**Nizar Sanchez(2018-6482)**

**Tarea 1 android**

1. Usar un relativelayout para listar una lista de 6 textview uno debajo del otro, intercalando uno a la derecha y otro a la izquierda los textview y colocándole los numero en letras **ejemplo, uno, dos, tres, cuatro y cinco** a cada textview
   1. **Ejemplo:**

ejercicio1.xml

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_height="match\_parent"

android:layout\_width="match\_parent"

>

<TextView

android:id="@+id/btn1"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="@string/one"

android:textAlignment="center"

android:layout\_alignParentLeft="true"

android:textSize="30dp"

>

</TextView>

<TextView

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="@string/two"

android:textSize="30dp"

android:textAlignment="center"

android:layout\_alignParentRight="true"

android:layout\_marginTop="50dp"

></TextView>

<TextView

android:id="@+id/btn3"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="@string/three"

android:textAlignment="center"

android:layout\_alignParentLeft="true"

android:textSize="30dp"

android:layout\_marginTop="100dp"

>

</TextView>

<TextView

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="@string/four"

android:textSize="30dp"

android:textAlignment="center"

android:layout\_alignParentRight="true"

android:layout\_marginTop="150dp"

></TextView>

<TextView

android:id="@+id/btn5"

android:layout\_width="200dp"

android:layout\_height="50dp"

android:text="@string/five"

android:textAlignment="center"

android:layout\_alignParentLeft="true"

android:textSize="30dp"

android:layout\_marginTop="200dp"

>

</TextView>

</RelativeLayout>

1. También usar un linearlayout vertical para hacer una lista de 5 filas con cinco comidas favoritas con el fondo del layout verde...

ejercicio2.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/green"

>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/pizza"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/taco"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/cake"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/ice\_scream"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hamburger"

>

</TextView>

</LinearLayout>

1. Crear una layout y colocar un framelayout y dentro colocar dos linearlayout cada uno con un pequeño párrafo cada uno como un pensamiento de superación y usar dos botones aceptar y cancelar debajo de cada layout

ejercicio3.xml

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="300dp"

>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="@string/sentence"

android:textSize="15dp"

android:textAlignment="center"

/>

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="330dp"

android:orientation="horizontal"

>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/ok"

android:layout\_marginLeft="125dp"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/cancel"

/>

</LinearLayout>

</FrameLayout>

1. Crear un framelayout y dentro tres botones llamados botón 1,2,3, uno al justo inicio otro en el medio y otro justo al final del layout

ejercicio4.xml

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="left"

android:text="1"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="right"

android:text="3"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="160dp"

android:text="2"

/>

</FrameLayout>

1. Usar un tablelayout y crearlo con 5 fila y 3 columnas, organizarlo con los números desde el 1 al 12 y en la fila 3 colocar un botón en el medio

ejercicio5.xml

<?xml version="1.0" encoding="utf-8"?>

<TableLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="1"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="2"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="3"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="4"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="5"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="6"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="7"

/>

<Button

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="8"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="9"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="10"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="11"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="12"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

</TableRow>

</TableLayout>

1. Usar un android:layout\_span  y colocarlo en el layout que corresponde de los tres ejercicios dejados más arriba, solo seria usar un ejemplo nuevo del ejercicio seleccionado con la nueva opción agregada y ponerlo a funcionar, y dejar el ejercicio de referencia seleccionado intacto.

ejercicio6.xml

<?xml version="1.0" encoding="utf-8"?>

<TableLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="1"

android:layout\_span="2"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="2"

android:layout\_span="2"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="3"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="4"

android:layout\_span="2"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="5"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="6"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="7"

android:layout\_span="2"

/>

<Button

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="8"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="9"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="10"

android:layout\_span="2"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="11"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="12"

/>

</TableRow>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=""

/>

</TableRow>

</TableLayout>

1. Explique que hace android:stretchColumns=”\*” en un tablelayout y haga un ejemplo de uso.

Es un atributo que indica el indice base 0 ”Es decir que empieza desde 0 hasta n” de las columnas para estirar. Los indices de las columnas obligatoriamente tienen que estar separado por comas.

Ej:

android:stretchColumns=“ 0,1,2”

Los indices erroneos o duplicados seran ignorados. Usando el valor “\*” podemos estirar todas las columnas a la vez.

Ejemplo de uso :

ejercicio7.xml

<?xml version="1.0" encoding="utf-8"?>

<TableLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:stretchColumns="0,1"

>

<TableRow >

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="0"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="1"

/>

<TextView

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="2"

/>

</TableRow>

</TableLayout>

Fuente: [https://developer.android.com/reference/android/widget/TableLayout.html#attr\_android:stretchColumns](https://developer.android.com/reference/android/widget/TableLayout.html" \l "attr_android:stretchColumns)

1. Explicarla diferencia de un relativelayout y un framelayout exacta en base a ejemplos y explicar en qué caso usaría uno y otro.

ejercicio8\_a.xml

Ejemplo:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/btn1"

android:text="1"

/>

<Button

android:id="@+id/btn2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/btn1"

android:text="2"

/>

<Button

android:id="@+id/btn3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/btn2"

android:text="3"

/>

</RelativeLayout>

ejercicio8\_b.xml

Ejemplo:

<?xml version="1.0" encoding="utf-8"?>

<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent" android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/hello\_world"

android:layout\_gravity="center"

/>

</FrameLayout>

Un relative layout es una disposicion en la cual permite mostrar las view hijas en posiciones relativas. La posicion de cada view puede ser especificada como relativa respecto a la view hermana(como lo puede ser a la izquierda o debajo de otra view) o por posiciones relativas al area del RelativeLayout padre(como lo puede ser a la izquierda, encima,abajo,etc).

Fuente: [https://developer.android.com/guide/topics/ui/layout/relative#Example](https://developer.android.com/guide/topics/ui/layout/relative" \l "Example)

Un FrameLayout es una disposición la cual bloquea un area de la pantalla para mostrar un solo objeto. Generalmente el FrameLayout se deberia usar para mostrar un solo objeto ya que puede ser dificl organizar las view hijas.Para poder controlar la posicion de cada objeto podemos usar la propiedad android:layout\_gravity.

Fuente : <https://developer.android.com/reference/android/widget/FrameLayout>

1. Explicar en qué caso y por qué usaría un tablelayout y un gridlayout según las informaciones entendidas y vistas.

TableLayout: es solo una disposicion,algo como en html.Este es usado para ordenar un grupo de view en filas y columnas.Este no hace scrolling por si solo.

Grid layout: ES basicamente como un ListView, cuyo objetos estan ordenados en regilla. Esta adjuntado a un Adapter y recupera los views desde el Adapter que tiene el usuario que hace scrolling a travez de el.

Fuente : <https://medium.com/pen-bold-kiln-press/tablelayout-vs-gridlayout-ae28be87b4b6>

1. Poner ejemplos de otros layout que existen y explicarlos, al menos de 3

ConstraintLayout: es un view group el cual permite posicionar y cambiar el tamano de los widgets de una manera flexible.

ejercicio10\_a.xml

Ejemplo:

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

xmlns:app="http://schemas.android.com/apk/res-auto">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

android:text="@string/hello\_world"

/>

</androidx.constraintlayout.widget.ConstraintLayout>

Fuente: <https://developer.android.com/reference/android/support/constraint/ConstraintLayout>

LinearLayout: Es un layout que organiza otras views ya sea horizontalmente en una sola columna o verticalmente en una sola fila.

ejercicio10\_b.xml

Ejemplo:

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical" android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="1"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="2"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="3"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="4"

>

</TextView>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="5"

>

</TextView>

</LinearLayout>

Fuente:<https://developer.android.com/reference/android/widget/LinearLayout>

TabLayout: provee un layout horizontal para mostrar pestanas.

ejercicio10\_b.xml

Ejemplo:

<?xml version="1.0" encoding="utf-8"?>

<!-- IMPORTANTE: Antes de probar el codigo de abajo, recordar aniadir la siguiente linea

al archivo build.gradle(Module:app): implementation 'com.android.support:design:28.0.0'

El ultimo numero indica la version del API en mi caso es la 28

-->

<com.google.android.material.tabs.TabLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/tabs"

>

<com.google.android.material.tabs.TabItem

android:text="@string/tab1"/>

<com.google.android.material.tabs.TabItem

android:text= "@string/tab2"/>

<com.google.android.material.tabs.TabItem

android:text="@string/tab3"/>

<com.google.android.material.tabs.TabItem

android:text="@string/tab4"

/>

</com.google.android.material.tabs.TabLayout>

Fuente: <https://developer.android.com/reference/android/support/design/widget/TabLayout>

**Subir el código en un documento de Word o pdf al virtual. Probar su emulador para que estén seguros que funcionan. Solo el código xml.**