Desarrollo de aplicaciones móviles I

Tema Nº10:Accediendo al Hardware en Android

Indicador de logro Nº10:Accede a componentes de Hardware del Dispositivo Android mediante la implementación de una aplicación que permite guardar imágenes tomadas con la cámara del dispositivo.

**TEMA 01 Teoría de los**

Imagen que contiene Icono

Descripción generada automáticamente

**TEMA Nº10:**

Accediendo al Hardware en Android

**Subtema 10.1:**

Accediendo a la cámara en Android

**Ejemplos:**

El código java asociado a esta vista se encuentra en el archivo "MainActivity.java" y es el siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Definimos las variables et1 e imagen1:

private ImageView imagen1;

private EditText et1;

En el método onCreate obtenemos la referencia de los objetos definidos en el archivo XML:

imagen1=(ImageView)findViewById(R.id.imageView);

et1=(EditText)findViewById(R.id.editText);

Cuando se presiona el botón para tomar la foto creamos un objeto de la clase Intent y pasamos como dato la constante ACTION\_IMAGE\_CAPTURE definida en la clase MediaStore:

Intent intento1 = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

Creamos un objeto de la clase File indicando en el primer parámetro el path donde se creará el archivo y lo obtenemos llamando al método getExternalFilesDir(null), y como segundo parámetro indicamos el nombre de archivo donde se guardará la foto:

File foto = new File(getExternalFilesDir(null), et1.getText().toString());

Pasamos al otro Activity la referencia del archivo que debe crear:

intento1.putExtra(MediaStore.EXTRA\_OUTPUT, Uri.fromFile(foto));

Finalmente lanzamos el Activity que permite tomar la foto:

startActivity(intento1);

Para recuperar una foto almacenada en la memoria externa procedemos a crear un objeto de la clase Bitmap llamando al método estático decodeFile de la clase BitmapFactory. Debemos pasar como referencia el path donde se encuentra el archivo jpg a recuperar:

Bitmap bitmap1 = BitmapFactory.decodeFile(getExternalFilesDir(null)+"/"+et1.getText().toString());

Para visualizar la imagen en el ImageView llamamos al método setImageBitmap:

imagen1.setImageBitmap(bitmap1);

Cuando se presiona el botón ver procedemos a crear un Activity que seguidamente procederemos a crear:

Intent intento1=new Intent(this,Actividad2.class);

startActivity(intento1);

**Subtema 10.2:**

Creando un layout para mostrar una foto en Android

**Ejemplos:**

Texto

Descripción generada automáticamente

Es decir agregamos la línea:

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

La interfaz visual de la ventana principal es:

Pantalla de un celular con la imagen de un video juego

Descripción generada automáticamente con confianza media

Disponemos un EditText, 3 botones y un objeto de la clase ImageView. El archivo "activity\_main.xml" que implementa la interfaz visual es:

Texto

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

La imagen inicial del ImageView la definimos en la propiedad src y disponemos el recurso por defecto propuesto en el proyecto (luego en tiempo de ejecución la modificamos con las imágenes de las fotos que sacamos):

android:src="@mipmap/ic\_launcher" />

Los tres botones tienen cargadas la propiedad onClick con el método respectivo:

android:onClick="tomarFoto" />

android:onClick="recuperarFoto" />

y

android:onClick="ver" />

**Subtema 10.3:**

Accediendo y mostrando fotos de la cámara en Android **Ejemplos:**

Ahora procederemos a crear la segunda actividad. Para esto presionamos el botón derecho del mouse sobre el nombre del paquete de nuestro proyecto y en el menú que aparece seleccionamos New -> Activity -> Empty Activity:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En el diálogo que aparece procedemos a cargar en "Activity Name" el valor "Actividad2":

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego de presionar "Finish" ya tenemos los dos archivos : XML y java de nuestra aplicación.

En el archivo activity\_actividad2.xml procedemos a crear la siguiente interfaz:

Pantalla de celular con imagen de la pantalla de un celular

Descripción generada automáticamente con confianza media

El contenido para crear esta pantalla está en el archivo activity\_actividad2.xml:

Texto, Carta

Descripción generada automáticamente

Texto, Carta

Descripción generada automáticamente

El archivo java asociado a esta actividad se llama "Actividad2.java" y su contenido es:

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Definimos cuatro variables:

private ListView lv1;

private ImageView iv1;

private String[] archivos;

private ArrayAdapter<String> adaptador1;

Recuperamos la lista de todos los archivos almacenados en la memoria externa y los guardamos en un vector:

File dir=getExternalFilesDir(null);

archivos=dir.list();

Creamos el ArrayAdapter con todos los nombres de archivos y los visualizamos en el ListView:

adaptador1=new ArrayAdapter<String>(this,android.R.layout.simple\_list\_item\_1,archivos);

lv1=(ListView)findViewById(R.id.listView1);

lv1.setAdapter(adaptador1);

Mostramos en un ImageView el contenido de la foto que seleccionamos del ListView:

public void onItemClick(AdapterView<?> arg0, View arg1, int arg2,

long arg3) {

Bitmap bitmap1 = BitmapFactory.decodeFile(getExternalFilesDir(null)+"/"+archivos[arg2]);

iv1.setImageBitmap(bitmap1);

}});

Otra parte muy importante para probar esta aplicación con el emulador de Android es su correcta configuración. Entremos a la opción AVD Manager:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En este diálogo entremos a modificar la configuración del emulador que estemos utilizando:

Tabla

Descripción generada automáticamente

Y activamos la cámara:

Interfaz de usuario gráfica

Descripción generada automáticamente

Y podemos proceder a ejectra la aplicación y probaos la funcionalidad implementada

**Actividad:**

1. CUESTIONARIO TÉCNICO

* Indique su opinión respecto al desarrollo de aplicaciones móviles híbridas y el impacto en la performance de las mismas al acceder al hardware de los dispositvos
* Detalle por qué utilizamos Intents al implementar la funcionalidad de acceder a la cámara del dispositivo móvil
* ¿En qué archivo configuramos los permisos para guardar archivos en la memoria externa?
* ¿Es correcto indicar que la forma más sencilla de utilizar la cámara en un dispositivo móvil Android es la que ofrece por defecto el sistema operativo o necesitamos una librería de terceros? Desarrolle su respuesta
* ¿Cuál es la utilidad del método estático decodeFile en el contexto de la imlementación de una aplicación que permita guardar imágenes tomadas con la cámra en la memoria externa?

1. CONCLUSIONES DE LA EXPERIENCIA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_