



Tarea 11. Las Clases Notification y NotificationCompat

CONCEPTOS

Las notificaciones son avisos de eventos importantes que ocurren en segundo plano. Las notificaciones se visualizan en el área de notificaciones en forma de icono descriptivo. Para ver la notificación se arrastra la barra de notificaciones hacia abajo y enseguida se muestran las notificaciones acumuladas hasta el momento.

Las notificaciones tienen cuatro partes fundamentales: El título, un contenido, el ícono y el tiempo de ocurrencia. Se pueden añadir más elementos, por ejemplo iconos combinados, la cantidad de notificaciones, botones, y otros.

Las notificaciones de aviso aparecen con prioridad alta o porque se definió que la notificación se proyecte de esta forma. Las llamadas entrantes o alarmas son eventos que producen notificaciones de aviso.

Las notificaciones utilizan la clase `Notification`, pero para tener compatibilidad con versiones anteriores se usa `NotificationCompat`. La clase `PendingIntent` contiene los datos de un `Intent` y la actividad que iniciará al presionarse la notificación. Las notificaciones pueden eliminarse desde la interfaz arrastrándola hacia el **drawer**. También, se puede digitar el botón para la eliminación de todas las notificaciones. Las notificaciones pueden expandir su contenido si se aplica el estilo `BigTextStyle`, que permite ampliar el detalle del texto del contenido. Las notificaciones permiten visualizar progresos de operaciones en curso relacionadas a otras aplicaciones. Para crear una notificación de aviso se utiliza el método `setFullScreenIntent()`.

Las notificaciones se muestran en la pantalla de bloqueo si el usuario así lo decide en la configuración de las preferencias de la plataforma. Las siguientes constantes definen si las notificaciones se muestran en la pantalla de bloqueo:

<code>VISIBILITY_PUBLIC</code>	Muestra todo el contenido de la notificación en la pantalla de bloqueo.
<code>VISIBILITY_PRIVATE</code>	Muestra el icono, el título y detalles mínimos de la notificación.
<code>VISIBILITY_SECRET</code>	No muestra ninguna información de la notificación.

La jerarquía de la clase `Notification`:

```
public class Notification extends Object implements Parcelable
java.lang.Object -> android.app.Notification
```

Esta clase representa la forma en que una notificación persistente se presenta al usuario utilizando el `NotificationManager`. Además, se ha agregado la clase `Notification.Builder` para facilitar la construcción de la notificación.

Métodos públicos.

Notification	<code>clone()</code>	Crea y regresa una copia de este objeto
int	<code>describeContents()</code>	Describe de objetos contenidas en la representación referenciada de la instancia <code>Parcelable</code>
String	<code>getGroup()</code>	Obtiene la clave para agrupar esta notificación en un grupo o pila con otras notificaciones en los dispositivos que soporten tal prestación
Icon	<code>getLargeIcon()</code>	El icono grande que se muestra en la vista del contenido de la presente notificación
Icon	<code>getSmallIcon()</code>	El icono pequeño que representa esta notificación en la barra de estado y el contenido.
String	<code>getSortKey()</code>	Obtiene la clave de clasificación que ordena la presente notificación entre otras notificaciones del mismo paquete.
String	<code>toString()</code>	Regresa la cadena que representa a este objeto.
void	<code>writeToParcel(Parcel p, int flags)</code>	Empaqueta la notificación en una clase <code>Parcel</code> .



DESARROLLO

EJEMPLO 1.

En el siguiente ejemplo se emplea la clase `NotificationCompat` para construir la notificación.

Paso 1. Crear un nuevo proyecto **Notificaciones**. En la carpeta `java/com.example.mipaquete`, abrir y modificar el archivo `MainActivity.java` con el siguiente código:

```
import android.os.Bundle;
import android.app.*;
import android.content.*;
import android.graphics.drawable.BitmapDrawable;
import android.support.v4.app.NotificationCompat;
import android.view.*;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends Activity {
    int t=200, i=0;
    boolean c=true;
    TextView jtv;
    Button jbnN;
    private static final int NOTIF_ALERTA_ID = 1;
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        jtv = (TextView) findViewById(R.id.xtv);
        jbnN = (Button) findViewById(R.id.xbnN);
        jbnN.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                NotificationCompat.Builder ncb =
                    new NotificationCompat.Builder(MainActivity.this)
                        .setSmallIcon(android.R.drawable.stat_sys_warning)
                        .setLargeIcon(((BitmapDrawable) getResources()
                            .getDrawable(R.mipmap.ic_launcher)).getBitmap())
                        .setContentTitle("Alerta de Notificación")
                        .setContentText("Uso de la notificación." + "i=" + ++i)
                        .setContentInfo("Un valor")
                        .setTicker("Mensaje de Alerta!");
                Intent in = new Intent(MainActivity.this, MainActivity.class);
                PendingIntent pi = PendingIntent.getActivity(MainActivity.this, 0, in, 0);
                ncb.setContentIntent(pi);
                NotificationManager nm = (NotificationManager) getSystemService(
Context.NOTIFICATION_SERVICE);
                nm.notify(NOTIF_ALERTA_ID, ncb.build());
                jtv.setText("Cuenta: i=" + i);
            }
        });
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }
}
```



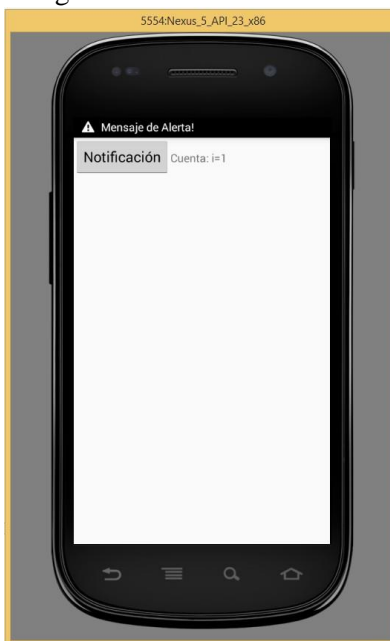
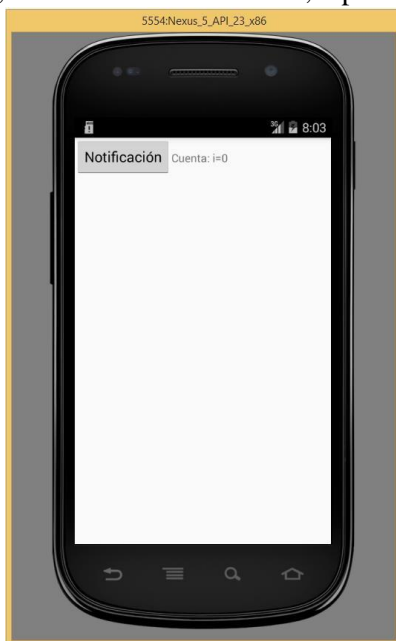
Paso 2. En la carpeta `res/layout`, abrir y modificar el archivo `activity_main.xml` con el siguiente código:

```
<?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"
    tools:context=".MainActivity" >
    <Button android:id="@+id/xbnN"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/notificacion" />
    <TextView
        android:id="@+id/xtv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Cuenta: i=0"/>
</LinearLayout>
```

Paso 3. En la carpeta `res/values`, abrir y modificar el archivo `strings.xml` con el siguiente código:


```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Notificaciones Android</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>
    <string name="notificacion">Notificación</string>
</resources>
```

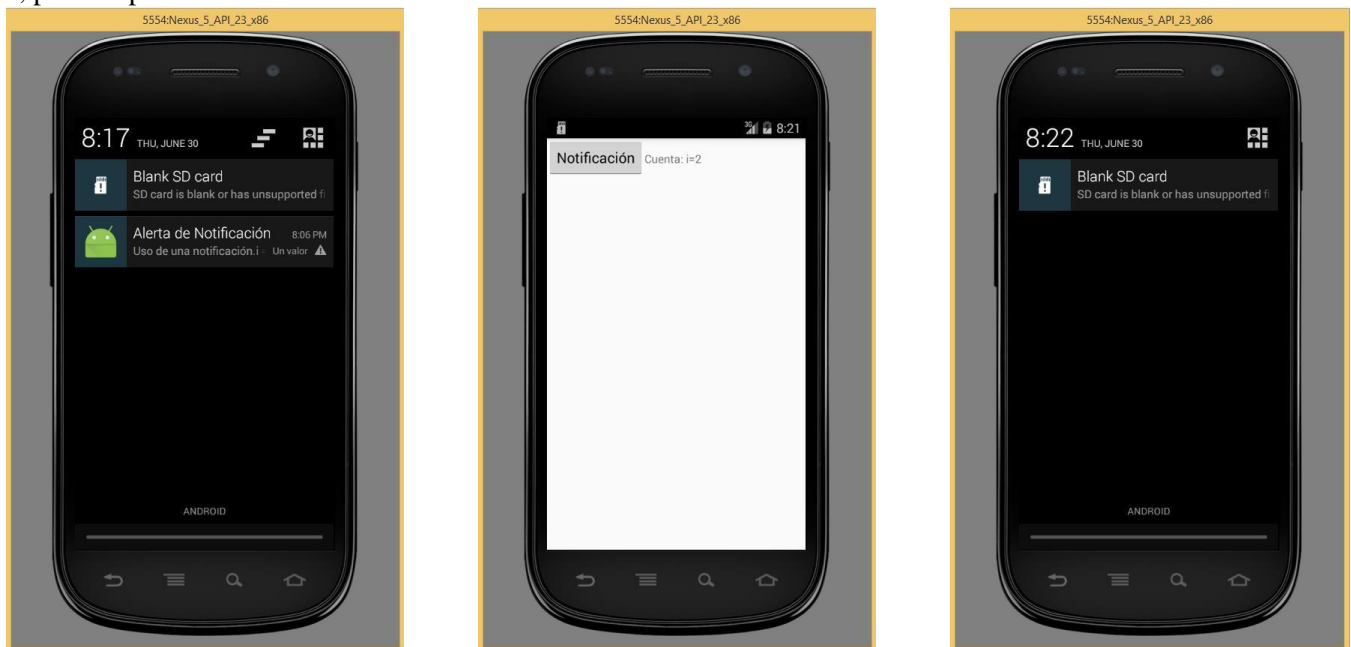
Paso 4. Por último, ejecutar la aplicación. Cuando por primera vez se digita el botón de Notificación, se muestra momentáneamente el mensaje **Mensaje de Alerta!**, y la cuenta de `i` se actualiza a 1. Si se digita el botón por segunda vez, se actualiza el valor de `i`, reportando una segunda notificación.



Enseguida, arrastrar hacia abajo la barra de notificaciones para visualizar los mensajes. Al digitar sobre el elemento **Alerta de Notificación**, se repliegan los mensajes y la cuenta de `i` vuelve a 0 lo que indica que se atendió la notificación, pero la notificación no desaparece de la lista.



Para borrar la notificación de la lista, arrastrar de nuevo hacia abajo la barra de notificaciones. Digitar el icono  de la barra, para limpiar la lista de las notificaciones.



EJERCICIO 1.

En el ejemplo anterior, realizar las modificaciones necesarias para que la variable `i` se muestre con 0 si, después de varias notificaciones, la lista de notificaciones se encuentra vacía.

EJEMPLO 2.

En el siguiente ejemplo, se realiza una notificación a través de otra actividad. El mensaje contenido se lee en un `TextView`.



Paso 1. Crear un nuevo proyecto **Notificaciones2**. En la carpeta `java/com.example.mipaquete`, abrir y modificar el archivo `MainActivity.java` con el siguiente código:

```
import android.app.*;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.support.v4.app.NotificationCompat;
public class MainActivity extends Activity {
    int id = 1;
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
    }
    public void onClick(View v){
        displayNotification();
    }
    protected void displayNotification(){
        Intent i = new Intent(this, NotificationView.class);
        i.putExtra("notificationID", id);
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, i, 0);
        NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        CharSequence ticker = "Nueva entrada en ESCOM";
        CharSequence contentTitle = "Notificaciones";
        CharSequence contentText = "Curso de Android";
        Notification noti = new NotificationCompat.Builder(this)
            .setContentIntent(pendingIntent)
            .setTicker(ticker)
            .setContentTitle(contentTitle)
            .setContentText(contentText)
            .setSmallIcon(R.mipmap.ic_launcher)
            .addAction(R.mipmap.ic_launcher, ticker, pendingIntent)
            .setVibrate(new long[] {100, 250, 100, 500})
            .build();
        nm.notify(id, noti);
    }
}
```

Paso 2. En la carpeta `java/com.example.mipaquete`, crear el archivo `NotificationView.java` y agregar el siguiente código:

```
import android.app.*;
import android.os.Bundle;
public class NotificationView extends Activity {
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.notificacion);
        NotificationManager nm = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);
        nm.cancel(getIntent().getExtras().getInt("notificationID"));
    }
}
```

Paso 3. En la carpeta `res/layout`, abrir y modificar el archivo `activity_main.xml` con el siguiente código:



```
<?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"
    tools:context=".MainActivity" >
    <Button
        android:id="@+id/xbn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="Enviar notificación" />
</LinearLayout>
```

Paso 4. En la carpeta `res/layout`, crear el archivo `notificación.xml` y agregar el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="\n\nDetalles de la notificacion\n...bla...bla" />
</LinearLayout>
```

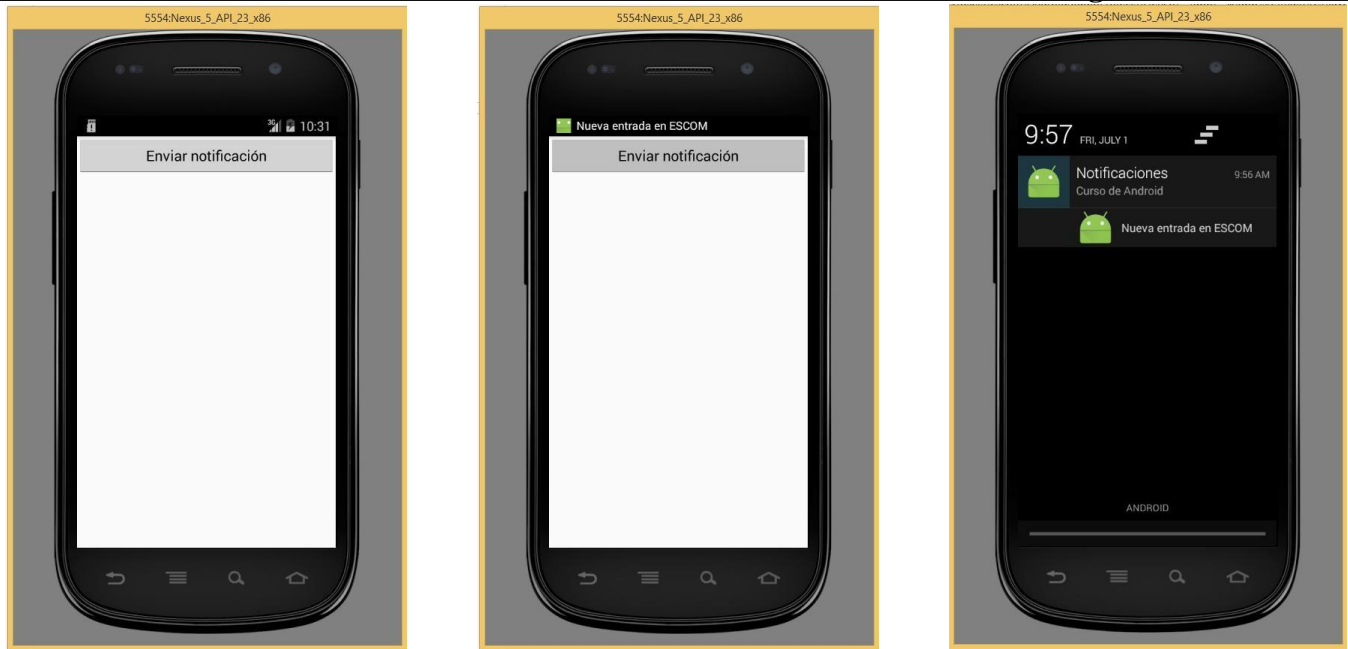
Paso 5. En la carpeta `app/manifests`, abrir el archivo predeterminado `AndroidManifest.xml` y agregar la etiqueta `<activity>` para la segunda actividad `NotificationView`, como se indica enseguida con letras negritas:


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.escom.lienzo">
    <application
        :
        >
        <activity
            android:name=".MainActivity"
            :
            >
            :
        </activity>

        <activity
            android:name=".NotificationView"
            android:label="Detalles de la Notificación">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.DEFAULT"/>
            </intent-filter>
        </activity>

    </application>
</manifest>
```

Paso 6. Por último, ejecutar la aplicación. Al inicio, al enviar una notificación con el botón, en la barra superior se muestra el mensaje de recepción. Para ver la notificación, la barra superior se arrastra hacia abajo.



Para leer la notificación, se selecciona y se muestra su contenido en el `TextView`. Para regresar a la pantalla principal, digitar el botón inferior  en este caso, al leer la notificación, ésta se borra de la lista automáticamente. En caso contrario, se siguen almacenando nuevas notificaciones. Observar que si la aplicación se cierra, debido a la persistencia, las notificaciones permanecen en la barra y se pueden atender después.



NOTA. Generar un reporte con las imágenes obtenidas al ejecutar las aplicaciones. Guardar el documento con la sintaxis `AlumnoTarea11Grupo.pdf` y enviarlo al sitio indicado por el profesor.