

## PROGRAMACION EN ANDROID CON JAVA

### PRACTICA No. 3

### NOTIFICACIONES

#### PARTE 1.

#### CREACION DEL PROYECTO

Ejecute el programa Android Studio. En la ventana que se despliega haga clic sobre el menú File y la Opción New Project.

En la nueva ventana seleccione la pestaña Phone and Table y dentro de esta la opción **Empty Activity** y haga clic en el botón Next.

En la siguiente ventana escriba como nombre del proyecto **Notificaciones** y como dominio escriba `ud.example.notificaciones`, como lenguaje seleccione Java y seleccione la mínima versión de Android de ejecución para su nueva aplicación (se recomienda la 6.0.0), haga clic sobre el botón Finish.

#### PARTE 2.

#### ADICION DE CODIGO

##### Modificación del archivo `strings.xml`

Haga doble clic sobre el archivo `string.xml` del paquete `values`, del paquete `res`, y modifique el código de la siguiente manera:

```
<resources>
    <string name="app_name">Notificaciones</string>
    <string name="mensaje01">Mensaje (Toast) por defecto</string>
</resources>
```

##### Modificación del archivo `activity_main.xml`

Nuestra actividad al final debería lucir de la siguiente manera:



Haga doble clic sobre el archivo activity\_main.xml del paquete layout, y cambie el modo de visualización a texto, por medio de la pestaña Text. Modifique el código de la siguiente manera:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:layout_editor_absoluteX="1dp"
        tools:layout_editor_absoluteY="1dp">
```

```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="noti01"
    android:text="TOAST 01" />

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="noti02"
    android:text="TOAST 02" />

<Button
    android:id="@+id/button3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="noti03"
    android:text="TOAST 03" />

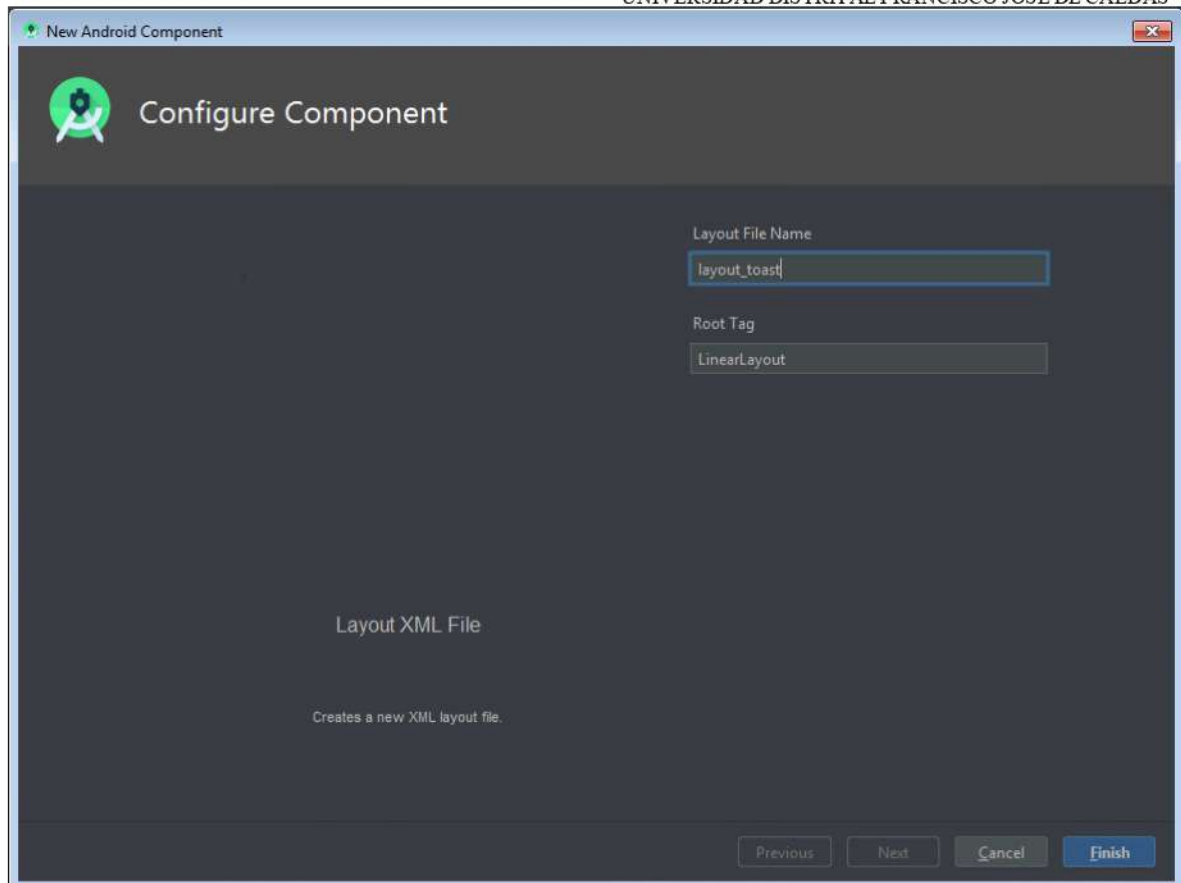
<Button
    android:id="@+id/button4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="noti04"
    android:text="BARRA DE NOTIFICACION" />

<Button
    android:id="@+id/button5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="noti05"
    android:text="SNACKBAR" />
</LinearLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

### Creación de archivo `layout_toast.xml`

Por medio de un explorador de Windows o Linux localice la carpeta `drawable` del proyecto, y dentro de esta copie el archivo `info.pgn` que acompaña la práctica.

Sobre el paquete `layout`, haga clic derecho y agregue un nuevo archivo `Layout XML File` al proyecto, este se llamara `layout_toast.xml`.



Haga doble clic sobre el archivo layout\_toast.xml del paquete layout, y cambie el modo de visualización a texto, por medio de la pestaña Text. Agregue el siguiente código:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/layoutToast"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/info" />

    <TextView
        android:id="@+id/mensajeLbl"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="10dp"
```



```
        android:layout_weight="1"  
        android:textColor="#000000"  
        android:text="TextView" />  
</LinearLayout>
```

### Modificación del archivo MainActivity.java

Haga doble clic sobre el archivo MainActivity.java del paquete java, y modifique el código de la siguiente manera:

Agregue los siguientes atributos privados a la clase

```
private static final int ID_MEN_BARRA_NOTIF = 1;
```

Agregue los siguientes métodos a la clase:

```
public void noti01(View v){  
    String mess = getResources().getString(R.string.mensaje01);  
    // Creación del mensaje  
    Toast toast1 = Toast.makeText(getApplicationContext(), mess,  
    Toast.LENGTH_SHORT);  
    // Mostrar el mensaje  
    toast1.show();  
}  
  
public void noti02(View v){  
    // Creación del mensaje  
    Toast toast1 = Toast.makeText(getApplicationContext(),  
        "Mensaje (Toast) con gravity", Toast.LENGTH_SHORT);  
    // Indicamos el posicionamiento  
    toast1.setGravity(Gravity.CENTER|Gravity.RIGHT,0,0);  
    // Mostrar el mensaje  
    toast1.show();  
}  
  
public void noti03(View v){  
    // Creamos el mensaje  
    Toast toast1 = new Toast(getApplicationContext());  
    // Inflamos el diseño de layout_toast.xml  
    LayoutInflater inflater = getLayoutInflater();  
    View layout = inflater.inflate(R.layout.layout_toast, (ViewGroup)  
    findViewById(R.id.layoutToast));  
    // Asignamos los componentes del diseño  
    TextView txtMsg = (TextView)layout.findViewById(R.id.mensajeLbl);  
    txtMsg.setText("Mensaje con diseño personalizado");  
    // Indicamos la duración corta para el mensaje  
    toast1.setDuration(Toast.LENGTH_SHORT);  
    // Asignamos el diseño al Toast
```

```

toast1.setView(layout);
// Mostramos el mensaje
toast1.show();
}

public void noti04(View v){
    //Obtenemos una referencia al servicio de notificaciones
    String ns = Context.NOTIFICATION_SERVICE;
    String CHANNEL_ID = "ud.com.ANDROID";
    NotificationManager notManager = (NotificationManager)
getSystemService(ns);

    //creación del canal de notificaciones, solo para API 26 o
superiores
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O){
        CharSequence name = "ANDROID CHANNEL";
        String descripcion = "Canal de Notificaciones ANDROID UD";
        int importance = NotificationManager.IMPORTANCE_DEFAULT;
        NotificationChannel channel = new
NotificationChannel(CHANNEL_ID, name, importance);
        channel.setDescription(descripcion);
        notManager.createNotificationChannel(channel);
    }

    //Configuramos la notificación que va a aparecer en la barra
    int icono = android.R.drawable.stat_sys_warning;
    CharSequence textoEstado = "¡Atención!";
    CharSequence titulo = "Mensaje de alerta";
    CharSequence descripcion = "Ejemplo de notificación.";
    long hora = System.currentTimeMillis();

    // Obtenemos el contexto de la aplicación
    Context contexto = getApplicationContext();
    //Configuramos el Intent
    Intent notIntent = new Intent(contexto, MainActivity.class);
    // Usamos una PendingIntent para crear la notificación
    PendingIntent contIntent = PendingIntent.getActivity(contexto, 0,
notIntent, 0);

    NotificationCompat.Builder mBuilder =
        (NotificationCompat.Builder) new
NotificationCompat.Builder(MainActivity.this, CHANNEL_ID)
        .setSmallIcon(icono)
        .setLargeIcon((((BitmapDrawable)getResources()
        .getDrawable(R.drawable.info)).getBitmap()))
        .setContentTitle(titulo)
        .setContentText(descripcion)
        .setContentInfo(textoEstado)
        .setWhen(hora)

```

```

        .setContentIntent(contIntent)
        .setAutoCancel(true) //AutoCancel: cuando se pulsa
la notificación desaparece

.setColor(getResources().getColor(R.color.colorPrimaryDark))
        //Para añadir sonido y vibración
        .setVibrate(new long[] {100, 250, 100, 500})

.setSound(RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION))
        ;

//Enviamos la notificación
notManager.notify(ID_MEN_BARRA_NOTIF, mBuilder.build());
}

public void noti05(View v){
    String mess = getResources().getString(R.string.mensaje01);
    //creación de Snackbar
    Snackbar.make(v, mess, Snackbar.LENGTH_SHORT)
        .setAction("Acción", new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Toast toast1 =
Toast.makeText(getApplicationContext(), "Hola Mundo!",
Toast.LENGTH_SHORT);
                toast1.show();
                Log.i("SNACKBAR", "Acción realizada!");
            }
        })
    })

.setActionTextColor(getResources().getColor(R.color.colorPrimary))
        .show();
}

```

Por último, **Ejecute y Evalúe la aplicación**

## TRABAJO ADICIONAL

1. Adicionar el código necesario para que su aplicación soporte los idiomas francés e inglés.