UNIVERSIDAD ESTATAL PENINSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TECNOLOGIAS DE LA INFORMACION

ASIGNATURA:

APLICACIONES MOVILES

TEMA:

INVESTIGACION 1 C1

ESTUDIANTE:

BENALCAZAR YAGUAL LUIS ARIEL

DOCENTE:

ING. JAIME OROZCO

CURSO:

TI 6/1

INTRODUCCION

Este es una documentación, donde se presentará paso a paso el proceso de creación de una aplicación para dispositivos móviles, Android, que logra leer códigos QR y cogidos de barra, al igual que interpretarlos.

Se inicializa el proyecto ejecutando los siguientes comandos en la terminal (anterior, ubicado en la carpeta donde queramos guardar el proyecto):

1. Ejecutar: Ionic start <Nombre de la App> <plantilla> --capacitor
    a. **Ionic start app_qr_barcode tabs –capacitor**
2. Seleccionamos como framework: Angular
3. Seleccionamos: Standalone

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\ariel\Documents\APPS_MOBILES_6-1\Nueva carpeta> ionic start app_qr_barcode tabs --capacitor
```

```
Please select the JavaScript framework to use for your new app. To bypass this prompt next time, supply a value for the
--type option.

? Framework: (Use arrow keys)
> Angular | https://angular.io
  React   | https://reactjs.org
  Vue     | https://vuejs.org
```

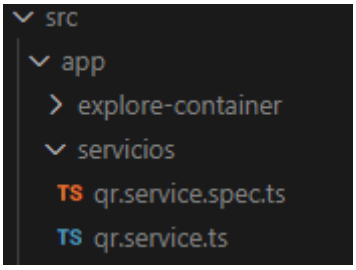
```
? Framework: Angular
? Would you like to build your app with NgModules or Standalone Components?
 Standalone components are a new way to build with Angular that simplifies the way you build your app.
 To learn more, visit the Angular docs:
 https://angular.io/guide/standalone-components


  NgModules
> Standalone
```

Luego abrimos la carpeta del proyecto con visual, tanto abriéndola desde VSC como en el mismo terminar entrar al carpeta y hacer un **code .**

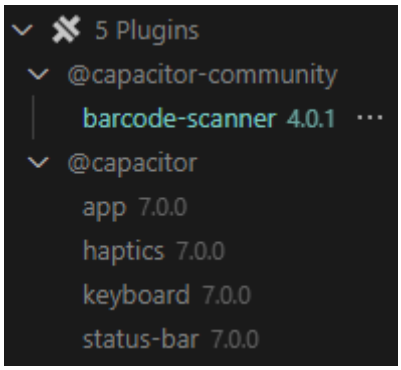
Proyecto

1. Como ya creamos un proyecto con Tabs, ya tenemos paginas para usar, asi que directamente vamos a crear un servicio con el comando:
    a. **npx g -s servicios/qr**

    Esto nos generara una carpeta con los archivos qr.services.spec.ts y qr.service.ts



2. Ahora vamos a instalar la siguientes dependencias:
    a. **npm install @capacitor-community/barcode-scanner**
    b. **npm install @capacitor/status-bar**



3. Ahora dentro de Nuestro main.ts agregamos:

```
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
import { StatusBar } from '@capacitor/status-bar';

import { AppModule } from './app/app.module';

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.log(err));

  const setStatusBar = async () => {
    await StatusBar.setOverlaysWebView({ overlay: false });
  };

  setStatusBar();
```

4. Luego en nuestro qr.services.ts agregamos:

```
import { Injectable } from '@angular/core';
import { BarcodeScanner } from '@capacitor-community/barcode-scanner';

@Injectable({
  providedIn: 'root'
})
export class QrService {
  scan: boolean = false;
  scanResult: string = '';

  constructor() {}

  async CheckPermission(): Promise<boolean> {
    try {
      const status = await BarcodeScanner.checkPermission({ force: true });
      return status.granted ?? false;  // Asegurar que retorna un booleano
    } catch (error) {
      console.error('Error al comprobar permisos:', error);
```

```
      return false;  // En caso de error, asumir que no hay permiso
    }
  }

  async StartScan() {
    if (this.scan) {
      this.StopScan();
      return;
    }

    this.scanResult = ''; // Limpiar el resultado anterior
    this.scan = true;

    this.scan = true;
    const permission = await this.CheckPermission();  // Esperar la
  respuesta de los permisos

    if (!permission) {
      alert('No se ha concedido permiso para acceder a la cámara');
      this.scan = false;
      this.scanResult = 'Error, no hay permisos';
      return;
    }

    try {
      await BarcodeScanner.hideBackground();
      document.querySelector('body')?.classList.add('scanner-
  active');

      const result = await BarcodeScanner.startScan();
      console.log('Resultado:', result);

      BarcodeScanner.showBackground();
      document.querySelector('body')?.classList.remove('scanner-
  active');
      this.scan = false;

      if (result.hasContent) {
        this.scanResult = result.content;
      } else {
        this.scanResult = 'No se detectó ningún código';
      }
    } catch (error) {
      console.error('Error en el escaneo:', error);
      this.scanResult = 'Error durante el escaneo';
      this.scan = false;
    }
  }

```

```
  StopScan() {
    BarcodeScanner.showBackground();
    BarcodeScanner.stopScan();
    document.querySelector('body')?.classList.remove('scanner-
  active');
    this.scan = false;
    this.scanResult = 'Escaneo cancelado';
  }
}

```

5. Luego en nuestro tab1.page.ts agregamos:

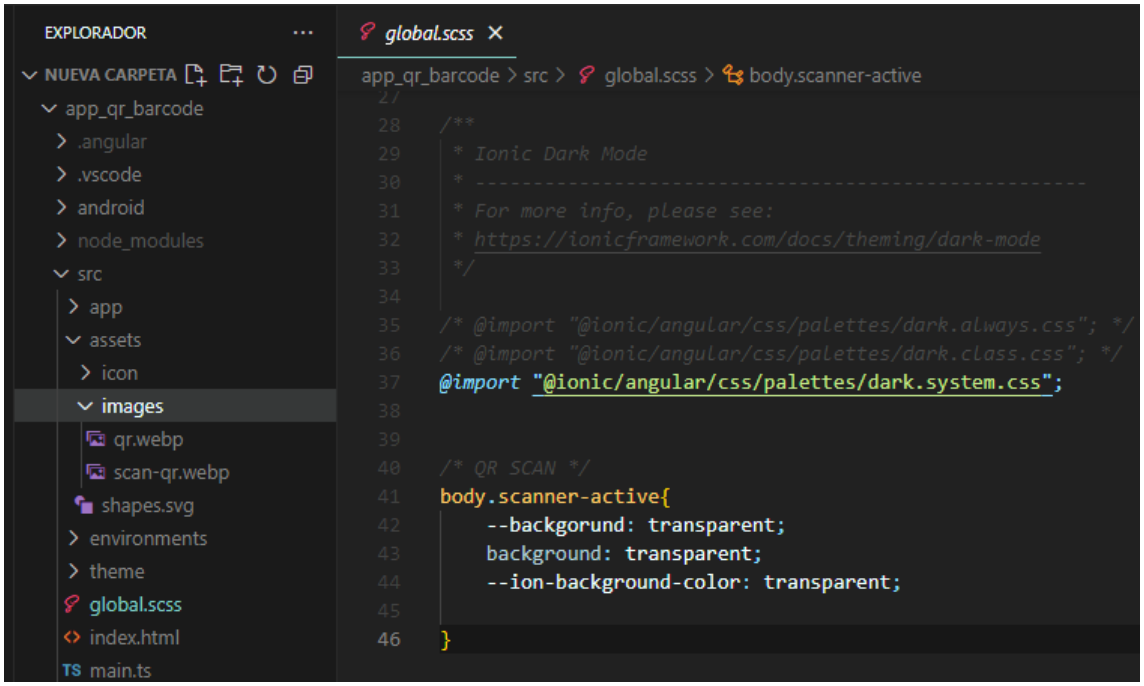
```
import { Component } from '@angular/core';
import { QrService } from '../servicios/qr.service';

@Component({
  selector: 'app-tab1',
  templateUrl: 'tab1.page.html',
  styleUrls: ['tab1.page.scss'],
  standalone: false
})
export class Tab1Page {

  objectJSON = false;
  JsonData : any;

  constructor(public qr: QrService) {}

  async scaner(){
    await this.qr.StartScan();
    try {
      let parseResult = JSON.parse(this.qr.scanResult);
      console.log(parseResult);
      if(parseResult.escaneo){
        this.objectJSON = true;
        this.JsonData = parseResult.data;
      }
    } catch (error) {
      console.log('No es un JSON');

    }

  }

}

```

6. Ahora podemos diseñar nuestro html y scss del tab1:

HTML	SCSS

<ion-header [translucent]="true">
<ion-toolbar>
<ion-title style="text-align: center;">
SCAN BAR APP
</ion-title>
</ion-toolbar>
</ion-header>
 	
<ion-content [fullscreen]="true">
 RESULTADO:
 	
<!-- Si hay un código escaneado, mostrar resultado -->
<div class="result_center" *ngIf="qr.scanResult">
 	
<a [href]="qr.scanResult" target="_blank">{{ qr.scanResult }}
 	
<div *ngIf="objectJSON">
 	
OBJECTOS QR
<ul *ngFor="let data of JsonData">
  {{ data }}
 	
<!-- Si NO hay un código escaneado y NO se está escaneando -->
<div class="no_result" *ngIf="!qr.scanResult && !qr.scan">
<img src="/assets/images/scan-qr.webp" alt="">
 No hay códigos escaneados aún
 Presiona el botón SCAN para iniciar un escaneo.
 	
</ion-content>
 	
<ion-footer>
<ion-button (click)="scaner()" id="btn_qr" expand="block" fill="clear" shape="round" class="btn_qr">
SCAN
<ion-icon slot="end" name="scan"></ion-icon>
</ion-button>
</ion-footer>
	.result_center{
	display: flex;
	justify-content: center;
	align-items: center;
	height: 100%;
	width: 100%;
	}
	
	.btn_qr{
	background-color: #000000 ;
	}
	
	.no-scan {
	display: flex;
	flex-direction: column;
	align-items: center;
	text-align: center;
	}
	
	img {
	
	width: 40%; */* Ajusta el tamaño de la imagen según necesites */*
	height: 40%;
	margin-bottom: 10px; */* Espaciado entre la imagen y el texto */*
	
	}
	
	.no_result{
	margin-top: 200px;
	display:flex;
	flex-direction: column;
	align-items: center;
	text-align: center;
	}
	
	.no-scan h2 {
	color: gray;
	}

7. Tambien agregamos una carpeta con imágenes y una modificación del global.scss:



Exportación a Android

Para realizar la exportación a Android, en la terminal del proyecto ejecutamos los siguientes comandos:

1. ionic build
2. **npm install @capacitor/Android**
3. npx cap add Android
4. npx cap sync

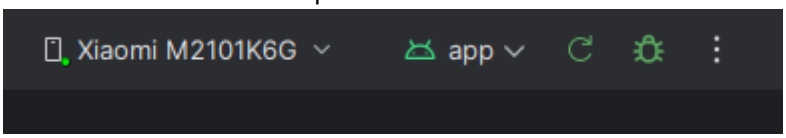
```
PS C:\Users\ariel\Documents\APPS_MOBILES_6-1\Nueva carpeta\app_qr_barcode> ionic build
> ng.cmd run app:build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.
```

```
Build at: 2025-03-17T15:06:13.461Z - Hash: b7e37feaf60e0eda - Time: 6261ms
PS C:\Users\ariel\Documents\APPS_MOBILES_6-1\Nueva carpeta\app_qr_barcode> npx cap add android
√ Adding native android project in android in 99.17ms
√ add in 100.30ms
√ Copying web assets from www to android\app\src\main\assets\public in 542.71ms
√ Creating capacitor.config.json in android\app\src\main\assets in 885.20µs
√ copy android in 581.79ms
√ Updating Android plugins in 20.78ms
[info] Found 5 Capacitor plugins for android:
       @capacitor-community/barcode-scanner@4.0.1
       @capacitor/app@7.0.0
       @capacitor/haptics@7.0.0
       @capacitor/status-bar@7.0.0
√ update android in 105.81ms
√ Syncing Gradle in 452.10µs
[success] android platform added!
Follow the Developer Workflow guide to get building:
https://capacitorjs.com/docs/basics/workflow
PS C:\Users\ariel\Documents\APPS_MOBILES_6-1\Nueva carpeta\app_qr_barcode> npx cap open android
[info] Opening Android project at: android.
```

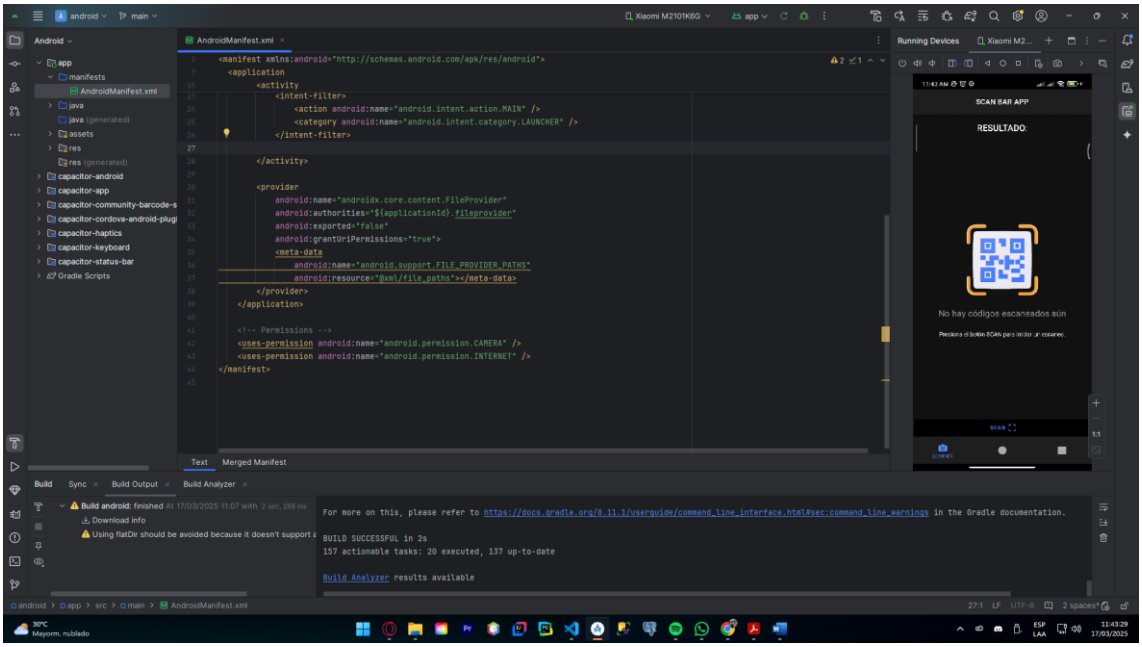
8. Agregamos esta línea en nuestro ***AndroidManifest.xml***

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

9. Ahora corremos nuestra aplicacion:

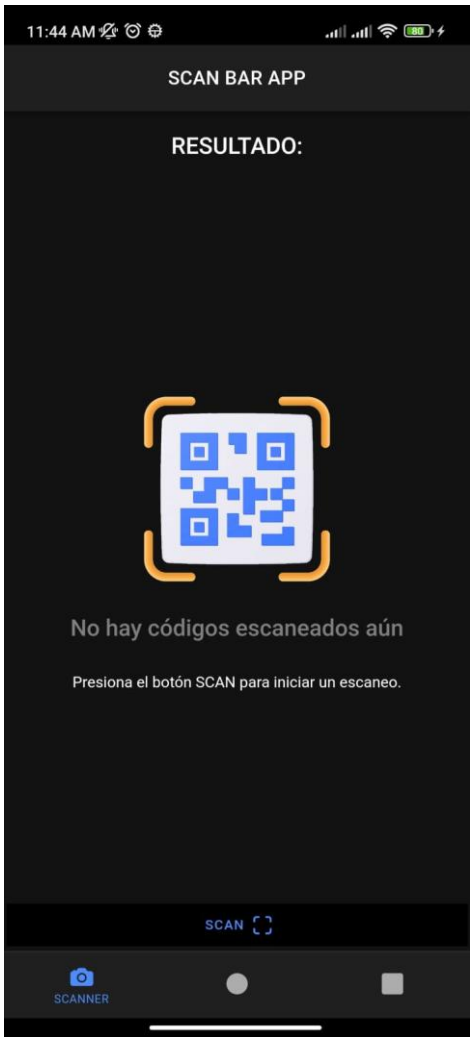


10. Y lo Podemos observar en el mirrorDevice



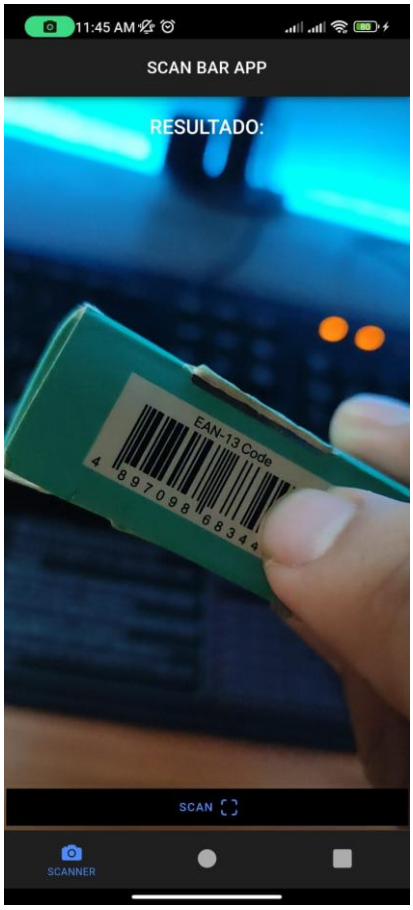
SCREENSHOTS DE LA APP

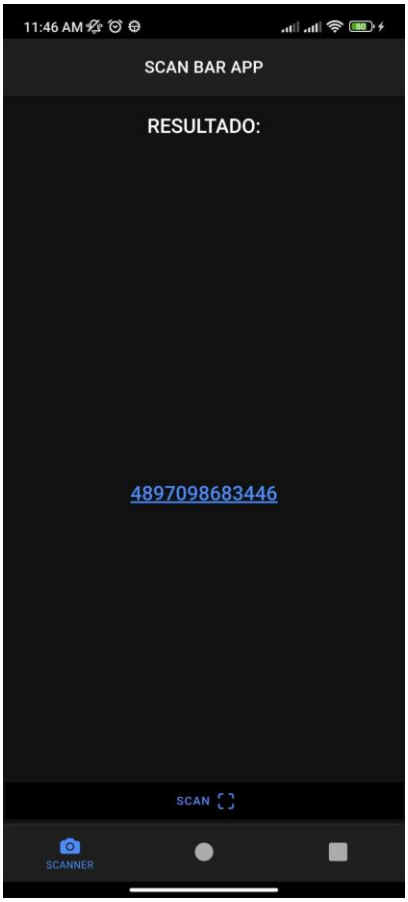
Esta es la interfaz principal de la aplicación

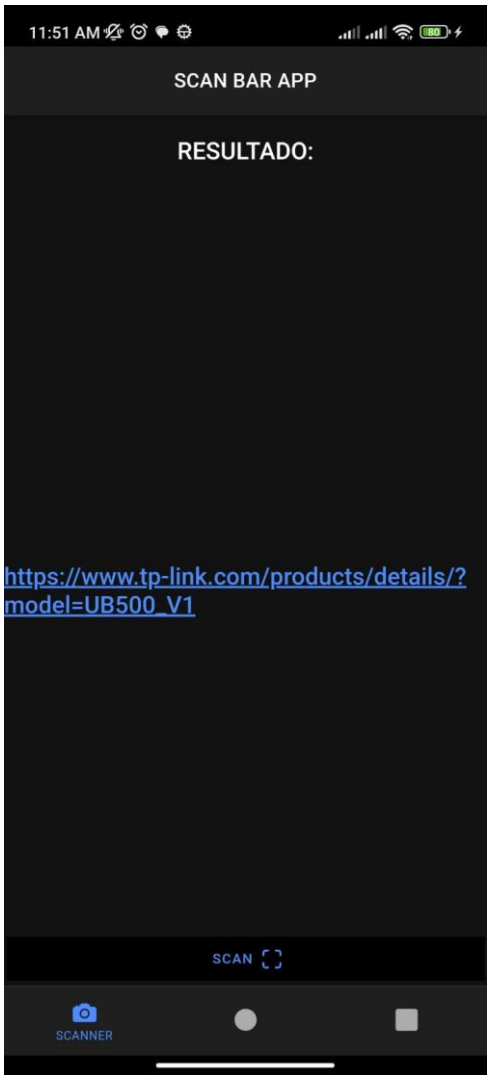


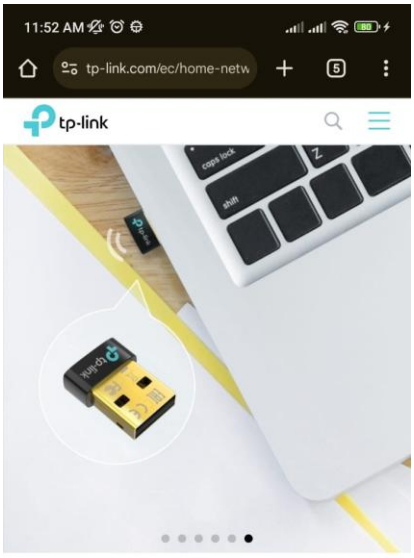
Al darle tap al botón scan, nos pedirá permisos para uso de la cámara:

Posterior se abrirá la cámara y podemos escanear Codigos de barra como QR
los códigos de barra nos dira el valor que contiene, mientras los QR nos mandara al enlace que contiene el QR:



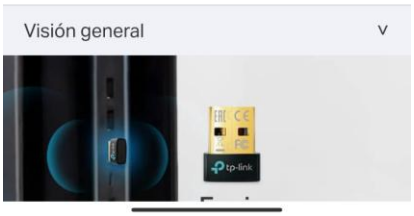






UB500

Adaptador Nano USB Bluetooth 5.0



Repositorio en GITHUB:

https://github.com/Artemis184/QrApp