



Asignatura: Application Development for Mobile Devices.

Tarea 11 Bis: Notificaciones (ANDROID CHANNEL)

Introducción.

La barra de notificaciones, o barra de estado, de Android permite notificar que las aplicaciones, que se ejecutan como tareas de fondo, adviertan al usuario sin interrumpir su uso actual del dispositivo. Esta barra almacena todas las notificaciones que se envían desde las aplicaciones.

En la parte superior en la pantalla se muestra la barra de notificaciones. En su lado izquierdo se muestran los iconos de las notificaciones recibidas. Deslizando la barra hacia abajo se muestran los detalles de cada notificación. La notificación se compone de un icono, un título, un mensaje y una fecha. Se puede realizar una acción si el usuario digita sobre la notificación.

Se pueden ejecutar otras funcionalidades, solas o combinadas entre sí, por ejemplo la muestra de un mensaje, la reproducción de un sonido, la vibración del dispositivo, o incluso el parpadeo de la luz del dispositivo.

Los canales de notificación permiten que los desarrolladores de aplicaciones agrupen las notificaciones en grupos, o canales, con lo que el usuario adquiere la habilidad de ajustar la notificación para el canal completo a la vez. Por ejemplo, para un determinado canal, el usuario puede bloquear completamente todas las notificaciones, anular el nivel de importancia, o permitir que la notificación se muestre, lo cual significa que ello mejora la experiencia del usuario en una aplicación.

Se debe considerar que el constructor `NotificationCompat.Builder` necesita un ID del canal, ya que se requiere por motivos de compatibilidad con Android 8.0 (API nivel 26) y versiones posteriores, pero se ignora en versiones anteriores.

Se debe registrar un canal de notificaciones de la aplicación en el sistema transfiriendo una instancia de `NotificationChannel` a `createNotificationChannel()`. Por ejemplo:

```
private void createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        CharSequence name = getString(R.string.channel_name);
        String description = getString(R.string.channel_description);
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new NotificationChannel(CHANNEL_ID, name,
importance);
        channel.setDescription(description);
        NotificationManager notificationManager =
getSystemService(NotificationManager.class);
        notificationManager.createNotificationChannel(channel);
    }
}
```

En el siguiente código se muestra el uso de un intento para abrir una actividad cuando el usuario digita sobre la notificación:

```
Intent intent = new Intent(this, AlertDetails.class); // Un intento para la Activity.
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_CLEAR_TASK);
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);
NotificationCompat.Builder builder = new NotificationCompat.Builder(this, CHANNEL_ID)
    .setSmallIcon(R.drawable.notification_icon)
    .setContentTitle("Mi notificacion")
    .setContentText("Hola Mundo!")
    .setPriority(NotificationCompat.PRIORITY_DEFAULT)
    .setContentIntent(pendingIntent)
    .setAutoCancel(true);
```



Para mostrar la notificación se invoca a `NotificationManagerCompat.notify()` y se le pasa un ID único para la notificación y el resultado de `NotificationCompat.Builder.build()`. Por ejemplo:

```
NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);
// El notificationId es un entero único para cada notificación que se desee definir.
notificationManager.notify(notificationId, builder.build());
```

DESARROLLO.

EJEMPLO

Paso 1. Crear un nuevo proyecto en Android Studio. Es importante que en la configuración del proyecto se seleccione al menos la Minimum SDK con la API 26: Android 8.0 (Oreo). Abrir el archivo `MainActivity.java` y pegar el siguiente código.

```
public class MainActivity extends AppCompatActivity {
    private NotificationUtils nu;
    EditText jetTit, jetAut;
    Button jbnEnv;
    @Override
    protected void onCreate(Bundle b) {
        super.onCreate(b);
        setContentView(R.layout.activity_main);
        nu = new NotificationUtils(this);
        jetTit = (EditText) findViewById(R.id.xetTit);
        jetAut = (EditText) findViewById(R.id.xetAut);
        jbnEnv = (Button) findViewById(R.id.xbnEnv);
        jbnEnv.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String titulo = jetTit.getText().toString();
                String autor = jetAut.getText().toString();
                if(!TextUtils.isEmpty(titulo) && !TextUtils.isEmpty(autor)) {
                    Notification.Builder nb = nu.
                        getAndroidChannelNotification(titulo, "Por: " + autor);
                    nu.getManager().notify(101, nb.build());
                }
            }
        });
    }
}
```

Paso 2. Crear un nuevo archivo `NotificationUtils.java` y pegar el siguiente código.

```
import android.app.*;
import android.content.*;
import android.graphics.Color;
public class NotificationUtils extends ContextWrapper {
    private NotificationManager mManager;
    public static final String ANDROID_CHANNEL_ID = "com.example.myapplication.ANDROID";
    public static final String ANDROID_CHANNEL_NAME = "ANDROID CHANNEL";
    public NotificationUtils(Context c) {
        super(c);
        createChannels();
    }
    public void createChannels() {
        // Se crea el canal android
        NotificationChannel androidChannel = new NotificationChannel(ANDROID_CHANNEL_ID,
            ANDROID_CHANNEL_NAME, NotificationManager.IMPORTANCE_DEFAULT);
        // Establece si las notificaciones publicadas en este canal deben mostrar luces
        de notificación
    }
}
```



```

        androidChannel.enableLights(true);
        // Establece si la notificación publicada en este canal debe vibrar.
        androidChannel.enableVibration(true);
        // Establece el color de la luz de notificación para las notificaciones
publicadas en este canal.
        androidChannel.setLightColor(Color.GREEN);
        // Establece si las notificaciones publicadas en este canal aparecen en la
pantalla de bloqueo o no
        androidChannel.setLockscreenVisibility(Notification.VISIBILITY_PRIVATE);
        getManager().createNotificationChannel(androidChannel);
    }
    public Notification.Builder getAndroidChannelNotification(String titulo, String
texto) {
        return new Notification.Builder(getApplicationContext(), ANDROID_CHANNEL_ID)
            .setContentTitle(titulo)
            .setContentText(texto)
            .setSmallIcon(android.R.drawable.stat_notify_more)
            .setAutoCancel(true);
    }
    NotificationManager getManager() {
        if (mManager == null) {
            mManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
        }
        return mManager;
    }
}

```

Paso 3. Abrir el archivo activity_main.xml y pegar el siguiente código.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="16dp" >
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Notificacion con Android Channel"
            android:layout_gravity="center_horizontal"
            android:textAppearance="@style/TextAppearance.AppCompat.Title"/>
        <EditText
            android:id="@+id/xetTit"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Título"/>

        <EditText
            android:id="@+id/xetAut"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Autor"/>
    </LinearLayout>
</LinearLayout>

```



```

<Button
    android:id="@+id/xbnEnv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Enviar"/>
</LinearLayout>
</LinearLayout>

```

Paso 4. Abrir el archivo `strings.xml` y actualizar el contenido con el siguiente código.

```

<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Notificaciones</string>
</resources>

```

Paso 5. Ejecutar la aplicación y seguir la secuencia de las siguientes imágenes. Al inicio, se muestra la solicitud de ingreso de algunos datos, ver fig.1. Se ingresan algunos datos, ver la fig.2. Al digitar el botón **Enviar** se envía una notificación y se actualiza la barra superior de notificaciones. Observar que se muestra un nuevo símbolo sobre la barra, ver fig.3. Para mostrar las notificaciones, se arrastra hacia abajo la barra de notificaciones, ver fig.4. Si se desea eliminar las notificaciones se digita el botón **CLEAR ALL** y se actualiza la barra de notificaciones; es decir, se borra el símbolo de la notificación, ver fig.5.



Figura 1.

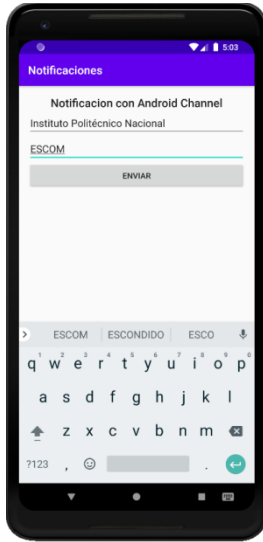


Figura 2.

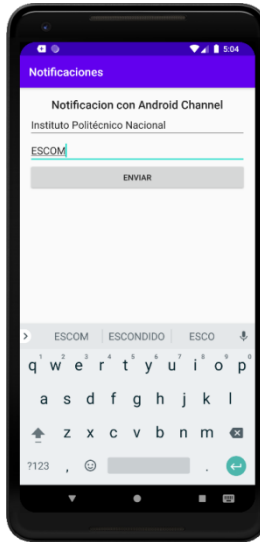


Figura 3.

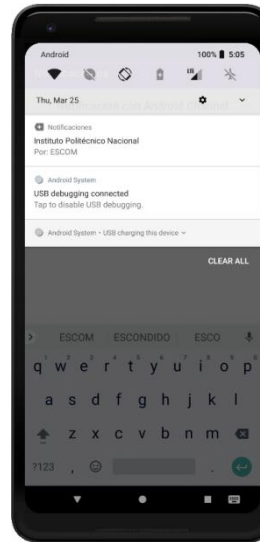


Figura 4.

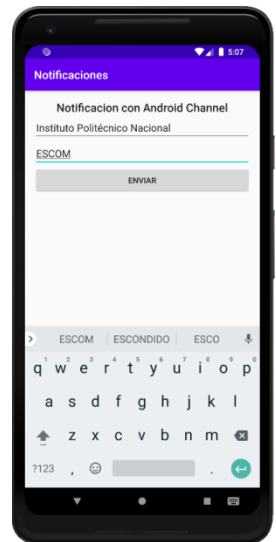


Figura 5.

EJERCICIO.

Modificar los dos ejercicios de la Tarea 11 Notificaciones para utilizar canales que permitan mostrar las notificaciones en la barra de notificaciones. Se sugiere basarse en el ejemplo de este archivo para adaptar el código conveniente en los dos ejemplos anteriores.

NOTA. Generar un reporte con las imágenes obtenidas al ejecutar las aplicaciones. Guardar el documento con la sintaxis `AlumnoTarea11BisGrupo.pdf` y enviarlo al sitio indicado por el profesor.