# UNIDAD 5: ContentResolver Imágenes

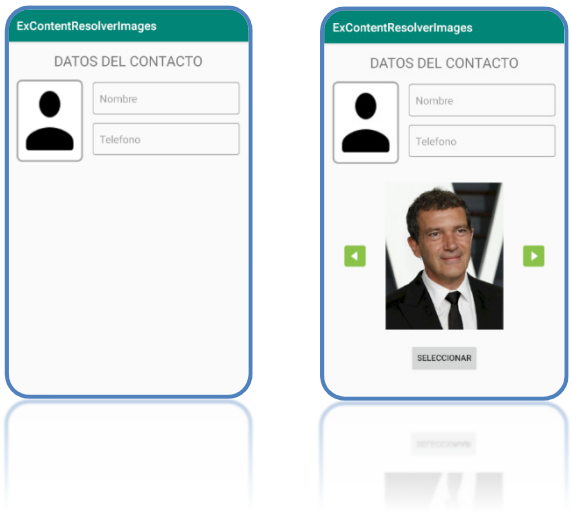
Uno de los recursos más utilizados en las aplicaciones son las imágenes.

Existen diversos métodos para realizar la selección de una imagen, pero todos se basan en la utilización de un *ContentResolver* que acceda a los datos del *ContentProvider* suministrado por Android, cuya *Uri* es la siguiente:

```
MediaStore.Images.Media.EXTERNAL_CONTENT_URI
```

Para probar el funcionamiento de este ContentResolver vamos a crear una aplicación con una única actividad en la que se mostrará un formulario para introducir los datos de un contacto que incluirá un *ImageView* para la foto de dicho contacto.

Al abrir la aplicación el *ImageView* contendrá una imagen genérica, y al hacer click sobre ella se cargarán las imágenes disponibles en nuestra galeria y se mostrarán los controles adecuados para seleccionar una de las imágenes en la parte inferior de la pantalla.



## CREACIÓN DE LA INTERFAZ

Como se ha señalado, la aplicación sólo va a contar con una actividad y por tanto tendremos que crear un único *layout*, que estará dividido en dos partes.

- Parte superior: formulario para introducir los datos del contacto. Como se han introducido componentes de *Material Design*, habrá que añadir la dependencia correspondiente en *Gradle*.
- Parte inferior: controles para seleccionar una imagen como foto de contacto. Al iniciar la aplicación estará oculta y se mostrará al hacer *click* en la foto del contacto.

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="16dp"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/etiTitulo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="DATOS DEL CONTACTO"
        android:textSize="24sp" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="16dp">

        <ImageView
            android:id="@+id/imgFoto"
            android:layout_width="0dp"
            android:layout_height="140dp"
            android:adjustViewBounds="true"
            android:scaleType="fitXY"
            android:layout_weight="1"
            android:src="@drawable/icon_user"/>

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:orientation="vertical"
            android:layout_weight="2"
            android:layout_marginLeft="16dp">

            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/layoutNombre"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">
                <com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/editNombre"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:hint="Nombre"/>
            </com.google.android.material.textfield.TextInputLayout>
```

```
            <com.google.android.material.textfield.TextInputLayout
                android:id="@+id/layoutTelefono"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_marginTop="8dp"
                style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">

                <com.google.android.material.textfield.TextInputEditText
                    android:id="@+id/editTelefono"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:hint="Telefono"
                    android:inputType="phone" />
            </com.google.android.material.textfield.TextInputLayout>

        </LinearLayout>

    </LinearLayout>

    <LinearLayout
        android:id="@+id/layoutImg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:visibility="invisible"
        android:layout_marginTop="32dp">

        <ImageButton
            android:id="@+id/botPrev"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:layout_gravity="center_vertical"
            android:src="@drawable/icon_prev"
            android:background="@android:color/transparent" />

        <ImageView
            android:id="@+id/imgPrev"
            android:layout_width="match_parent"
            android:layout_height="250dp"
            android:layout_weight="1"/>

        <ImageButton
            android:id="@+id/botSig"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:layout_gravity="center_vertical"
            android:src="@drawable/icon_sig"
            android:background="@android:color/transparent"/>

    </LinearLayout>

    <Button
        android:id="@+id/botSeleccionar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="24dp"
        android:visibility="invisible"
        android:text="Seleccionar"/>

</LinearLayout>
```

## MANIFIESTO

Para poder acceder a las imágenes necesitamos el permiso "READ_EXTERNAL_STORAGE" así que no deberemos olvidar declararlo en nuestro manifiesto:

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

El archivo de manifiesto deberá ser similar al siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.excontentresolverimages">

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

## CLASE: MAINACTIVITY

En la clase *MainActivity* declararemos los objetos relativos a la vista que vamos a manejar dentro de la aplicación (*Button*, *ImageView*, etc), las variables necesarias para almacenar las imágenes (*ArrayList* y posición), y una constante que identificará la solicitud de permiso.

```
public class MainActivity extends AppCompatActivity {
  protected static final int REQUEST_READ_EXTERNAL_STORAGE = 101;
  protected LinearLayout layoutImg;
  protected ImageView imgFoto, imgPrev;
  protected ImageButton botSig, botPrev;
  protected Button botSeleccionar;
  protected ArrayList<String> fotos;
  protected int pos;
  ...
}
```

- ***onCreate***

En primer lugar, obtendremos las referencias de los objetos de la vista e iniciaremos las variables para el manejo de la lista de imágenes:

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

  // obtenemos las referencias d elos objetos
  layoutImg = (LinearLayout)findViewById(R.id.layoutImg);
  imgPrev = (ImageView)findViewById(R.id.imgPrev);
  imgFoto = (ImageView)findViewById(R.id.imgFoto);
  botSeleccionar = (Button)findViewById(R.id.botSeleccionar);
  botSig = (ImageButton)findViewById(R.id.botSig);
  botPrev = (ImageButton)findViewById(R.id.botPrev);


  // Iniciamos las variables para el manejo del array
  fotos = null;
  pos = 0;
  ...
}
```

Y a continuación establecemos el comportamiento de los controles de la aplicación:

- ***ImageFoto***: al hacer *click* en este *ImageView* se desplegará la parte del formulario que permite seleccionar una imagen de nuestra galería.

  Tendremos que comprobar si disponemos del permiso de lectura de archivos externos y de ser así llamaremos al método obtenerImagenes, que se encargará de almacenar la ruta de todas las imágenes disponibles en nuestra galería.

  En el terminal virtual vamos a trabajar con un número reducido de imágenes y la carga será rápida, pero si el número de imágenes es muy elevado habría que implementar algún mecanismo que realice la tarea en segundo plano como un cargador *CursorLoader*.

  NOTA: se ha añadido una comprobación inicial para evitar repetir la carga de las imágenes en caso de que ya se hubieran cargado previamente. Hay que tener en cuenta que de esta forma no se cargarán las imágenes que se hubieran añadido posteriormente, así que si existe esa posibilidad deberíamos eliminar dicha comprobación.

```
imgFoto.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    if (fotos == null) {
      // Comprobamos el permiso READ_EXTERNAL_STORAGE y obtenemos las imagenes
      if ((android.os.Build.VERSION.SDK_INT>=android.os.Build.VERSION_CODES.M)
      && ContextCompat.checkSelfPermission(MainActivity.this,
      Manifest.permission.READ_EXTERNAL_STORAGE)
      != PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(MainActivity.this,
        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
        REQUEST_READ_EXTERNAL_STORAGE);
      } else {
        obtenerImagenes();
      }
    }

    if (fotos != null) {
      layoutImg.setVisibility(View.VISIBLE);
      botSeleccionar.setVisibility(View.VISIBLE);
      mostrarImagen(imgPrev, pos);
    }
  }
});
```

- ***botSig***: selecciona la siguiente imagen de nuestro *ArrayList* y la muestra en la vista previa.
  Para ello basta con incrementar la variable "pos" que indica la imagen seleccionada, y si hemos alcanzado el final de la lista volvemos al principio.
  Una vez seleccionada la imagen la mostramos con ayuda del método "mostrarImagen" que recibe como parámetros el *ImageView* donde queremos mostrar la imagen y la posición de la imagen que queremos mostrar.

```
botSig.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    pos++;
    if (pos == fotos.size()) {
      pos = 0;
    }
    mostrarImagen(imgPrev, pos);
  }
});
```

- ***botPrev***: es análogo al método anterior pero seleccionando la imagen anterior.

```
botPrev.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    pos--;
    if (pos < 0) {
      pos = fotos.size()-1;
    }
    mostrarImagen(imgPrev, pos);
  }
});
```

- ***botSeleccionar***: permite seleccionar la imagen que se está mostrando en la vista previa como foto del contacto.

```
botSeleccionar.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View v) {
    mostrarImagen(imgFoto,pos);
    layoutImg.setVisibility(View.INVISIBLE);
    botSeleccionar.setVisibility(View.INVISIBLE);
  }
});
```

Hecho esto ya tenemos finalizado el método *onCreate*.

- ***onRequestPermissionsResult***

Se encarga de gestionar el resultado obtenido tras la solicitud del permiso.

```
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
  super.onRequestPermissionsResult(requestCode, permissions, grantResults);
  switch (requestCode) {
    case REQUEST_READ_EXTERNAL_STORAGE:
      if (grantResults.length>0 && grantResults[0]==PackageManager.PERMISSION_GRANTED)
      {
        obtenerImagenes();
      } else {
        int dura = Toast.LENGTH_SHORT;
        String mensa = "No dispone de permisos para acceder a las imagenes";
        Toast.makeText(getApplicationContext(),mensa,dura).show();
      }
      return;
  }
}
```

- ***obtenerImagenes()***

Por medio del ContentResolver, ejecutamos la consulta sobre la URI asociada a las imágenes (MediaStore.Images.Media.***EXTERNAL_CONTENT_URI***).

A continuación, comprobamos que la consulta haya obtenido algún resultado y si es así recorremos el cursor quedándonos con el primer elemento de cada registro (ruta de la imagen) y la almacenamos en nuestro *ArrayList* fotos.

```
public void obtenerImagenes() {
  fotos = new ArrayList<String>();
  ContentResolver cr = getContentResolver();
  String[] projection = {MediaStore.Images.Media.DATA};
  Cursor curImg = cr.query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI, projection,
  null, null, null);
  if (curImg!=null) {
    while (curImg.moveToNext()) {
      String rutaImg = curImg.getString(0);
      fotos.add(rutaImg);
    }
    curImg.close();
  } else {
    String mensa = "No hay imagenes";
    Toast.makeText(MainActivity.this, mensa, Toast.LENGTH_SHORT).show();
  }
}
```

- ***mostrarImagen (ImageView marco, int pos)***

Este método recibe como parámetros el *ImageView* donde queremos mostrar la imagen y la posición de la imagen a mostrar, de manera que obtendremos la ruta de la imagen y la mostraremos en el *ImageView* por medio del método *setImageBitmap*.

El método *setImageBitmap* espera recibir un objeto *Bitmap* pero lo que tenemos es la ruta de la imagen, así que deberemos utilizar el método *decodeFile* de la clase *BitmapFactory* para obtener el *Bitmap* a partir de la ruta de la imagen.

```
public void mostrarImagen (ImageView marco, int pos) {
    marco.setImageBitmap(BitmapFactory.decodeFile(fotos.get(pos)));
}
```

## PRUEBAS

Para poder probar la aplicación desde el dispositivo virtual deberemos disponer de imágenes almacenadas. Puedes utilizar el navegador Chrome del propio dispositivo virtual para descargar algunas imágenes y/o tomar fotografías con ayuda de la cámara virtual.

## MEJORAS

Investiga el funcionamiento del *Intent ACTION_PICK* para mejorar la selección de la imagen.