Layout e Interfaces de usuarios

Como se acomodan los componentes

https://goo.gl/ZazdmE

AD HOC TALENT

Layouts e Interfaces de usuario

- XML de una interfaz de usuario
- Interfaces de usuario desde el IDE
- La clase View
- Tipos de Layouts
- Introducción al material Design

AD HOC TALENT

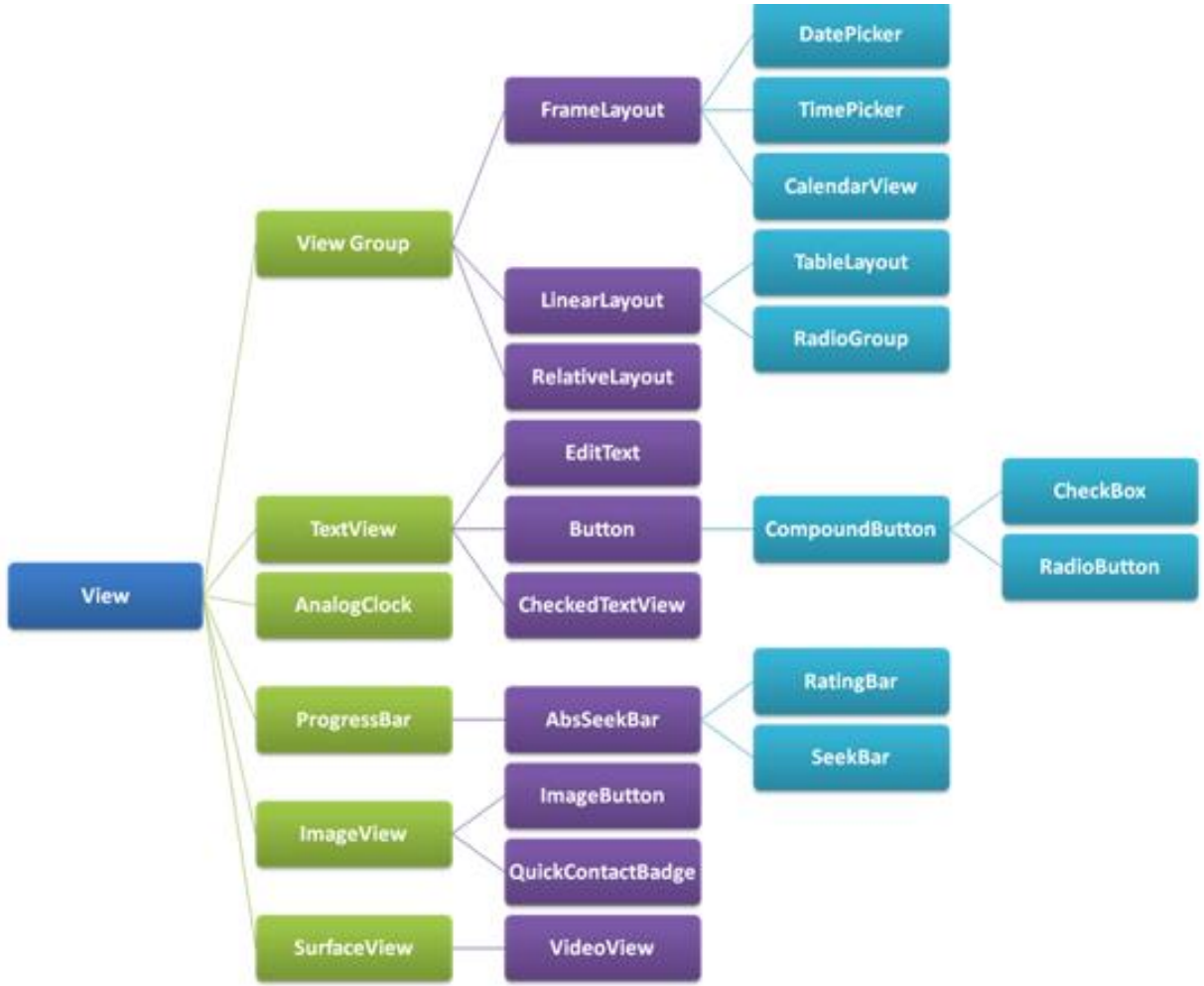
Objetivo

- Conocer el uso de los XML en un diseño
- Componente de un diseño
- Conocer el concepto de Material design

AD HOC TALENT

XML de una interfaz de usuario

- Los componentes pueden crearse a través de código como si fueran objetos de Java, sin embargo Android implementa una forma en la cual el diseño puede ser trazado a través de XML y son interpretadas como diseños de pantallas.



AD HOC TALENT

Clase View

- Es probable poder crear interfaces directamente desde código de Java pero realmente no es lo adecuado. Al los componentes gráficos se les puede instanciar directamente ya que interno los xml de las vistas que se crean (layouts) son también clases que dependen de View

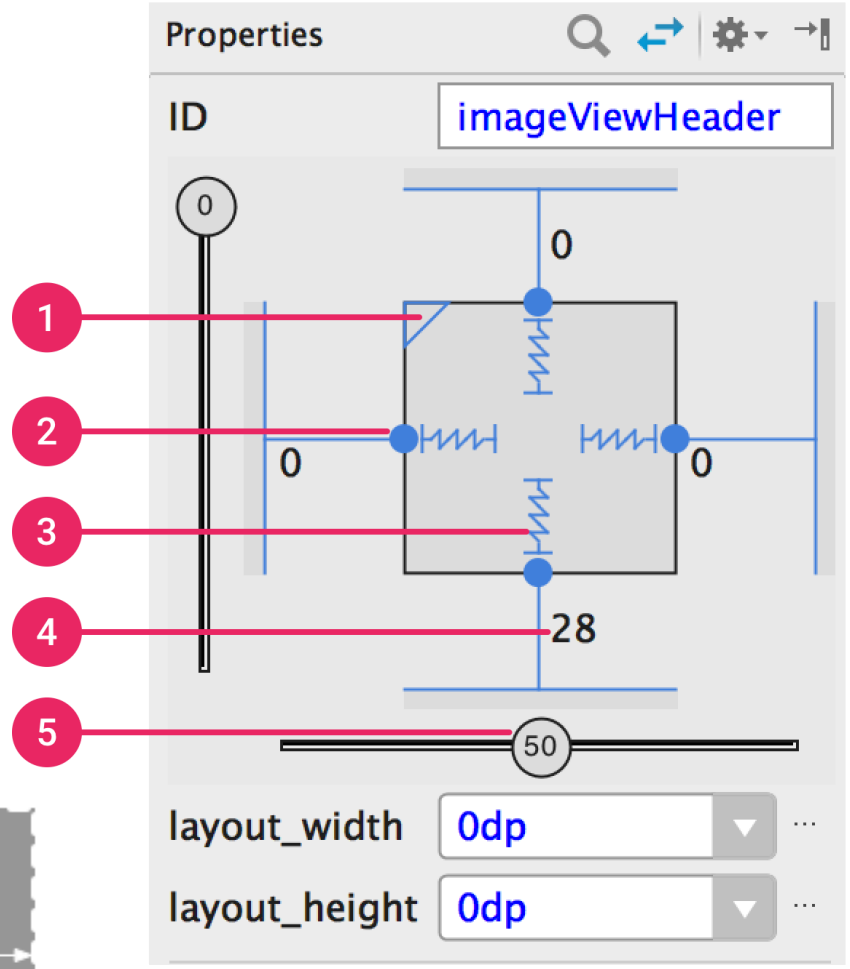
```
public void onCreate(Bundle savedInstanceState)
{
	super.onCreate(savedInstanceState);
	TextView texto = new TextView(this);
	texto.setText("Hello, Android");
	setContentView(texto);
}
```

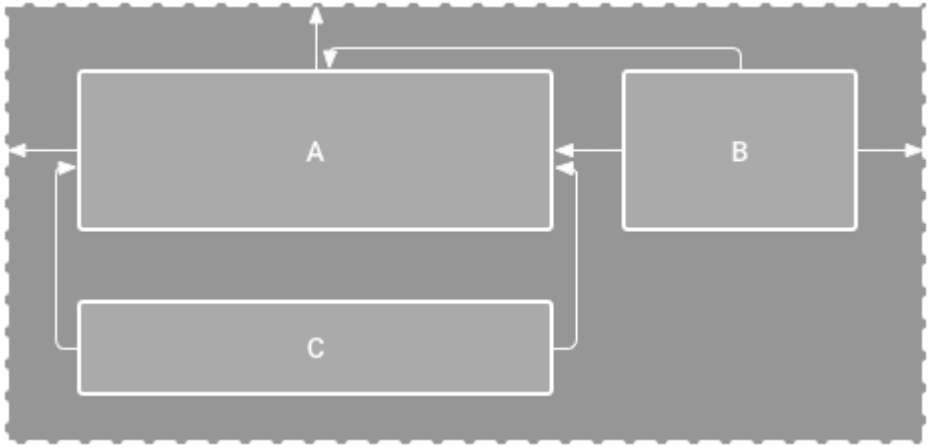
Tipos de Layouts

- Si queremos combinar varios elementos de tipo vista tendremos que utiliza un objeto de tipo Layout. Un Layout es un contenedor (ViewGroup) de una o más vistas y controla su comportamiento y posición. Hay que destacar que un Layout puede contener a otro Layout y que es un descendiente de la clase View.
  - **ConstraintLayout**
  - **GridLayout**
  - **FrameLayout**
  - **LinearLayout**
  - **RelativeLayout**
  - **TableLayout**
  - **CoordinatorLayout**
  - **TabLayout**
  - **TextInputLayout**

ConstraintLayout

- Acomodo de los componentes por medio de objetos, lados de la vista padre de una forma muy similar a RelativeLayout

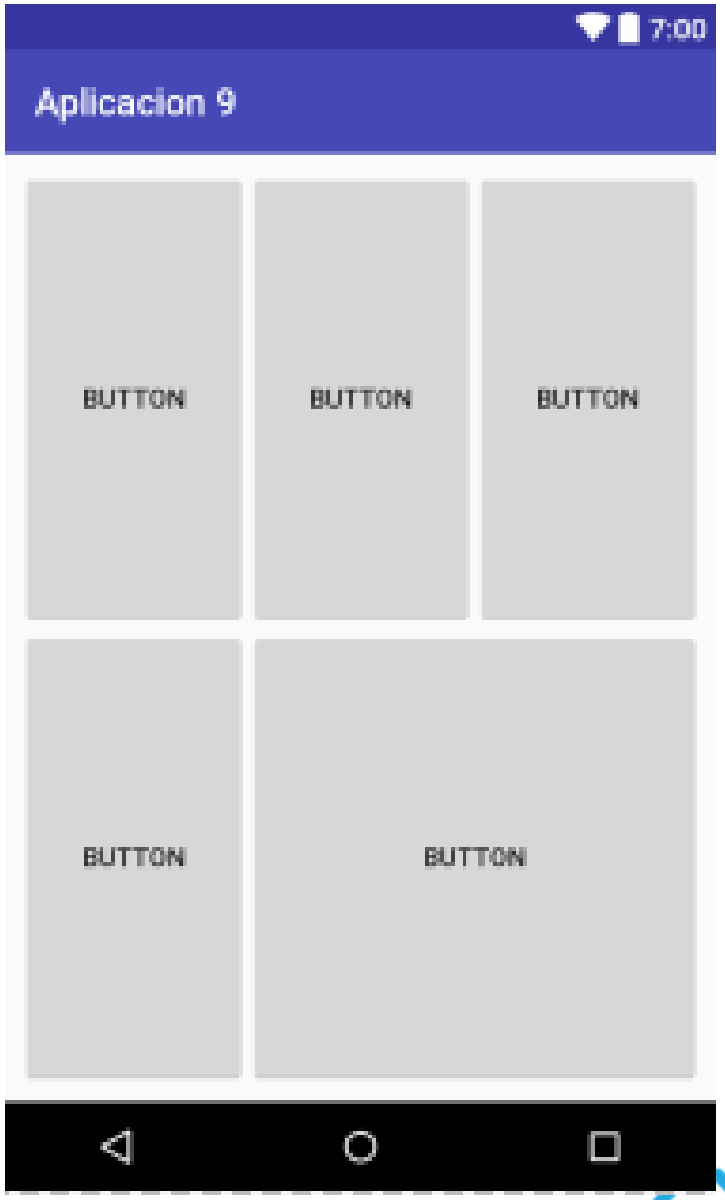




AD HOC TALENT

GridLayout

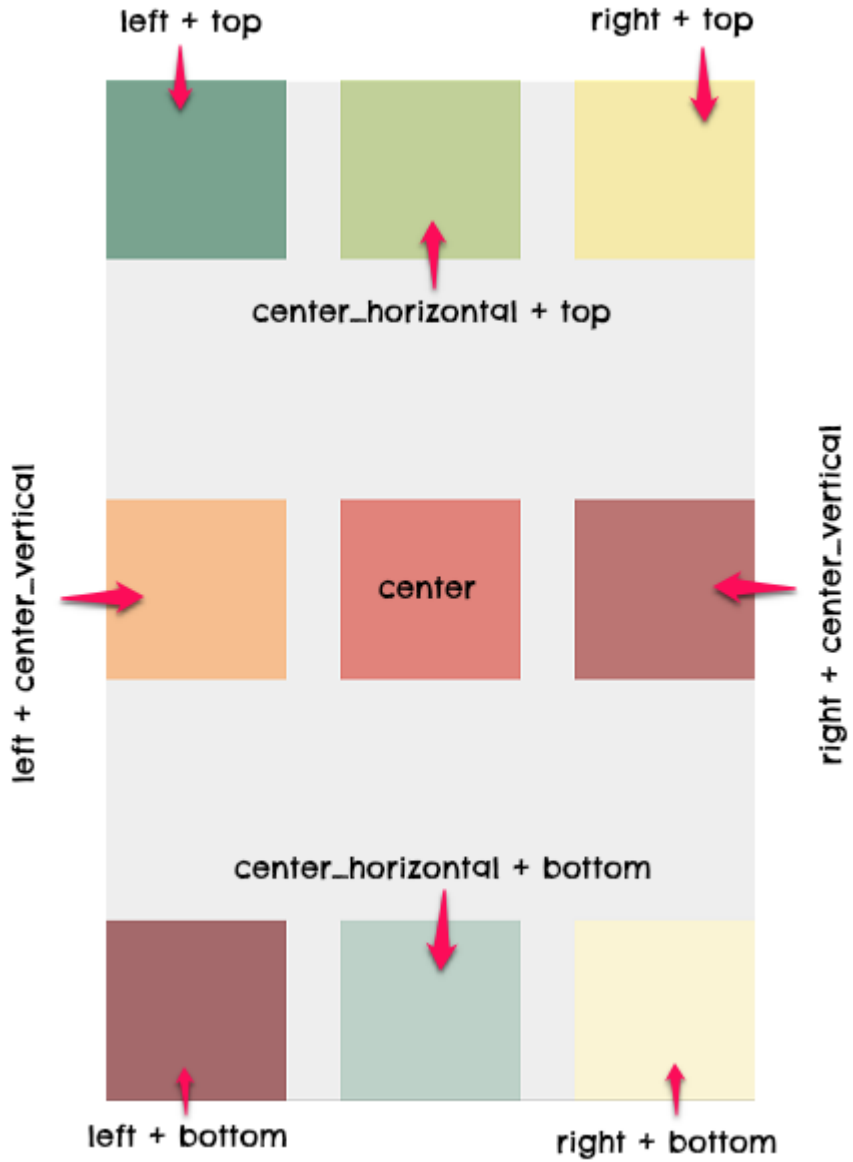
- Parecido al Tablelayout permite organizar tus componentes rejillas o cuadricula en la pantalla. Los componentes se acomodan según la orientación y cada componente se acomoda en una celda



AD HOC TALENT

FrameLayout

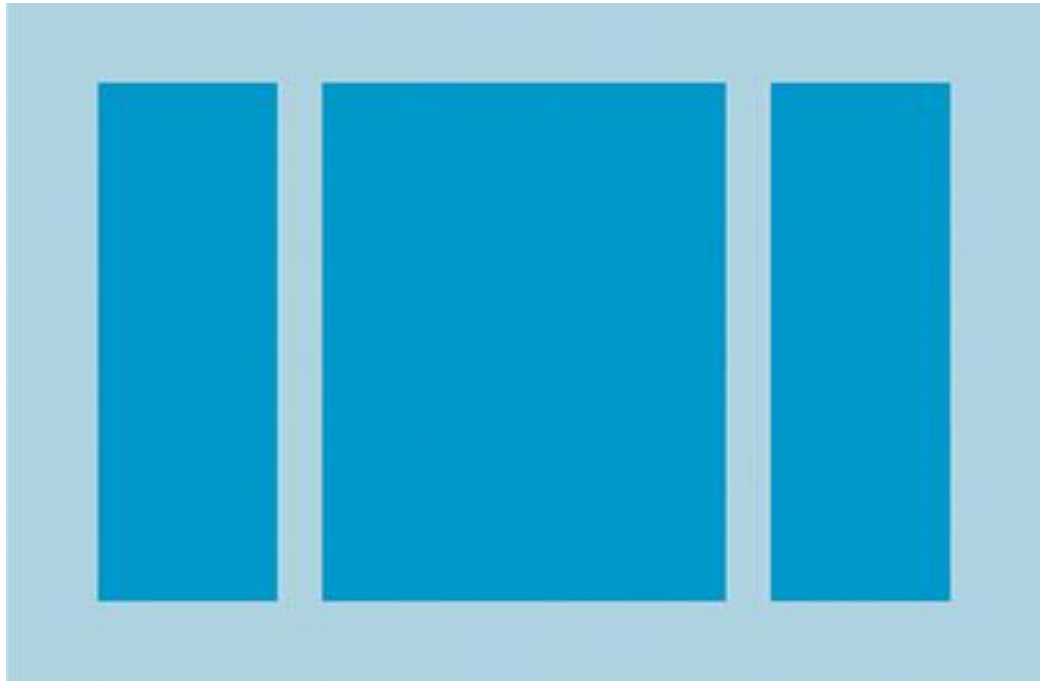
- Framelayout es una ViewGroup que forma una parte de la pantalla que puede ser utilizada para poner o colocar un fragmento aunque se puedo anidar mas componente, en este caso por medio de programación puede cambiar.
  - Utiliza una propiedad de Layout_gravity para acomodar a los componentes



AD HOC TALENT

LinearLayout

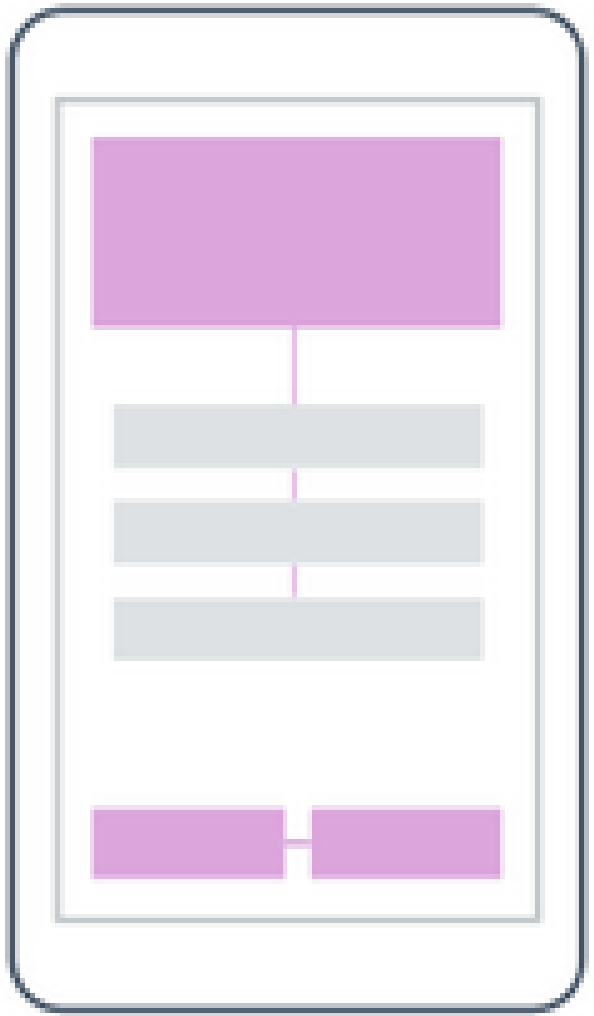
- Es un ViewGroup que distribuye los componente uno tras otro según la orientación que se vea reflejada, en este caso cada componente lo apila y solo permite un solo componente.



AD HOC TALENT

RelativeLayout

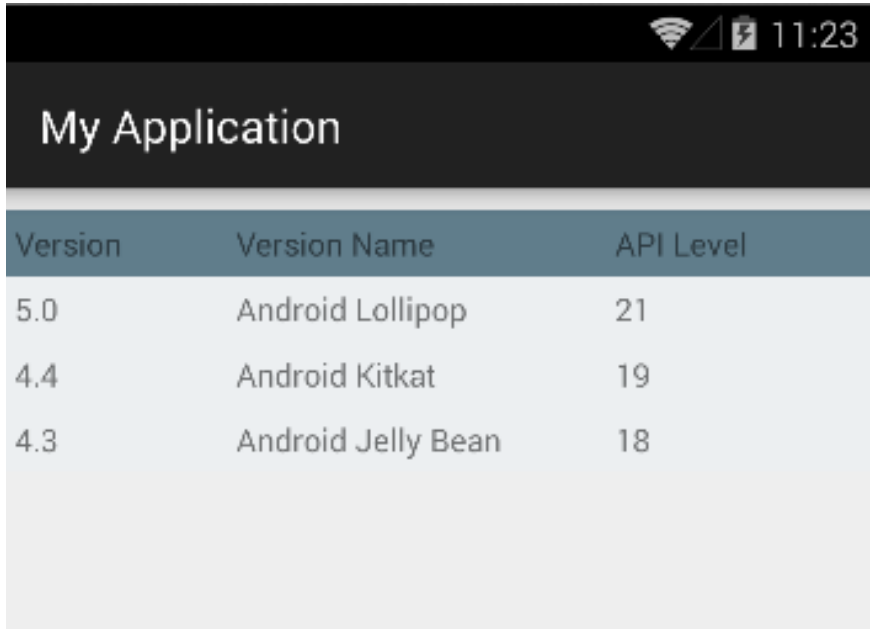
- Te permite especificar la ubicación de los objetos a partir de otros o en función de ellos, regularmente tienen características como ponlo debajo de, o a la derecha de ..



AD HOC TALENT

TableLayout

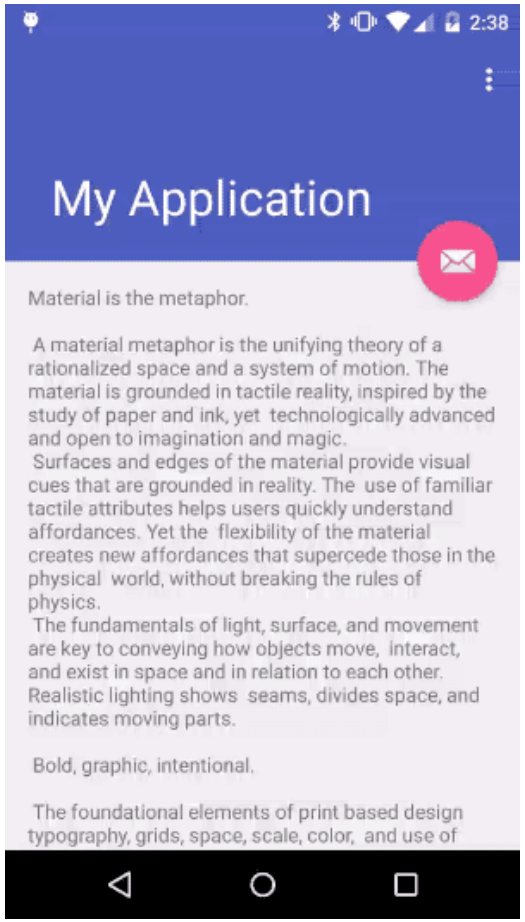
- Organiza tu vista en filas y columnas, muy similar a como trabaja GridLayout, a diferencia que en esta vista se tiene que especificar los componentes que pertenecen a las filas mediante un TableRow



AD HOC TALENT

CoordinatorLayout (MD)

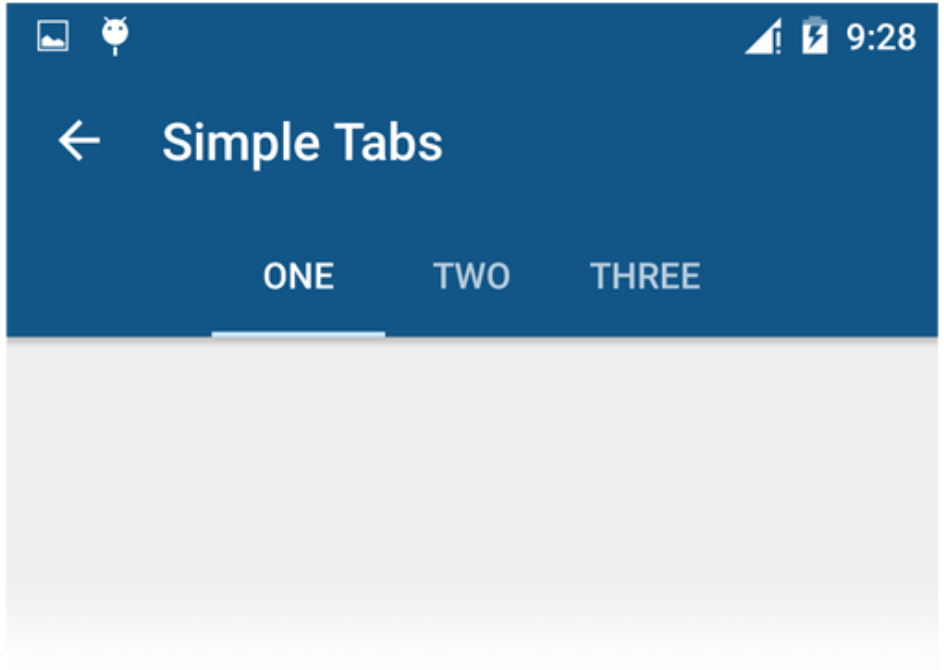
- El CoordinatorLayout pone enfasis, en el comportamiento que tenga cada uno de los elementos, su interacción con algunas de las actividades que se tienen en la interfaz como lo puede ser el SCROLL



AD HOC TALENT

TabLayout

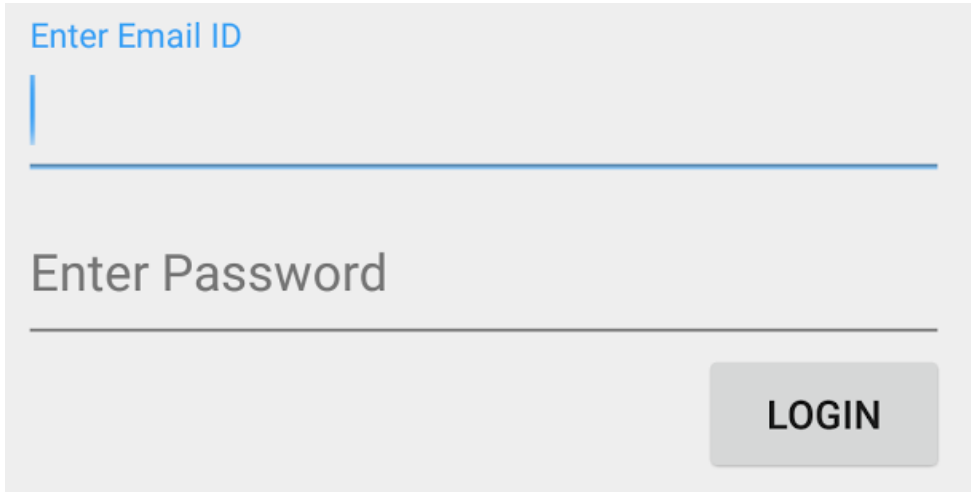
- TabLayout es una vista que permite separar diferentes fragmentos basados en pestañas, es muy comada cuando tienes que mostrar mucha información.



AD HOC TALENT

TextInputLayout (MD)

- Este tiene un manejo del comportamiento en un cuadro de texto, el cual permite animar el texto de ayuda hint, de manera que no desaparesca. Ademas de algunas otras propiedades



AD HOC TALENT

Algunos otros layouts

- **ScrollView**: Visualiza una columna de elementos; cuando estos no caben en pantalla se permite un deslizamiento vertical.
- **HorizontalScrollView:** Visualiza una fila de elementos; cuando estos no caben en pantalla se permite un deslizamiento horizontal.
- **TabHost:** Proporciona una lista de ventanas seleccionables por medio de etiquetas que pueden ser pulsadas por el usuario para seleccionar la ventana que desea visualizar. Se estudia al final del capítulo.
- **ListView:** Visualiza una lista deslizable verticalmente de varios elementos. Su utilización es algo compleja. Se verá un ejemplo en el capítulo siguiente.
- **GridView**: Visualiza una cuadrícula deslizable de varias filas y varias columnas.
- **ViewFlipper:** Permite visualizar una lista de elementos de forma que se visualice uno cada vez. Puede ser utilizado para intercambiar los elementos cada cierto intervalo de tiempo.

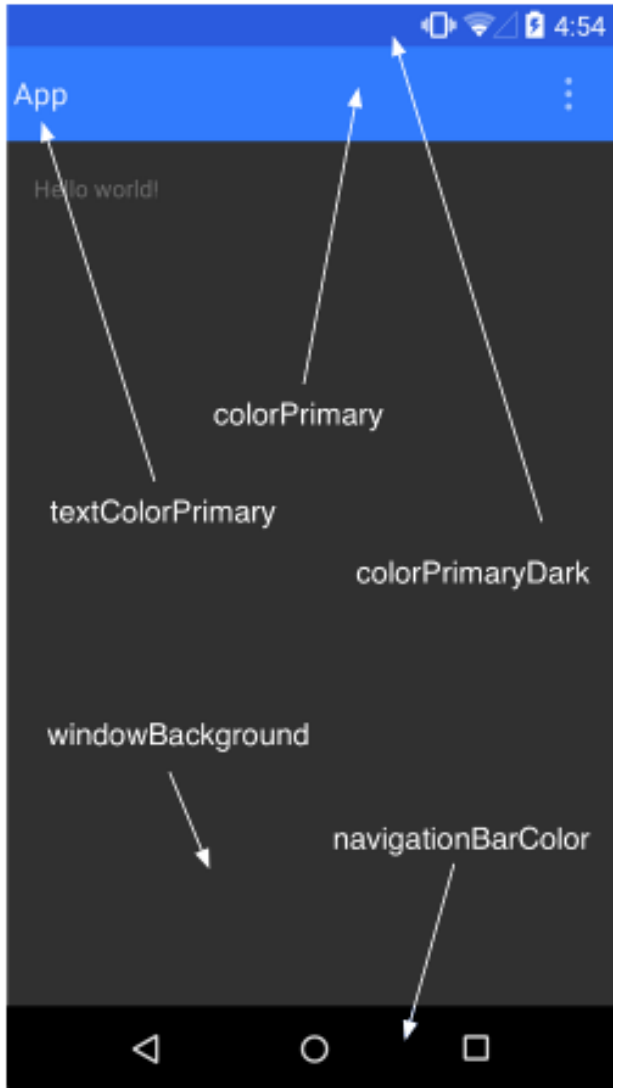
Material Design

- Es una apuesta que hace android para el diseño de sus aplicaciones donde tiene la filosofía de ver las interfaces de una manera más real.
  - colores planos
  - colores vivos
  - animaciones diferentes
- La principal caracteristica de este concepto de diseño es que viene acompañado por una libreria de diseño, la cual permite tratar a los componentes como objetos.

AD HOC TALENT

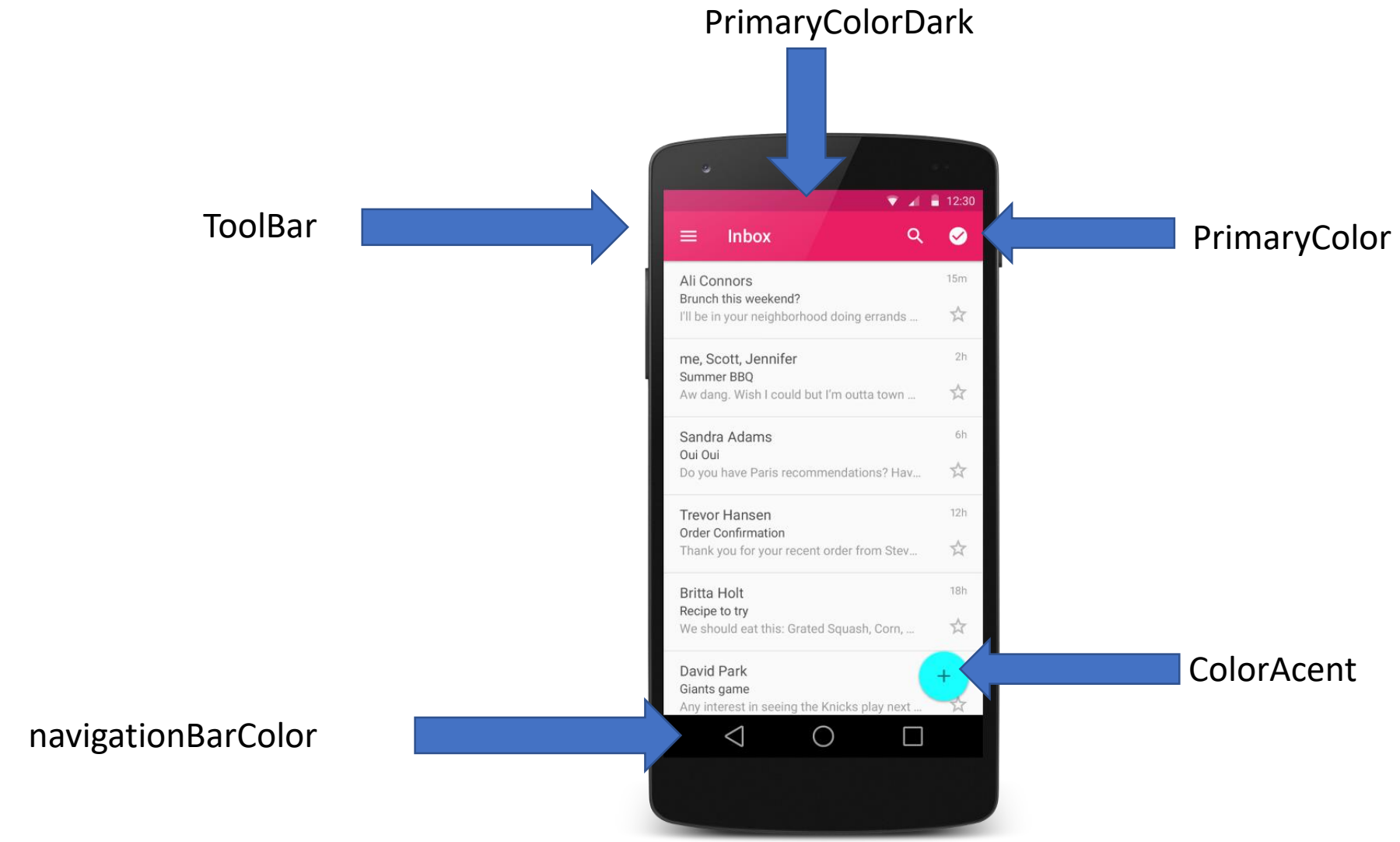
Esquema de colores

- Material refleja simplicidad en los colores para esto maneja solo tres colores principales aunque puedes agregar mas
  - Sitios te ayudan a escoger un esquema de colore correcto.
  - https://www.materialpalette.com/



AD HOC TALENT

Partes



AD HOC TALENT

ToolBar

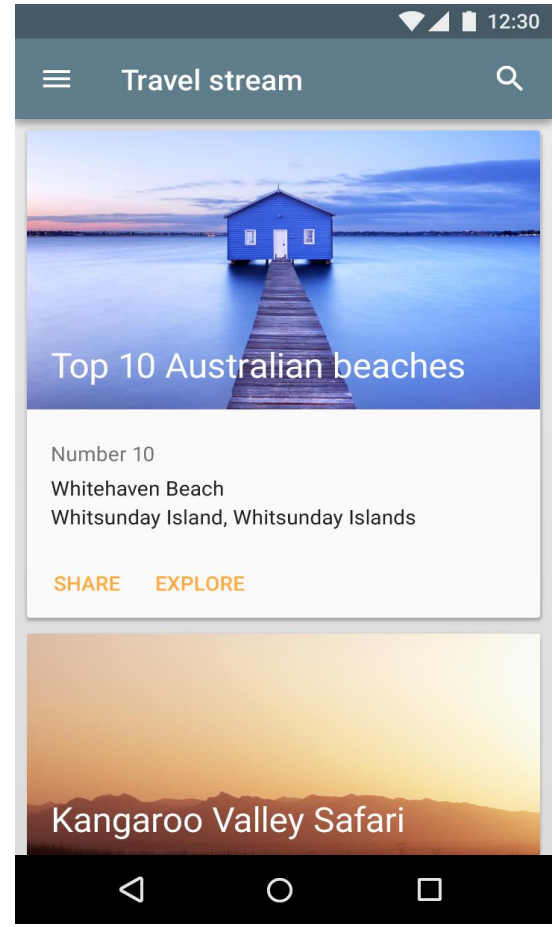
```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity);

    Toolbar toolbar = (Toolbar)
findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
}
```

```
<android.support.v7.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:elevation="4dp"
    app:theme="@style/ToolbarTheme" />
```

AD HOC TALENT

CardView

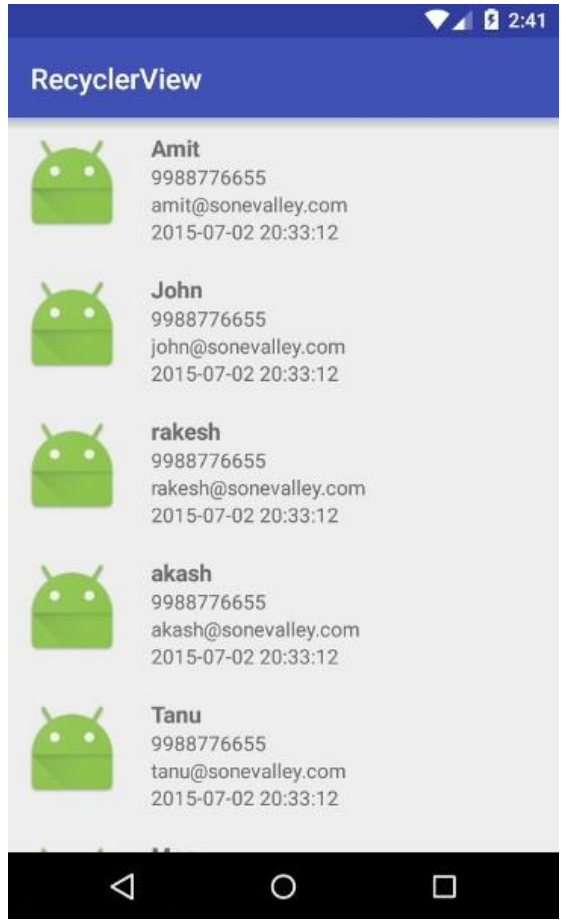
```
<android.support.v7.widget.CardView
    android:layout_width="172dp"
    android:layout_height="192dp"
    android:orientation="vertical"
    app:cardBackgroundColor="@color/primary_dark"
    app:cardCornerRadius="4dp"
    app:cardElevation="4dp">
```



AD HOC TALENT

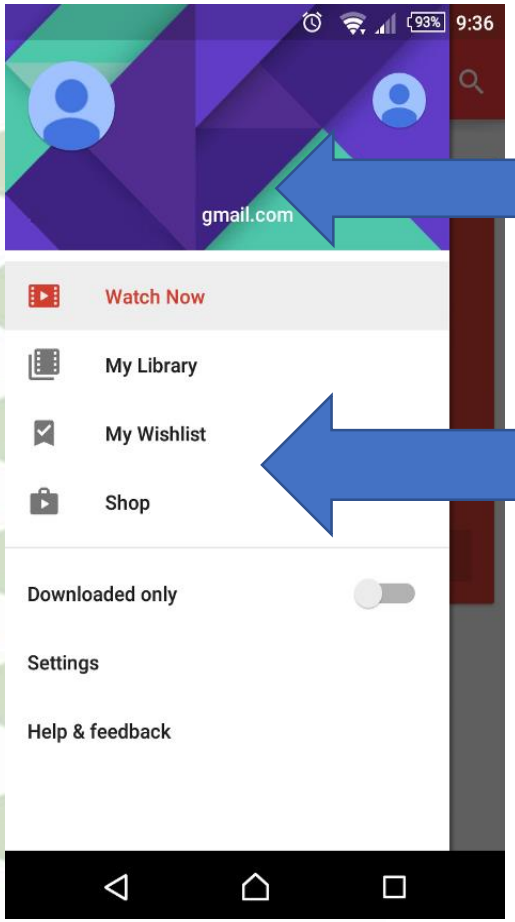
RecyclerView

- LayoutManager
- RecyclerView.Adapter
- ItemAnimator
- ItemDecorator



AD HOC TALENT

NavigationDrawer



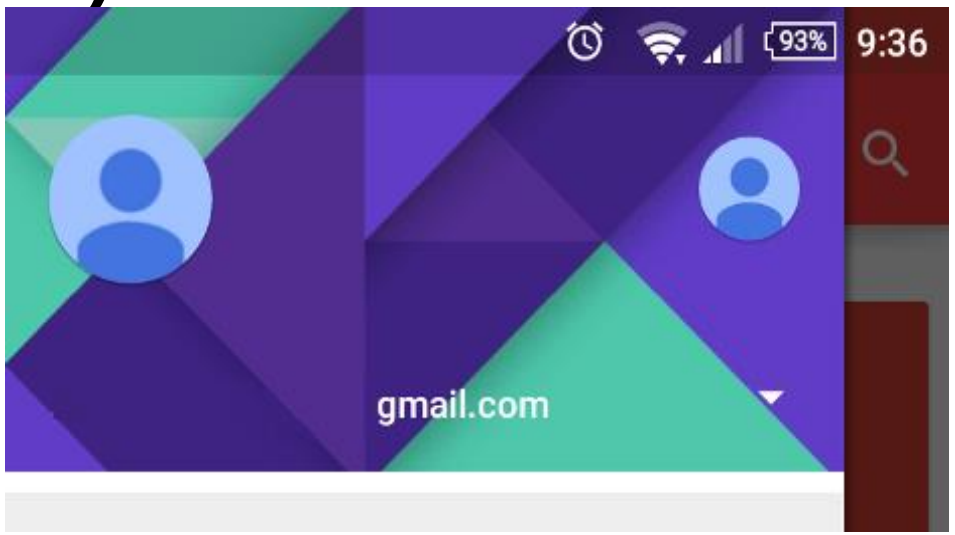
Header

Items

- <android.support.v4.widget.DrawerLayout
- android:layout_width="match_parent"
- android:layout_height="match_parent"
- android:fitsSystemWindows="true">
- <android.support.design.widget.NavigationView
- android:layout_width="wrap_content"
- android:layout_height="match_parent"
- android:layout_gravity="start"
- app:headerLayout="@layout/drawer_header"
- app:menu="@menu/drawer"/>
- </android.support.v4.widget.DrawerLayout>

AD HOC TALENT

NavigationDrawer (Header)

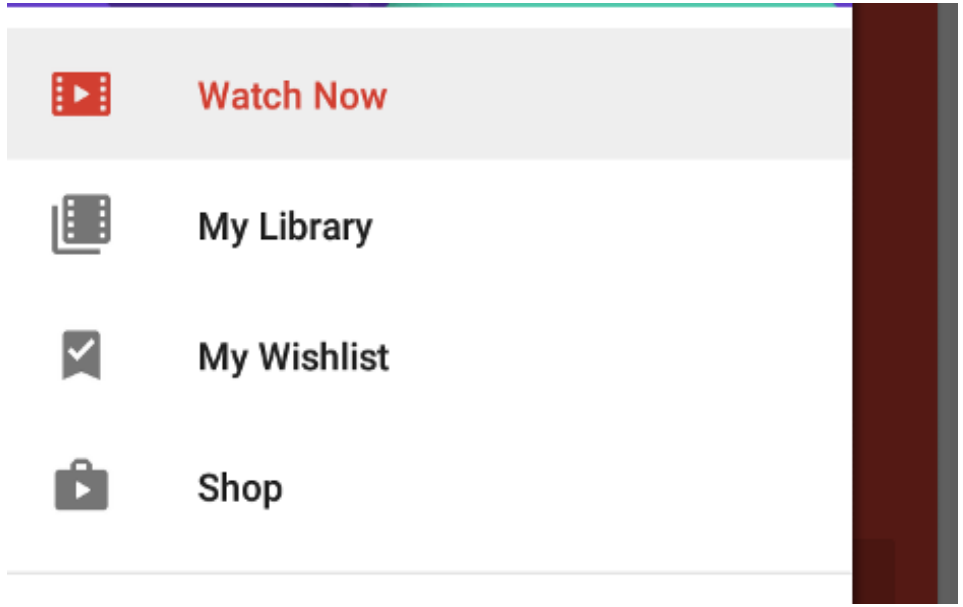



```
<FrameLayout
   android:layout_width="match_parent"
 android:layout_height="wrap_content">
   <ImageView ... />
   <de.hdodenhof.circleimageview.CircleImageView ...
/>
   <LinearLayout ...>
      <TextView .../>
      <TextView .../>
   </LinearLayout>
</FrameLayout>
```

AD HOC TALENT

NavigationDrawer (Menu)

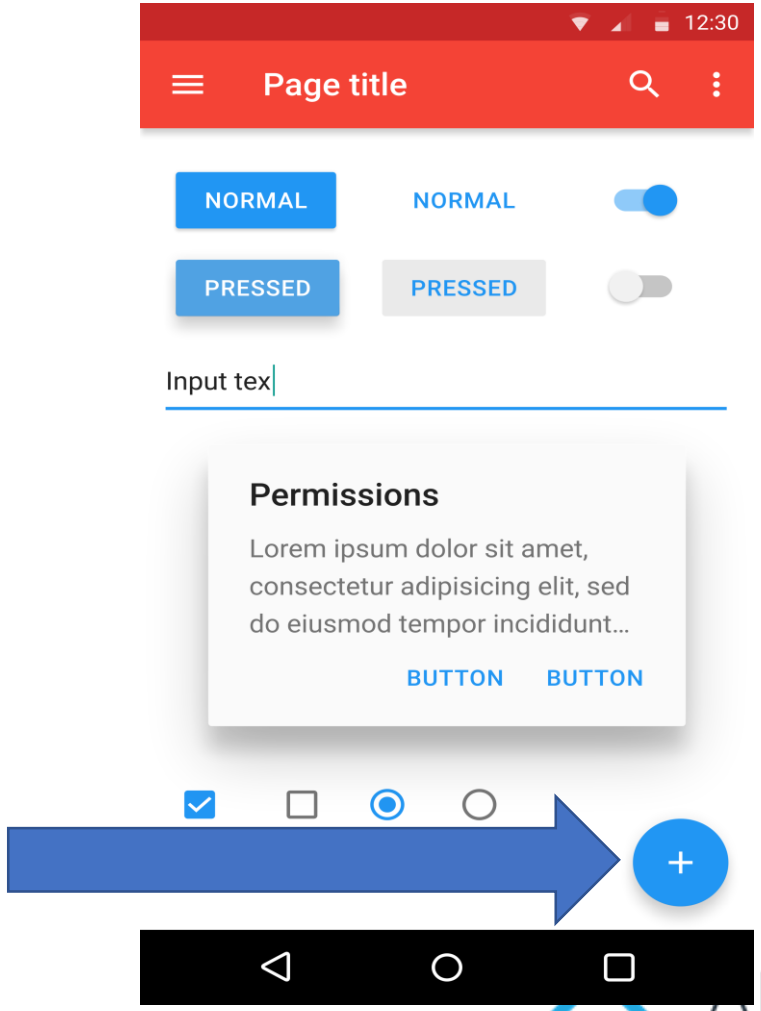
```
<menu>
    <group android:checkableBehavior="single">
        <item android:checked="true"
            android:id="@+id/item_nav_explore"
android:icon="@drawable/ic_explore_black_24dp"
            android:title="@string/menu_schedule" />
        <item  android:id="@+id/item_nav_people"
android:icon="@drawable/ic_people_black_24dp"
            android:title="@string/menu_guests" />
        <item android:id="@+id/item_nav_map"
            android:icon="@drawable/ic_map_black_24dp"
            android:title="@string/menu_map" />
        <item android:id="@+id/item_nav_info"
android:icon="@drawable/ic_info_outline_black_24dp"
            android:title="@string/menu_about" />
    </group>
</menu>
```



AD HOC TALENT

FloatingActionButton

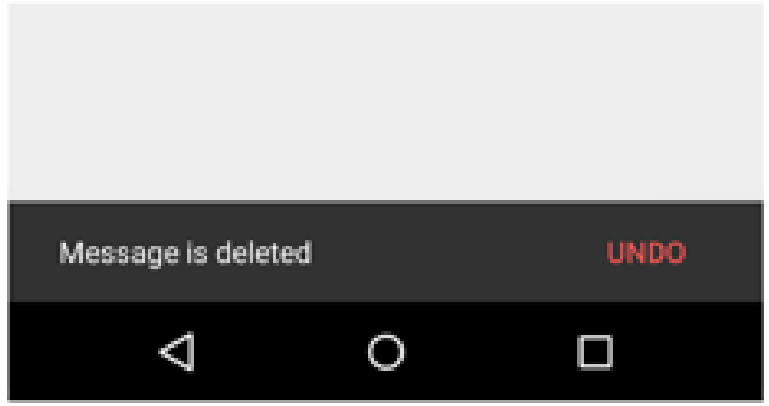
```
<android.support.design.widget.FloatingActionButton

android:id="@+id/floatingActionButton"

android:layout_width="wrap_content"

android:layout_height="wrap_content"
    android:layout_margin="16dp"

android:layout_gravity="end|bottom"
android:src="@drawable/ic_search_white_24dp"
    app:borderWidth="0dp" />
```



AD HOC TALENT

SnackBar

```
Snackbar.make(itemView,
R.string.message_selected_session,
Snackbar.LENGTH_LONG)
        .setAction(R.string.ok, new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                //Do something
            }
        }).show();
```



AD HOC TALENT

Desarrollo de aplicaciones en Android

XML de una interfaz de usuario
Interfaces de usuario desde el IDE
La clase View
Tipos de Layouts
Introducción al material Design

AD HOC TALENT

Referencias

- https://developer.android.com/training/constraint-layout/index.html
- https://developer.android.com/reference/android/widget/FrameLayout.html
- https://developer.android.com/training/material/index.html
- https://developer.android.com/reference/android/support/design/widget/TabLayout.html
- http://developer.android.com/reference/android/widget/Toolbar.html
- http://developer.android.com/reference/android/support/v7/widget/CardView.html
- http://developer.android.com/intl/es/training/implementing-navigation/nav-drawer.html
- http://developer.android.com/reference/android/support/design/widget/Snackbar.html
- http://developer.android.com/reference/android/support/design/widget/TextInputLayout.html

AD HOC TALENT