|  |  |  |
| --- | --- | --- |
| **Título componente** | **Acordeón** | |
| **Título** | ***Layouts*** | |
| **Texto descriptivo** | Existen diferentes tipos de *Layout*, entre los más comunes están: *LinearLayout*, *TableLayout* y *ConstraintLayout*. A continuación, se profundiza en cada uno: | |
| **Título** | **Texto** | **Imagen de referencia (opcional)** |
| ***LinearLayout*** | Un *LinearLayout* dispone las vistas de la interfaz una debajo de otra (orientación vertical) o una al lado de la otra (orientación horizontal). Sus propiedades principales son:  a) **Orientation**: es la propiedad que define la disposición de sus vistas, ya sea horizontal o vertical.  b) **Layout\_height**: esta propiedad especifica el alto del *ViewGroup* (contenedor). c) **Layout\_width**: esta propiedad especifica el ancho del *ViewGroup*.  Los valores que pueden tomar las propiedades *height* y *width* son:   * **match\_parent**: para que ocupe el mismo tamaño de la pantalla. * **wrap\_content**: para que se adapte al tamaño de su contenido. * Valores personalizados que se miden con dp (píxeles de densidad independiente).   Estas propiedades de *height* y *width* se aplican a todos los contenedores y vistas.   |  | | --- | | <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  xmlns:tools="http://schemas.android.com/tools"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".MainActivity">  <!-- Dentro del contenedor se agregan los view en este caso Button -->  <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="45dp"  android:text="@string/boton1" />  <Button  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="@string/boton2" />    </LinearLayout>  <!-- Cierre del contenedor LinearLayout --> |   El código XML para la creación de un *layout* de tipo *LinearLayout* con orientación vertical, el cual contiene dos vistas de tipo *Button.*  **<LinearLayout:** tipo de contenedor lineal   * **orientation=vertical:** propiedad que define la ubicación de los *views* de forma vertical. * **layout\_width:** propiedad que define el ancho del contenedor. * **layout\_height:** propiedad que define el alto del contenedor >   **<Button:** etiqueta para crear un *view* de tipo *Button* (Botón).   * **layout\_width:** propiedad que define el ancho del *view.* * **layout\_height:** propiedad que define el alto del *view* * **text**: propiedad que define el título del button /*>*   El código anterior mostraría el *LinearLayout* de la siguiente forma: | Vista de Diseño LinearLayout |
| ***TableLayout*** | Es un contenedor *ViewGroup* que permite mostrar los elementos *view* en columnas y filas. Para crear las filas se utiliza el objeto *TableRow*. Cada fila puede tener una o más celdas, o ninguna, y cada celda se crea mediante un tipo de objeto *view*. Es decir, las celdas de una fila pueden formarse por una diversidad de objetos *view*, como *ImageView* o *TextView*.  El siguiente código XML de diseño muestra dos filas con tres celdas en cada una, y una tercera fila con dos celdas:   |  | | --- | | <!-- Crea la tabla -->  <TableLayout  xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:stretchColumns="0,1,2">  <!-- Crea la fila -->  <TableRow>  <TextView android:text="Celda 1.1" /> <!-- Crea la Columna -->  <TextView android:text="Celda 1.2" /> <!-- Crea la Columna -->  <TextView android:text="Celda 1.3" /> <!-- Crea la Columna -->  </TableRow>  <!-- Crea la fila -->  <TableRow>  <TextView android:text="Celda 2.1" /> <!-- Crea la Columna -->  <TextView android:text="Celda 2.2" /> <!-- Crea la Columna -->  <TextView android:text="Celda 2.3" /> <!-- Crea la Columna -->  </TableRow>  <!-- Crea la fila -->  <TableRow>  <TextView android:text="Celda doble de ancho 3.1"  android:layout\_span="2" />  <TextView android:text="Celda 3.2" />  </TableRow>    </TableLayout> | |  |
|  | El código anterior mostraría el***TableLayout***de la siguiente forma: |  |
| ***ConstraintLayout*** | *ConstraintLayout* es el contenedor más utilizado actualmente y es el que se configura por defecto al crear una actividad, ya que permite diseñar los *view* directamente desde las herramientas visuales del editor de diseño. Con este contenedor se puede crear todo el diseño arrastrando y soltando elementos, en lugar de editar el XML manualmente.  Cada posición de un *view* en un *ConstraintLayout* se define agregando al menos una restricción horizontal y una vertical. Cada restricción representa una conexión o alineación con otra vista. Estas restricciones determinan la posición de la vista a lo largo del eje vertical u horizontal, por lo que cada vista debe tener al menos una restricción para cada eje. Sin embargo, normalmente se requieren más restricciones para lograr un diseño preciso.  Vista de Diseño ConstraintLayout: |  |