Observaciones

Tipos de elementos "*text*"

- Los distintos tipos de texto (*date*, *time*, *password*, etc) son hijos de "*EditText*" y lo único que tienen consigo son configuraciones específicas de etiquetas entre ellas el tipo de *input* que reciben.
- El más diferente de todos en configuración es el tipo multilínea, pero asumo que eso es debido a ciertas características inherentes, como ser extensible o no tener un tipo de input definido.

Tipos de elementos "*button*"

- Los distintos tipos de botones son solo diseños visuales con, esencialmente nada diferente en cuanto a funcionalidad. Seguramente provienen todos de la misma clase padre "button" la cual tiene todas las funcionalidades y cada hijo lo que cambia es la presentación de esas funcionalidades.
  Ej.: la forma en la cual se activa cada botón (xml) será implementada por cada hijo, pero las acciones lógicas disponibles (java) son las del padre (a menos que se requiera alguna funcionalidad extra para el hijo en específico).
- Una clara excepción en esta categoría son son los grupos de botones, que entiendo que sirven para realizar/facilitar funciones lógicas avanzadas. Como solo poder activar uno de varios botones, o activar varios al activar el botón padre en un árbol de opciones.

Tipos de elementos "*Widget*"

- Estos son elementos con funcionalidades útiles para el funcionamiento de una app. Se podrían considerar como elementos precocinados, que usarás cuando no necesites/quieras algo personalizado para ahorrarte tiempo y ganas de morirte.
- Estos *widgets* van desde relojes y calendarios, a sliders y barras de progreso, pasando por ventanas a webs/img/videos y la campos de búsqueda (la clásica lupa). Asumo que aunque en xml son muy similares en cuanto a atributos base, la diferencia radica en atributos y funciones específicas para la utilización de cada uno.

Tipos de elementos "*Layout*"

- Son básicamente usados como herramientas para ordenar las demás vistas de forma visualmente coherente para el usuario. Se pueden crear estructuras de marcos, tablas, filas, columnas, etc.
- Algo interesante es que la parte visual de mi app utiliza el *layout* de tipo *constraint* como base de los demás elementos. Lo cual implica que estos son la base de todos los demás elementos desde un punto de vista estructural.

Tipos de elementos "*Containers*"

- Continúan donde los elementos *layout* se detienen. Sirven para crear contenedores visuales más complejos para los demás elementos. Muchos de estos son sacados directamente de las bibliotecas de Google, así que no son nativos y por lo tanto podrían entrar en la misma categoría espiritual de los widgets, ya que su objetivo es permitir abstraerse a los desarrolladores de Android de las complejidades del diseño de estos elementos. En su lugar solo necesita uno aprender a usar las interfaces proporcionadas. En resumen, son APIs, como los elementos del grupo *widgets*.
- Los únicos que no son APIs son *Spinner*, *ScrollView* y *HorizontalScrollView*. Los cuales sirven como para poder mostrar más de lo que normalmente se podría mostrar en el espacio que ocupan. En el caso de *Spinner* siendo un contenedor desplegable y los otros dos siendo contenedores deslizables.

Tipos de elementos "*Helpers*"

- Simplemente son utilidades para ayudar con la composición de las vistas. Dependen del *layout* en el que se coloquen. Sirven para agrupar (*Group*), listar (*Flow*), crear capas (*Layer*), poner límites (*Guideline* & *Barrier*), crear filtros (*ImageFilterView* & *ImageFilterButton*) y prototipos de vistas (*MockView*).

Tipos de elementos "*Google*"

- Son APIs para poder incluir Google Maps y Google ADs fácilmente en tus aplicaciones.

Tipos de elementos "*Legacy*"

- Son elementos ya no usados actualmente pero que se conservan a forma de permitir retrocompatibilidad. Son las estructuras *GridLayout*, *RelativeLayout* y *TabHost*, y los contenedores *ListView*, *GridView*.