



Instituto Tecnológico Superior de Jerez



Ingeniería en Sistemas Computacionales

Alumno: Marco Enrique Villegas Ulloa
No. Control: 14070094

Docente: Jorge Manjarrez

Programación Móvil

Tema 4

Actividad: Aplicación de uso de la Cámara

Fecha: 24/05/2020

Aplicación de uso de Cámara con intent implícito

Para el correcto funcionamiento de la aplicación y nos permita utilizar la cámara necesitamos permisos. Agregamos dos permisos en el manifest uno para el uso de la cámara y el otro para almacenaje interno.

```
<uses-feature android:name="android.hardware.camera" android:required="true" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

A continuación, agregamos esta etiqueta dentro de la etiqueta de Android el cual en forma general nos permitirá almacenar las fotos como un proveedor.

```
<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="${applicationId}.fileprovider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths" />
</provider>
```

Lo anterior nos marcara un error en 2 @xml/file_path porque no existe este archivo así creamos un xml o activity con el nombre de file_paths.

En el archivo file_paths agregamos lo siguiente.

```
package com.example.appcamara;

import androidx.core.content.FileProvider;

public class MyFileProvider extends FileProvider {
}
```

En MainActivity agregamos 5 variables de instancia dos constantes, un ImageButton, un ImageView y un String que guardara el path de la foto.

```
public class MainActivity extends AppCompatActivity {

    static final int REQUEST_IMAGE_CAPTURE = 1;
    static final int REQUEST_TAKE_PHOTO = 1;

    private ImageButton tomarCamara;
    private ImageView vistaPrevia;

    String currentPhotoPath;
```

En el onCreate inicializamos el ImageButton y el ImageView, agregamos los permisos para el usuario y el evento al tocar el ImageView.

```

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

    tomarCamara = (ImageButton) findViewById(R.id.btnFoto);
    vistaPrevia = (ImageView) findViewById(R.id.vistaPrevia);

    if (ContextCompat.checkSelfPermission(context: MainActivity.this, Manifest.permission.WRITE_EXTERNAL_STORAGE) != PackageManager.PERMISSION_GRANTED ||
        ActivityCompat.checkSelfPermission(context: MainActivity.this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(@dwtby: MainActivity.this, new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE, Manifest.permission.CAMERA}, requestCode: 1000);
    }

    tomarCamara.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            tomarFoto();
        }
    });
}

```

Creamos el método tomarFoto para que mediante un intent implícito llame la aplicación por defecto del dispositivo para tomar una foto. Al tomar la foto el if verifica que si hay una imagen y lo guarda dos copias uno en el directorio de la aplicación y la otra copia en la galería.

```

private void tomarFoto() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            Log.e(tag: "Error", ex.getMessage());
        }
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(context: this,
                authority: "com.example.android.fileprovider",
                photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent, REQUEST_TAKE_PHOTO);
        }
    }
}

```

Al activarse el método tomarFoto este método sobre escrito detecta información en el intent y obtiene el bitmap para colocarlo en el imageView.

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        vistaPrevia.setImageBitmap(imageBitmap);
    }
}

```

Este método es llamado por tomar foto para crear la imagen tomado por el usuario para darle nombre y formato.

```

private File createImageFile() throws IOException {
    String timeStamp = new SimpleDateFormat(pattern: "yyyyMMdd_HH:mm:ss").format(new Date());
    String imageFileName = "JPEG_" + timeStamp + "_";
    File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(imageFileName, suffix: ".jpg", storageDir);

    currentPhotoPath = image.getAbsolutePath();
    return image;
}

```

La interface de la aplicación

