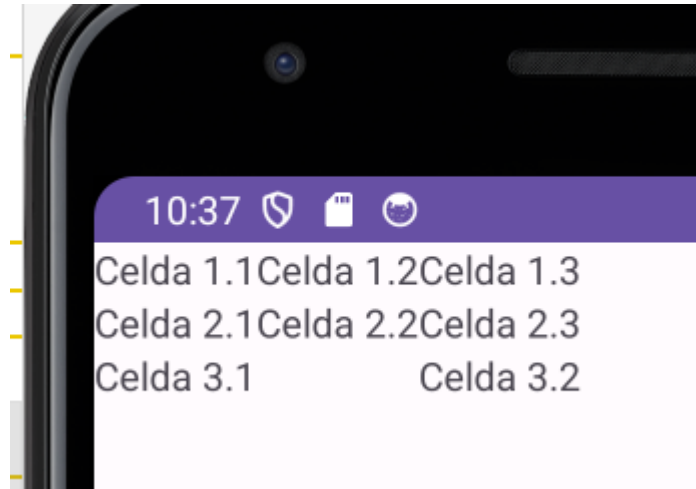


# TableLayout y GridLayout



```
<TableRow>
```

```
<AnalogClock
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"/>
```

```
<CheckBox
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Un checkBox"/>
```

```
</TableRow>
```

```
<TableRow>
```

```
<Button
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
```

```
android:text="Un botón"/>
```

```
<TextView
```

```
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="Un texto cualquiera"/>
</TableRow>
```

## GRIDLAYOUT

Similar al `TableLayout` pero se especifica el número de filas y columnas como propiedades del layout, mediante `android:rowCount` y `android:columnCount`.

También dispone de las propiedades `android:layout_rowSpan` y `android:layout_columnSpan` para conseguir que una celda ocupe el lugar de varias filas o columnas.

Se puede indicar de forma explícita la fila y columna que debe ocupar un determinado elemento hijo contenido en el `GridLayout`, y se consigue utilizando los atributos `android:layout_row` y `android:layout_column`.

```
Toast.makeText(this, "Botón presionado", Toast.LENGTH_SHORT).show();
```