Instituto Tecnológico de Costa Rica.

Escuela de Ingeniería en Computación.

Curso: Desarrollo de Aplicaciones Móviles

Profesora: Adriana Álvarez

Estudiantes: Pablo Baltodano, Víctor Saborío.

Laboratorio: Creación y lector de códigos QR.

Requisitos:

1. Dispositivo móvil con cámara.

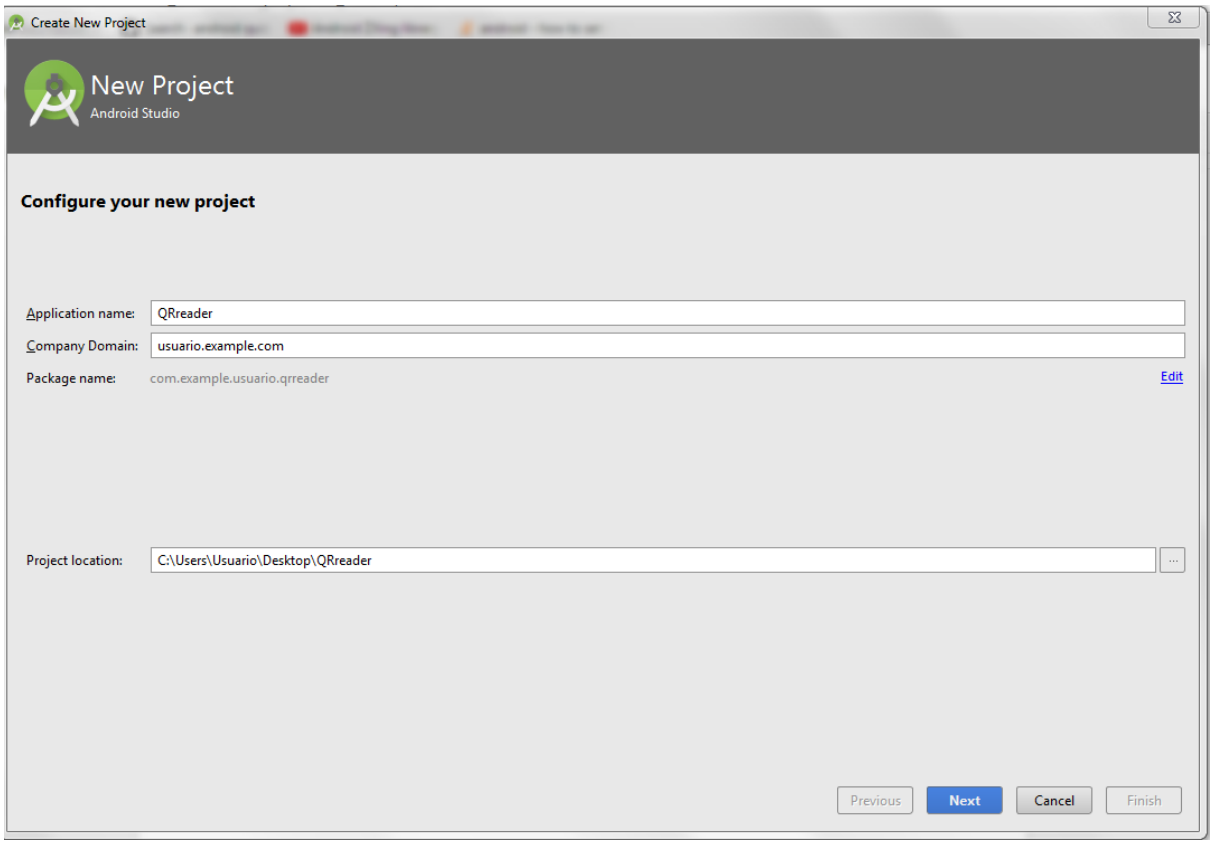
Link de referencia:

https://github.com/zxing/zxing

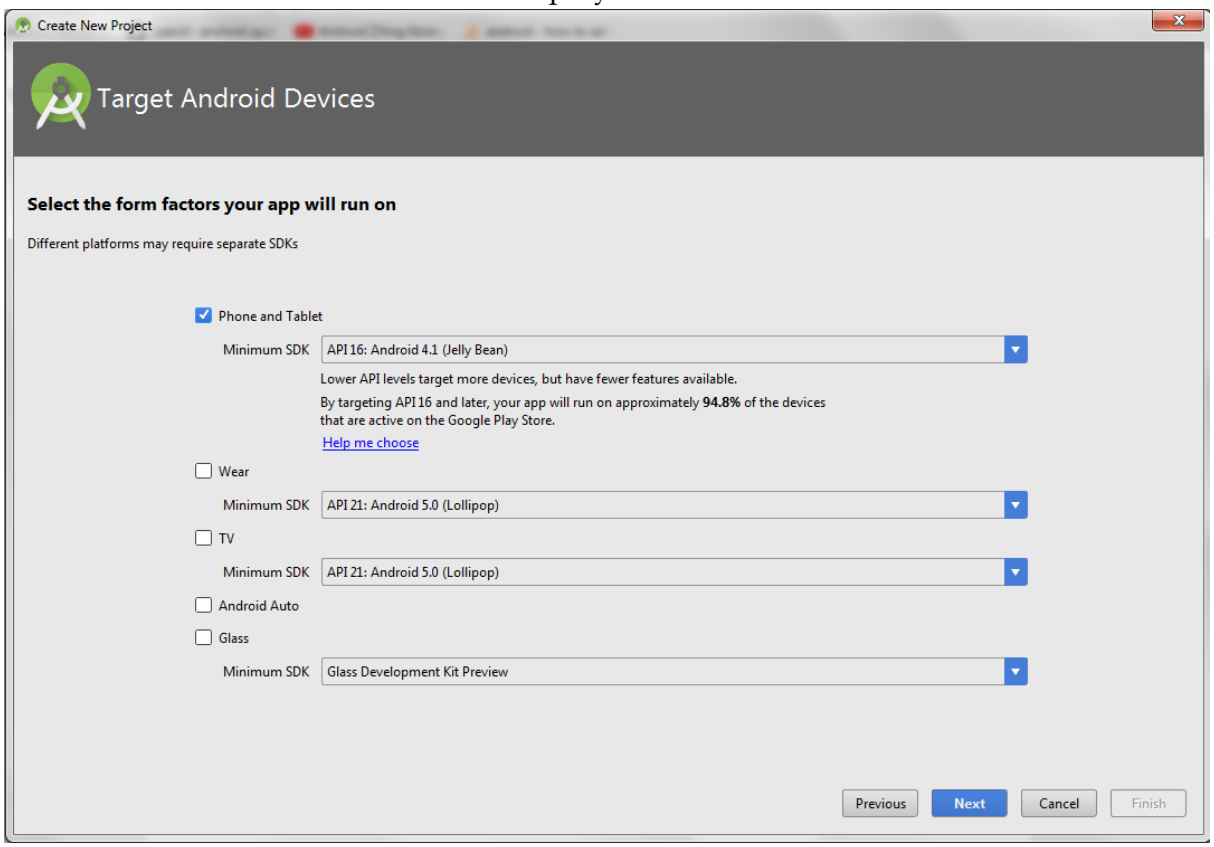
Nota: Este laboratorio estará compuesto por dos aplicaciones, una para crear el código QR y la otra que se usará como lector.

Primera Parte

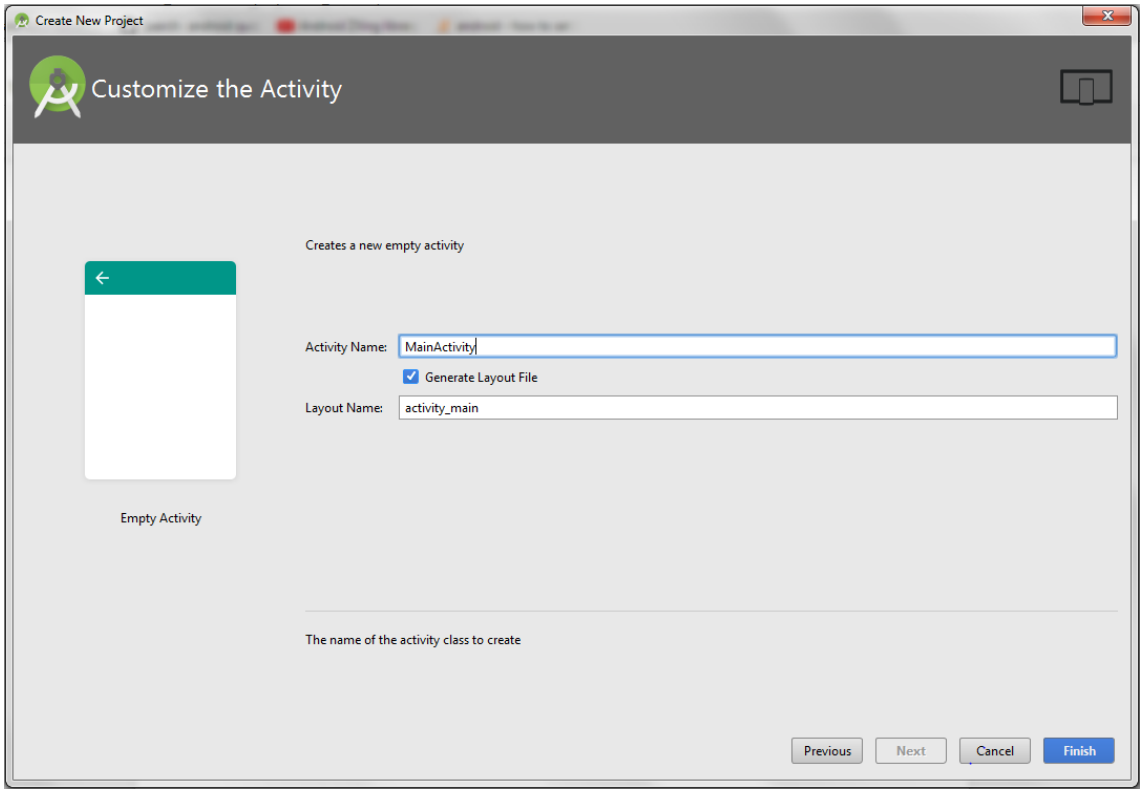
1. Primeramente, crearemos una aplicación que se encargue de leer código QR, la cual llamaremos QRreader



2. Para ello debemos crear un nuevo proyecto con min API 16.



3. Crearemos un Empty Activity



4. Llamaremos al activity principal MainActivity.

5. Para empezar es necesario agregar el siguiente código en el gradle (2 dependencias):

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
    compile 'com.google.zxing:core:3.2.1'
    compile 'com.journeyapps:zxing-android-embedded:3.2.0@aar'
}
```

6. Una vez hecho los pasos anteriores, debemos dirigirnos al activity_main en la carpeta Layout. Se agregará un botón y un textview como se muestra en el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.usuario.qrmaker.MainActivity"
    android:background="@color/abc_search_url_text_normal">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Escanear"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="169dp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Lector de código QR"
        android:id="@+id/textView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textColor="#081abc"
        android:textSize="35dp"
        android:textStyle="bold" />
</RelativeLayout>
```

7. Una vez que hayamos agregado la información anterior en el layout, debemos dirigirnos al MainActivity.java y copiar la siguiente información.

   - Paso 1: importar las bibliotecas necesarias:

```
package com.example.usuario.qrmaker;

import android.app.Activity;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

import com.google.zxing.integration.android.IntentIntegrator;
import com.google.zxing.integration.android.IntentResult;
```

   - Paso 2: Generación del código

```
public class MainActivity extends AppCompatActivity {

    private Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        button = (Button)this.findViewById(R.id.button);
        final Activity activity = this;
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                IntentIntegrator integrator = new IntentIntegrator(activity);
                integrator.setDesiredBarcodeFormats(IntentIntegrator.ALL_CODE_TYPES);
                integrator.setPrompt("scan");
                integrator.setCameraId(0);
                integrator.setBeepEnabled(false);
                integrator.setBarcodeImageEnabled(false);
                integrator.initiateScan();
            }
        });
    }
```

```
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        IntentResult result = IntentIntegrator.parseActivityResult(requestCode,resultCode,data);
        if (result != null) {
            if (result.getContents() == null){
                Log.d("MainActivity", "Cancelled scan");
                Toast.makeText(this, "Cancelled: ", Toast.LENGTH_LONG).show();
            } else {
                Log.d("MainActivity", "scanned");
                Toast.makeText(this, "scanned: "+ result.getContents(), Toast.LENGTH_LONG).show();
            }
        } else {
            super.onActivityResult(requestCode, resultCode, data);
        }
    }
}
```

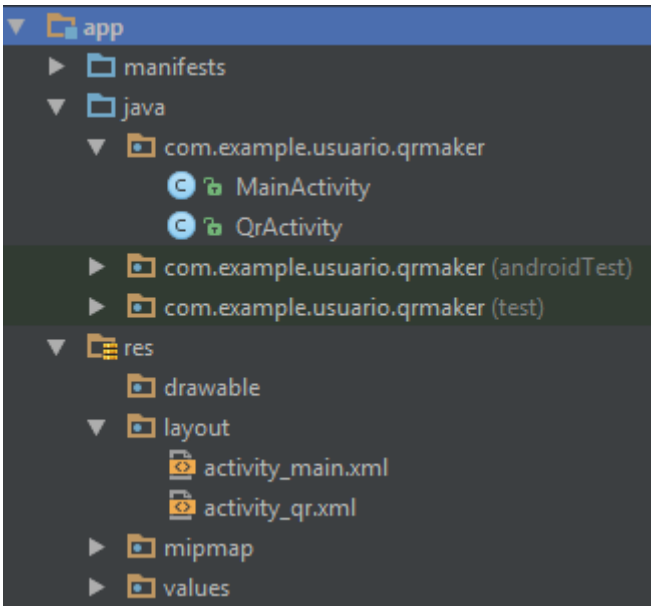
Al terminar de copiar el código anterior, terminamos de crear la primera parte del laboratorio que sería el lector de códigos QR.

Segunda Parte

1. Para iniciar se debe crear un nuevo proyecto igual a como se mostró con la aplicación anterior, a esta nueva aplicación le llamaremos QRrmaker.
2. Debemos de ingresar las mismas dependencias que se mostraron en la primera parte.

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    testCompile 'junit:junit:4.12'
    compile 'com.android.support:appcompat-v7:23.3.0'
    compile 'com.google.zxing:core:3.2.1'
    compile 'com.journeyapps:zxing-android-embedded:3.2.0@aar'
}
```

3. La siguiente será la estructura de la nueva aplicación (2 actividades):



4. Para la actividad QrActivity necesitaremos importar los siguiente:

```
import android.graphics.Bitmap;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.drawable.Drawable;
import android.os.Environment;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.File;
import java.io.FileOutputStream;
```

5. Luego debemos de escribir el siguiente código:

```
public class QrActivity extends AppCompatActivity {

    private ImageView imageView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_qr);
        Button button;
        imageView = (ImageView ) this.findViewById(R.id.imageView);
        Bitmap bitmap = getIntent().getParcelableExtra("pic");
        imageView.setImageBitmap(bitmap);
        button = (Button)  this.findViewById(R.id.button2);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                File root = Environment.getExternalStorageDirectory();
                File cachePath = new File(root, "QR.png");
                Drawable image = imageView.getDrawable();
                if(image != null && image instanceof BitmapDrawable) {
                    BitmapDrawable drawable = (BitmapDrawable) image;
                    Bitmap bitmap = drawable.getBitmap();
                    try {
                        FileOutputStream stream = new FileOutputStream(cachePath);
                        bitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);
                        stream.flush();
                        stream.close();
                        Toast.makeText(getApplicationContext(), "Se guardó el QR",
                                Toast.LENGTH_LONG).show();
                    } catch (Exception e) {
                        Toast.makeText(getApplicationContext(), "No se guardó el QR",
                                Toast.LENGTH_LONG).show();
                    }
                }
                }
        });
    }
}
```

6. Lo anterior es la clase que nos mostrará la imagen generada (código QR) del texto ingresado en el main_activity.
7. Además tendremos la opción de guardar el QR generado en nuestro celular, éste estaría almacenado en la ruta "root" del dispositivo.

8. El siguiente código corresponde al layout de la clase anterior:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.usuario.qrmaker.QrActivity">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/imageView"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Almacenar"
        android:id="@+id/button2"
        android:layout_alignParentBottom="true"
        android:layout_alignRight="@+id/imageView"
        android:layout_alignEnd="@+id/imageView" />
</RelativeLayout>
```

9. Luego de terminado los códigos anteriores, seguiremos con el main_activity.
10. Debemos importar las siguientes bibliotecas para el uso de ciertas funciones que se necesitan:

```
import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import com.google.zxing.BarcodeFormat;
import com.google.zxing.MultiFormatWriter;
import com.google.zxing.WriterException;
import com.google.zxing.common.BitMatrix;
import com.journeyapps.barcodescanner.BarcodeEncoder;
```

11. El siguiente código corresponde al main_activity, que sería donde se realiza toda la creación del bitmap que representará el código QR del texto que se ingresó en el layout que corresponde a esta clase:

```
public class MainActivity extends AppCompatActivity {

    private Button button;
    private EditText editText;
    private String textoVacio;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        final Context context = this;
        editText = (EditText) this.findViewById(R.id.editText);
        button = (Button)  this.findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                textoVacio = editText.getText().toString();
                if (textoVacio.matches("")) {
                    Toast.makeText(context, "Debe ingresar algún texto", Toast.LENGTH_SHORT).show();
                } else {
                    String text20r = editText.getText().toString();
                    MultiFormatWriter multiFormatWriter = new MultiFormatWriter();
                    try {
                        BitMatrix bitMatrix = multiFormatWriter.encode(text20r, BarcodeFormat.QR_CODE, 200, 200);
                        BarcodeEncoder barcodeEncoder = new BarcodeEncoder();
                        Bitmap bitmap = barcodeEncoder.createBitmap(bitMatrix);
                        Intent intent = new Intent(context, QrActivity.class);
                        intent.putExtra("pic", bitmap);
                        context.startActivity(intent);
                    } catch (WriterException e) {
                        e.printStackTrace();
                    }
                }
            }
        });
    }
}
```

12. Ya finalizando, el siguiente código corresponde al layout del main_activity:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.usuario.qrmaker.MainActivity">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Generar"
        android:id="@+id/button"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/editText"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
</RelativeLayout>
```

13. El último paso sería agregar este permiso, que se necesita para almacenar el código QR generado, dentro de nuestro dispositivo móvil.

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

Conteste las siguientes preguntas:

1. ¿Qué significan las siglas QR?
2. ¿Qué informaciones contiene un código QR? Mencione 3.
3. ¿Cuál es la librería utilizada en este proyecto para leer y crear códigos QR?
4. Mencione utilidades que generalmente se le da al código QR.
5. Mencione 2 campos donde son utilizados los QR y explique para qué son utilizados.
6. Investigue y explique las diferencias entre el código QR y un código de barras.
7. Genere un QR, publíquelo en el grupo de Facebook y pídale a algún otro compañero que escanee su código y escriba como comentario el significado de dicho QR.