



SEMANA 1 – PRACTICA 2

Instrucciones:

Los Layouts permiten ubicar los Views dentro de un Activity, básicamente un Layout es un elemento que representa el diseño de la interfaz de usuario de componentes gráficos como una actividad, fragmento o widget. Los layouts pueden ser creados a través de archivos XML o con código Java de forma programática

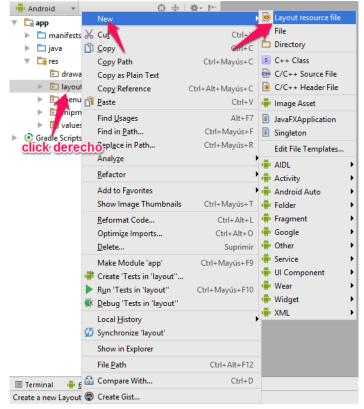
Algunos de los view groups más populares son:										
	LinearLay	out								
	FrameLayout									
	RelativeLayout									
	TableLayout									
	GridLayout									
En la LinearL	•	actividad	generaremos	interfaces	utilizando	Views	у	las	propiedades	del

I. Procedimientos

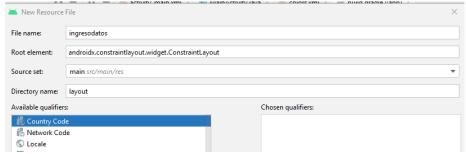
Instrucciones:

- 1. Crea un nuevo proyecto, denominado Layouts_XYZ, donde xyz representan tus iniciales (Esto con fines de evaluación).
- 2. Luego de haber creado el proyecto elige en las plantillas de actividades la de Empty Activity.
- 3. A continuación, vamos a agregar un nuevo layout:

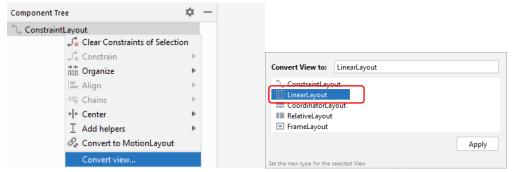




4. Proporcionarle el nombre Ingresodatos

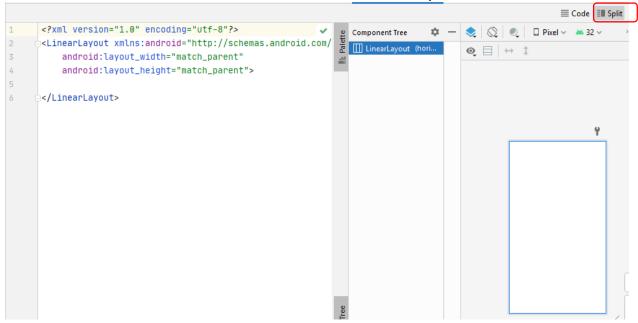


5. Efectúa clic derecho sobre el árbol de componentes y elige Convert View a continuación LinearLayout



6. Configura la vista del diseñador para que se muestren tanto el código xml, como el diseño obtenido, efectúa clic sobre el botón Split.





7. En la vista código inserta la siguiente codificación:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    android:layout_width="match_parent"
    android:layout height="match parent"
    android:orientation="vertical"
    android:padding="48dp">
    <TextView
        android:id="@+id/texto conectar"
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout gravity="center horizontal"
        android:layout weight="1"
        android:text="Conectar"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <EditText
        android:id="@+id/input usuario"
        android:layout width="match parent"
        android:layout height="0dp"
        android: layout gravity="center horizontal"
        android:layout_weight="1"
        android:drawableLeft="@drawable/ic mail"
        android:gravity="center"
        android:hint="Correo"
        android:inputType="textEmailAddress"
    <EditText
        android:id="@+id/input contrasena"
        android:layout width="match parent"
        android:layout_height="0dp"
        android:layout gravity="center horizontal"
        android:layout weight="1"
        android:ems="10"
        android:hint="Contraseña"
```



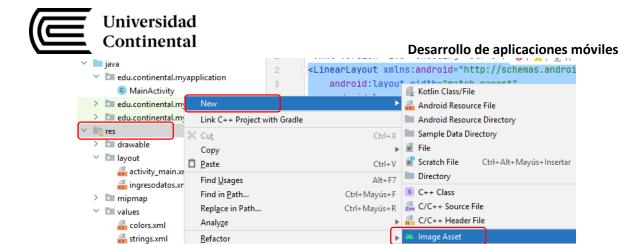


```
android:inputType="textPassword"
        android:drawableLeft="@drawable/ic text"
        />
    <Button
        android:id="@+id/boton iniciar sesion"
        style="?android:attr/buttonStyleSmall"
        android:layout width="match parent"
        android:layout_height="0dp"
        android:layout_gravity="center_horizontal"
        android:layout_weight="1"
        android:text="Iniciar Sesion" />
    <TextView
        android:id="@+id/texto olvidaste contrasena"
        android:layout width="wrap content"
        android:layout_height="0dp"
        android:layout_gravity="center_horizontal"
        android:layout weight="1"
        android:gravity="center vertical"
        android:text="¿Olvidaste tu constraseña?"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:textColor="#0E8AEE" />
</LinearLayout>
```

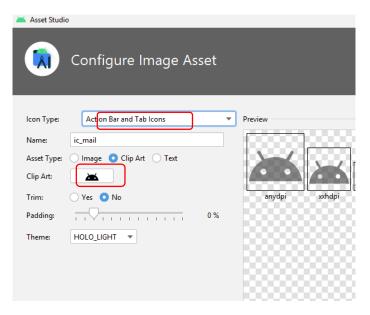
8. Debe obtener una interface como la siguiente:



9. Agregue los recursos faltantes, para ello siga la siguiente secuencia:



10. En la pantalla de Configure Image Asset, en IconType elija Action Bar and tab icons, en Name inserte ic_mail, luego para elegir el ícono haga clic en Clip Art

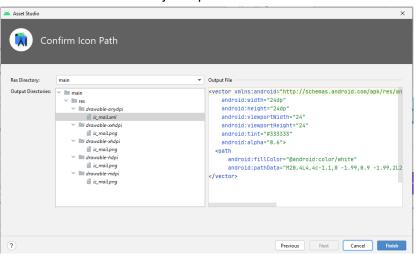


11. Para facilitar la ubicación de un ícono inserte el texto mail en el buscador





12. En la pantalla de Confirm Icon Path elija la opción Finish.



- 13. Repita el proceso para el ícono de ic_text.
- 14. El diseño final debería verse como se muestra:



