

GUI

Existen dos conceptos alrededor de la interfaz de usuario:

➤ **UI**

Se enfoca en lo que está dentro de la pantalla, se resuelve el **diseño**: selección y distribución de los elementos de la interfaz (ej. textos y campos del formulario), **consistencia del diseño** (con la plataforma, con otras pantallas).

➤ **UX**

UX es la experiencia de usuario y se enfoca en lo que *experimenta* el usuario antes, durante y después de interactuar con el artefacto. Sin incorporar al usuario, no se puede hacer UX. Por eso, resulta fundamental en el diseño de la experiencia.

UI



UX

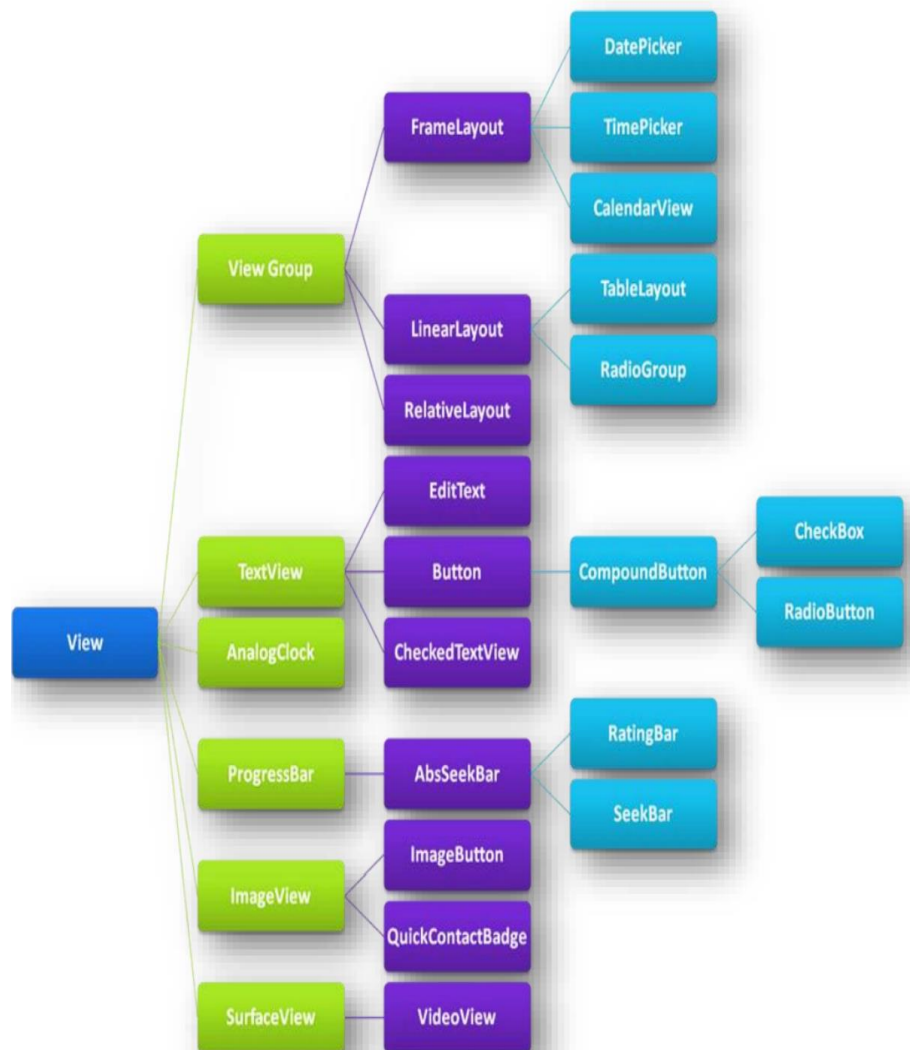


La Interfaz de Usuario es la parte de la aplicación que ve y con la interactúa el usuario

Las *vistas* van a ser objetos descendientes de la clase View a continuación mostramos la jerarquía de las clases de vistas.

Jerarquía de clases

La siguiente figura muestra la jerarquía de clases relacionadas con los elementos visuales



Valores de medida

Unidad	Significado
dp	Densidad de pixeles (se adaptan automáticamente)
sp	Pixeles escalonados (con base a la letra preferente)
px	Pixeles
in	Pulgadas
mm	Milímetros

Layout

- Contenedor de una o más vistas y controla su comportamiento y posición.

Distribuye los elementos en:

- **LinearLayout:** Fila o columna.
- **TableLayout:** Forma tabular.
- **RelativeLayout:** En relación a otros.
- **AbsoluteLayout:** Forma absoluta.

Atributos Comunes de los Objetos View

Nombre.	Descripción.
android:layout_width	Especifica el ancho del elemento, se puede especificar alguna de las constantes predefinidas (match_parent y wrap_content) o un valor específico.
android:layout_height	Especifica el alto del elemento, se puede especificar alguna de las constantes predefinidas (match_parent y wrap_content) o un valor específico con las unidades deseadas (ver al final de la tabla las posibles unidades).
android:layout_gravity	Especifica como el elemento estará alineado dentro de su contenedor.
android:gravity	Especifica como el contenido del elemento estará alineado. Los posibles valores para este y el anterior atributo son: top, bottom, left, right, center_vertical, center_horizontal, center, fill_vertical, fill_horizontal, fill, clip_vertical, clip_horizontal.
android:layout_marginBottom, android:layout_marginTop, android:layout_marginLeft, android:layout_marginRight	Especifican los márgenes del elemento (espacio alrededor del elemento) con las unidades deseadas.
android:paddingBottom, android:paddingTop, android:paddingLeft, android:paddingRight	Especifican el espacio de relleno del elemento (espacio entre el borde del elemento y su contenido) con las unidades deseadas.
android:id	Especifica "un identificar" a ser asignado al elemento, se usa si se requiere poder referirse a algún elemento en el código Java y en los RelativeLayout para indicar con respecto a cuál elemento se colocará algún otro. Para asignar un nuevo id el valor a usar sería "@+id/nombre_identificador" donde nombre_identificador es el identificador que se le desea asignar al elemento.
android:background	Especifica el color de fondo, se puede hacer referencia a algún color definido en algún archivo XML de valores simples ("@color/nombre_color"), o bien, directamente usando alguno de los formatos comunes ("#rrggbb", "#aarrggbb", donde rr es el valor hexadecimal para rojo, gg es el valor hexadecimal para verde, bb es el valor hexadecimal para azul y aa es el valor hexadecimal de alfa o transparencia).
android:textColor	Especifica el color del texto, se usan las mismas reglas que para el color de fondo.