

INDICE

<u>1.1.</u>	<u>INTRODUCCIÓN</u>	<u>1</u>
<u>1.2.</u>	<u>CASOS PRÁCTICOS</u>	<u>2</u>
<u>1.2.1.</u>	<u>TABLA SIMPLE</u>	<u>2</u>
<u>1.2.2.</u>	<u>OCULTAR COLUMNAS</u>	<u>3</u>
<u>1.2.3.</u>	<u>CELDA OCUPANDO VARIAS COLUMNAS</u>	<u>4</u>
<u>1.2.4.</u>	<u>EXPANSIÓN DE COLUMNAS</u>	<u>5</u>

1.1. Introducción

- Este Layout permite disponer de los elementos visuales en una tabla de filas e columnas.
- Esta estructura se crea de forma similar a HTML:
 - ✓ Se indican las filas de la tabla: <TableRow>
 - ✓ Se crean objetos dentro de la fila, cada objeto será una columna.
- Por lo general, el ancho de cada columna corresponde con el ancho del mayor del objeto de esa columna.
- Existen propiedades que permiten modificar esos anchos:
 - ✓ Atributos del Layout:
 - **android:stretchColumns:** indica que columna/s se expande/n para ocupar el espacio libre que queda a la derecha de la tabla en la pantalla.
 - ...1. Las columnas comienzan a numerarse en 0.
 - ...2. Ejemplo 1: **android:stretchColumns="0"**, se expande a la primera columna de la tabla.
 - ...3. Ejemplo 2: **android:stretchColumns="0,2"**, se expande a la primera y tercera columnas de la tabla.
 - ...4. Ejemplo 3: **android:stretchColumns="*"**, se expande todas las columnas de la tabla.
 - **android:shrinkColumns:** indica que columna/s se pueden/n encoger para dejar espacio del lado derecho de la tabla.
 - ...1. Se define del mismo hecho que stretchColumns.
 - **android:collapseColumns:** indica que columna/s de la tabla se pueden ocultar.
 - ...1. Se define del mismo hecho que stretchColumns.
 - ✓ Atributo de objeto dentro de una fila:
 - **android:layout_span:** una celda X,Y puede ocupar un espacio de varias columnas.
 - ...1. Se indica el número de columnas que debe ocupar la celda.

1.2. Casos Prácticos

- Comenzamos creando un nuevo proyecto: **U2_06_Table**.

1.2.1. Tabla simple

- La imagen muestra la distribución que se quiere tener de la tabla:



- Cada celda tiene un ancho que viene determinado por la longitud de su contenido.
- Observar como el ancho de la columna 2 se adapta al objeto más ancho.
- El XML do layout es:

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow>

        <Button android:text="1" />

        <Button android:text="2" />

        <Button android:text="3" />
    </TableRow>

    <TableRow>

        <Button android:text="4" />
```

```

        <Button android:text="abcdef" />
    </TableRow>

    <TableRow>

        <Button android:text="7" />
    </TableRow>

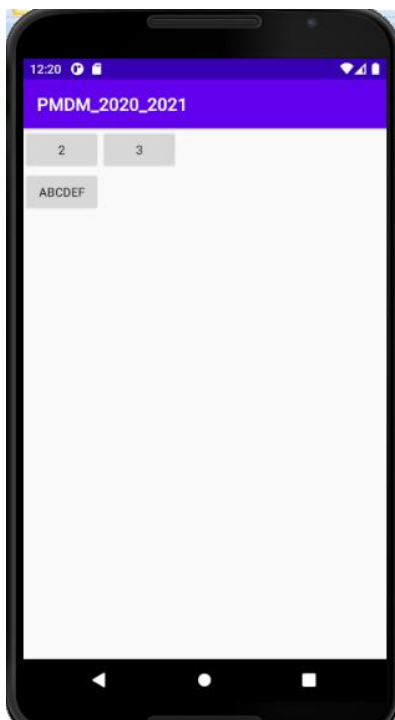
</TableLayout>

```

- Observar que solo se definen filas y luego se crean los objetos dentro de estas, que serán las columnas

1.2.2. Ocultar columnas

- Vamos a ocultar, en este caso, solo la primera columna (índice 0). Recordar que el layout numera las columnas comenzando en 0.



- Nuestra columna 1 desapareció.
- El XML del layout:

```

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:collapseColumns="0" >

    <TableRow>

        <Button android:text="1" />

        <Button android:text="2" />

```

```

        <Button android:text="3" />
    </TableRow>

    <TableRow>

        <Button android:text="4" />

        <Button android:text="abcdef" />
    </TableRow>

    <TableRow>

        <Button android:text="7" />
    </TableRow>

</TableLayout>

```

1.2.3. Celda ocupando varias columnas

- La celda 3,1 que ocupe todo el ancho de la tabla:



- El XML del layout. Ojo!!!! que no está la propiedad de ocultar columnas.

```

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

```

```

    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TableRow>

        <Button android:text="1" />

        <Button android:text="2" />

        <Button android:text="3" />
    </TableRow>

    <TableRow>

        <Button android:text="4" />

        <Button android:text="abcdef" />
    </TableRow>

    <TableRow>
        <Button android:text="7" android:layout_span="3"/>
    </TableRow>

</TableLayout>

```

- Observar la línea marcada.

1.2.4. Expansión de columnas

- Vamos a hacer que las columnas 1 (índice 0) y 3 (índice 2) se expandan ocupando o ancho de la derecha la pantalla que deja libre la tabla.



- Observar que la celda 3,1, sigue ocupando todo el ancho de la tabla.

➤ El XML del layout:

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:stretchColumns="0,2">

    <TableRow>

        <Button android:text="1" />

        <Button android:text="2" />

        <Button android:text="3" />
    </TableRow>

    <TableRow>

        <Button android:text="4" />

        <Button android:text="abcdef" />
    </TableRow>

    <TableRow>

        <Button android:text="7" android:layout_span="3"/>
    </TableRow>

</TableLayout>
```

➤ Para terminar, se propone al estudiante, crear el layout asociado a la siguiente imagen:

- ✓ Usando un TableLayout.
- ✓ Sin modificar el tamaño de cada objeto explícitamente.

