

1 TableLayout

1.1 Sumario

- 1 Introducción
- 2 Casos prácticos
 - ◆ 2.1 Táboa simple
 - ◆ 2.2 Ocultar columnas
 - ◆ 2.3 Cella ocupando varias columnas
 - ◆ 2.4 Expansión de columnas

1.2 Introducción

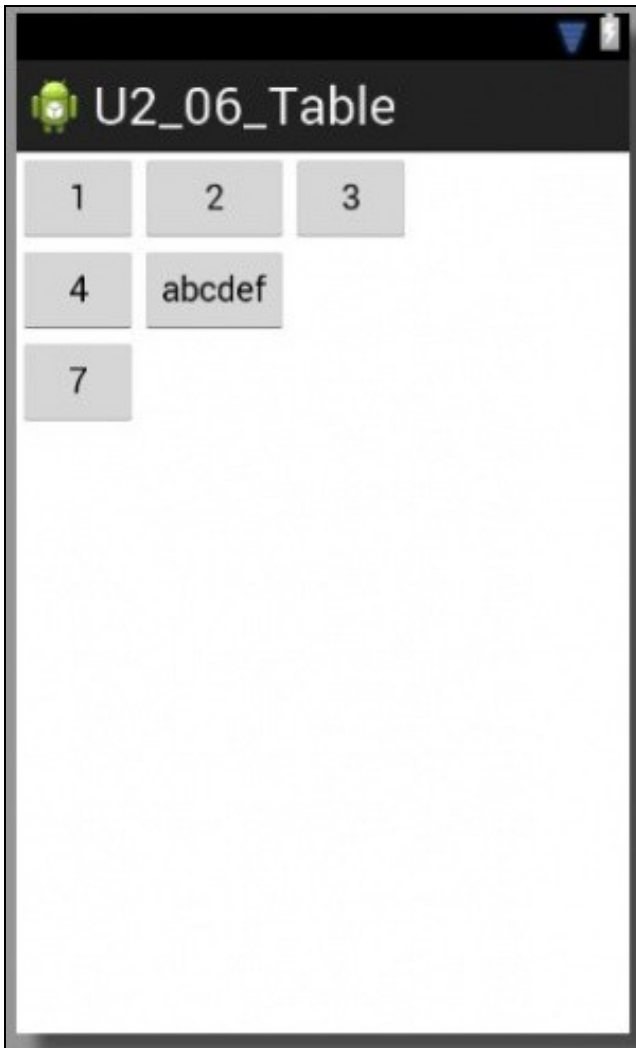
- Este Layout permite dispoñer os elementos visuais nunha táboa de filas e columnas.
- Esta estrutura créase de xeito similar a HTML:
 - ◆ Indícanse as filas da táboa: **<TableRow>**
 - ◆ Créanse obxectos dentro da fila, cada obxecto será unha columna.
- Polo xeral, o ancho de cada columna corresponde co ancho do maior obxecto desa columna.
- Existen propiedades que permiten modificar eses anchos:
 - ◆ Atributos do Layout:
 - ◇ **android:stretchColumns**: indica que columna/s se expande/n para ocupar o espacio libre que queda á dereita da táboa na pantalla.
 - As columnas comezan a numerarse en 0.
 - Exemplo 1: **android:stretchColumns="0"**, expándese a primeira columna da táboa.
 - Exemplo 2: **android:stretchColumns="0,2"**, expándense a primeira e terceira columnas da táboa.
 - Exemplo 3: **android:stretchColumns="**"**, expándense tódalas columnas da táboa.
 - ◇ **android:shrinkColumns**: indica que columna/s se pode/n encoller para deixar espazo ao lado dereito da táboa.
 - Defínese do mesmo xeito que stretchColumns.
 - ◇ **android:collapseColumns**: indica que columna/s da táboa se poden ocultar.
 - Defínese do mesmo xeito que stretchColumns.
 - ◆ Atributo de obxecto dentro dunha fila:
 - ◇ **android:layout_span**: unha cela X,Y pode ocupar o espazo de varias columnas.
 - Indícase o número de columnas que debe ocupar a cela.

1.3 Casos prácticos

- Comezamos, como sempre, creando un novo proxecto: **U2_06_Table**

1.3.1 Táboa simple

- A imaxe amosa a distribución que se quere ter da táboa:



- Cada cela ten un ancho que vén determinado pola lonxitude do seu contido.
- Observar como o ancho da columna 2 se adapta ao obxecto máis ancho.

- O XML do layout é:

```
<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 só se definen filas e logo créanse obxectos dentro destas, que serán as columnas.

1.3.2 Ocultar columnas

- Imos ocultar, neste caso, só a primeira columna (índice 0). Lembrar que o layout numera as columnas comezando en 0.



- A nosa columna 1 desapareceu.

- O XML do 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" />
    </TableRow>
</TableLayout>
```

```

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

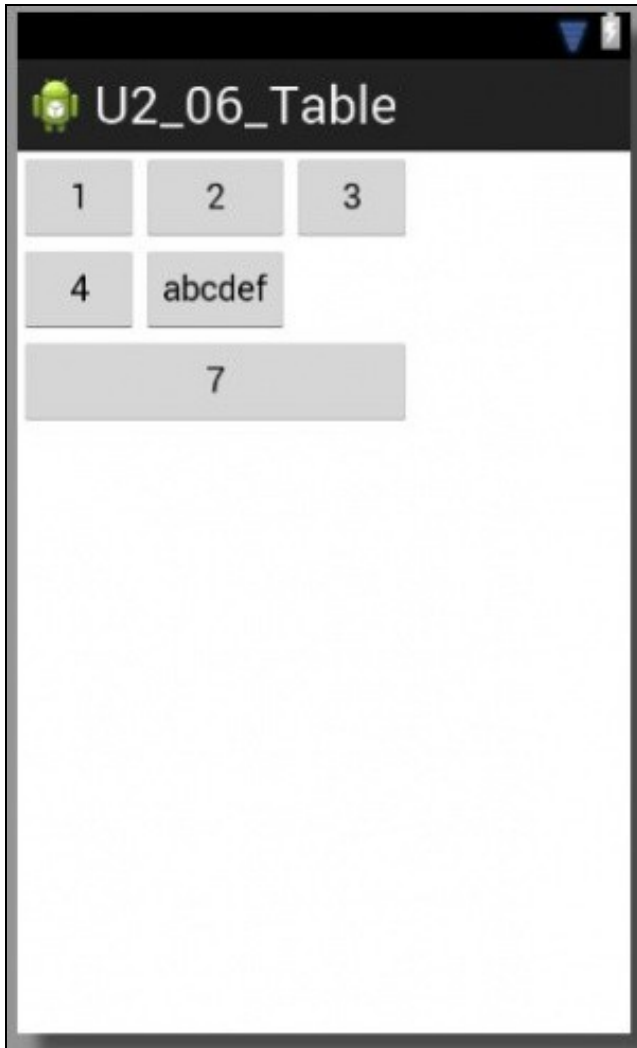
    <TableRow>

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

```

1.3.3 Cella ocupando varias columnas

- A cela 3,1 que ocupe todo o ancho da táboa:



- O XML do layout. Olo que non está a propiedade 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>

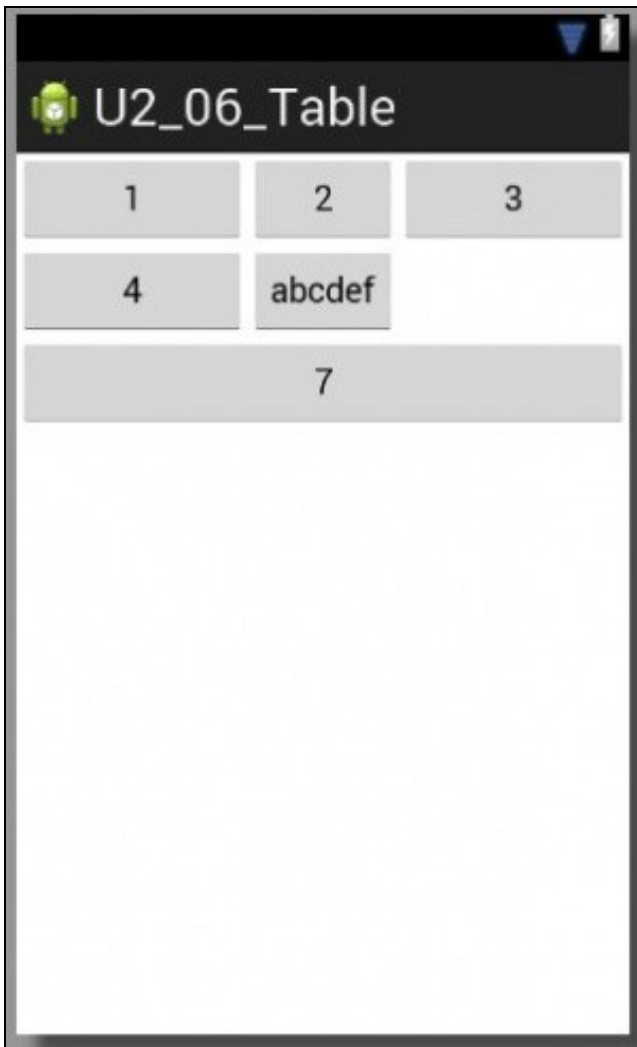
</TableLayout>

```

- Observar a liña marcada.

1.3.4 Expansión de columnas

- Imos facer que as columnas 1 (índice 0) e 3 (índice 2) se expandan ocupando o ancho da dereita da pantalla que deixa libre a táboa.



- Observar que a cela 3,1 segue ocupando todo o ancho da táboa.

- O XML do 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" />

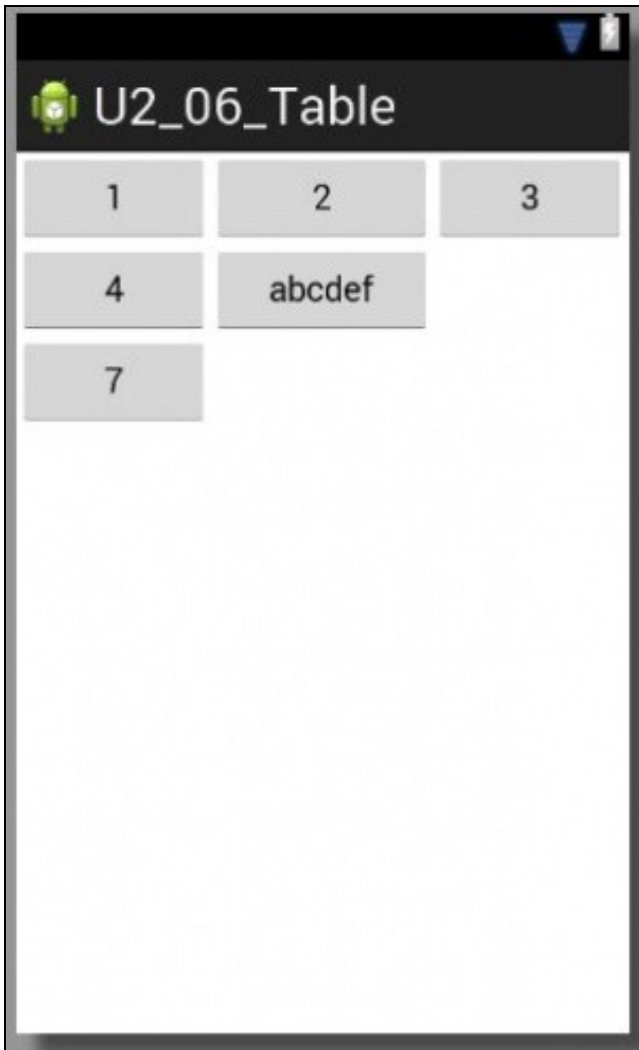
    <Button android:text="abcdef" />
</TableRow>

<TableRow>

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

</TableLayout>
```

- Finalmente, para o alumno: trata de crear o layout asociado á seguinte imaxe:
 - ◆ Usando un TableLayout
 - ◆ Sen modificar o tamaño de cada obxecto explicitamente
 - ◆ Hai dúas formas posibles de facelo



-- Ángel D. Fernández González e Carlos Carrión Álvarez -- (2015).