



La Clase Intent y los Códigos de Barras y QR.

CONCEPTOS

Todos los dispositivos móviles Android, tiene la capacidad de leer códigos QR y códigos de barras, los cuales poseen una gran cantidad de información sobre los productos, ayuda para visitar una URL del sitio web, o guardar un contacto dentro de la memoria de nuestro dispositivo.

En este ejemplo, se revisa cómo funciona el Scanner de Android, para el código de barras y QR, que se implanta con el uso de la biblioteca ZXing (Zebra Crossing), lo que ayudará a escanear el código de barras dentro de una aplicación de Android. Se invocan recursos en esta biblioteca abierta apk dentro de la aplicación, para recuperar y procesar los datos devueltos.

El Barcode Scanner escanea productos con código de barras o códigos Data Matrix y QR, que contienen direcciones URL, información de contactos, etc. La aplicación permite compartir contactos, aplicaciones, y marcadores en código QR. Esto es el por qué se necesitan permisos. Si el dispositivo no escanea, se prueban primero las soluciones para los errores del dispositivo en **Ajustes**. Habilitarlos todos, y luego desactivar uno a la vez para determinar cuál de ellos es necesario. Si esto no funciona, tratar de borrar la memoria caché del dispositivo y los ajustes en los **Ajustes** de Android.

El escaneo se inicia con la biblioteca `com.google.zxing.client.android.SCAN`, desde un Intent:

```
public void scanBarra(View v) {
    try {
        //Inicia escaneo desde: com.google.zxing.client.android.SCAN
        Intent i1 = new Intent(ACTION_SCAN);
        i1.putExtra("SCAN_MODE", "PRODUCT_MODE");
        startActivityForResult(i1, 0);
    } catch (ActivityNotFoundException anfe) {
        //Al obtenerlo, muestra diálogo de descarga
        showDialog(MainActivity.this, "No hay scanner", "Descargar scanner?", "Si", "No").show();
    }
}
```

Cuando se termina la actividad de escaneo, se muestra el resultado con el método `onActivityResult` y extras con el formato del código:

```
public void onActivityResult(int requestCode, int resultCode, Intent i) {
    //Resultado
    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            //Obtiene extras regresados por el intento
            String sr = i.getStringExtra("SCAN_RESULT");
            String sf = i.getStringExtra("SCAN_RESULT_FORMAT");
            Toast t = Toast.makeText(this, "Content:" + sr + " Format:" + sf, Toast.LENGTH_LONG);
            t.show();
        }
    }
}
```

DESARROLLO

EJEMPLO 1.

Paso 1. Crear un proyecto `CodigoBarrasQR`. En la carpeta `java/com.example.mipaquete`, abrir y modificar el archivo Java predeterminado `MainActivity.java`, con el siguiente código:

```
import android.app.*;
import android.content.*;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class MainActivity extends Activity {
    static final String ACTION_SCAN = "com.google.zxing.client.android.SCAN";
    @Override
    public void onCreate(Bundle b) {
        super.onCreate(b);
```



```

        setContentView(R.layout.activity_main);
    }
    public void scanBarra(View v) {
        try {
            //Inicia escaneo desde: com.google.zxing.client.android.SCAN
            Intent i1 = new Intent(ACTION_SCAN);
            i1.putExtra("SCAN_MODE", "PRODUCT_MODE");
            startActivityForResult(i1, 0);
        } catch (ActivityNotFoundException anfe) {
            //Al obtenerlo, muestra diálogo de descarga
            showDialog(MainActivity.this, "No hay scanner", "Descargar scanner?", "Si", "No").show();
        }
    }
    public void scanQR(View v) {
        //Modo QR del producto
        try {
            //Inicia escaneo desde: com.google.zxing.client.android.SCAN
            Intent i2 = new Intent(ACTION_SCAN);
            i2.putExtra("SCAN_MODE", "QR_CODE_MODE");
            startActivityForResult(i2, 0);
        } catch (ActivityNotFoundException anfe) {
            //Al obtenerlo, muestra diálogo de descarga
            showDialog(MainActivity.this, "No hay scanner", "Descargar scanner?", "Si", "No").show();
        }
    }
    private static AlertDialog showDialog(final Activity a, CharSequence st, CharSequence mn,
CharSequence bnSi, CharSequence bnNo) {
        AlertDialog.Builder adb = new AlertDialog.Builder(a); //Alert para descarga
        adb.setTitle(st);
        adb.setMessage(mn);
        adb.setPositiveButton(bnSi, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface di, int i) {
                Uri uri = Uri.parse("market://search?q=pname:" + "com.google.zxing.client.android");
                Intent in = new Intent(Intent.ACTION_VIEW, uri);
                try {
                    a.startActivity(in);
                } catch (ActivityNotFoundException anfe) {
                }
            }
        });
        adb.setNegativeButton(bnNo, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface di, int i) {
            }
        });
        return adb.show();
    }
    public void onActivityResult(int requestCode, int resultCode, Intent i) {
        //Resultado
        if (requestCode == 0) {
            if (resultCode == RESULT_OK) {
                //Obtiene extras regresados por el intento
                String sr = i.getStringExtra("SCAN_RESULT");
                String sf = i.getStringExtra("SCAN_RESULT_FORMAT");
                Toast t = Toast.makeText(this, "Content:" + sr + " Format:" + sf, Toast.LENGTH_LONG);
                t.show();
            }
        }
    }
}

```

Paso 2. En la carpeta `res/layout`, abrir y modificar el archivo XML predeterminado `activity_main.xml`, con el siguiente código:

```

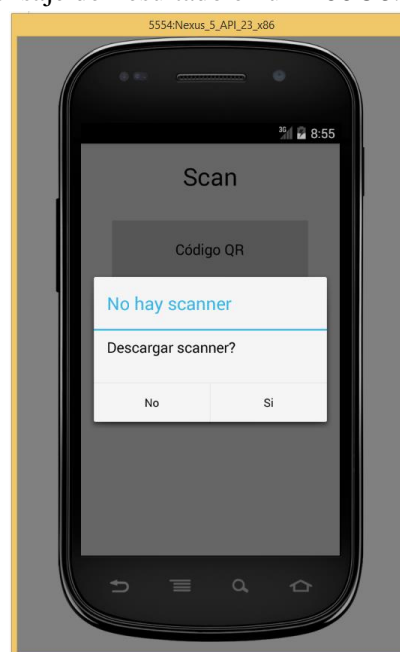
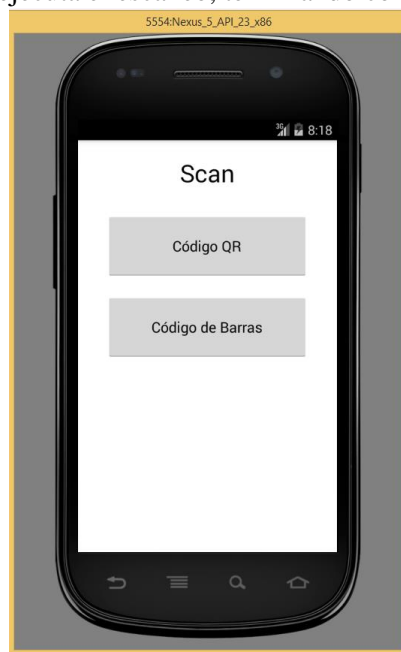
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ffffff"
    android:orientation="vertical"

```



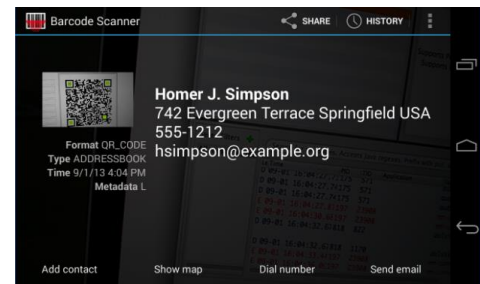
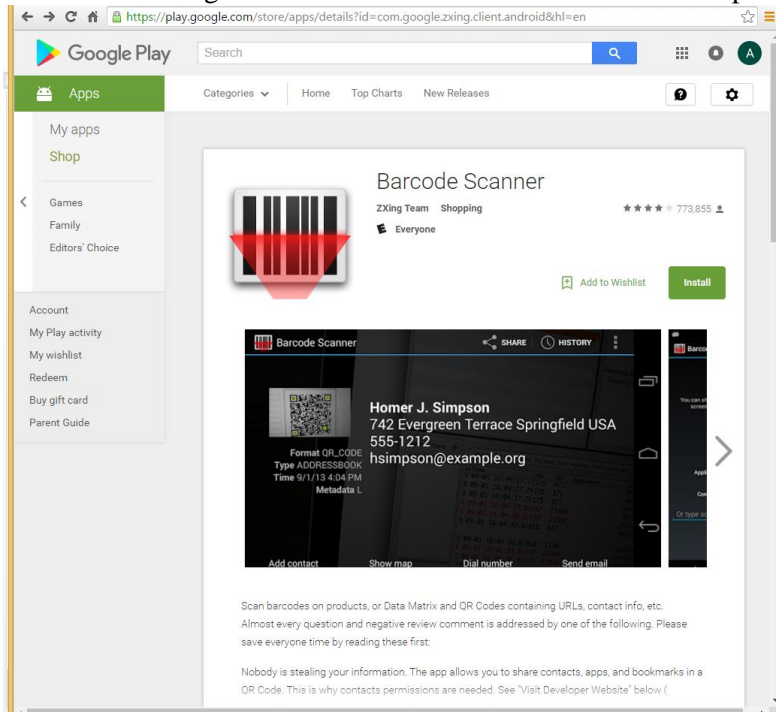
```
tools:context="com.javacodegeeks.androidstartactivityforresultexample.ActivityOne" >
<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center|top"
    android:layout_margin="20dp"
    android:text="Scan"
    android:textColor="#000000"
    android:textSize="30dp" />
<Button
    android:id="@+id/scanner"
    android:layout_width="250dp"
    android:layout_height="80dp"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:gravity="center"
    android:onClick="scanQR"
    android:text="Código QR"
    android:textSize="18dp" >
</Button>
<Button
    android:id="@+id/scanner2"
    android:layout_width="250dp"
    android:layout_height="80dp"
    android:layout_gravity="center"
    android:layout_margin="10dp"
    android:gravity="center"
    android:onClick="scanBarra"
    android:text="Código de Barras"
    android:textSize="18dp" >
</Button>
</LinearLayout>
```

Paso 3. Por último, ejecutar la aplicación. Digitar uno de los dos botones, según el tipo de código elegido. Si el móvil posee alguna actividad con escáner, se ejecutará y mostrará el resultado del escaneo. El móvil muestra el mensaje de descarga, se acepta y al terminar se ejecuta el escaneo, terminando con el mensaje del resultado en un `Toast`.





Si no se encuentra algún escáner, entonces se muestra la opción de descarga y, si se tiene acceso a Internet, se tendrá acceso al sitio de Google Store: <https://play.google.com/store/apps/details?id=com.google.zxing.client.android&hl=en>, para efectuar la descarga del Barcode Scanner en el móvil. En la pantalla del móvil acepta la opción de Instalar.



Al terminar la descarga, se inicia la aplicación, ejecutando el Intent, el cual inicia la Activity de escaneo. Tratar de enfocar clara y completamente la imagen del código dentro del marco mostrado por el escáner, como se indica en las siguientes figuras. Cuando el escáner detecta adecuadamente el código, se muestra la información contenida.

