

CÓDIGOS QR



- 1. ¿Qué es un Código Qr?.*
- 2. ¿Cómo funciona un Código Qr?.*
- 3. ¿Para qué se usa un Código Qr?.*
- 4. ¿Cómo se utiliza un Código Qr?.*
- 5. Cómo crear y/o generar Códigos Qr.*
- 6. Mi App*

Los códigos están muy presentes actualmente en nuestro día a día ya que estos nos sirven como herramienta de almacenamiento y también para poder distribuir información, otra ventaja clave que tienen es que son rápidos y accesibles para todas las personas.

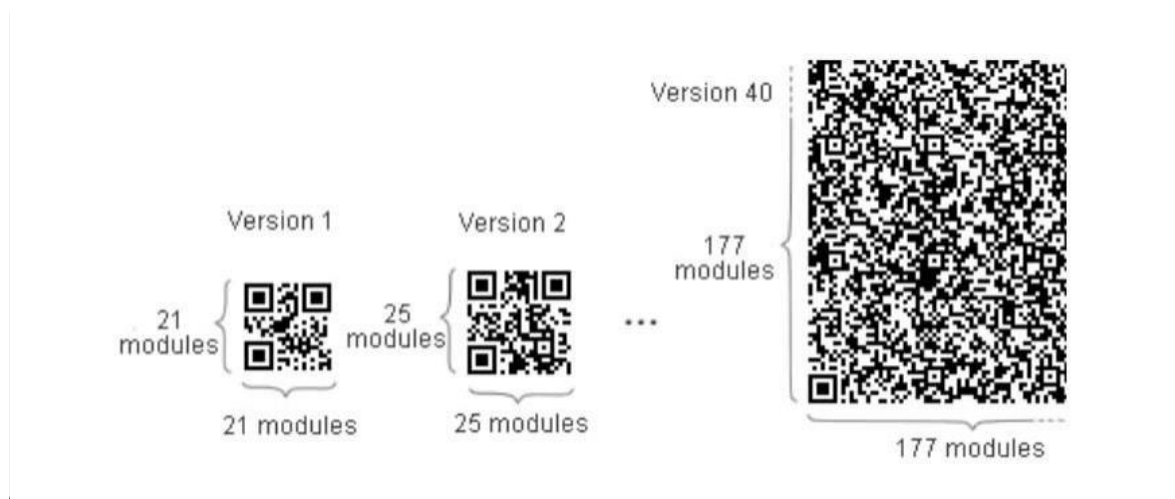
Conforme el paso del tiempo el uso que se le ha ido dando a esta herramienta ha sido más amplio, llegando al punto de ser una herramienta imprescindible para ciertos ámbitos. El caso más reciente sería el pasaporte COVID.

1. ¿Qué es un código Qr?

Un código QR es un código de respuesta rápida (*Quick Response*) que es una evolución de los códigos de barras debido a la necesidad de aumentar la cantidad de información y posibilidades de uso que éstos podían ofrecer. Fueron creados en Japón, en 1994, por una subsidiaria de Toyota y se extendieron de forma masiva por el país asiático a finales de la década. En junio de los 2000 se aprobó la estandarización internacional ISO que usamos hoy en día.

Como todo cualquier código de barras, un Qr tiene como función principal el almacenar información en una etiqueta óptica legible por algún tipo de máquina o dispositivo. Para ello, incluye una matriz de puntos bidimensionales en formato cuadrado comúnmente en blanco y negro, aunque también hay colores, y con tres cuadrados en las esquinas que permiten al lector que se use detectar la posición del código.

La capacidad de información que pueden incluir esta definida por niveles según la cantidad de módulos que forman la matriz: desde el nivel 1 con 21 x 21 módulos hasta el 10 que puede incluir hasta 177 x 177 módulos. Los códigos Qr más comunes son los de **25 x 25** y los de 29 x 29 debido a su equilibrio entre tamaño y capacidad de alojar información.



Ventajas principales del código Qr respecto a los códigos de barras:

- **Gran capacidad de información:** Un código de barras puede almacenar hasta unos 20 dígitos mientras que un Qr puede almacenar hasta varios cientos de veces más información.
- **Tamaño de impresión pequeño:** Al tener información tanto en vertical como en horizontal se puede codificar la misma información en la décima parte de un código de barras tradicional.
- **Resistente a daños y suciedad:** Al llevar claves de corrección de errores se puede recuperar hasta un 30% de bytes según el daño que haya sufrido el código.
- **Legible desde cualquier dirección en 360°:** Gracias a los patrones de detección de dirección situados en las 3 de las esquinas del código se puede leer el Qr desde cualquier posición.
- **Concatenación del código:** Un Qr se puede dividir en códigos Qr más pequeños que al concatenarlos contienen la misma información que el código original.



2. ¿Cómo funciona un código Qr?

Un código Qr a la vista del ojo humano puede parecer un conjunto de cuadritos aleatorios y símbolos raros sobre un fondo blanco, pero esa estructura tan rara es la que les permite almacenar información y que puedan ser codificables para un escáner.



Aunque la mayoría de los Qr son en blanco y negro también se permite añadir ciertos colores a ellos. En estos casos, los colores deben de mantener un contraste que los haga legibles para los sistemas como para los lectores de códigos. Cómo podemos apreciar el contraste tiene que ser notorio, priorizar más colores oscuros que resalte sobre el fondo blanco.

También se puede añadir grafos o logotipos para que sean únicos y atractivos de cara al público. Por ejemplo, puedes incluir el logo de tu empresa que te redireccione a su web para obtener más información de ella a modo de carta de presentación. O como advertencia de enfermedades o alergias que pueden ser leídos por sanitarios o fuerzas de seguridad en casos de accidente o emergencia. Por último, el uso de textos incrustados es otra ampliación a los genéricos gracias a la redundancia de información basada en la corrección de errores *Reed-Solomon* que incluyen.

- ***Código Qr con logotipo:***

Código Qr con un logotipo en el centro.



¿Es escaneable con logotipo?

Si, debido a que el código tiene corrección de error, que ayuda al código a seguir siendo escaneable incluso si el 30% del código esta dañado.

El nivel de daño depende del nivel de error correction aplicado: L(7%), M(14%), Q(21%), y H(30%).

Así siempre se recomienda seleccionar el nivel Q o H mientras se crea este tipo de códigos, le añadirá más filas y columnas de módulos de datos, pero aseguras una exploración más alta.

- ***Código Qr con imagen de fondo:***

Uso de una imagen como el fondo y añade un código Qr en él.

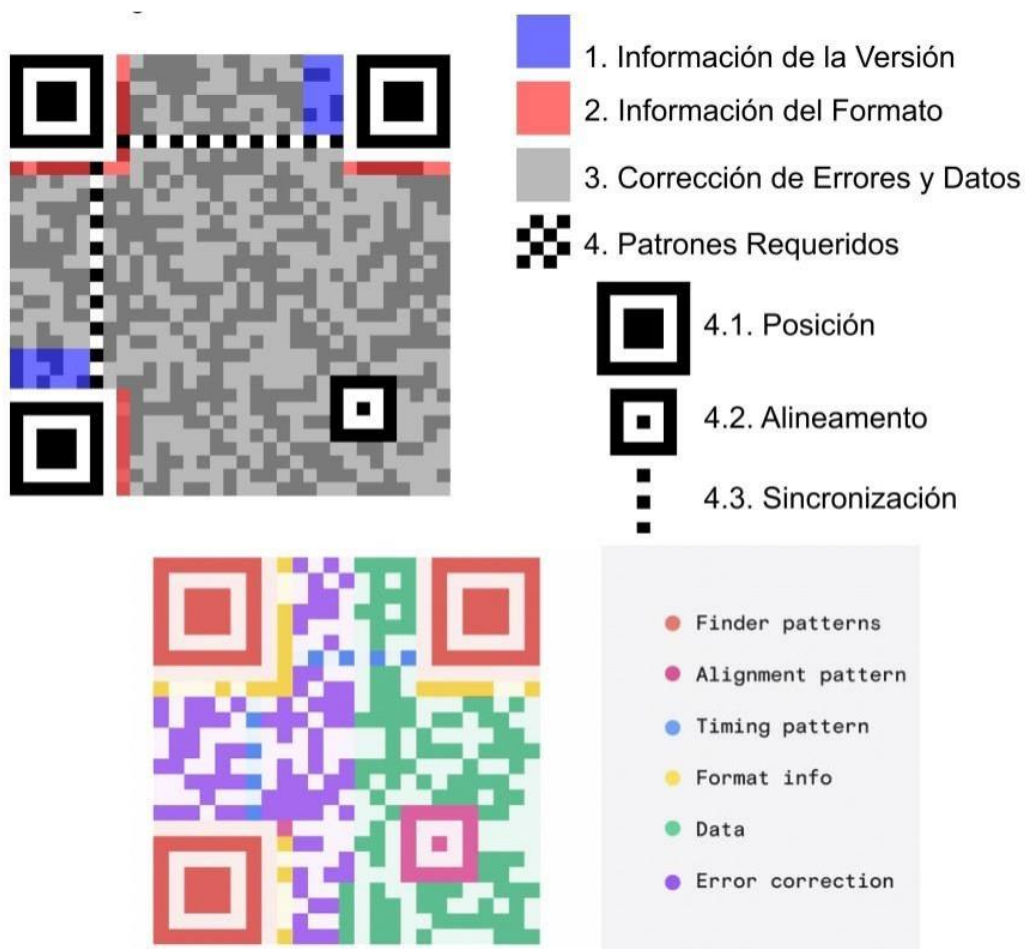


¿Es escaneable con una imagen de fondo?

Si, debido a que se sigue manteniendo un alto contraste entre los elementos del código y el fondo. También, a diferencia de un código Qr de diseño con un logotipo, el código de fonde de imagen no elimina los patrones de datos asique puede mantener el nivel de corrección de errores bajo.

- Estructura de un código Qr:

- 1. Información sobre la versión:** estos patrones especifican la versión del código QR. Actualmente hay 40, que van desde la versión 1 que tiene 21×21 módulos, hasta la versión 40 con 177×177 módulos. Las versiones 1 a 7 son las más utilizadas para el marketing móvil.
- 2. Información de formato:** estos patrones almacenan información sobre la corrección de errores y la máscara de datos. La función de corrección de errores garantiza que el código QR pueda decodificarse incluso si está parcialmente cubierto o dañado.
- 3. Datos y claves de corrección de errores:** estos patrones almacenan la información real del código QR.
- 4.1. Patrón de posición:** estos patrones indican la dirección correcta del código QR.
- 4.2. Patrón de alineación:** este patrón permite que el código QR se decodifique en 360°.
- 4.3 Patrón de temporización:** estos patrones permiten que el decodificador determine el ancho de la matriz de datos.
- 5. Zona tranquila:** este espacio crucial permite al decodificador identificar el código QR de su entorno.



- ***Función de corrección de errores:***

A diferencia de los códigos de barras, los códigos 2d si admiten una función integrada de corrección de errores. En la era digital, un error en la industria de los negocios te puede hacer perder millones de dinero e incluso ir a la quiebra por eso hay que buscar una solución, algo que disminuya la cantidad de errores que hay, especialmente en herramientas de almacenamiento de información.

Gracias al algoritmo de corrección de errores Reed-Solomon, los códigos Qr aún se pueden escanear incluso teniendo errores, ya sea por desgaste o porque estén sucios, etc.

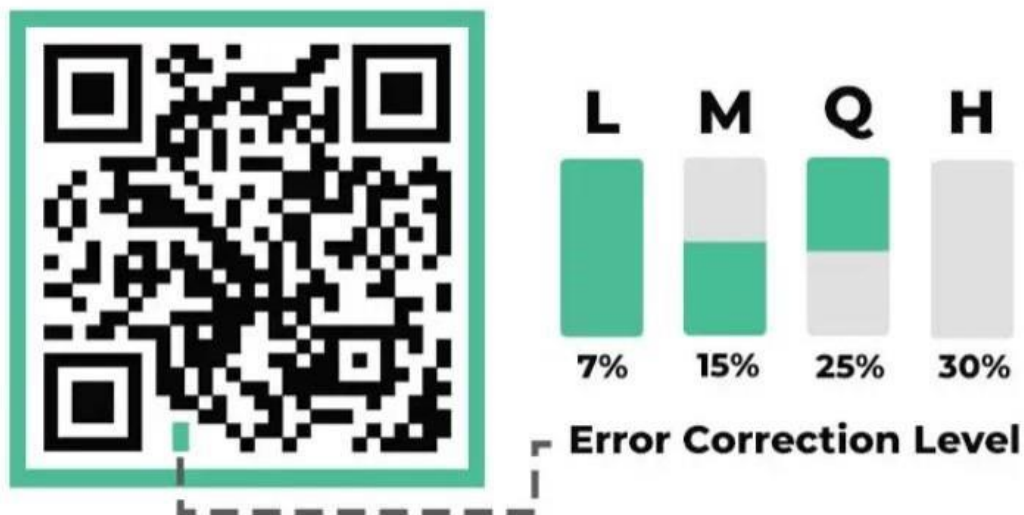
Esto lo que te proporciona a nivel de herramienta es seguridad debido a que tu información va a estar ahí siempre, por ejemplo, si es información de alguien esa info estará ahí segura, si es una URL, mientras no caduque o cambien de dominio, estará segura.

¿Cómo funciona la función de corrección de errores del Código Qr?

La función de corrección de errores funciona debido al algoritmo de Reed-Solomon, este algoritmo duplica los datos originales para una capacidad de escaneo óptimo.

De esta manera, seguirá siendo escaneable incluso si el código ha sufrido un desgarró o daño debido a condiciones climáticas externas.

Dependiendo del desgarró o daño existen varios niveles:



NIVEL L



Es el nivel más bajo de tasa de corrección de errores que puede tener un código Qr. El software del código usa éste nivel cuando el usuario tiene la intención de generar una imagen de código menos densa.

Tiene la tasa de corrección de error más alta de aproximadamente un **7%**.

NIVEL M

Es el nivel medio del nivel de corrección de errores que los expertos en Qr recomiendan para uso de marketing. Debido a esto, los especialistas en marketing pueden corregir sus códigos a un nivel medio.

Tiene la tasa de corrección de error más alta de aproximadamente **15%**.



NIVEL Q



Es el segundo al nivel más alto de corrección de errores, tiene la tasa de corrección más alta aproximadamente **25%**.

NIVEL H

Es el nivel de corrección de errores más alto que puede soportar un nivel extremo de daño en su código.

Los niveles de corrección de errores de nivel Q y H son los más recomendados para empresas industriales y manufactureras.

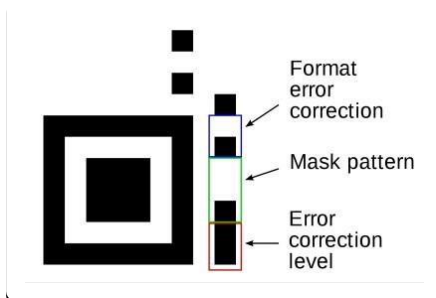
Estos códigos pueden experimentar condiciones ambientales y provocadas por el hombre que conducen a daños inevitables.

Tiene una tasa de corrección de error más alta de aproximadamente **30%**.



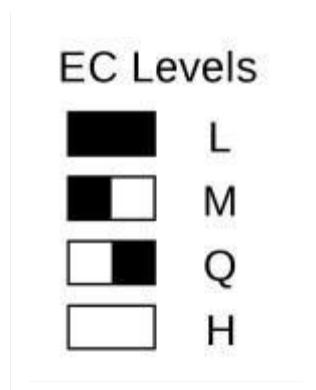
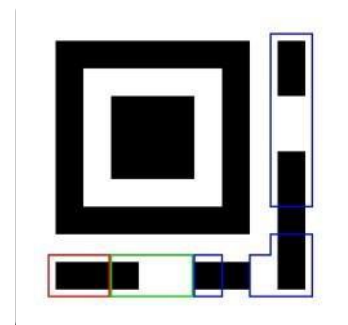
¿Cómo detectar estos niveles de corrección de errores?

Miramos en el área de corrección de errores del Qr, por ejemplo, te puedes fijar en la parte inferior izquierda, donde tenemos uno de los 3 símbolos de posición del Qr, pues a su derecha



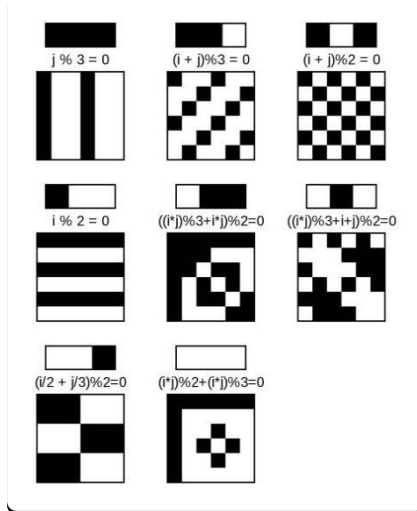
tenemos la línea de información de formato, que está compuesto por el formato de error, la máscara de patrón y luego el nivel de error de corrección.

También podemos observarlo en el símbolo de posición de arriba a la izquierda, que en éste caso tendremos la línea que nos interesa justo debajo de ella, la parte que muestra el nivel de corrección de error sería la de la izquierda del todo.



Una vez localizado nos fijamos si está sombreado o no esa parte, si está sombreado entero es Nivel L, si está sombreado la parte izquierda y la derecha no es M, si es al revés, izquierda no sombreada y derecha sombreada, es Q y si no lo está ninguna de las dos es Nivel H.

- **Patrón de Máscara:**



Al codificar un código QR, hay 8 patrones de máscara que se pueden usar para cambiar la matriz de salida. Cada patrón de máscara cambia los bits según sus coordenadas en la matriz QR.

El propósito de un patrón de máscara es hacer que el código QR sea más fácil de leer para un escáner QR.

Cada patrón de máscara utiliza una fórmula para determinar si se cambia o no el color del bit actual. Pones las coordenadas del bit actual en la fórmula, y si el resultado es 0, usas el bit opuesto en esa coordenada. Por ejemplo, si el bit para la coordenada (0,3) es 1 y la fórmula es igual a 0 para esa coordenada, entonces pones un 0 en (0,3) en lugar de un 1.

- **Capacidad de Almacenamiento de un código Qr:**

La cantidad de datos que un código puede almacenar depende de la versión, su nivel de corrección de errores y tipo de datos.

Hay 4 tipos:

- **Alfanumérico:** puede almacenar hasta 4.296 caracteres.
- **Numérico:** puede almacenar hasta 7,089 caracteres.
- **Binario:** puede almacenar hasta 2,953 bytes.
- **Kanji:** puede almacenar hasta 1.817 caracteres.

3. ¿Para qué se usa un código Qr?

Como pueden almacenar diferentes tipos de datos e información, los códigos Qr se utilizan para muchos propósitos.

- Como texto simple, por ejemplo, para mensajes de bienvenida en conferencias.
- Direcciones: domicilio personal, domicilio comercial, etc.
- Números de teléfono: número de teléfono personal o de empresa.
- Direcciones de correo electrónico: cuentas personales o comerciales.
- URL con direcciones de sitios web o páginas web específicas.
- Vínculos a apps, por ejemplo, las que dirigen a las tiendas Google Play o la Apple App Store.
- Pagos: los códigos QR pueden almacenar información sobre su cuenta bancaria o tarjeta de crédito.
- Autenticación de cuentas en línea. Los sitios web pueden mostrar un código QR que un usuario registrado puede escanear con su teléfono inteligente e iniciar sesión automáticamente.
- Autenticación Wi-Fi. Los códigos QR se pueden usar para almacenar detalles de autenticación de redes inalámbricas como SSID, contraseña y tipo de cifrado.
- Claves de acceso de verificación de dos pasos. Se utilizan durante la configuración de la seguridad de autenticación 2FA un grupo cada vez mayor de sitios web y aplicaciones.
- Otros usos diversos. Por ejemplo, de marketing; para ver la carta de un restaurante; en manejo de monedas virtuales o hasta usos funerarios que en Japón se incluyen en lápidas que apuntan a páginas web que contienen información sobre los fallecidos.

4. ¿Cómo se utiliza un código Qr?

Su primer uso en Japón fue para registrar los repuestos en fábricas de automóviles y hoy se usan masivamente en todo tipo de industrias para gestión del inventario, al igual que en programas de marketing, los de asistencia de clientes o simplemente para incluir la identidad visual de una empresa. Sin embargo, la inclusión de software capaz de leer códigos QR y la explosión de ventas de teléfonos móviles, ha conseguido que su uso en consumo sea también masivo.

La pandemia del coronavirus que nos está tocando sufrir también los ha impulsado, incluidos en los 'pasaporte COVID' o simplemente para digitalizar los menús de un restaurante, evitar el contacto con la carta física y reducir los riesgos de transmisión comunitaria. Herramientas avanzadas como el Contact Less Menu permiten a través de la lectura de los códigos QR no solo visualizar los distintos productos sino generar directamente un pedido tal y como haríamos en comercio electrónico.

Si en sus inicios necesitaban dispositivos industriales fijos o de mano para leer los códigos, hoy se pueden usar casi cualquier aparato electrónico, PCs, tablets o wearables, aunque lo que ha facilitado enormemente su uso ha sido el impresionante despliegue de smartphones.

De hecho, su uso más común hoy (dejando un lado el segmento industrial) es usando la cámara de un móvil inteligente para escanear el código y un software especializado que lo traduzca.

Muchos fabricantes de smartphones ofrecen soluciones propias nativas y hay decenas de terceros en las tiendas oficiales Play Store para Android o la App Store para iPhones. Como ejemplo, en Android nos gusta el lector y escáner de Kaspersky, mientras que para iOS una gratuita y que funciona muy bien es esta de TapMedia.



Todo lo que hay que hacer es apuntar la cámara de tu smartphone hacia la matriz del código QR y la app que uses hará el resto, generalmente en consumo con el propósito de vincularlo a alguna página web, un mapa de localización, un correo electrónico, un perfil en una red social o cualquier cosa.

Hay que destacar que la aplicación que se use para leer el código QR juega un papel muy importante en el proceso de obtención de los datos codificados en el código. Esto es debido a que la lectura del código se realiza en dos pasos: a) la cámara de un teléfono móvil se utiliza para obtener una imagen del código y b) el software utiliza diferentes algoritmos para decidir que ciertos puntos de la imagen corresponden a módulos del código. Si los algoritmos son buenos, el programa podrá compensar la falta de nitidez provocada por una mala óptica de la cámara, nuestro pulso a la hora de sujetar el móvil, la falta de luz y contraste en la imagen y usar los datos de corrección de errores para obtener la información que se ha codificado.

5. Cómo crear y/o generar Códigos Qr

Existen distintas plataformas y herramientas para crear tu propio QR, que pueden alcanzar a muchos de los usos que hemos listado, desde una página web, pasando por un menú o tarjeta de presentación.

En algunas plataformas su uso es tan sencillo como elegir el contenido que tendrá tu código, personalizarlo y diseñarlo a medida, y descargarlo en distintos formatos para que puedas imprimirlos o mostrarlos en formato digital. Estos servicios especializados añaden la posibilidad de crear QR dinámicos y editables; analíticas.

Algunos de los generadores de códigos QR más conocidos son los siguientes:

- **goQR.me** (Gratuito/Comercial). Permite codificar Texto, URL, Teléfono, SMS y vCard mediante la API de QRServer. El resultado se puede descargar en formato PNG o incrustar en una página web. Se puede seleccionar el nivel de corrección de errores, los colores de los módulos y el fondo, el tamaño de la imagen resultante en píxeles y la zona libre alrededor del código.
- **ZXing** (Gratuito). Permite codificar Evento de Calendario, vCard/MECARD, Email, Posición GPS, Teléfono, SMS, Texto, URL y Red WiFi. El resultado se puede descargar en formato PNG o incrustar en una página web. Únicamente se puede seleccionar el nivel de corrección de errores y tres tamaños fijos de la imagen resultante (small, medium, large). A destacar que el tamaño de la zona libre alrededor del código generado es mayor de lo estrictamente necesario.
- **Delivr** (Gratuito/Comercial). Permite codificar URL, Servicios (Android Market, Facebook, iPhone App Store, Twitter, YouTube, etc.), MECARD, Email, SMS, Texto y otros. Se puede seleccionar el nivel de corrección de errores, el tamaño del margen (0-10 módulos), el tamaño de la imagen (1x-4x) y el color del código. El resultado se puede descargar en formato PNG, EPS y SVG.

También tenemos la opción de programarlo mediante código, por ejemplo, usando código de lenguaje R, nos haría falta hacer uso de la librería “qrcode” de la cual podemos usar para crear los códigos qr, cuando ya la tenemos instalada la librería es cuestión de cargarla y usar la función `qrcode_gen()`.

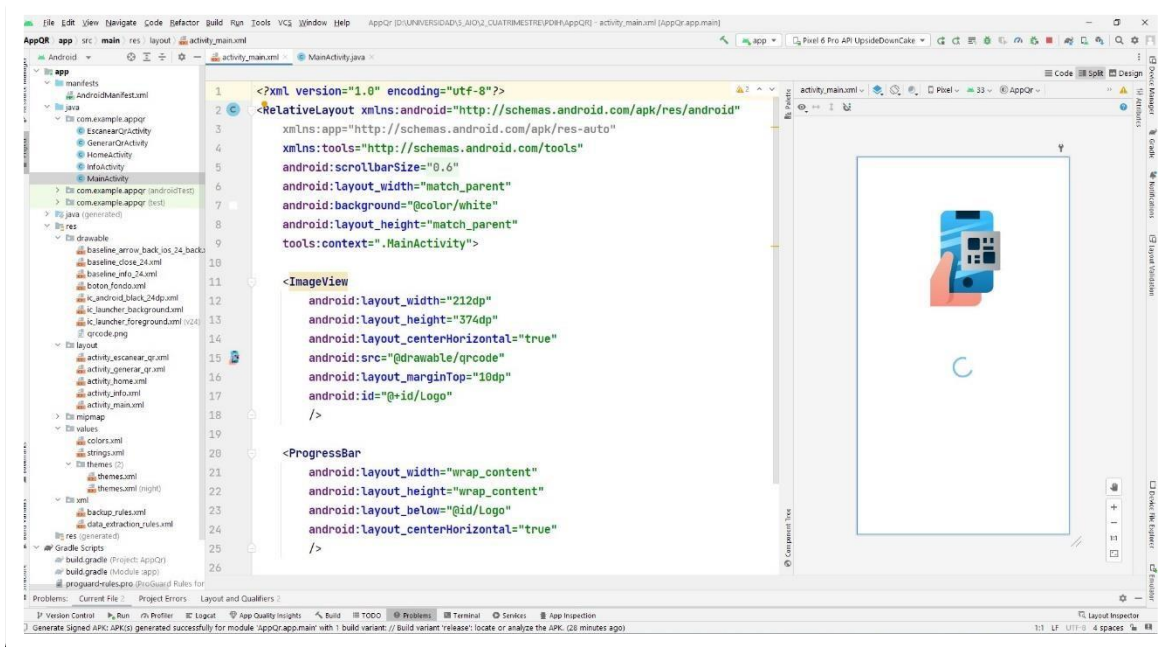
También tenemos la opción de Python en el cual se haría uso de los módulos OpenCV y qrcode para poder usar las funciones correspondientes, ya lo demás sería un proceso prácticamente igual que con R.

6. Mi App

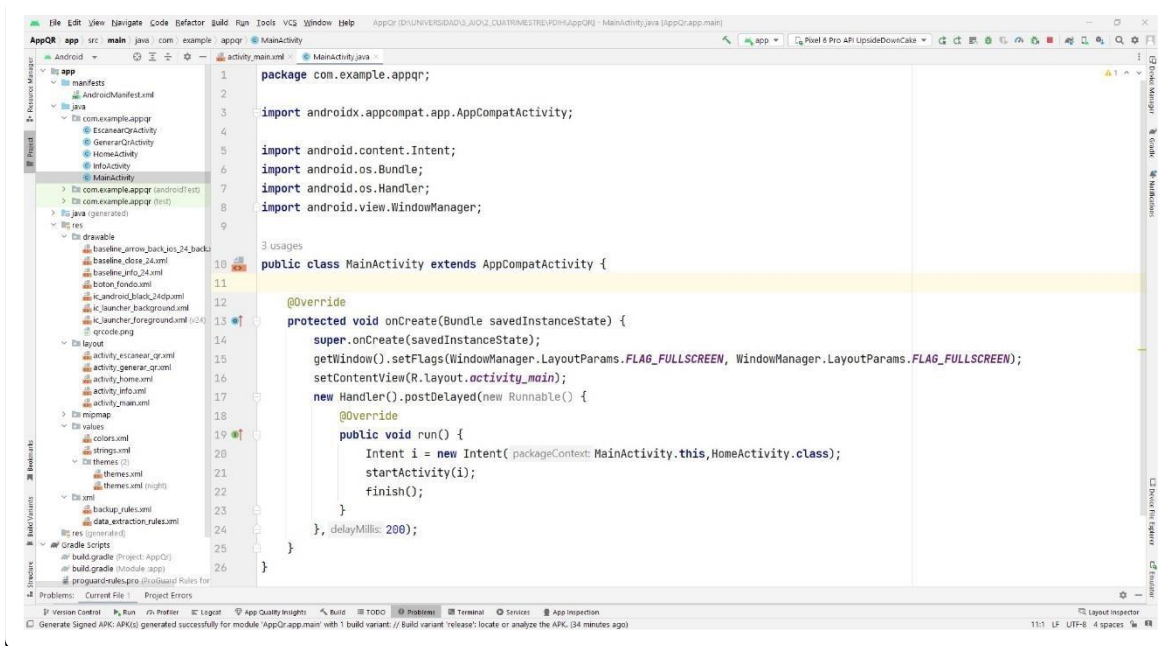
Yo he decidido diseñar un App de Android la cual consta de 2 funciones, la primera de ella es la de generar un código qr a partir de un texto que introduzcamos, ya sea un texto, números o url. La siguiente función es la más importante que es la de escanear códigos qr ya sea desde la cámara de forma directa haciendo una foto a dicho qr o desde la galería importando la imagen al programa.

Para desarrollar esta App he usado el entorno de desarrollo Android Studio, un potente editor de código y con múltiples herramientas disponibles para desarrollar apps. El lenguaje que he usado para el código ha sido Java, que como se cursa en el 2º Año del grado, nos bien bien su uso ya que manejamos el tema de interfaces en Java.

La estructura es simple, App se divide en varias ventanas, cada ventana tiene 2 archivos, uno con extensión .xml y otra con extensión .java, la primera es para el diseño gráfico y el segundo para el código y funcionamiento de los elementos gráficos.

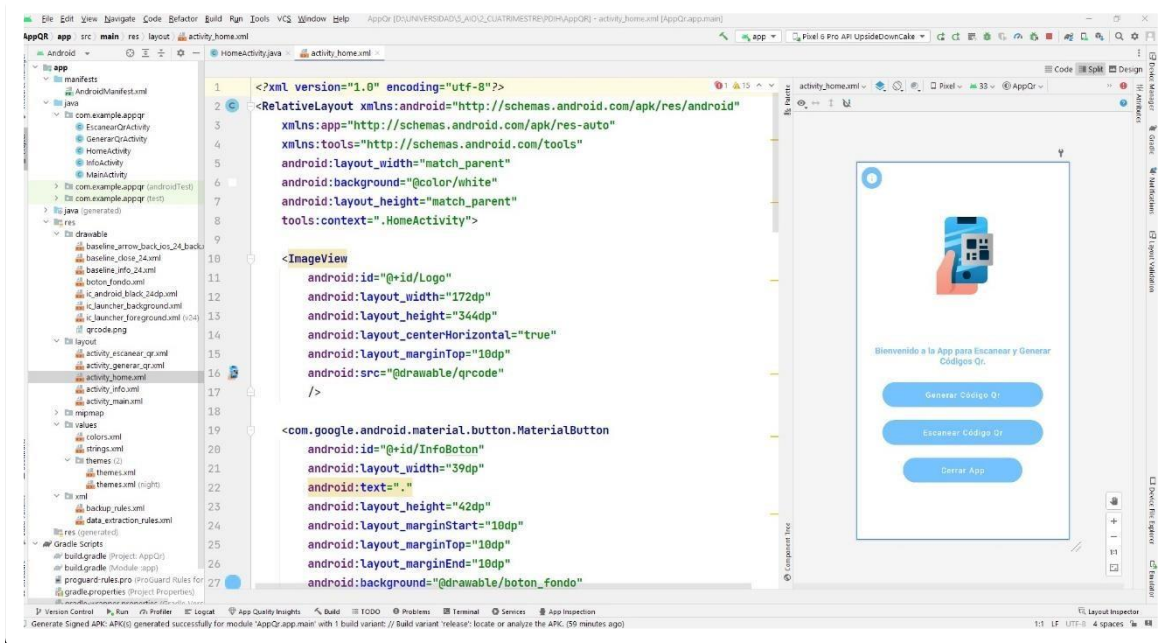


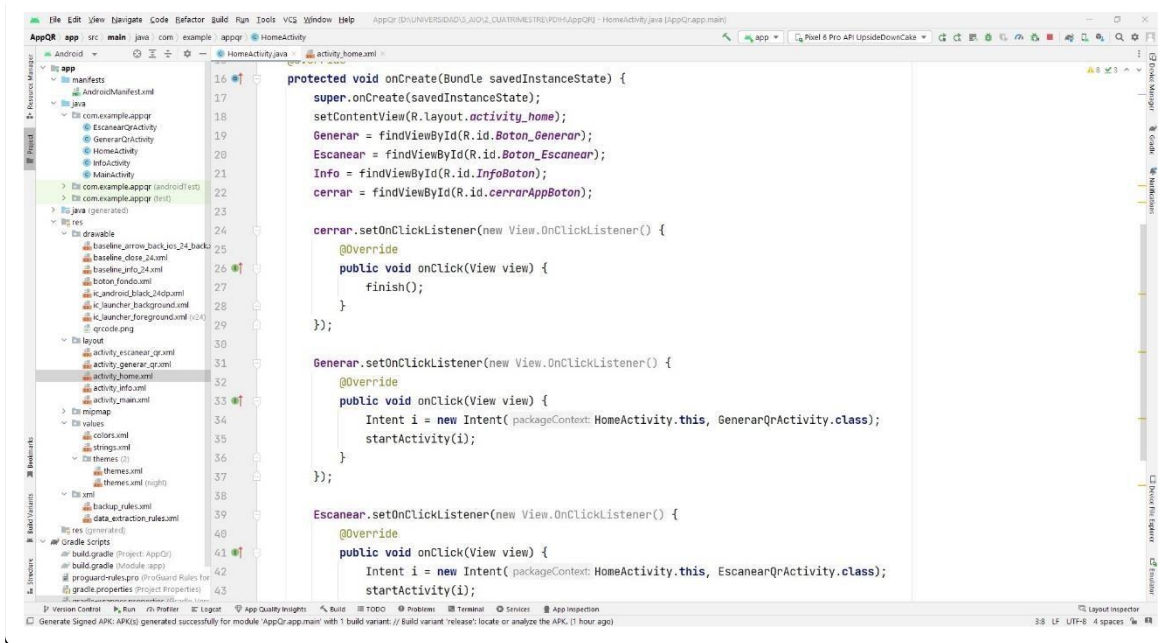
Cómo se puede ver en esta captura de imagen, esto es un ejemplo del diseño en xml de la parte gráfica de la “pantalla de bienvenida” de mi app”, tenemos la opción de poder tener una previsualización de como va quedando desarrollo de la misma mientras vamos estableciendo valores y ajustando medidas y opciones varias de los distintos herramientas que nos proporciona Android Studio.



En el segundo archivo de la “pantalla de bienvenida” tenemos la parte de programación en Java, en esta parte lo que hacemos básicamente es darle el código necesario para hacer lo que queremos que haga los objetos de la parte gráfica, en este caso no hay gran cosa sólo que cargue toda la app y pase a la siguiente ventana donde estará la app funcional en cuestión.

XML y JAVA de la pantalla principal de la App

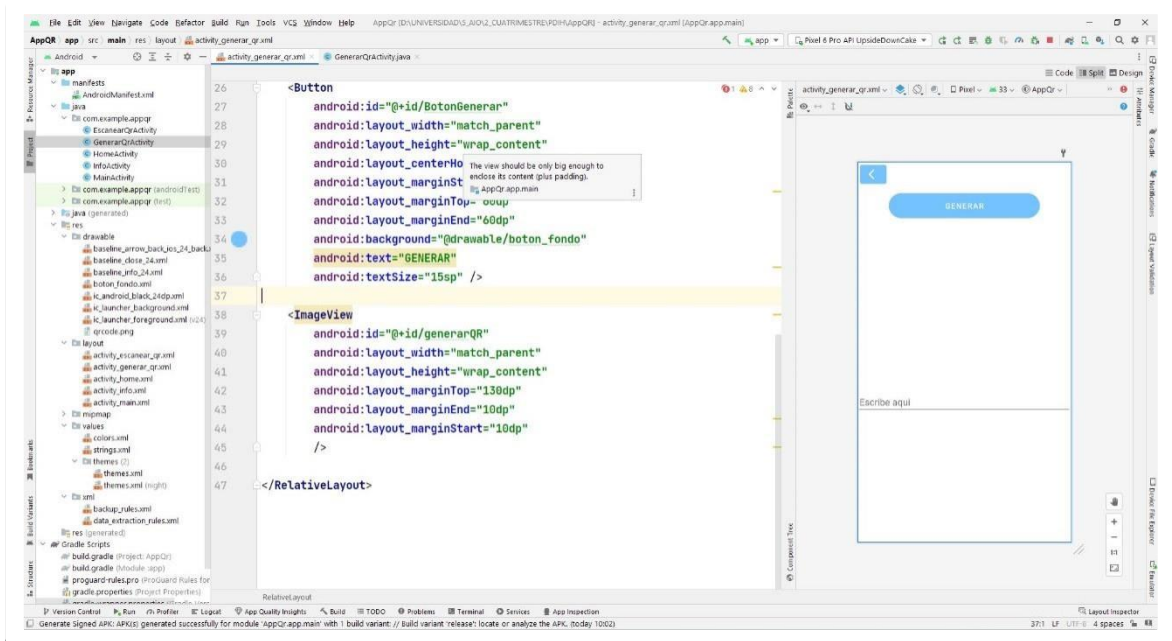


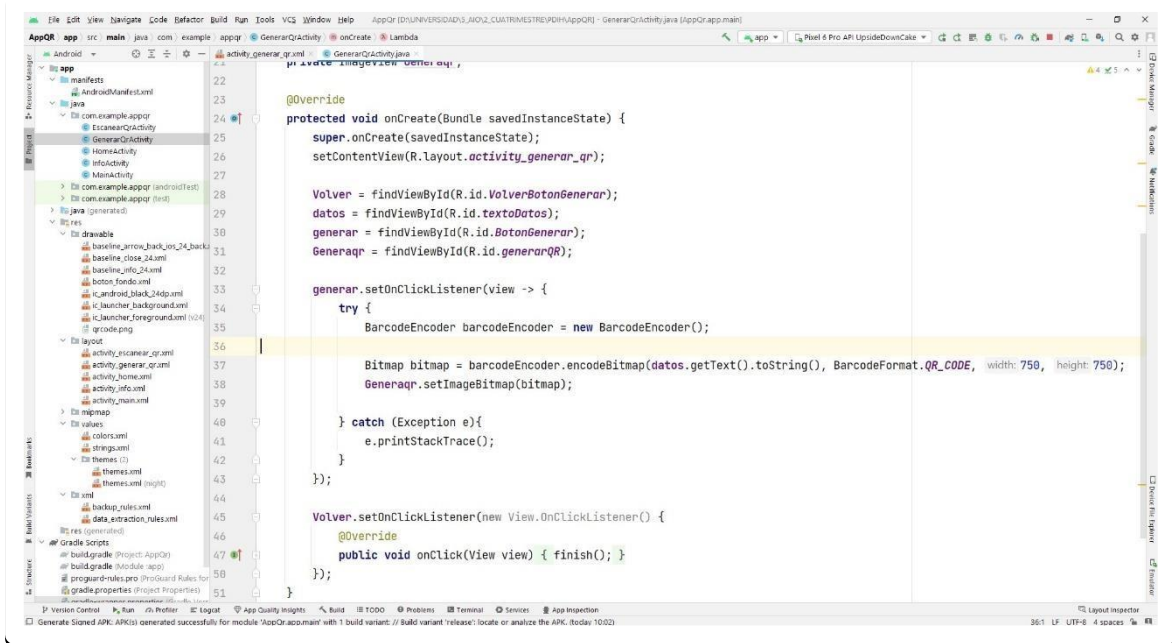


Como podemos observar en el .xml ya vemos el cuerpo de lo que será la pantalla principal de la app, conformada por el logo de la app en medio y 4 botones, uno arriba a la izquierda, el cual si pulsas en él te mandará a una pantalla nueva con la info de la app y poco más. Luego tenemos los 3 botones principales, Generar Qr, Escanear Qr y Salir de app.

En la parte del código Java podemos ver que le asigno su función respectiva a cada botón del xml, por ejemplo el boto Salir detiene la ejecución del programa y se cierra, luego los botones de Escanear y Generar Qr nos redirecciona a sus respectivas pantallas.

XML y JAVA de la pantalla Generar Qr de la App

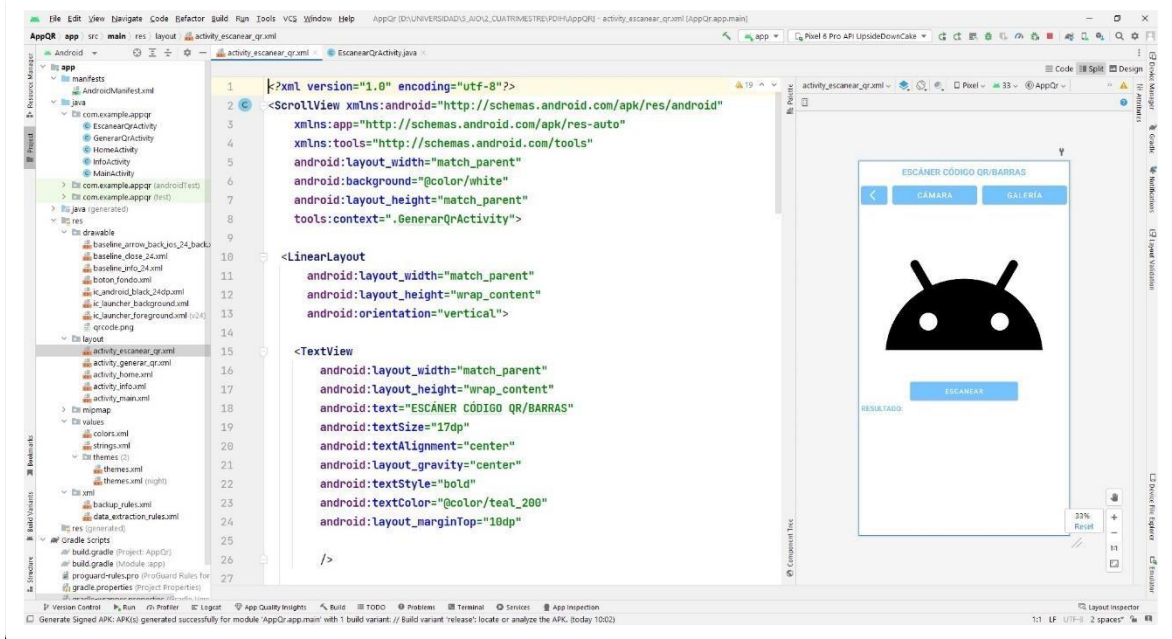


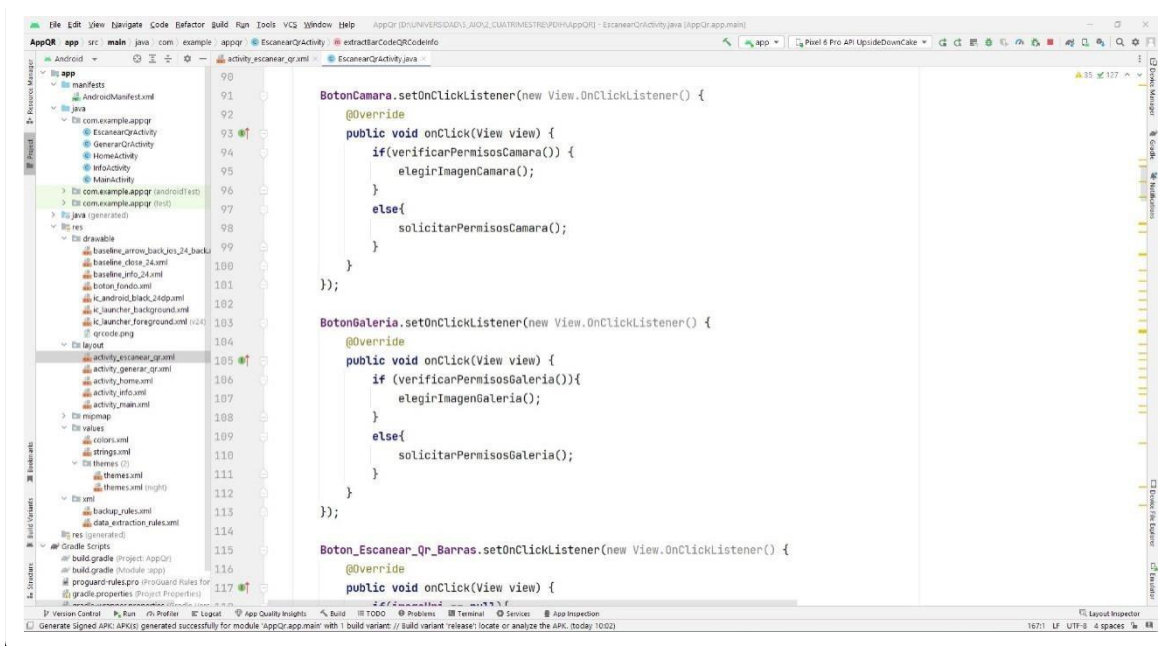


En esta ventana partimos conque tenemos el botón de ir hacia atrás, el botón de generar el Qr y una sección donde escribir texto.

En la parte del Código Java la parte a destacar es el tema de generar el qr cuando introducimos un texto, con una función de la biblioteca importada BarcodeEncoder generamos un código Qr y lo mostramos en tiempo real, en caso de cambiar algo del texto si le damos a generar se sobrescribe todo y se actualiza el Qr.

XML y JAVA de la pantalla Escanear Qr de la App





Vale, en esta parte es donde más he dedicado tiempo debido a que la biblioteca necesaria me ha ofrecido más herramientas y funciones. En el xml tenemos 4 botones, un visor de imágenes y luego una sección que nos mostrará la info del Qr escaneado.

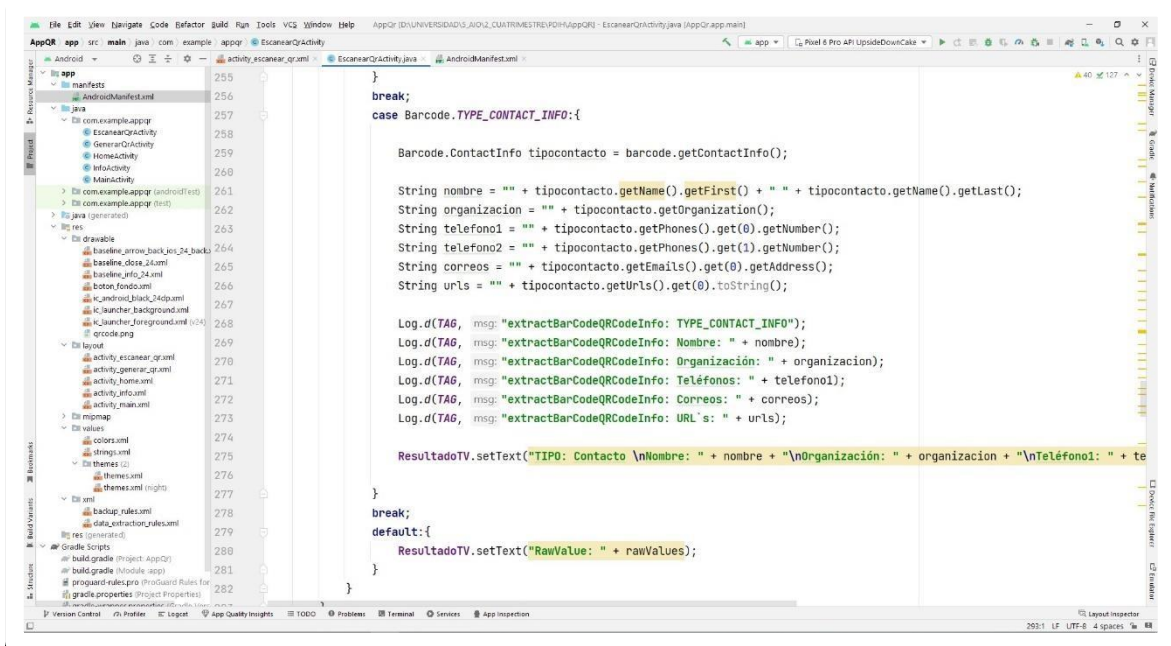
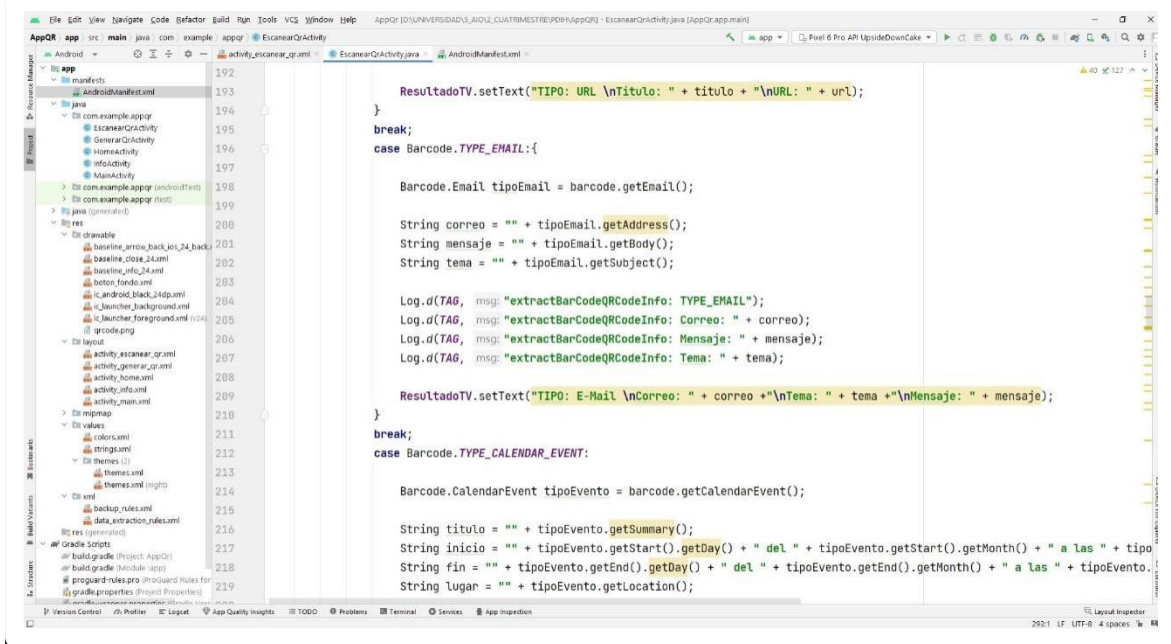
El botón de cámara consiste en acceder y abrir la cámara del móvil con la cual haremos la foto del Qr y acto seguido podremos escanearlo. Antes hay que verificar que tenga los permisos de acceso a cámara por temas de seguridad y privacidad, si tiene los permisos acceder a ella en caso negativo solicitará dichos permisos.

En el caso del botón galería es prácticamente parecido, cuando pulsamos en él comprobará si tiene los permisos de acceso, en caso de que no los solicitará, en caso afirmativo accede a la galería y podremos seleccionar la imagen a importar y escanear.

El botón escanear consiste en una vez ya tenemos la imagen en el visor de imágenes, ya estaría lista para ser escaneada, una vez le damos al botón procedemos a escanear la imagen importada y mostrar el resultado abajo con toda su información obtenida.

Cabe destacar que para poder tener acceso a funciones de la cámara del móvil y a la galería he tenido que darle permisos al programa para que pueda acceder a ellos y poder usar las funciones que dichos permisos proporcionan.

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



Y por último pero lo más importante del programa, la función encargada de sacar toda la información que nos proporciona el código Qr, consiste en mostrar por pantalla los datos de distintos tipos de códigos Qr, por ejemplo, de un Qr de tipo WiFi te muestra el SSID, la contraseña, y su encriptado, así con varios tipos, los cuales son: WiFi, URL, E-Mail, Evento del Calendario, Número de teléfono, SMS, Contacto y en caso de ser algo no relacionado con alguno de los tipos mencionados, muestra valores que pueda obtener del Qr.

CAPTURAS DEL USO DE LA APP

