PROPIEDADES

- ✓ **android:layout:width=”match_parent”… y android:height**

  Sirve para indicar el ancho y el alto de un elemento. Se puede utilizar los constantes match_parent para indicar que queremos todo el alto o el ancho o bien wrap_content para indicar que reservamos el espacio que solo necesita. También se puede indicar el ancho/alto **que se necesite.**
- ✓ **android:visibility= “invisible” o visible**

  Oculta una vista
- ✓ **android:textColor= “@color/..”**

  Cambia el color de un texto
- ✓ **android_background =”@android:color/..”**

  Color de fondo
- ✓ **android:layout_marginStart=”xx dp” o Android:layout_marginTop=”xx dp “ …**

  Sirve para agregar espacio hacia fuera del elemento
- ✓ **android:padding=”xx dp”**

  Espacio entre contenedor y los elementos intermos.(dp-> densidad por pulgada)

LINEAR LAYOUT:

- ✓ **android:orientation=”vertical” o android:orientation=”horizontal**

  es obligatorio incluirla cuando hay mas de dos elementos en el layout e indica la disposición de las vistas.
- ✓ **android:gravity=”center_horizontal|top”**

  Esta propiedad nos indica como podemos centrar los controles en el layout (linear) , en el caso de que se ponga varios deben de ir separados |.

FRAME LAYOUT

FrameLayout no se rige por ningún orden especifico se va colocando los componentes unos encima de otros**.**

- ✓ **android:layout_gravity= “center”**

  Con esta propiedad podemos centrar con respecto al padre**.**

CONSTRAINTLAYOUT:

- ✓ **app:layout_constraintStart_toStartOf=”…”**

  Significa Inicia al inicio del …
- ✓ **app:layout_constraintStart_toEndOf=”…”**

  Empiece al final de

- ✓ **app:layout_constraintEnd_toEndOf="…"**

  Finaliza al final de…
- ✓ **app:layout_constraintBottom-toBottomOf="…"**

  Finaliza la parte inferior con referencia al inferior de…
- ✓ **app:layout_constraintTop-toTopOf="…"**

  Empieza en la parte superior de…
- ✓ **app:layout_constraintTop-toBottomOf="…"**

  Empieza en la parte superior al final de

- ✓ **app:layout_marginStart o app:layout_marginEnd o app:layout_marginTop o app:layout_marginEnd**

  Estos sirven para poner márgenes, pero tiene que haber un constraint del mismo lado para que tengan efecto

MATERIAL DESIGN

BOTTOMAPPBAR (https://material.io/components/app-bars-bottom)

Navegación principal en la parte inferior (Barra acciones)

- ✓ **android:layout_gravity= "bottom"**
  Le indicamos que se vaya al final de la pantalla
- ✓ **app:navigationIcon**
  Le asignamos un icono de navegación
- ✓ **style= @ style/….."**
  Damos un style a la barra de navegación
- ✓ **app:menu="@menu…."**

  Para asignarle un menú de opciones
- ✓ **hideOnScroll =true**
  Ocultamos nuestro appBar cuando estemos realizando un scroll

FLOATINGACTIONBUTTON (https://material.io/components/buttons-floating-action-button)

- ✓ **app:layout_anchor**
  enlazar nuestro FloatingActionButton con la barra inferior (BotomAppBar).
  El efecto es que se transparente y se hace un corte redondeado.

- ✓ **contentDescription**
  propiedad para hacerla accesible.
- ✓ **importantForAccessibility ="no"**
  No tenemos en cuenta la accesibilidad

MATERIAL CARDVIEW (https://material.io/components/cards)

Otro tipo contenedor pero especializado en mostrar información en otras áreas más pequeñas a modo de tarjeta

- ✓ **app:layout_anchor**
  enlazar nuestro FloatingActionButton con la barra inferior (BotomAppBar).
  El efecto es que se transparente y se hace un corte redondeado.
- ✓ **android:clickable=true**
  Hacemos que la tarjeta se clickable
- ✓ **android:focusable=true**
  Clickable y focusable van de la mano

TEXTVIEW

- ✓ **android:textAppearance="?attr/textAppearandeHeadLine"**

es la apariencia del texto para los títulos, hay varios dependiendo de lo que se quiera

- ✓ **android:maxLines= numero**

máximo de líneas que se quieren que se muestren

- ✓ **android:ellipsize=end**

Pone tres puntos sucesivos, indicando que hay más texto

EDITTEXT, TEXTINPUTLAYOUT Y TEXTINPUTEDITTEXT
(https://material.io/components/text-fields)

EDITTEXT (TEXTINPUTEDITTEXT en MaterialDesign)

- ✓ **android:inputType**

Tipo de entrada(textpassword, textUri,…)

TEXTOINPUTLAYOUT

- ✓ **android:hint**

Para que el usuario sepa que es lo que debe de introducir. (esta en la parte de arriba, para que el usuario siempre sepa lo que tiene que hacer)

- ✓ **app:helperText=””**

se muestra en la parte de abajo con la cual tenemos la opción de decirle al usuario con más énfasis algo de lo que pueda ayudar al usuario.

- ✓ **app:endIconMode=”clear text· (password_toggle, custom,none, dropdow_menu)**

none-> por defecto, no tiene nada
dropdowmenu es para un despeglar
clear_text -> limpia todo el texto que ha escrito
pasword_toggle -> ocultar lo que se está escribiendo
custom-> para le agregemos el icono y la acción que se quiera

- ✓ **style= @ style/ Widget.MaterialComponents.TextInputLayout.OutlinedBox.Dense"**

Para cambiarle el diseño. En OutLinedbox viene de TextInputLayout. Tiene dos el normal y el denso (El dense es más compacto.)

CHECKBOX https://material.io/components/checkboxes

button-> podemos indicarle el icono personalizado para su botón
buttonTint -> cambiamos el color del icono

DIVIDERS https://material.io/components/dividers#usage

MATERIAL TOGGLE BUTTON

Serie de botones agrupados

SHARED PREFENCES

SISTEMA DE ALMACENAMIENTO INTERNO DE FORMA PERMANENTE, FUNCIONA EN BASE A VALORES CLAVE - VALOR

Como guardar datos de pares clave-valor

HTTPS://DEVELOPER.ANDROID.COM/TRAINING/DATA-STORAGE/SHARED-PREFERENCES?HL=ES-419