
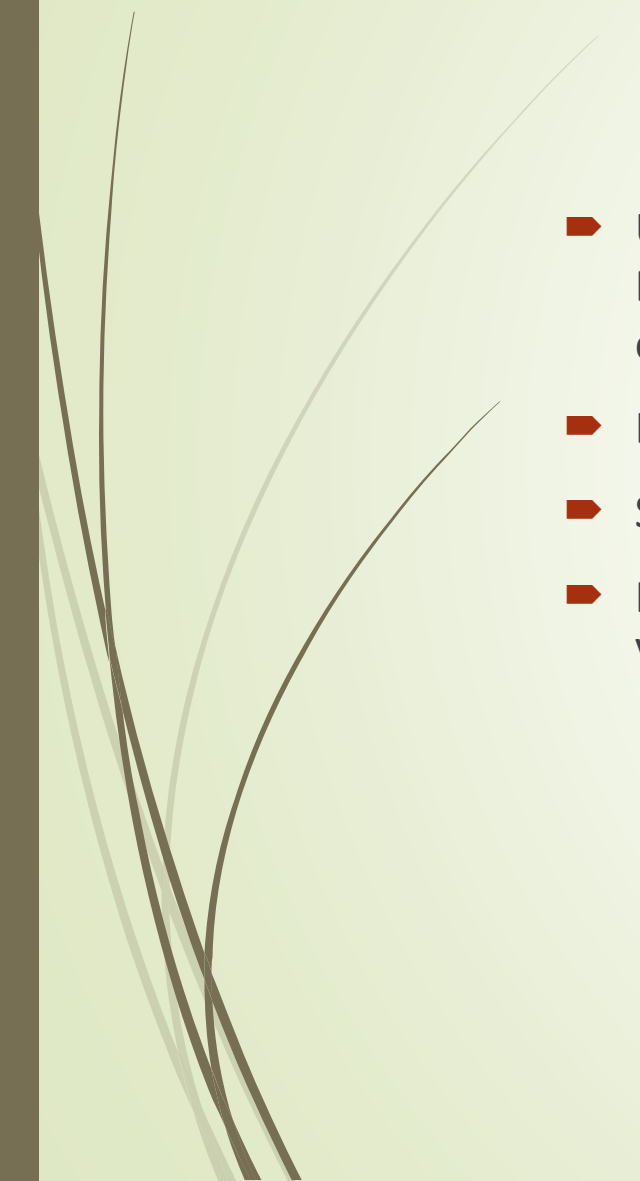



Layouts



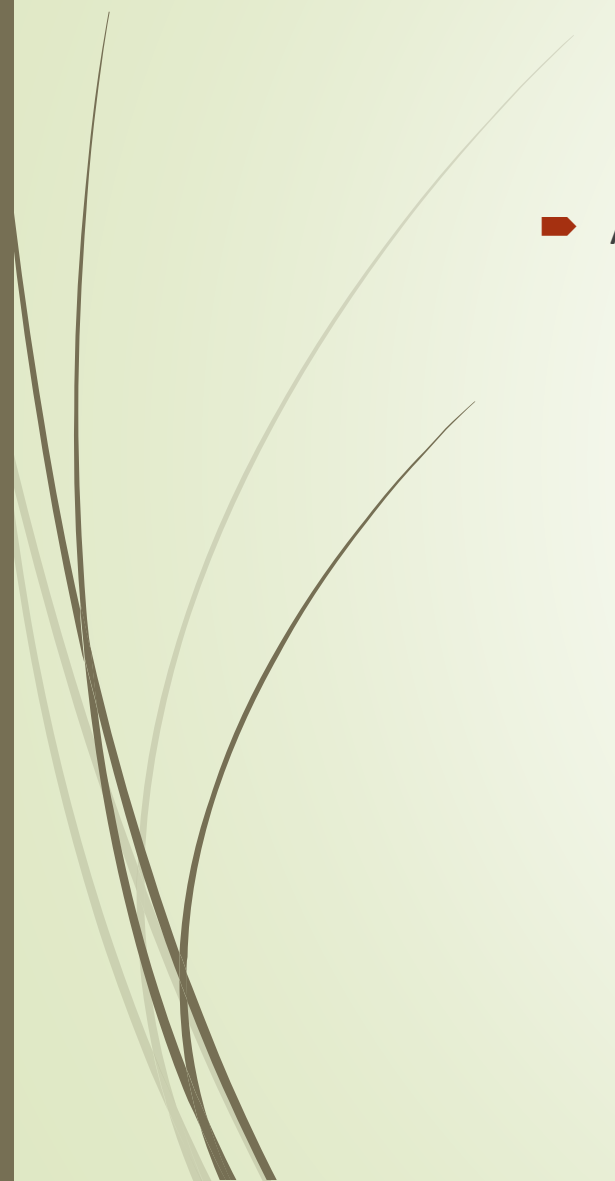
Layouts



- Un layout es un contenedor de una o más vistas y controla su comportamiento y posición. Desciende de la clase View. Son elementos no visibles que establecen cómo se distribuyen en la interfaz del usuario los componentes que incluyamos en su interior.
- La estructura general de un archivo de diseño de interfaz XML de Android es simple.
- Se trata de un árbol de elementos XML, donde cada nodo es el nombre de una clase View.
- Puedes utilizar el nombre de cualquier clase de tipo vista (View) de android o una clase View personalizada.




Layouts



➤ Atributos básicos:

- **xmlns:android:** Esta declaración indica que vamos a usar el espacio de nombres (la terminología) de Android para referirnos a los atributos que se definen a continuación.
- **android:id:** Asigna un identificador único para el elemento correspondiente. Este identificador sirve para poder acceder al componente desde el código fuente o desde las declaraciones de otros elementos en este archivo XML.
- **android:layout_width:** Define el ancho que debe ocupar la Vista. Es posible asignarle medidas absolutas o ajustar automáticamente mediante `wrap_content` o `match_parent`.
- **android:layout_height:** Define el largo que debe ocupar la Vista.
- **android:text:** Establece el texto que la view debe mostrar.



Layouts

- Los principales tipos de layouts:
 - `FrameLayout`
 - `LinearLayout`
 - `RelativeLayout`
 - `TableLayout`
 - `GridLayout`



FrameLayout

- Un FrameLayout es un view group creado para mostrar un solo elemento en pantalla. Sin embargo se puede añadir varios hijos con el fin de superponerlos, donde el ultimo hijo agregado es el que se muestra en la parte superior y el resto se pone por debajo en forma de pila. Para alinear cada elemento dentro del FrameLayout usa el parámetro **android:layout_gravity**.
- El parámetro gravity se basa en las posiciones comunes de un view dentro del layout. Su valores pueden ser:
 - Top
 - Left
 - Right
 - Bottom
 - Center_horizontal
 - Center_vertical
 - Center

FrameLayout

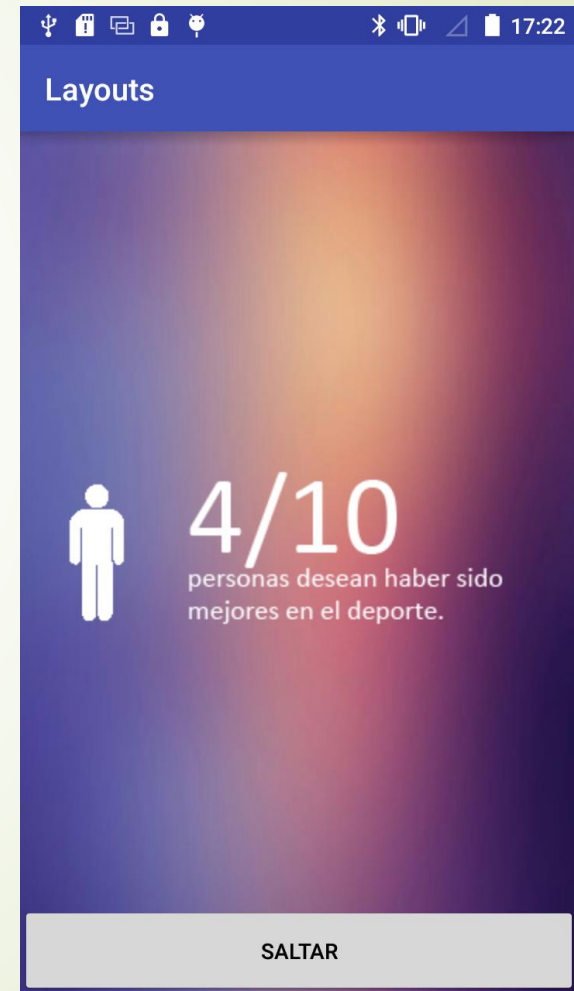
```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.android.curso.layouts.FrameActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:text="Saltar"
        android:id="@+id/boton_salutar"
        android:layout_gravity="center_horizontal|bottom"/>

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/image_background"
        android:layout_gravity="top|center"
        android:src="@drawable/background_frame_layout"
        android:scaleType="centerCrop"/>

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/image_estadistica"
        android:layout_gravity="center"
        android:src="@drawable/imagen_estadistica"
        android:padding="16dp"/>

</FrameLayout>
```





LinearLayout

- Un LinearLayout es un view group que distribuye sus hijos en una sola dimensión establecida. Todos están organizados en una sola columna (vertical) o en una sola fila (horizontal). La orientación puede elegirla a través del atributo **android:orientation**.
- Al igual que el FrameLayout, el LinearLayout permite asignar una gravedad a cada componente según el espacio que ocupa.
- Existe un parámetro llamado **android:layout_weight**, el cual define la importancia que tiene un view dentro del linear layout. A mayor importancia, más espacio podrá ocupar.
- Es posible definir la suma total del espacio con el atributo **android:weightSum**.

LinearLayout

The screenshot displays an Android application interface with a blue header bar labeled "Layouts". The main content area is light gray and contains a login form. At the top of the form is the text "Conectar". Below it are two input fields: "Correo" with a pink underline and "Contraseña" with a black underline. A gray button labeled "INICIAR SESION" is positioned below the password field. At the bottom of the form is a blue link that reads "¿Olvidaste tu contraseña?". The top of the screen shows a standard Android status bar with various icons and the time "17:23".



RelativeLayout

- Este elemento es el más flexible y elaborado de todos los view groups. El RelativeLayout permite especificar la posición de cada componente de forma relativa a su elemento padre o a cualquier otro elemento incluido en el propio layout.
- Posición relativa al layout padre:
 - **android:layout_alignParentLeft:** alinear a la izquierda.
 - **android:layout_alignParentRight:** alinear a la derecha.
 - **android:layout_alignParentTop:** alinear arriba.
 - **android:layout_alignParentBottom:** alinear abajo.
 - **android:layout_centerHorizontal:** alinear horizontalmente al centro.
 - **android:layout_centerVertical:** alinear verticalmente al centro.
 - **android:layout_centerInParent:** centrar.



RelativeLayout

- Propiedades: Posición relativa a otro control.
 - **android:layout_above**: arriba.
 - **android:layout_below**: debajo.
 - **android:layout_toLeftOf**: a la izquierda de.
 - **android:layout_toRightOf**: a la derecha de.
 - **android:layout_alignLeft**: alinear a la izquierda.
 - **android:layout_alignRight**: alinear a la derecha.
 - **android:layout_alignTop**: alinear arriba.
 - **android:layout_alignBottom**: alinear abajo.
 - **android:layout_alignBaseline**: alinear en la base.



RelativeLayout

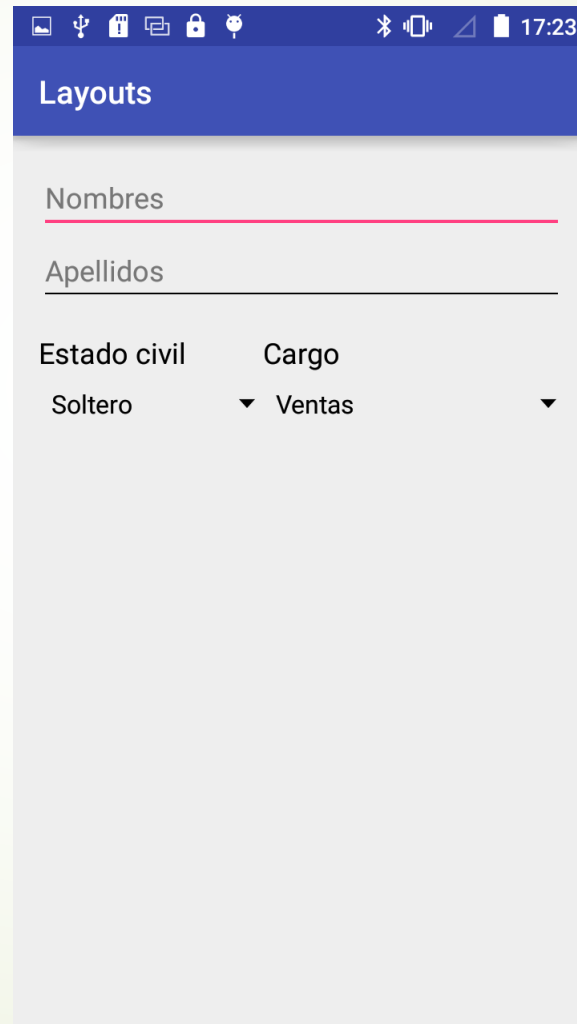
- Propiedades: Opciones de margen(también disponibles en el resto de layouts).
 - **android:layout_margin:** establece el margen.
 - **android:layout_marginBottom:** establece el margen inferior.
 - **android:layout_marginTop:** establece el margen superior.
 - **android:layout_marginLeft:** establece el margen izquierdo.
 - **android:layout_marginRight:** establece el margen derecho.



RelativeLayout

- Propiedades: Opciones de espaciado o padding.
 - **android:padding:** establece la separación entre componentes.
 - **android:paddingBottom:** establece la separación inferior.
 - **android:paddingTop:** establece la separación superior.
 - **android:paddingLeft:** establece la separación izquierda.
 - **android:paddingRight:** establece la separación derecha.

RelativeLayout



The screenshot displays an Android application interface with a blue header bar labeled "Layouts". Below the header, there is a form with the following elements:

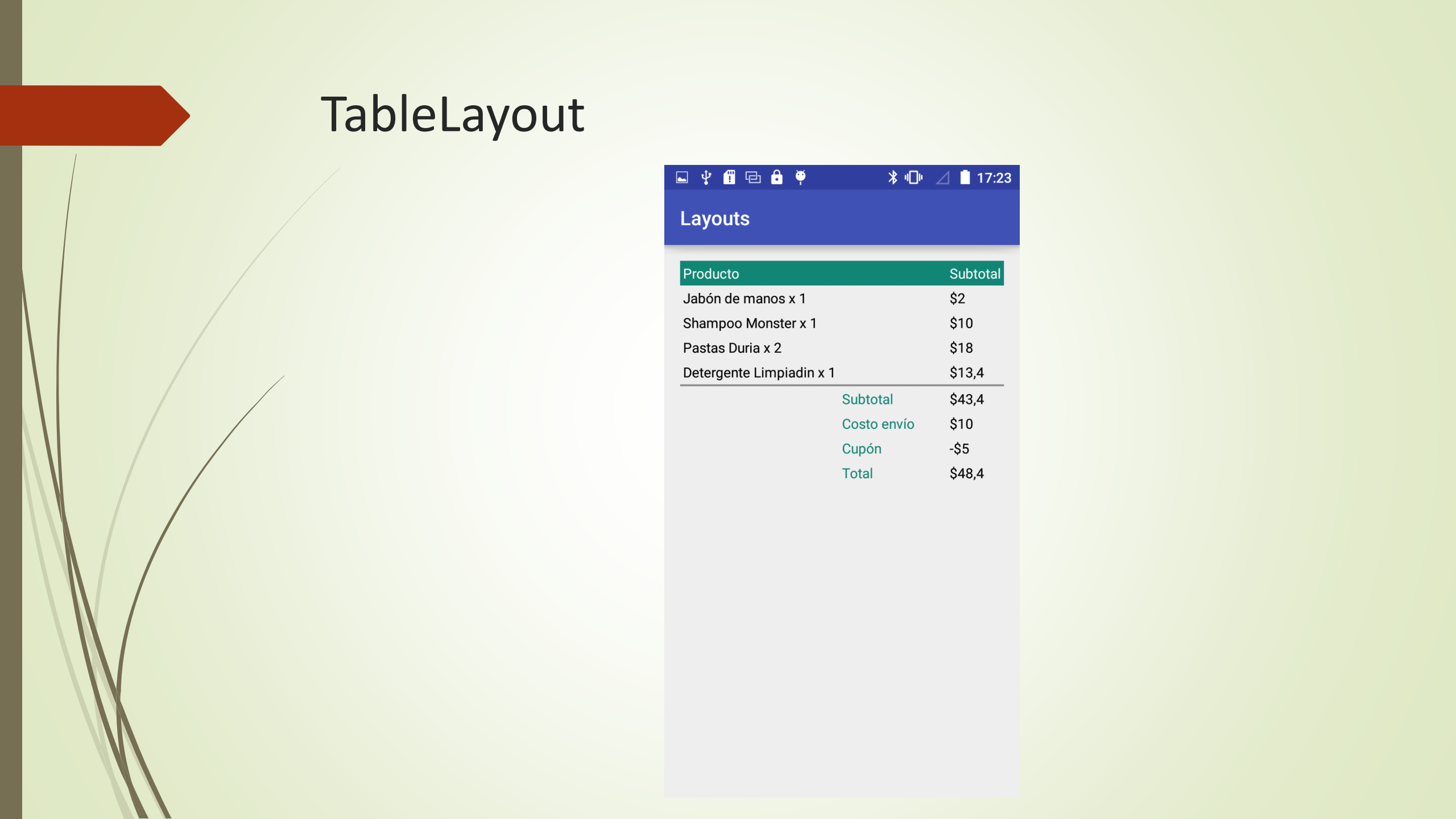
- A text input field labeled "Nombres" with a pink underline.
- A text input field labeled "Apellidos" with a black underline.
- A section with two labels: "Estado civil" and "Cargo".
- Under "Estado civil", there is a text input field containing "Soltero".
- Under "Cargo", there is a dropdown menu showing "Ventas" with a downward arrow on the right.



TableLayout

- El TableLayout organiza views en filas y columnas de forma tabular.
- Para crear filas se utiliza el componente TableRow dentro del TableLayout. Cada celda es declarada como un view de cualquier tipo dentro de la fila.
- El parámetro **android:layout_column** asigna la columna a la que pertenece cada celda en su interior. Se puede asignar peso a las celdas mediante **weight**. El ancho de cada columna es definido tomando como referencia la celda más ancha. Pero también se puede definir su comportamiento como los atributos:
 - **android:shrinkColumns**: Reduce el ancho de la columna seleccionada hasta ajustar la fila al tamaño del padre.
 - **android:stretchColumn**: Permite rellenar el espacio vacío que queda en el TableLayout, expandiendo la columna seleccionada.

TableLayout



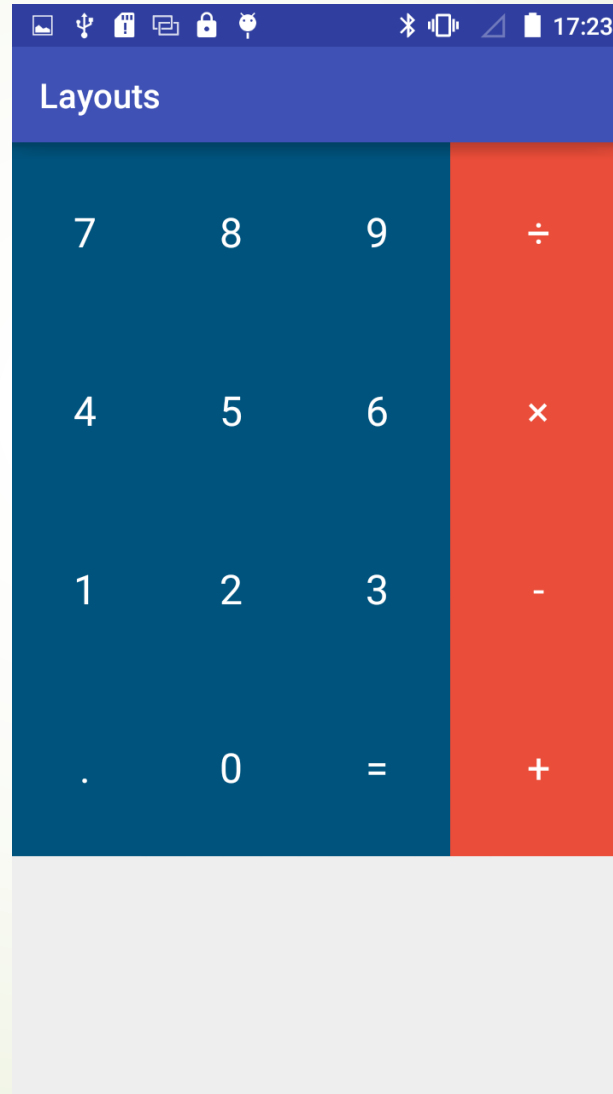
Layouts	
Producto	Subtotal
Jabón de manos x 1	\$2
Shampoo Monster x 1	\$10
Pastas Duria x 2	\$18
Detergente Limpiadin x 1	\$13,4
<hr/>	
Subtotal	\$43,4
Costo envío	\$10
Cupón	-\$5
Total	\$48,4



GridLayout

- Un GridLayout es un ViewGroup que alinea sus elementos hijos en una cuadrícula.
- Los atributos más importantes del GridLayout son:
 - **columnCount**: Cantidad de columnas que tendrá.
 - **rowCount**: Cantidad de filas de la cuadrícula.
 - **useDefaultMargins**: Si asignas el valor de true para establecer márgenes predeterminados entre los ítems.
- Es posible especificar la cantidad de filas y columnas que ocupará una celda a través de los atributos **android:layout_rowSpan** y **android:layout_columnSpan**.

GridLayout





➡ <https://github.com/rononunsys/Layouts.git>