

UNIDAD 3: Trabajar con imágenes

Android trata las imágenes como un recurso de tipo “*drawable*”, por ello cuando queramos incorporar imágenes a nuestra interfaz en primer lugar tendremos que guardar dichas imágenes dentro de la carpeta de recursos de nuestro proyecto: “app/src/main/res/drawable”.

Si sabemos la imagen concreta que vamos a utilizar podemos acceder a ella a través de la clase R como a cualquier otro recurso:

```
R.drawable.nombre_imagen
```

Pero si por el contrario, no se conoce la imagen a obtener hasta el momento de la ejecución tendremos que obtener su id a partir de su nombre utilizando el método *getIdentifier*:

```
Context c = getApplicationContext();  
String imgName = "image01"; // Nombre sin extensión  
String paquete = c.getPackageName();  
int id = c.getResources().getIdentifier(imgName, "drawable", paquete);
```

También es posible llamar al método *getIdentifier* concatenando los tres valores (paquete, tipo, nombre) en un único *String* y dejar como *null* el resto de parámetros:

```
String recurso = c.getPackageName() + ":drawable/" + imgName;  
int id = c.getResources().getIdentifier(recurso, null, null);
```

En el caso de tener varias imágenes que queramos utilizar podemos implementar varios métodos que nos permitan localizar las imágenes y guardar sus “id” para utilizarlos cuando sea necesario.

MÉTODO 1: NOMBRE COMÚN SEGUIDO DE UN ÍNDICE

En primer lugar, renombraremos todas las imágenes que queramos utilizar utilizando un nombre común, por ejemplo “imagen” seguida de un índice consecutivo: “imagen1”, “imagen2”, “imagen3”, etc.

Una vez renombradas las imágenes, dentro de un bucle intentaremos obtener el “id” a partir de los nombres de cada una de esas imágenes “imagen1”, “imagen2”, etc. almacenando los “id” obtenidos en un *ArrayList* hasta que dicho que “id” sea 0, lo que significará que no hay una imagen con ese nombre y por tanto hemos terminado.

```
public class MainActivity extends Activity {

    protected int contador;
    protected ArrayList<Integer> imagenes;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        contador = 0;
        imagenes = obtener_ids();

        Button btn = (Button) findViewById(R.id.botNext);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                contador++;
                int numImg = imagenes.size();
                int id = imagenes.get(contador % numImg);
                ImageView im = (ImageView) findViewById(R.id.imagen);
                im.setImageResource(id);
            }
        });
    }

    protected ArrayList obtener_ids() {
        ArrayList<Integer> ids = new ArrayList<Integer>();
        Context c = getApplicationContext();
        String paquete = c.getPackageName();
        String imageName;
        int id;
        int i = 0;
        do {
            imageName = paquete + "drawable/imagen" + i;
            id = c.getResources().getIdentifier(imageName, null, null);
            if (id != 0) {
                ids.add(id);
                i++;
            }
        } while (id != 0);
        return ids;
    }
}
```

MÉTODO 2: OBTENER TODOS LOS RECURSOS DE TIPO DRAWABLE Y DISCRIMINAR LOS QUE QUEREMOS UTILIZAR POR SU NOMBRE

En este caso no es necesario que todas las imágenes tengan el mismo nombre seguido de un índice, pero sí que haya algo en su nombre que permita diferenciarlas de otros recursos “*drawable*” que no se quieran utilizar como imágenes a mostrar. Para este ejemplo que ha añadido a los nombres las imágenes la partícula “img_” antes del propio nombre.

En este caso nos apoyaremos en una clase “*Reflection*”, en concreto “*Field*” que nos permitirá acceder a la clase que queramos “*R.drawable.class*” y obtener todos sus campos por medio del método *getFields()*. A partir de ahí recorreremos el array obtenido y si empieza por la raíz que identifica a nuestras imágenes almacenaremos su “id” en nuestro *ArrayList*.

```
public class MainActivity extends Activity {

    protected int contador;
    ArrayList<Integer> imagenes;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        contador = 0;
        imagenes = obtenerIDs(R.drawable.class, "img_");

        Button btn = (Button) findViewById(R.id.botNext);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                contador++;
                int numImg = imagenes.size();
                int id = imagenes.get(contador % numImg);
                ImageView im = (ImageView) findViewById(R.id.imagen);
                im.setImageResource(id);
            }
        });
    }
}
```

```
// Este método devuelve un ArrayList con los id de todos los elementos
// de un tipo de recurso (string, drawable, etc) cuyo nombre empiece
// por la cadena pasada como parámetro

protected ArrayList obtenerIDs(Class<?> aClass, String inicio)
    throws IllegalArgumentException{
    ArrayList<Integer> ids = new ArrayList<Integer>();

    // Obtenemos los miembros de la clase pasada como argumento
    Field[] elementos = aClass.getFields();

    // Recorremos el array y nos quedamos con aquellos que
    // empiecen por la cadena pasada como argumento ("img_")
    for(int i = 0; i < elementos.length; i++){

        // Tratamos la excepción que se produce si el tipo recibido
        // como argumento no es válido (no es una clase)
        try {
            if (elementos[i].getName().startsWith(inicio)) {
                ids.add(elementos[i].getInt(null));
            }
        } catch (Exception e) {
            throw new IllegalArgumentException();
        }
    }
    return ids;
}
```