

SEMINARIO DE LENGUAJES

OPCIÓN ANDROID

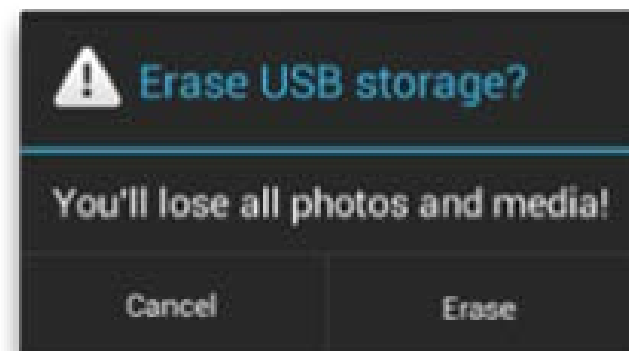
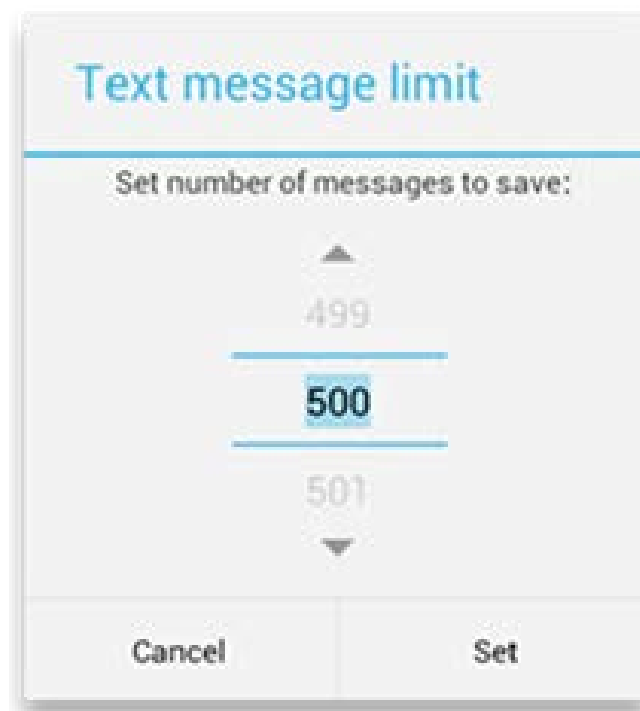


Diálogos y Notificaciones

Esp. Delia Lisandro, Mg. Corbalán Leonardo

Diálogos

- Ventana pequeña que le indica al usuario que debe tomar una decisión o ingresar información adicional.
- No ocupa toda la pantalla y generalmente se usa para eventos modales que requieren que los usuarios realicen alguna acción para poder continuar.



Diálogos

- Es posible diseñar nuestros propios diseños de dialogo.
 - Lectura adicional: <https://material.io/design/components/dialogs.html?hl=es-419>
- Existen diálogos predefinidos:
 - AlertDialog
 - Un dialogo que muestra un título, hasta tres botones, una lista de ítems para seleccionar o un layout específico.
 - DatePickerDialog
 - Dialogo para seleccionar una fecha
 - TimePickerDialog
 - Dialogo para seleccionar una hora.

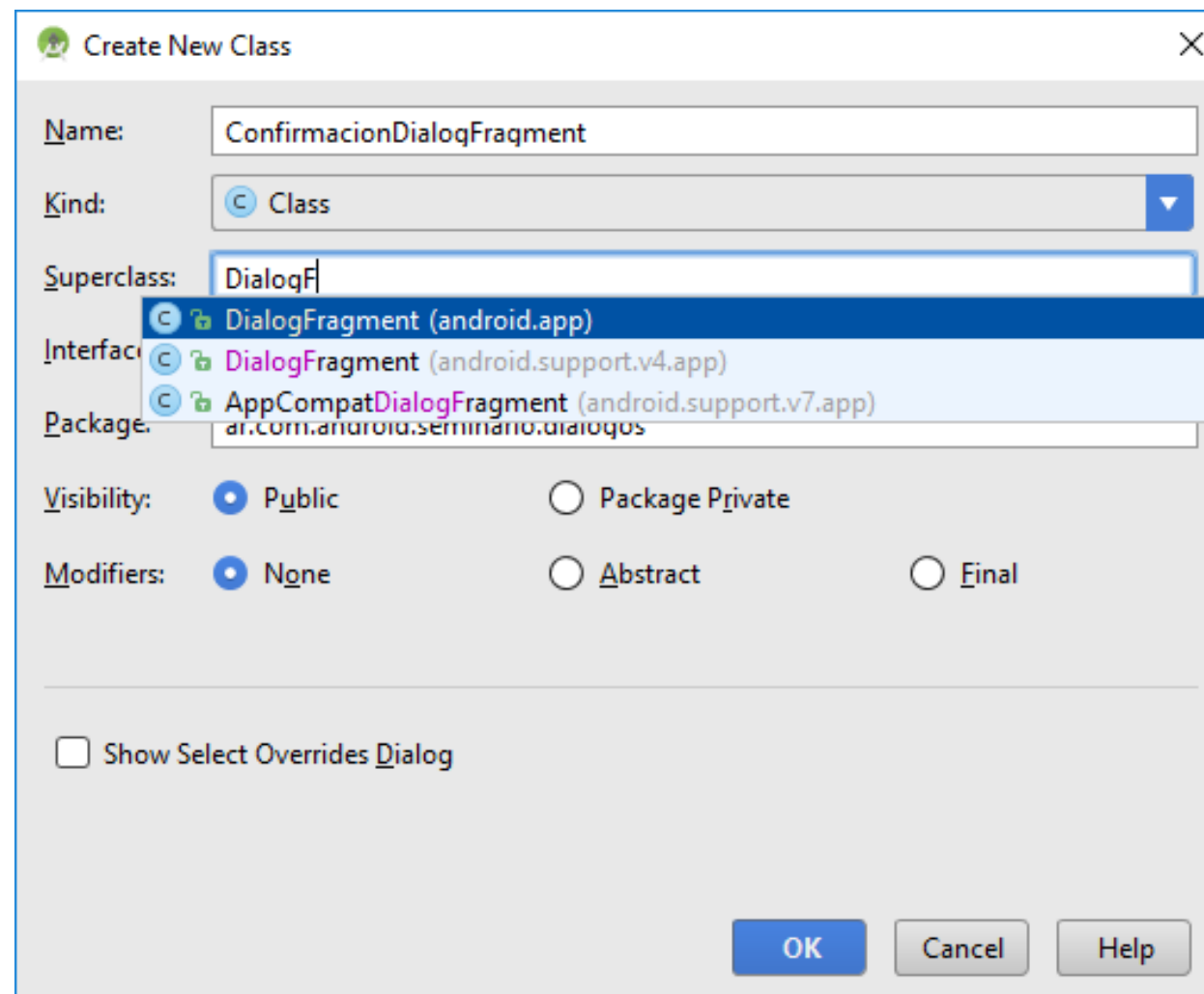
Diálogos

- Los diálogos se crean extendiendo la clase `DialogFragment`, en el método `onCreateDialog`.

```
public class ConfirmacionDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setMessage(R.string.confirmacion)
            .setPositiveButton(R.string.aceptar, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    // el usuario aceptó el dialogo !
                }
            })
            .setNegativeButton(R.string.cancelar, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    // el usuario canceló el dialogo !
                }
            });
        return builder.create();
    }
}
```

Dialogos - Actividad guiada

- Actividad: Abrir un diálogo al presionar un botón.
- Crear un nuevo proyecto desde Android Studio
- Crear una clase Java, especificarle un nombre y la superclase DialogFragment



Dialogos - Actividad guiada

- Definir strings en el archivo de recursos

```
<resources>
```

```
...
```

```
<string name="salir">Salir</string>
```

```
<string name="confirmacion">¿Está seguro que desea salir?</string>
```

```
<string name="aceptar">Si, muy seguro!</string>
```

```
<string name="cancelar">Mejor no, me quedo!</string>
```

```
</resources>
```

Dialogos - Actividad guiada

```
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.DialogFragment;
import android.os.Bundle;
import android.content.DialogInterface;

public class ConfirmacionDialogFragment extends DialogFragment
{
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setMessage(R.string.confirmacion)
            .setPositiveButton(R.string.aceptar, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                }
            })
            .setNegativeButton(R.string.cancelar, new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                }
            });
        return builder.create();
    }
}
```

Dialogos - Actividad guiada

- Se invocará al dialogo al hacer click en un botón.
- Editar el archivo activity_main.xml

```
<LinearLayout
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/salir"
        android:onClick="salir"/>
</LinearLayout>
```


Dialogos - Actividad guiada

- Se invocará al dialogo al hacer click en un botón.
- Editar el archivo activity_main.xml

```
<LinearLayout
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/salir"
        android:onClick="salir" />
</LinearLayout>
```

Se debe implementar el manejador del click,
para lanzar el diálogo

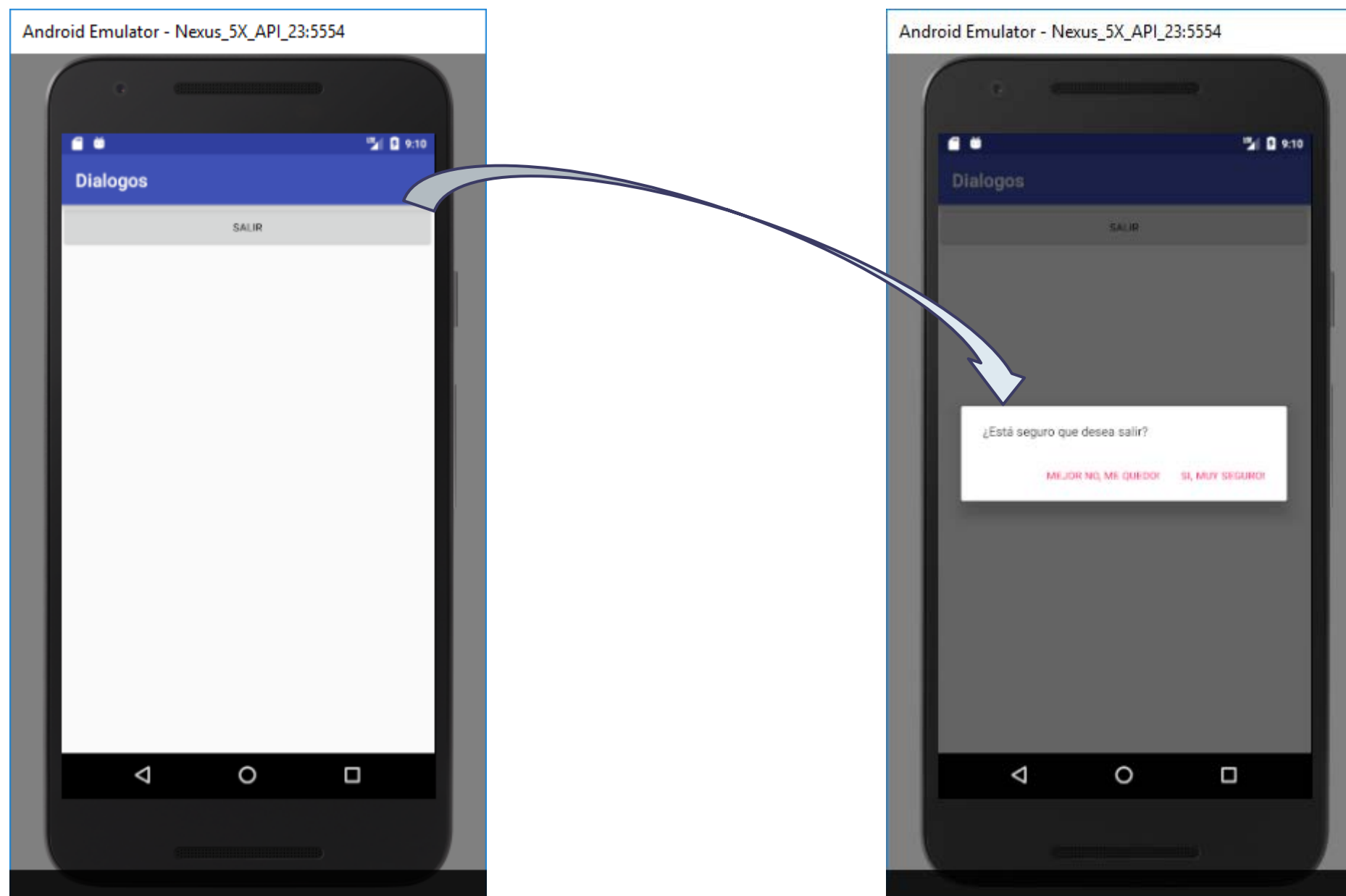
Dialogos - Actividad guiada

- Agregar a la clase MainActivity el método salir

```
public class MainActivity extends AppCompatActivity {  
  
    ...  
  
    public void salir(View view){  
        ConfirmacionDialogFragment dialogo = new ConfirmacionDialogFragment();  
        dialogo.show(getFragmentManager(), "tagConfirmacion");  
    }  
}
```

Dialogos - Actividad guiada

- Probar en emulador



Manejando la respuesta del usuario - Actividad guiada

- Agregue las sentencias sombreadas en el método **onCreateDialog** de la clase **ConfirmacionDialogFragment**

```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
    builder.setMessage(R.string.confirmacion)  
        .setPositiveButton(R.string.aceptar, new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                getActivity().finish();  
            }  
        })  
        .setNegativeButton(R.string.cancelar, new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                Toast.makeText(getActivity(), "El usuario decidió quedarse", Toast.LENGTH_SHORT).show();  
            }  
        });  
    return builder.create();  
}
```

Manejando la respuesta del usuario - Actividad guiada

```
public Dialog onCreateDialog(Bundle savedInstanceState) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());  
    builder.setMessage(R.string.confirmacion)  
        .setPositiveButton(R.string.aceptar, new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                getActivity().finish();  
            }  
        })  
        .setNegativeButton(R.string.cancelar, new DialogInterface.OnClickListener() {  
            public void onClick(DialogInterface dialog, int id) {  
                Toast.makeText(getActivity(), "El usuario decidió quedarse", Toast.LENGTH_SHORT).show();  
            }  
        });  
    return builder.create();  
}
```

Información complementaria

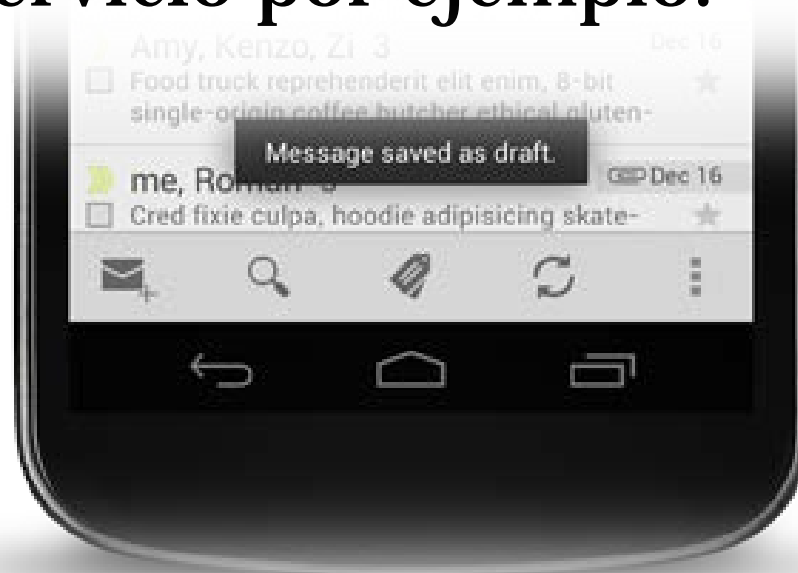
Puede mejorarse el diseño de la solución usando el *patrón de diseño Observer*, el cual delega el comportamiento a la clase Actividad (se verá en otro curso)

Notificaciones

- Se conocerán tres tipos de notificaciones
 - Toast
 - Barra de estados
 - Snackbar

Toast

- Mensaje que se muestra en pantalla durante unos segundos.
- No requiere intervención por parte del usuario.
- No interfiere con las acciones que está llevando a cabo el usuario
- Desaparece automáticamente.
- No requiere de una actividad en foreground. Puede ser lanzado desde un servicio por ejemplo.




Toast

- Son ideales para mostrar mensajes rápidos y sencillos al usuario.
- Al no requerir confirmación, no se debería utilizar para hacer avisos importantes
- Invocación:

```
Toast toast = Toast.makeText(getApplicationContext(), "Hola mundo!", Toast.LENGTH_LONG);  
toast.show();
```

- La duración puede ser:
 - `Toast.LENGTH_LONG`
 - `Toast.LENGTH_SHORT`



Agregar un segundo botón a la activity que al ser presionado muestre este mensaje

Toast

- Aparece por defecto en la parte inferior de la pantalla. Esto puede ser modificado mediante el método `setGravity`.
 - Ejemplo ubicándolo en la esquina superior izquierda:

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);  
toast.show();
```

Toast

- Aparece por defecto en la parte inferior de la pantalla. Esto puede ser modificado mediante el método `setGravity`.
 - Ejemplo ubicándolo en la esquina superior izquierda:

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);  
toast.show();
```

Ubicación. En la clase `Gravity` están definidas constantes para indicar diferentes posiciones

Toast

- Aparece por defecto en la parte inferior de la pantalla. Esto puede ser modificado mediante el método `setGravity`.
 - Ejemplo ubicándolo en la esquina superior izquierda:

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);  
toast.show();
```



Desplazamiento desde el borde superior

Toast

- Aparece por defecto en la parte inferior de la pantalla. Esto puede ser modificado mediante el método `setGravity`.
 - Ejemplo ubicándolo en la esquina superior izquierda:

```
toast.setGravity(Gravity.TOP | Gravity.LEFT, 0, 0);  
toast.show();
```



Desplazamiento desde el borde izquierdo

Toast

- Por defecto se visualiza en un rectángulo gris, ligeramente traslúcido y redondeado. Esto puede ser modificado definiéndole un layout específico

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/custom_toast_container"
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:padding="8dp"
    android:background="@color/colorAccent"
    >
    <ImageView android:src="@drawable/droid"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginRight="8dp"
        />
    <TextView android:id="@+id/texto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#FFF"
        />
</LinearLayout>
```

Layout definido en
“res/layout/toast_layout.xml”

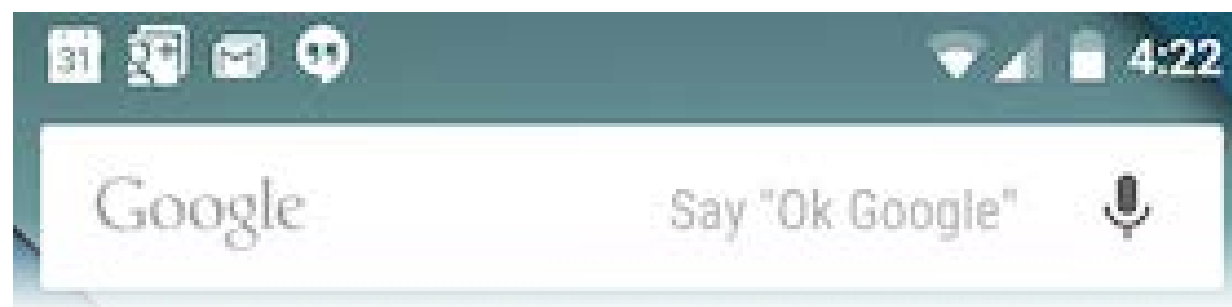
Toast

- Asociación de un layout específico a un toast

```
LayoutInflater inflater = getLayoutInflater();  
View layout = inflater.inflate(R.layout.toast_layout,  
    (ViewGroup) findViewById(R.id.custom_toast_container));  
  
TextView text = (TextView) layout.findViewById(R.id.texto);  
text.setText("Redefiniendo el diseño del toast");  
  
Toast toast = new Toast(getApplicationContext());  
toast.setView(layout);  
toast.show();
```

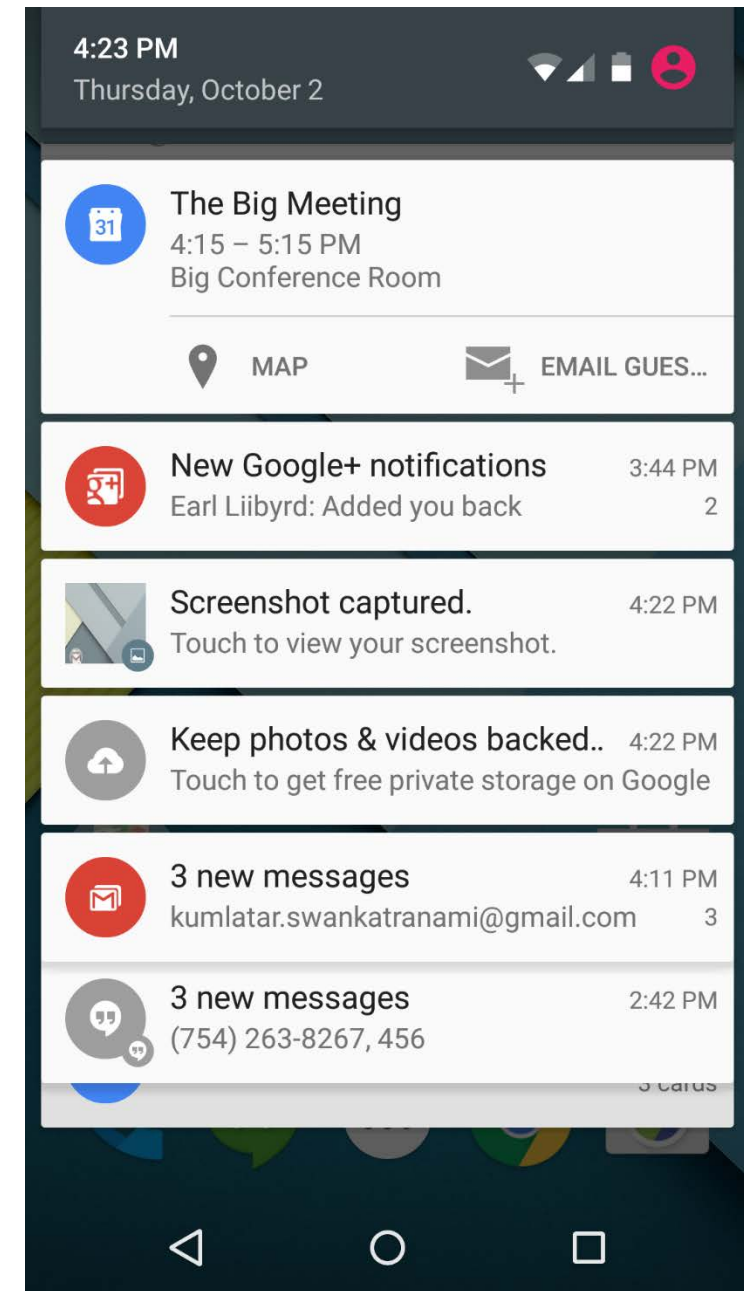
Notificaciones en la barra de estado

- Permite mostrar información al usuario, de un modo más persistente que mediante Toast.
- La notificación se muestra fuera de la interfaz de usuario de la aplicación, en la barra de estados.
- Inicialmente se muestra el aviso mediante un ícono, en el área de notificaciones.



Notificaciones en la barra de estado

- Las notificaciones constan de un icono, un título, una fecha y una descripción.
- El usuario puede expandir el área de notificaciones y ver el detalle de la notificación.
- Las notificaciones pueden tener acciones asociadas.



StatusBar - Actividad guiada

- Crear un nuevo proyecto desde Android Studio
- Agregar un ícono como recurso al proyecto.
 - File > New > Image Asset
 - IconType: ActionBar and TabIcons
 - Name: notificacion
 - AssertType: Clip Art
 - ClipArt: seleccionar uno de los disponibles
 - Next y finish

StatusBar - Actividad guiada

- Definir un botón en el layout de la actividad y asociarle un manejador al click

```
<LinearLayout
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Mostrar notificación"
        android:onClick="mostrarNotificacion" />
</LinearLayout>
```

StatusBar - Actividad guiada

- Lanzar la notificación desde la actividad

```
import android.app.NotificationManager;
import android.content.Context;
import android.support.v4.app.NotificationCompat;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    public static final int NOTIFICACION_ID = 1;

    // ...

    public void mostrarNotificacion(View view){
        NotificationCompat.Builder mBuilder =
            new NotificationCompat.Builder(this)
                .setSmallIcon(R.drawable.notificacion)
                .setContentTitle("Mi notificación")
                .setContentText("Hola Curso!");

        NotificationManager mNotificationManager =
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
    }
}
```

StatusBar - Actividad guiada

- Lanzar la notificación desde la actividad

```
import android.app.NotificationManager;
import android.content.Context;
import android.support.v4.app.NotificationCompat;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    public static final int NOTIFICACION_ID = 1;

    // ...

    public void mostrarNotificacion(View view){
        NotificationCompat.Builder mBuilder =
            new NotificationCompat.Builder(this)
                .setSmallIcon(R.drawable.notificacion)
                .setContentTitle("Mi notificación")
                .setContentText("Hola Curso!");

        NotificationManager mNotificationManager =
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
    }
}
```

Se define la notificación,
con un ícono, un título y
un texto

StatusBar - Actividad guiada

- Lanzar la notificación desde la actividad

```
import android.app.NotificationManager;  
import android.content.Context;  
import android.support.v4.app.NotificationCompat;  
import android.view.View;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    public static final int NOTIFICACION_ID = 1;
```

```
    // ...
```

```
    public void mostrarNotificacion(View view){
```

```
        NotificationCompat.Builder mBuilder =
```

```
            new NotificationCompat.Builder(this)
```

```
                .setSmallIcon(R.drawable.notificacion)
```

```
                .setContentTitle("Mi notificación")
```

```
                .setContentText("Hola Curso!");
```

```
        NotificationManager mNotificationManager =
```

```
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
```

```
        mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
```

```
    }
```

```
}
```

Se lanza la notificación

StatusBar - Actividad guiada

- Lanzar la notificación desde la actividad

```
import android.app.NotificationManager;
import android.content.Context;
import android.support.v4.app.NotificationCompat;
import android.view.View;

public class MainActivity extends AppCompatActivity {

    public static final int NOTIFICACION_ID = 1;

    // ...

    public void mostrarNotificacion(View view){
        NotificationCompat.Builder mBuilder =
            new NotificationCompat.Builder(this)
                .setSmallIcon(R.drawable.notificacion)
                .setContentTitle("Mi notificación")
                .setContentText("Hola Curso!");

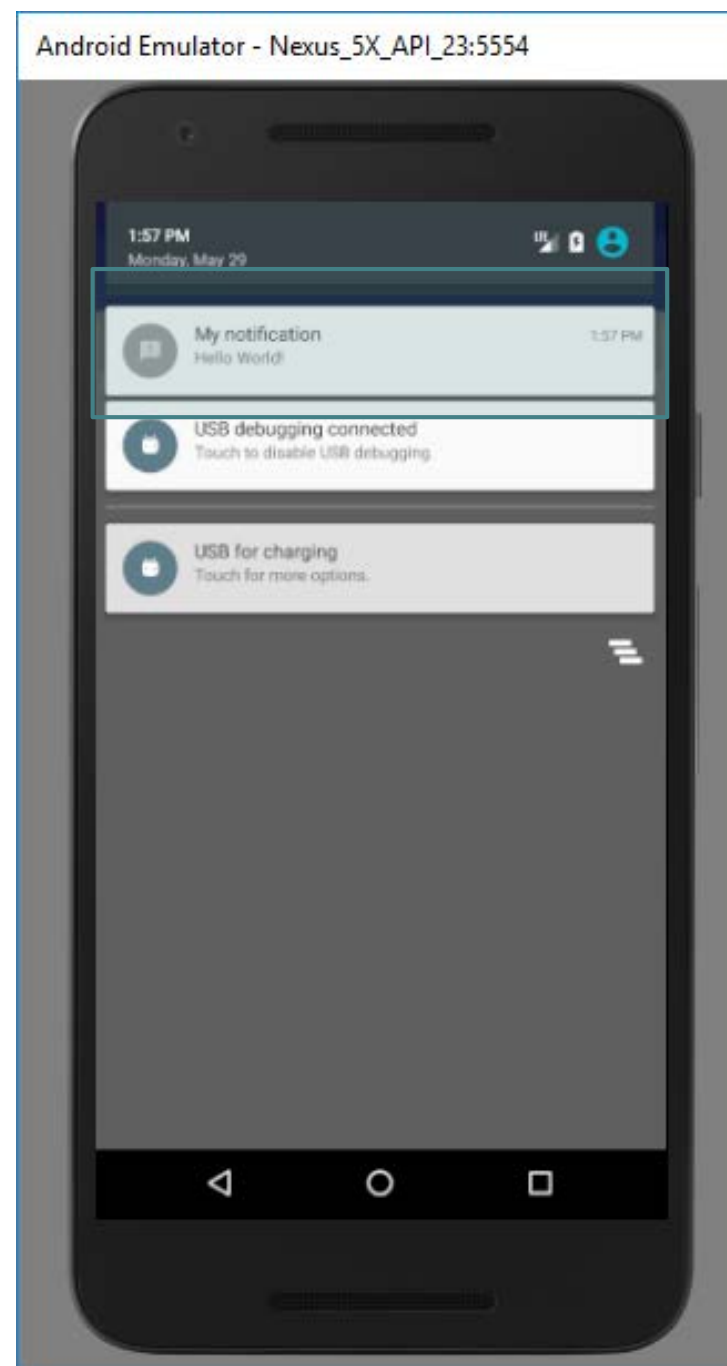
        NotificationManager mNotificationManager =
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
    }
}
```

Permitirá modificar/reemplazar luego la notificación lanzada

StatusBar - Actividad guiada

- Probar la aplicación en el emulador



StatusBar - Actividad guiada

- Dar comportamiento a la notificación

```
import android.app.PendingIntent;
import android.content.Intent;

public class MainActivity extends AppCompatActivity {

    // ...

    public void mostrarNotificacion(View view){
        NotificationCompat.Builder mBuilder =
            new NotificationCompat.Builder(this)
                .setSmallIcon(R.drawable.notificacion)
                .setContentTitle("Mi notificación")
                .setContentText("Hola curso!");

        Intent intent = new Intent(this, MainActivity.class );

        PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);

        mBuilder.setContentIntent(pendingIntent);

        NotificationManager mNotificationManager =
            (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);

        mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
    }
}
```


StatusBar - Actividad guiada

- Dar comportamiento a la notificación

```
import android.app.PendingIntent;
import android.content.Intent;

public class MainActivity extends AppCompatActivity {
```

```
// ...
```

```
public void mostrarNotificacion(View view){
    NotificationCompat.Builder mBuilder =
        new NotificationCompat.Builder(this)
            .setSmallIcon(R.drawable.notificacion)
            .setContentTitle("Mi notificación")
            .setContentText("Hola curso!");
```

```
Intent intent = new Intent(this, MainActivity.class );

PendingIntent pendingIntent = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);

mBuilder.setContentIntent(pendingIntent);
```

```
NotificationManager mNotificationManager =
    (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE);
```

```
mNotificationManager.notify(NOTIFICACION_ID, mBuilder.build());
```

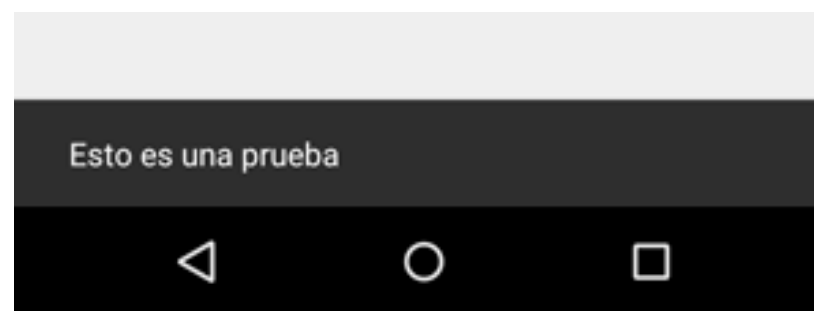
```
}
```

```
}
```

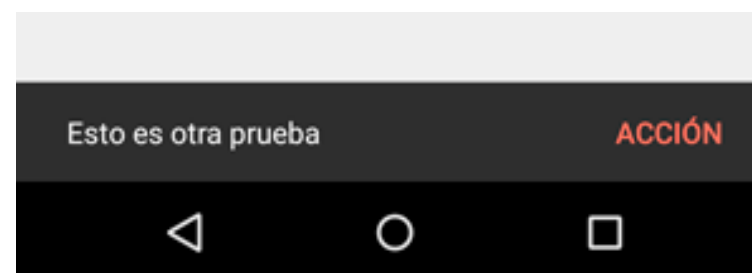
Se debe asociar un PendingIntent.
Representa la intención a futuro de ejecutar un Intent concreto

Snackbar

- Permite mostrar información al usuario, de forma similar a un Toast.
- La notificación desaparece luego de un período de tiempo, similar al Toast.

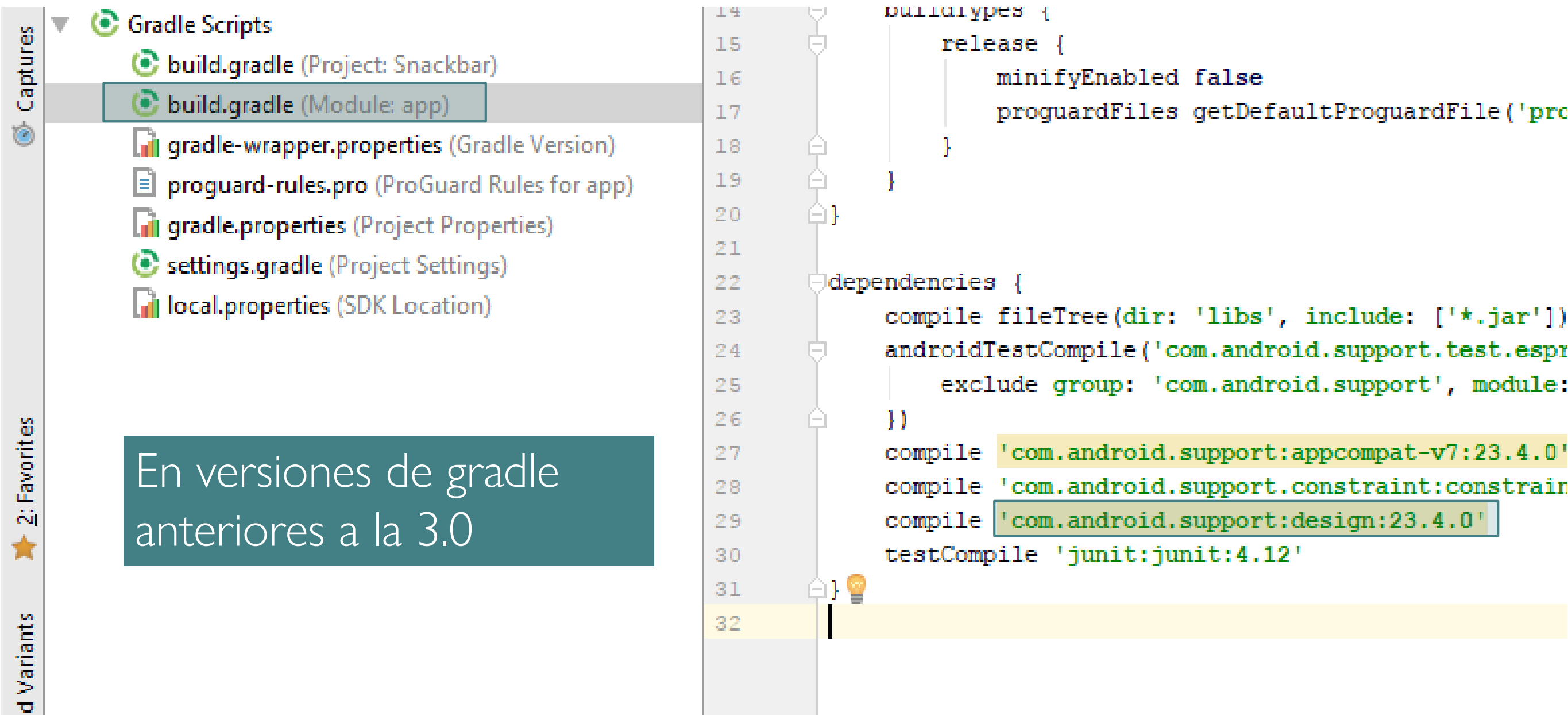


- Requiere ser mostrada dentro de una actividad.
- Da la posibilidad de asociarle una acción
- Puede ser descartada haciendo swipe.



Snackbar - Actividad guiada

- Crear un nuevo proyecto desde Android Studio
- Incluir la **com.android.support:design:23.4.0** desde el gradle



The screenshot shows the Android Studio interface. On the left, the 'Gradle Scripts' panel is open, displaying a list of files for the 'app' module. The file 'build.gradle (Module: app)' is selected and highlighted. Below this panel, a dark teal box contains the text: 'En versiones de gradle anteriores a la 3.0'. The main editor area shows the content of the selected 'build.gradle' file. The file contains Gradle build script code. The 'dependencies' block is visible, showing the inclusion of 'com.android.support:design:23.4.0' for the 'compile' configuration. The line numbers 14 through 32 are visible on the left side of the code editor.

Gradle Scripts

- build.gradle (Project: Snackbar)
- build.gradle (Module: app)
- gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for app)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

En versiones de gradle anteriores a la 3.0

```
14 buildTypes {
15     release {
16         minifyEnabled false
17         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
18             'proguard-rules.pro'
19     }
20 }
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
25         exclude group: 'com.android.support', module: 'support-annotations'
26     })
27     compile 'com.android.support:appcompat-v7:23.4.0'
28     compile 'com.android.support.constraint:constraint-layout:1.0.2'
29     compile 'com.android.support:design:23.4.0'
30     testCompile 'junit:junit:4.12'
31 }
32
```

Snackbar - Actividad guiada

- Crear un nuevo proyecto desde Android Studio
- Incluir la **com.android.support:design:23.4.0** desde el gradle

En versiones de gradle anteriores a la 3.0

Usar **compile**

Importante: Estos números deben coincidir

```
14 buildTypes {
15     release {
16         minifyEnabled false
17         proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
18     }
19 }
20
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
25         exclude group: 'com.android.support', module: 'support-annotations'
26     })
27     compile 'com.android.support:appcompat-v7:23.4.0'
28     compile 'com.android.support.constraint:constraint-layout:1.0.2'
29     compile 'com.android.support:design:23.4.0'
30     testCompile 'junit:junit:4.12'
31 }
32
```

Snackbar - Actividad guiada

Android

app

Gradle Scripts

- build.gradle (Project: SnackBar3)
- build.gradle (Module: app)
- gradle-wrapper.properties (Gradle Version)
- proguard-rules.pro (ProGuard Rules for app)
- gradle.properties (Project Properties)
- settings.gradle (Project Settings)
- local.properties (SDK Location)

```
17 }
18 }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:26.1.0'
24
25     implementation 'com.android.support.constraint:constraint-layout:1.0.2'
26     testImplementation 'junit:junit:4.12'
27     androidTestImplementation 'com.android.support.test:runner:1.0.1'
28     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.1'
29     implementation 'com.android.support:design:26.1.0'
30 }
```

A partir de la versión 3.0 de gradle

Estos números deben coincidir

Usar **implementation**

Build Sync

Build: completed successfully at 22/5/2018 15:26

- Run build C:\Users\leona\AndroidStudioProjects\SnackBar3
 - Load build
 - Configure build
 - Calculate task graph
 - Run tasks

Snackbar - Actividad guiada

- Definir un botón en el layout de la actividad y asociarle un manejador al click

```
<LinearLayout
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:android="http://schemas.android.com/apk/res/android">
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Mostrar notificación"
        android:onClick="mostrarNotificacion" />
</LinearLayout>
```

Snackbar - Actividad guiada

- Lanzar la notificación desde la actividad

```
import android.view.View;
import android.support.design.widget.Snackbar;

public class MainActivity extends AppCompatActivity {

    public void mostrarNotificacion(View view){
        Snackbar.make(view, "Esto es una prueba", Snackbar.LENGTH_LONG)
            .show();
    }
}
```

Snackbar - Actividad guiada

- Probar la aplicación en el emulador



Snackbar - Actividad guiada

- Incorporar un acción al Snackbar

```
import android.view.View;
import android.support.design.widget.Snackbar;

public class MainActivity extends AppCompatActivity {

    public void mostrarNotificacion(View view){
        Snackbar.make(view, "Esto es una prueba", Snackbar.LENGTH_LONG)
            .setAction("Acción", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Log.i("Snackbar", "Se invocó la acción");
                }
            })
            .show();
    }
}
```

Snackbar - Actividad guiada

- Probar la aplicación en el emulador



Snackbar - Actividad guiada

- Probar la aplicación en el emulador



Por defecto el color de la acción es el “accent” del tema de la aplicación

Snackbar - Actividad guiada

- Cambiar el color de la acción

```
import android.view.View;
import android.support.design.widget.Snackbar;

public class MainActivity extends AppCompatActivity {

    public void mostrarNotificacion(View view){
        Snackbar.make(view, "Esto es una prueba", Snackbar.LENGTH_LONG)
            .setActionTextColor(Color.GREEN)
            .setAction("Acción", new View.OnClickListener() {
                @Override
                public void onClick(View view) {
                    Log.i("Snackbar", "Se invocó la acción");
                }
            })
            .show();
    }
}
```

Snackbar - Actividad guiada

- Es posible incorporar la animación de descartar la notificación arrastrándola hacia la derecha
- Solo hay que definir un nuevo elemento raíz en el layout de la actividad

```
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <LinearLayout
        android:layout_height="match_parent"
        android:layout_width="match_parent">
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Mostrar notificación"
            android:onClick="mostrarNotificacion" />
        </LinearLayout>
    </android.support.design.widget.CoordinatorLayout>
```

Snackbar - Actividad guiada

- Probar la aplicación en el emulador

