

Periféricos y dispositivos de interfaz humana

Proyecto QR



**UNIVERSIDAD
DE GRANADA**

Jordi Pereira Gil
DNI: 35674006V

Introducción.....	3
Menú.....	3
Leer QR.....	3
Crear QR.....	4

Introducción

Para este proyecto he creado una aplicación android la cual va a permitir leer códigos QR obteniendo su texto y generarlos desde la misma.

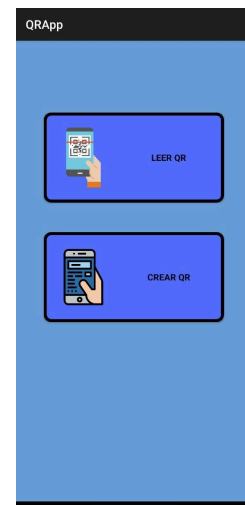
Esta aplicación tiene 3 pantallas principales (y una de splash que sale al iniciar la aplicación), que son la de menú, la de leer QRs y la de crearlos.

A continuación se hablará de cada página y de los algoritmos pertinentes para el correcto funcionamiento de la aplicación.

Menú

Esta página es la primera que nos aparece después de ver el splash inicial. Como podemos apreciar se compone de 2 botones que nos llevan a la ventana de leer QR y crearlo respectivamente.

En lo que respecta a la creación de los botones se ha usado un objeto *Button* los cuales tienen una imagen a la izquierda y un texto a la derecha.

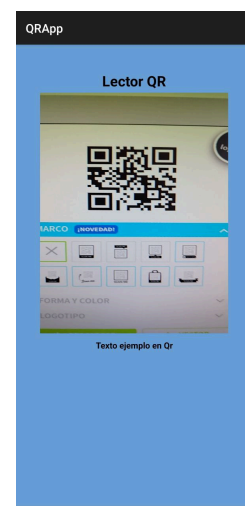


Leer QR

Esta página nos permite leer códigos QR de manera sencilla, gracias a que como vemos en la imagen hay una vista de la cámara con la que hemos leído el contenido del QR el cual nos ha dado como resultado "Texto de ejemplo en Qr" el cual es correcto.

A continuación explicaremos el flujo y funcionamiento de la página de lectura de códigos.

Lo primero que hacemos es comprobar que el usuario tenga activados los permisos para el uso de la cámara en la aplicación, en caso afirmativo saldrá un mensaje en la parte de abajo del móvil que dirá que todo es correcto, si no nos saldrá un mensaje pidiéndonos el permiso de



la cámara en este momento deberemos darle a permitir, en caso contrario no podremos usar la página.

Una vez permitido el uso de la cámara se iniciará el QR con el método *startQR()* encargado de iniciar la vista de la cámara y llamar a la función de lectura de códigos QR.

Para iniciar la vista previa de la cámara comprobaremos si se tiene permisos de uso de la cámara o no, en caso de tenerlos inicializará la cámara.

Para escanear los QR lo primero que hace el código es crear un procesador para la clase *BarcodeDetector* utilizando una clase anónima que implementa la interfaz *Detector.Processor*. En esa clase anónima se implementan el método *release()* y el *receiveDetections()*, este último se llama cuando se detectan nuevos códigos QR. En el caso de detectar un código QR, el código comprueba si es diferente al último código QR detectado y los almacena en el objeto *SparseArray<Barcode>* llamado *barcodes*. Si su longitud es mayor que 0 se coge el valor del qr con la siguiente línea de código:

```
qr = barcodes.valueAt(0).displayValue;
```

Si es anterior al último código leído el programa no hará nada, pero si es nuevo éste se almacena en la variable *last_qr* e imprime su valor por pantalla como podemos apreciar en la imagen de arriba. Además si el texto obtenido es un enlace, se abre una ventana de navegador a esa página.

Crear QR

Esta página nos permite crear códigos QR a partir de una entrada de un texto que escribamos en la aplicación. El uso de la página es sencillo, escribes un texto en el input le das al botón de generar y se te genera un QR con el contenido adecuado.

Para poder crear códigos fácilmente se ha importado la librería *zxing* para ello en hemos tenido que poner la siguiente línea en el archivo *build.gradle.kts*:

```
implementation ("com.google.zxing:core:3.4.1")
```

Una vez implementado podemos importar las dependencias necesarias para la generación de QRs



```
import com.google.zxing.BarcodeFormat;
import com.google.zxing.EncodeHintType;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.google.zxing.qrcode.QRCodeWriter;
```

Una vez tengamos todo a punto podemos empezar a programar la página.

Lo primero es crear un evento de pulsado en el botón, en este introduciremos toda la lógica, debemos definir el alto y el ancho del código QR, hemos decidido 300 de ancho y alto para que sea cuadrado.

Lo siguiente que haremos será obtener valor del input de la página y comprobamos que no esté vacío, con éste creamos un *bitmap* usando el método *generateQRCode()*.

Este método lo primero que hace es crear un objeto *HashMap* llamado *hints* para almacenar los parámetros de codificación del código QR, en este caso *UTF-8*.

Lo siguiente que hacemos es crear una matriz de bits con la clase *QrCodeWriter* importada anteriormente. Esta matriz representa el código QR correspondiente al contenido proporcionado, el formato de código de barras y las dimensiones especificadas, utilizando los parámetros de codificación almacenados en *hints*.

Creamos un objeto de la clase *Bitmap* del mismo tamaño que la matriz de bits y vamos rellenando cada pixel de la matriz de color negro si el valor de la matriz de bits es verdadero y blanco si es falso.

Finalmente el *Bitmap* es devuelto al lugar de donde se llamó el método.

Una vez creado el QR añadimos la imagen a la página y hacemos desaparecer el teclado de la pantalla.