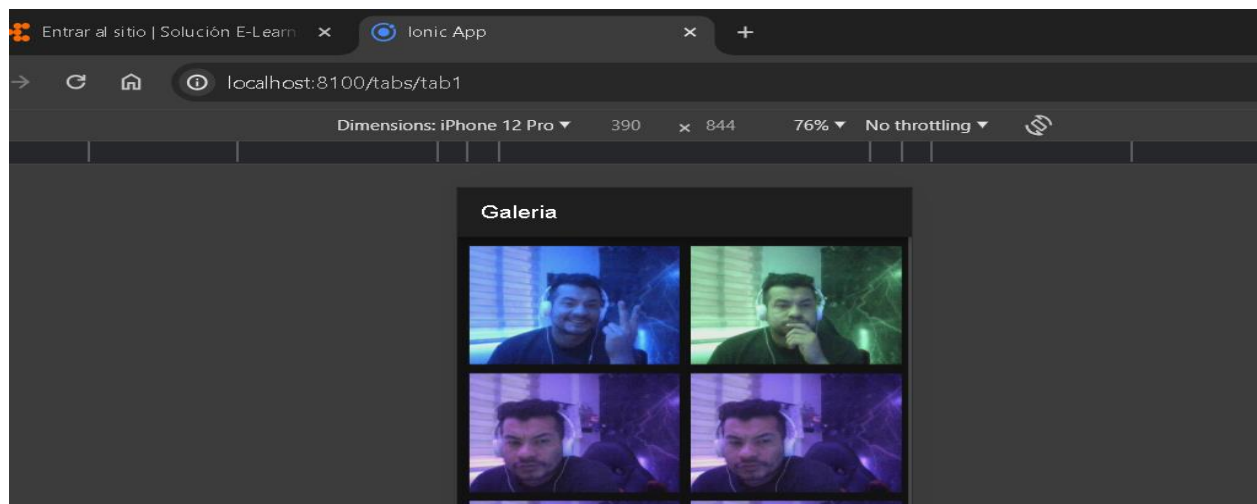
public async savePicture(cameraPhoto: CameraPhoto){
```

Debemos llamar a la función “public async loadSaved()” y crear una función asíncronica para que cargue lo que está en el sistema de archivos.

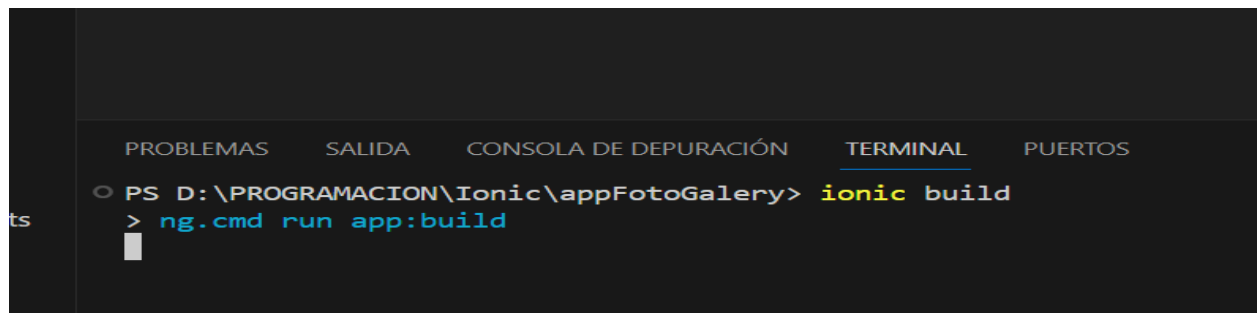
```
13      addPhotoToGallery(){ //propiedad para llamar a la funcion
14
15          this.FotoService.addNewToGallery()
16      }
17
18      async ngOnInit(){
19          await this.FotoService.loadSaved()
20      }
21
22    }
23
```

Sin embargo, la codificación que hemos hecho, no esta guardando el total de las fotos tomadas, y al recargar la pagina esta solo trae la ultima foto capturada, para ello, debemos ajustar y ampliar el método “addNewToGallery” en el servicio “FotoService” y agregar el método que guardara la foto capturada y el método que guarda todas las fotos en la cache.

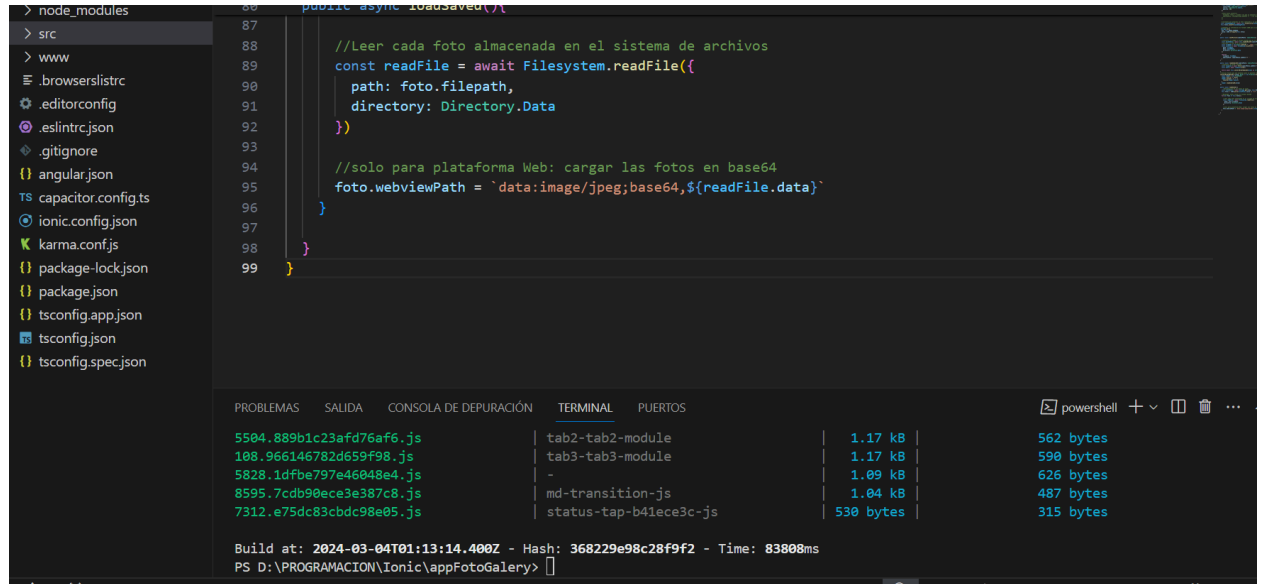
### Ejemplo de funcionalidad y pruebas de Browser



Procedemos a crear la app para la web con el comando “ionic build” en nuestra terminal.



Esto crea una carpeta “www” en donde se instalan todos los elementos necesarios para trabajar en la web.

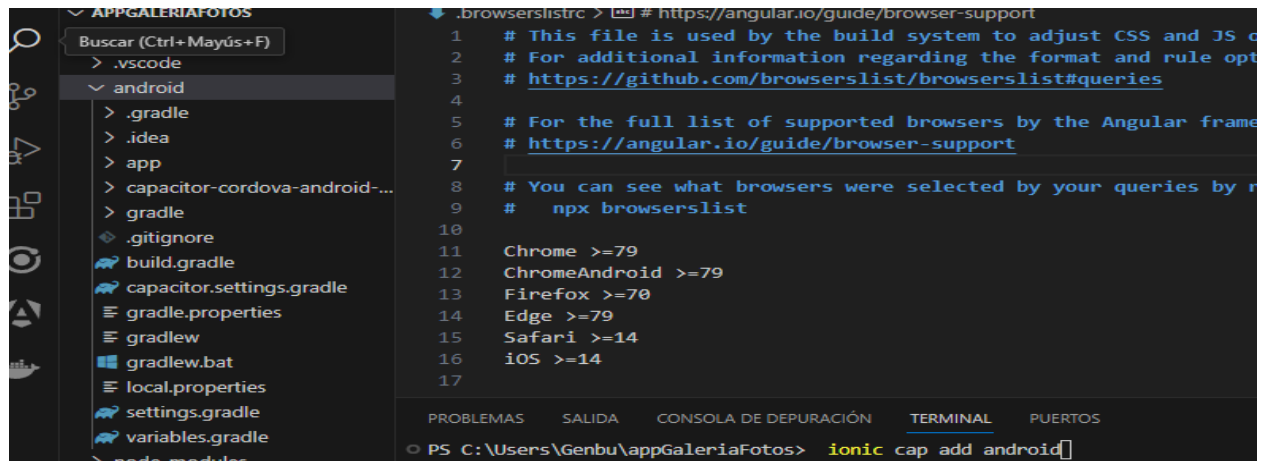


```
87 public async loadSaved() {
88
89     //Leer cada foto almacenada en el sistema de archivos
90     const readFile = await Filesystem.readFile({
91         path: foto.filepath,
92         directory: Directory.Data
93     })
94
95     //solo para plataforma Web: cargar las fotos en base64
96     foto.webviewPath = `data:image/jpeg;base64,${readFile.data}`
97 }
98
99 }
```

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	TERMINAL	PUERTOS
5504.889b1c23afd76af6.js			tab2-tab2-module	1.17 kB   562 bytes
108.966146782d659f98.js			tab3-tab3-module	1.17 kB   590 bytes
5828.1dfbe797e46048e4.js			-	1.09 kB   626 bytes
8595.7cdb90ece3e387c8.js			md-transition-js	1.04 kB   487 bytes
7312.e75dc83cbdc98e05.js			status-tap-b41ece3c-js	530 bytes   315 bytes

Build at: 2024-03-04T01:13:14.400Z - Hash: 368229e98c28f9f2 - Time: 83808ms  
PS D:\PROGRAMACION\Ionic\appFotoGaleria>

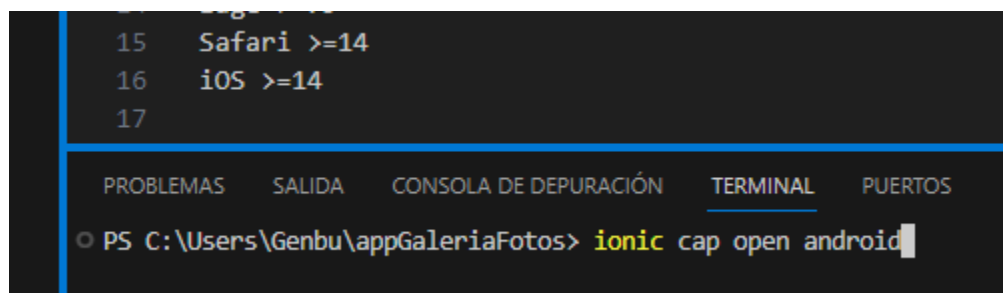
Ahora se crea la carpeta para Android con el comando “ionic cap add android” y deberá crear la carpeta con el mismo nombre



```
1 # https://angular.io/guide/browser-support
2 # This file is used by the build system to adjust CSS and JS
3 # For additional information regarding the format and rule opt
4 # https://github.com/browserslist/browserslist#queries
5 # For the full list of supported browsers by the Angular frame
6 # https://angular.io/guide/browser-support
7
8 # You can see what browsers were selected by your queries by r
9 # npx browserslist
10
11 Chrome >=79
12 ChromeAndroid >=79
13 Firefox >=70
14 Edge >=79
15 Safari >=14
16 iOS >=14
17
```

PS C:\Users\Genbu\appGaleriaFotos> ionic cap add android

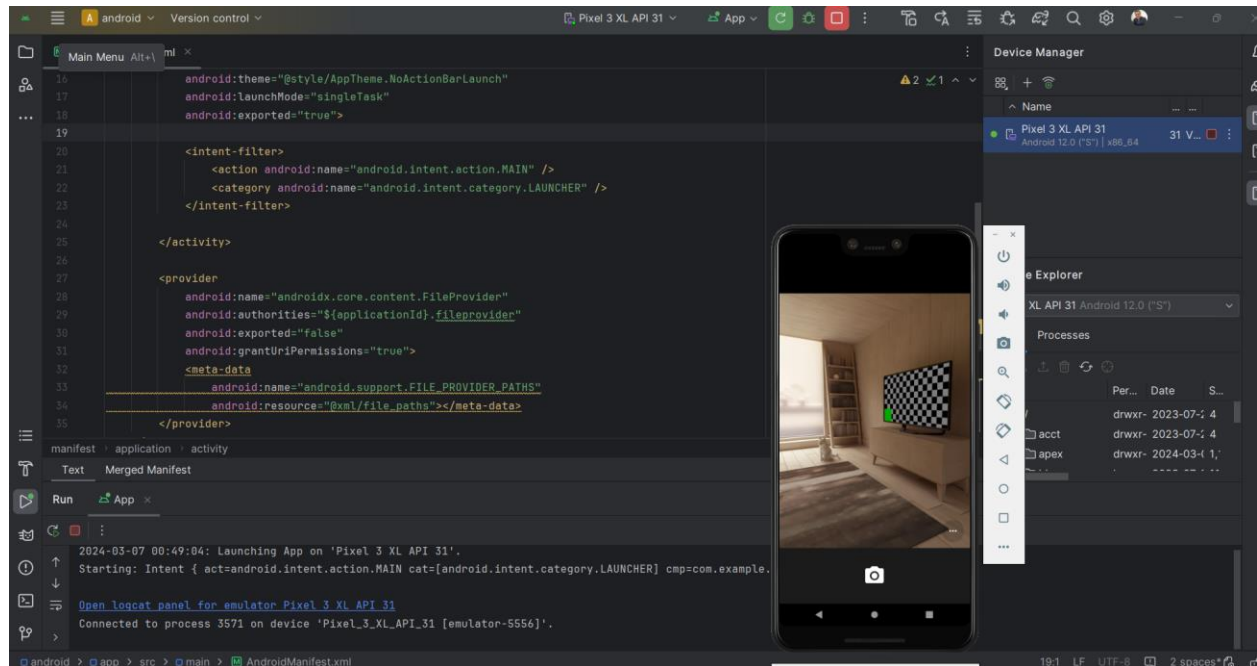
Posterior a esto se ejecuta el comando “Ionic cap open Android” que ejecutara la aplicación en ANDROID STUDIO.



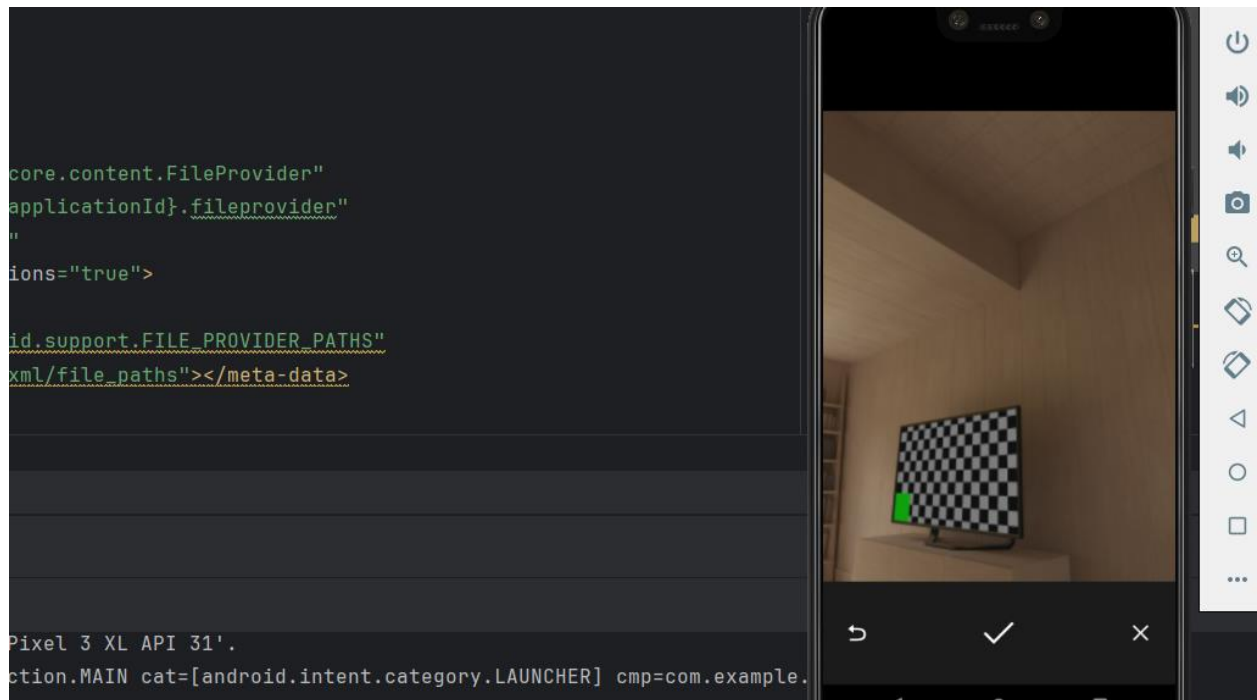
```
15 Safari >=14
16 iOS >=14
17
```

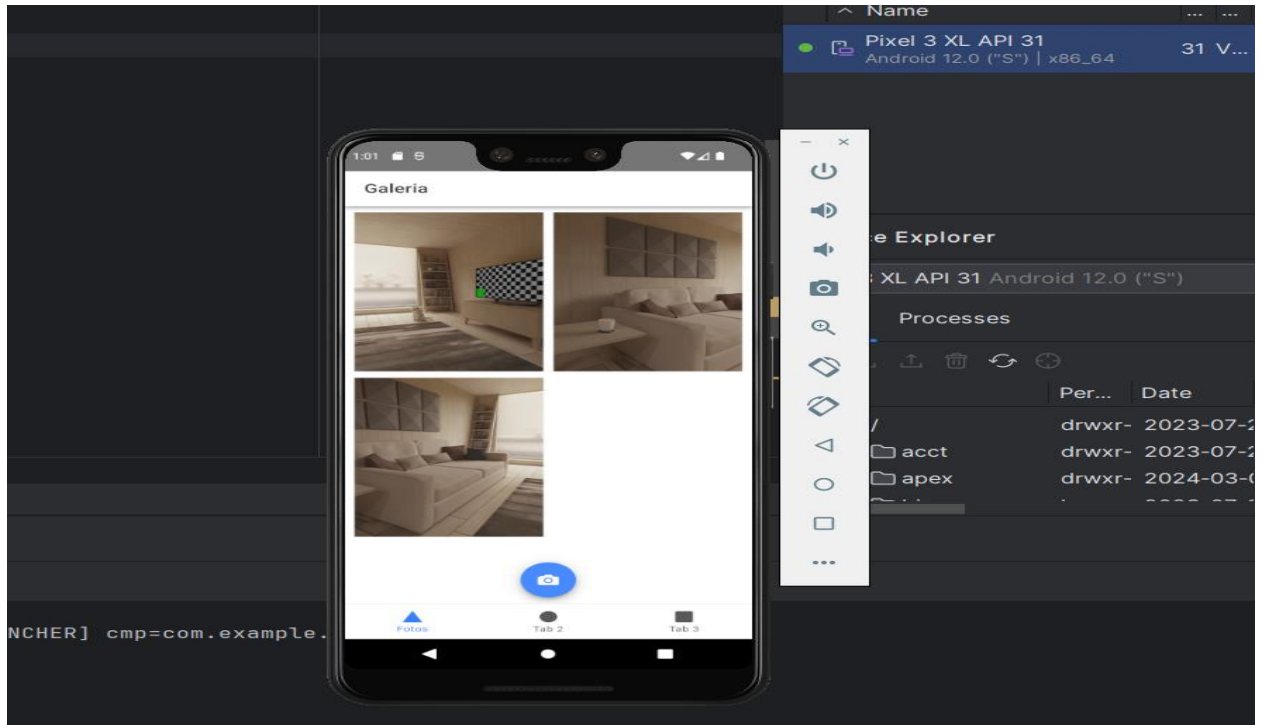
PS C:\Users\Genbu\appGaleriaFotos> ionic cap open android

Luego de parametrizar el “emulador” con el que se va trabajar se ejecuta la aplicación y se puede visualizar una cámara virtual para hacer pruebas de la app.

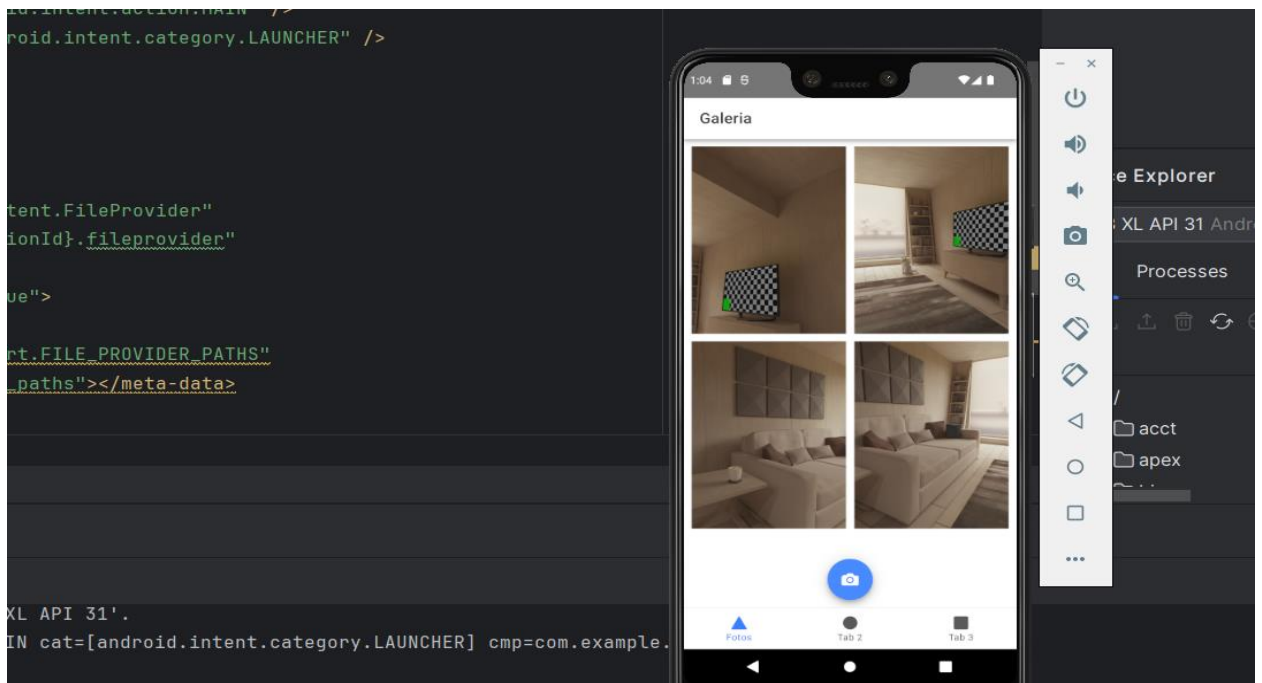


Con la tecla Alt (presionado) y el cursor(mouse) moveremos la cámara, tomamos la foto y le damos al check de aceptar





## Prueba de que se guardan en la galería



## **Recursos Bibliograficos**

Ingeniería de Software – UTA. (10 de agosto de 2021). App Ionic galeria de fotos y cámara [Archivo de video]. Youtube.

<https://youtu.be/jGEAqQbHNic?si=LmRKXL6ZIBek1ML>

Code Bless You, (7 de marzo de 2022). Npm ERR code ERESOLVE react | Npm install error in visual studio code [Fast] [Archivo de video]. Youtube.

<https://youtu.be/vn97BdYx2K8?si=UKiXPMppmoQlkXl8>

Tutorials Home. (25 de diciembre de 2022). how to fix android studio emulator is not working [Archivo de video]. Youtube

<https://youtu.be/P3B2opeHpV8?si=kBNe51E2DDukm4LP>

Cristian Henao. (31 de enero de 2019). Como resolver problema del Emulador en AndroidStudio (Hardware acceleration HAXM) [Archivo de video]. Youtube

[https://www.youtube.com/watch?v=qjXxvzfedaA&t=29s&ab\\_channel=CristianHenao](https://www.youtube.com/watch?v=qjXxvzfedaA&t=29s&ab_channel=CristianHenao)

Natasa Tech. (22 de agosto de 2023). #Solution - 10 | Emulator Process for AVD has terminated error | #androidstudio #natasatech [Archivo de video]. Youtube

[https://www.youtube.com/watch?v=1nfiqHTsioc&ab\\_channel=NATASATech](https://www.youtube.com/watch?v=1nfiqHTsioc&ab_channel=NATASATech)