

Evidencia Día 4 Semana 11

Tarea en clases: Crear método para guardar una foto capturada.

Siguiendo con la última etapa ejecutada de nuestra aplicación donde se realizó una captura de la cámara, vamos a crear el método para guardar la foto y verla como un archivo en el teléfono.

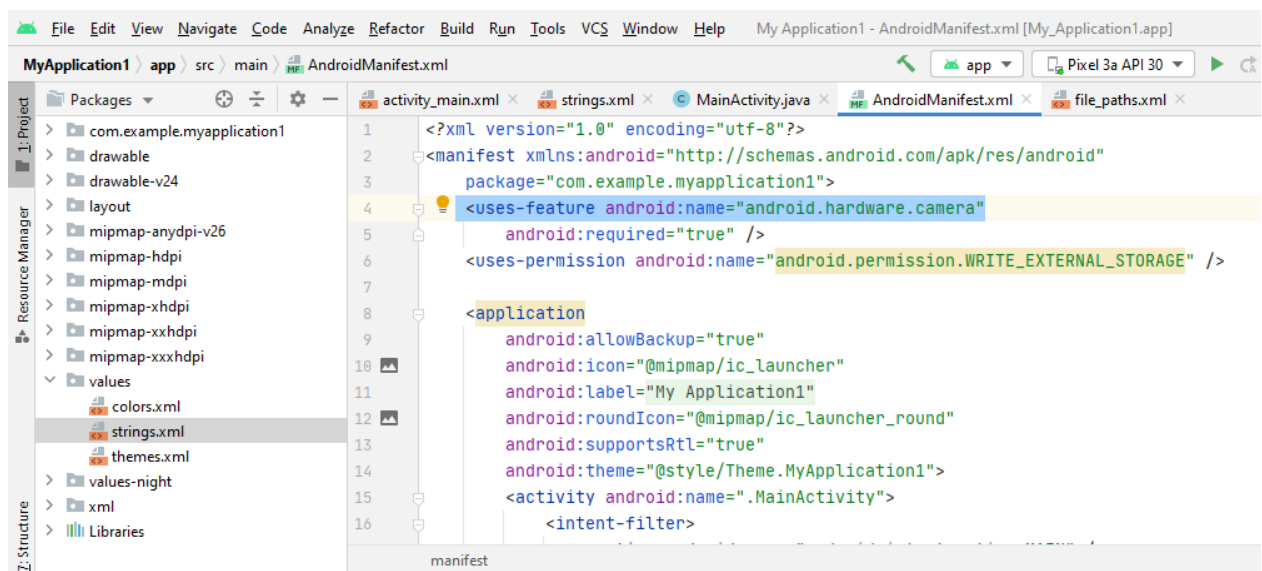
En general, las fotos que el usuario captura con la cámara del dispositivo deben guardarse en el almacenamiento externo público del dispositivo para que todas las apps puedan acceder a ellas. `getExternalStoragePublicDirectory()` proporciona el directorio adecuado para las fotos compartidas, con el argumento `DIRECTORY_PICTURES`.

Debido a que el directorio proporcionado por este método se comparte con todas las apps, para leer y escribir allí, se requieren los permisos `READ_EXTERNAL_STORAGE` y `WRITE_EXTERNAL_STORAGE`, respectivamente. El permiso de escritura permite de manera implícita la lectura, por lo que si se necesita escribir en el almacenamiento externo, se debe solicitar sólo un permiso:

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

...

</manifest>
```



Luego vamos al MainActivity y posterior al último método visto en clase para crear el archivo (Bitmap); creamos un nuevo método para guardar el archivo:

```

public void saveTempBitmap(Bitmap bitmap) {

    if (isExternalStorageWritable()) {
        saveImage(bitmap);
        Toast.makeText(this, "Archivo guardado", Toast.LENGTH_LONG).show();

    }

    //prompt the user or do something

}

public void saveImage(Bitmap finalBitmap) {

    String root = Environment.getExternalStorageDirectory().toString();
    File myDir = new File(root + "/Download"); //ubicación del archivo
    String timeStamp = new SimpleDateFormat("yyyyMMdd_HH:mm:ss").format(new
Date());
    String fname = "Shutta_" + timeStamp + ".jpg";

    File file = new File(myDir, fname);
    if (file.exists()) file.delete ();
    try {
        FileOutputStream out = new FileOutputStream(file);
        finalBitmap.compress(Bitmap.CompressFormat.JPEG, 100, out);
        out.flush();
        out.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}

/* Checks if external storage is available for read and write */

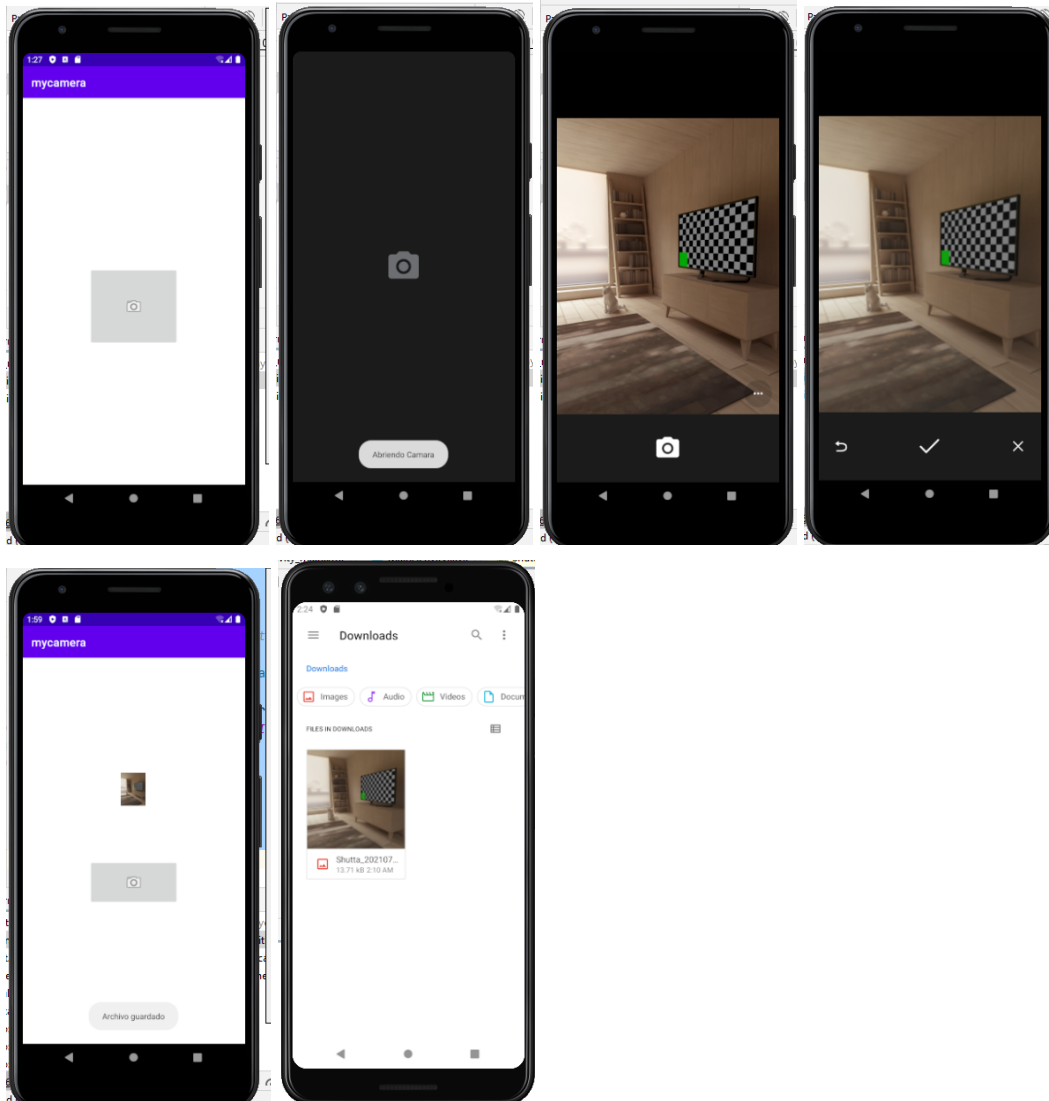
public boolean isExternalStorageWritable() {

    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

}

```

Ejecutamos el emulador:



Reflexión: La función guardar foto, fue más complicada para encontrar información al respecto, por lo que un compañero tuvo que prestarme apoyo en el código.