Notificaciones

Profesor: Ana Isabel Vegas

INDICE

1.- TOAST

INTRODUCCIÓN

Una notificación o un toast que es la forma como solemos referirnos es un mensaje que mostramos al usuario durante un espacio de tiempo. Transcurrido ese tiempo el mensaje desaparece.

NOTIFICACIONES POR DEFECTO

Para generar un toast utilizamos el método estático makeText al cual le pasamos como argumentos un contexto, el mensaje a mostrar y la duración de la exposición.

Un toast por defecto siempre se muestra en la parte inferior de la pantalla.

```
// Toast por defecto que se muestra en la parte inferior de la pantalla
Toast toast1 = Toast.makeText(getApplicationContext(),
        "Este es un toast por defecto", Toast.LENGTH_LONG);
toast1.show();
```

Gráfico 1. Mensaje Toast por defecto

DURACION DE LA NOTIFICICACION

La duración de la notificación se puede establecer a través de dos constantes:

- **Toast.LENGTH_LONG**; Muestra el mensaje durante un periodo más largo.
- **Toast.LENGTH_SORT**; Muestra el mensaje durante un breve periodo.

UBICACION DEL MENSAJE

Con el método setGravity podemos establecer la ubicación donde mostrar la notificación. Se pasa como argumentos la alineación vertical y horizontal por este orden.

```
// Toast prsonalizado que se muestra en centro vertica|Horizontal de la pantalla
Toast toast2 = Toast.makeText(getApplicationContext(),
        "Este es un toast personalizado", Toast.LENGTH_LONG);
toast2.setGravity(Gravity.CENTER|Gravity.CENTER, 0, 0);
toast2.show();
```

Gráfico 2. Toast mostrado en el centro de la pantalla

TOAST PERSONALIZADO CON LAYOUT

Otra forma de crear un toast es utilizando nuestro propio layout. De esta forma podemos personalizarlo añadiendo por ejemplo un icono al mensaje a mostrar.

```
Toast toast = new Toast(getApplicationContext());

LayoutInflater inflater = getLayoutInflater();

View layout = inflater.inflate(R.layout.toast, (ViewGroup) findViewById(R.id.lytToast));

TextView mensaje = (TextView) layout.findViewById(R.id.texto);
mensaje.setText("Toast con layout");
toast.setDuration(Toast.LENGTH_LONG);
toast.setView(layout);
toast.show();
```

Gráfico 3. Toast personalizado con un layout

En el layout de este ejemplo mostramos una imagen, el testo y otra imagen al final. Establecemos un identificador al layout de tal forma que lo podamos referenciar al mostrar el toast.

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/lytToast"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

    <ImageView
        android:id="@+id/imagen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/warning" />

    <TextView
        android:id="@+id/texto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Toast con layout"/>

    <ImageView
        android:id="@+id/imagen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/warning" />

</LinearLayout>
```

Gráfico 4. Layout que se mostrará en el toast

2.- NOTIFICACIONES EN LA BARRA DE ESTADO

Si queremos mostrar la notificación en la barra de estado. Esta está situada en la parte superior de la pantalla.

```
NotificationCompat.Builder builder =
        new NotificationCompat.Builder(getApplicationContext());
builder.setSmallIcon(android.R.drawable.stat_sys_warning);
builder.setContentTitle("Mensaje de alerta");
builder.setContentText("Ejemplo de notificacion");
builder.setTicker("Alerta!!");


NotificationManager manager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
manager.notify(1, builder.build());
```

Gráfico 5. Notificación en la barra de estado

ASOCIAR UN SONIDO

Si queremos asociar un sonido a la notificación lo haremos tal como se muestra en la imagen.

```
// Asociar un sonido a la notificacion
Notification notificacion = builder.build();
notificacion.defaults = Notification.DEFAULT_SOUND;
```

Gráfico 6. Asignar sonido a la notificación

AÑADIENDO VIBRACION

Si queremos asociar una vibración a la notificación lo haremos tal como se muestra en la imagen.

```
// Asociar una vibracion a la notificacion
Notification notificacion = builder.build();
notificacion.defaults = Notification.DEFAULT_VIBRATE;
```

Gráfico 7. Asignar vibración a la notificación

CONFIGURANDO PARPADEO DE LED

Otra opción que tenemos para llamar la atención del usuario es añadir un parpadeo con LED.

En la imagen vemos como hacerlo utilizando un parpadeo por defecto.

```
// Añadir parpadeo LED  por defecto a la notificacion
Notification notificacion = builder.build();
notificacion.defaults = Notification.DEFAULT_LIGHTS;
```

Gráfico 8. Parpadeo LED

También podemos personalizar las acciones con el parpadeo LED como por ejemplo, cambiar el color, establecer la duración, ...etc.

```
// Añadir parpadeo LED  personalizado a la notificacion
Notification notificacion = builder.build();
// Indicamos que el LED sea de color verde;
notificacion.ledARGB = 0xff00ff00;
// Se ilumina durante 300 mseg.
notificacion.ledOnMS = 300;
// Se apaga durante 1 segundo
notificacion.ledOffMS = 1000;
notificacion.flags = Notification.FLAG_SHOW_LIGHTS;
```

Gráfico 9. Parpadeo LED Personalizado

OCULTAR NOTIFICACIONES

Con el método cancel pasando como argumento el identificador de la notificación podemos cancelar el toast y de esta forma ocultarlo.

```
// Cancelar y ocultar la notificacion
manager.cancel(1);
```

Gráfico 10. Cancelar la notificación

3.- NOTIFICACIONES CON CUADROS DE DIALOGO

Otra opción para mostrar notificaciones son los cuadros de diálogo. Hasta ahora las notificaciones que hemos visto no requerían ninguna actividad por parte del usuario. Si necesitamos pedir una confirmación, pedir al usuario una selección o simplemente mostrar un mensaje los cuadros de diálogo son nuestra solución.

CUADROS DE ALERTA

Un cuadro de alerta es un cuadro de dialogo formado por un titulo, el texto a mostrar y un botón de aceptación.

Para construir el cuadro de alerta creamos un clase que heredará de DialogFragment y sobrescribimos el método onCreateDialog().

En este método es donde creamos el cuadro de diálogo utilizando la clase AlertDialog y su método builder().

Una vez creado ya podemos establecer el título, el texto del mensaje y el botón.

El botón define el manejador de eventos onClick(), el cual dará respuesta de cancelar l y cerrar el cuadro de diálogo al ser pulsado.

En la siguiente imagen mostraos como crear la clase Alerta.

```
public class Alerta extends DialogFragment{


    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("AVISO");
        builder.setMessage("Pulsa aceptar si te has enterado de la alerta");
        builder.setPositiveButton("ACEPTAR",
                new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog,
                            int which) {
                        dialog.cancel();

                    }
                });
        return builder.create();
    }

}
```

Gráfico 11. Cuadro de diálogo de Alerta

Una vez creado el cuadro de diálogo ya podremos mostrarlo desde la actividad principal.

Al crear la actividad heredamos de FragmentActivity y no de Activity como hacemos habitualmente. Al pulsar el botón se mostrará la alerta.

```
public class MainActivity extends FragmentActivity {

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

        Button boton = (Button) findViewById(R.id.boton);

        boton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Alerta alerta = new Alerta();
                FragmentManager fm = getSupportFragmentManager();
                alerta.show(fm, "identificador");
            }
        });

    }
```

Gráfico 12. Actividad desde la que se muestra el cuadro de diálogo de Alerta

CUADROS DE SELECCION

Un cuadro de selección muestra al usuario varias opciones de las cuales deberá elegir una para continuar.

El tratamiento es muy similar al ejemplo anterior.

```
public class Seleccion extends DialogFragment{

    final String[] items = {"Español", "Ingles", "Frances"};


    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("SELECCIONA");
        builder.setItems(items, new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog, int item) {
                        Log.i("Ejemplo DialogoSeleccion",
                                "Idioma seleccionado: " + items[item]);
                        System.out.println("Ejemplo DialogoSeleccion :: " +
                                "Idioma seleccionado: " + items[item]);
                        dialog.cancel();

                    }
                });

        return builder.create();
    }

}
```

Gráfico 13. Cuadro de diálogo de selección

Nuevamente utilizamos la actividad principal para mostrar el cuadro de diálogo.

```java
public class MainActivity extends FragmentActivity {

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

        Button boton = (Button) findViewById(R.id.boton);

        boton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Seleccion seleccion = new Seleccion();
                FragmentManager fm = getSupportFragmentManager();
                seleccion.show(fm, "identificador");
            }
        });

    }
```

Gráfico 14. Actividad desde la que se muestra el cuadro de diálogo de selección

CUADROS DE CONFIRMACION

Un cuadro de confirmación es muy similar al de alerta que hemos visto.

En este caso vamos a tener dos botones: uno para aceptar y otro para cancelar tal como muestra la imagen.

```
public class Confirmacion extends DialogFragment{

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setTitle("RESPONDE");
        builder.setMessage("Estas de acuerdo?");

        builder.setPositiveButton("ACEPTAR",
                new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog,
                            int which) {
                        Log.i("Ejemplo DialogoConfirmacion",
                                "El usuario acepta");
                        System.out.println("Ejemplo DialogoConfirmacion :: " +
                                "El usuario acepta");
                        dialog.cancel();
                    }
                });

        builder.setNegativeButton("CANCELAR",
                new DialogInterface.OnClickListener() {

                    @Override
                    public void onClick(DialogInterface dialog,
                            int which) {
                        Log.i("Ejemplo DialogoConfirmacion",
                                "El usuario no esta de acuerdo");
                        System.out.println("Ejemplo DialogoConfirmacion :: " +
                                "El usuario no esta de acuerdo");
                        dialog.cancel();
                    }
                });
        return builder.create();
    }
}
```

Gráfico 15. Cuadro de diálogo de confirmación

```java
public class MainActivity extends FragmentActivity {

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

        Button boton = (Button) findViewById(R.id.boton);

        boton.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
                Confirmacion confirmacion = new Confirmacion();
                FragmentManager fm = getSupportFragmentManager();
                confirmacion.show(fm, "identificador");
            }
        });

    }
```

Gráfico 16. Actividad desde la que se muestra el cuadro de diálogo de confirmación

ÍNDICE DE GRÁFICOS