

Material Design

Dra. MPuerto Paule Ruiz

Qué es?

- Es un lenguaje de diseño desarrollado por Google y anunciado en la conferencia de Google I/O en Junio del 2014
- Diseño basado en coordinator-layouts, animaciones y transiciones, y con efectos de luces y sombras
- Objetivo: Definir patrones de interacción con el usuario

Especificaciones de diseño

- Revisión de especificaciones de diseño:
<http://www.google.com/design/spec/material-design/introduction.html>

Material Theme

1. Tres tipos de Material Theme:

- `@android:style/Theme.Material` (dark version)
- `@android:style/Theme.Material.Light` (light version)
- `@android:style/Theme.Material.Light.DarkActionBar`

2. Posibilidad de customizar la paleta de colores

(<http://developer.android.com/training/material/theme.html>)

3. Customizar el Status Bar:

<http://developer.android.com/training/material/theme.html>

Diseño de Layouts

- Cambio de filosofía. Mismos componentes pero se añaden atributos como:
 - Elevación de las vistas: Utilizar la elevación como un elemento que ayuda a entender la relativa importancia de cada objeto y centra la atención en la tarea en cuestión
 - Nueva componente Z a la x e y

Animaciones customizadas

- Enriquecen la interacción del usuario con la app
- Elementos a los que se aplica: botones y transiciones entre las activitys

Librería

- Añadir en **build.gradle** (Module app)
compile '**com.android.support:design:26.+**'
 - Instalar si es necesario
- Esta librería permite trabajar con componentes como FAB, TextInputLayout, ...
- Algunos componentes está disponibles en CustomView
- Permite la compatibilidad hacia atrás (API menores a la 21 soportan Material Design)

FAB

- Disponible en CustomView

```
<android.support.design.widget.FloatingActionButton  
    android:id="@+id/fab"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@android:drawable/ic_input_add"  
    android:onClick="lanzaSegundaActividad" />
```


TextInputLayout

- `<android.support.design.widget.TextInputLayout`
 `android:id="@+id/Usuario"`
 `>`
 `<EditText`
 `android:inputType="textEmailAddress"`
 `/>`
 `</android.support.design.widget.TextInputLayout`
 `out>`

TextInputLayout (cont)

```
t1=(TextInputLayout)findViewById (R.id.Usuario);  
t2=(TextInputLayout)findViewById (R.id.password);  
t3=(TextInputLayout)findViewById (R.id.password2);
```

Mensaje

```
Snackbar.make(view, "Contraseña Incorrecta",  
Snackbar.LENGTH_LONG)  
.setAction("Action", null).show();
```

Ejercicio

- Verificar Clave: Al pulsar el botón de aceptar la aplicación nos dirá si hemos introducido correctamente la contraseña

