

1 LinearLayout

1.1 Sumario

- 1 Introducción
- 2 Casos prácticos
 - ♦ 2.1 A propiedade `android:orientation`
 - ♦ 2.2 A propiedade `android:layout_weight`
 - ♦ 2.3 As propiedades `android:gravity` e `android:margin`
 - ♦ 2.4 Padding

1.2 Introducción

- É un dos Layouts máis comúns e máis usados.
- Dispón as vistas (os elementos fillos) nunha sola orientación: ou en vertical ou en horizontal. Un a continuación do outro.
- Para iso ten o atributo: **`android:orientation`**



- Este layout ten a propiedade **`android:layout_weight`** para as vistas.
- Esta propiedade indica canto debe ocupar ese view do espazo restante do seu contenedor, de forma proporcional ao weight (peso) que se lle asine.
- Así, se temos dúas caixas de texto e asinamos un weight (peso) de 2 e 1 respectivamente, estaremos establecendo un ancho o dobre de grande na primeira caixa.
- Se temos 3 caixas e a unha delas lle poñemos peso e as outras non, a que ten o peso asinado ocupará todo o espazo que poida do seu contedor pai, quedando as outras dúas axustadas o tamaño indicado no arquivo xml.
- Se asignamos un `layout_weight`, normalmenteponse un `layout_width=?0dp?` ou `layout_height=?0dp?`.

- Referencias:

- ♦ LinearLayout: <http://developer.android.com/guide/topics/ui/layout/linear.html>

1.3 Casos prácticos

- Comezamos creando un novo proxecto: **`U2_04_Linear`**.

1.3.1 A propiedade android:orientation

- Comezamos cambiando o xml do layout por este outro:

```
<LinearLayout 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:orientation="horizontal" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 1"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 3" />

</LinearLayout>
```

- Observar como se van orientar os elementos fillos (vistas) no layout. Neste caso en horizontal.
- Neste exemplo os elementos fillos (vistas) son botóns.



- Agora cambiamos a orientación a **vertical**

```
<LinearLayout 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:orientation="vertical" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```

        android:text="Preme 1"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Preme 2" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Preme 3" />

</LinearLayout>

```



1.3.2 A propriedade `android:layout_weight`

- Engadimos/modificamos as propriedades marcadas no seguinte ficheiro XML.

```

<LinearLayout 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:orientation="vertical" >

    <Button
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:text="Preme 1"/>

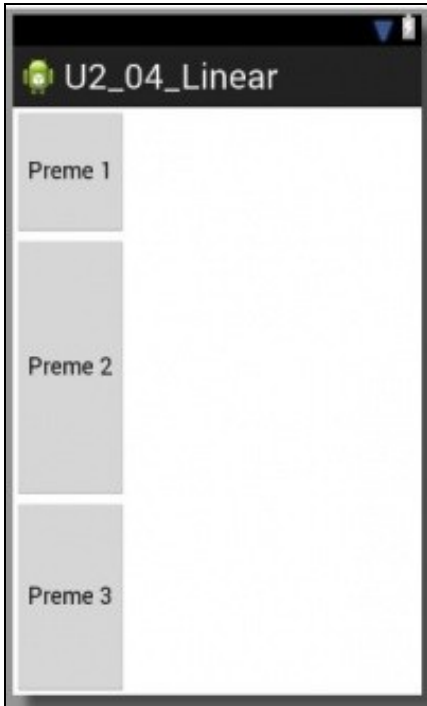
    <Button
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_weight="3"
        android:text="Preme 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="2"
        android:text="Preme 3" />

```

```
</LinearLayout>
```

- Preténdese que se o espazo sobranse no layout en vertical se pode dividir en 6 anacos (a suma dos pesos), pois que a primeira view ocupe 1 anaco, a segunda view 3 anacos e a terceira view o restante, 2 anacos.
- Observar como dá igual o alto que se indique en calquera das tres vistas **android:height**.



1.3.3 As propiedades android:gravity e android:margin

- Modificamos o layout anterior para deixalo como:

```
<LinearLayout 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:orientation="vertical" >

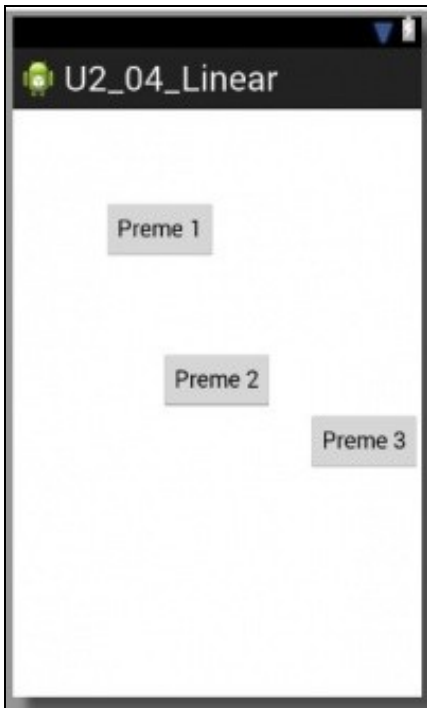
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="70dp"
        android:text="Preme 1"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Preme 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="Preme 3" />

</LinearLayout>
```

- Observar na seguinte imaxe a ubicación dos botóns.



1.3.4 Padding

- Vaise realizar recheo no Layout e logo nalgún dos botóns.
- No seguinte XML realizase padding no propio Layout.
- Co cal, cada elemento do interior estará a 20dp do borde do layout.

```
<LinearLayout 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:paddingBottom="20dp"
    android:paddingLeft="20dp"
    android:paddingRight="20dp"
    android:paddingTop="20dp"
    android:orientation="vertical" >

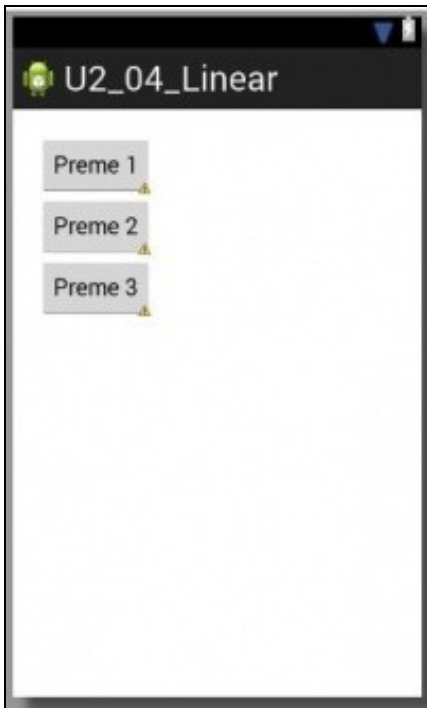
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 1"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Preme 3" />

</LinearLayout>
```

- A que saberías como substituír as liñas marcadas por unha soa propiedade?
- Observar como os botóns están separados dos bordes da esquerda e superior.



- Finalmente imos facer padding nunha vista.

```
<LinearLayout 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:padding="20dp"
    android:orientation="vertical" >

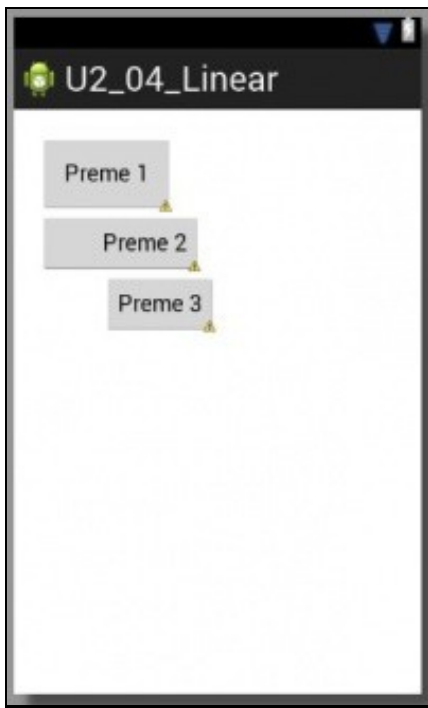
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="20dp"
        android:text="Preme 1"/>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingLeft="50dp"
        android:text="Preme 2" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="50dp"
        android:text="Preme 3" />

</LinearLayout>
```

- Observar as liñas marcadas e o seu efecto no contido das vistas na seguinte imaxe. Apreciar a diferenza entre marxe e recheo, entre os Botóns **Preme 2** e **Preme 3**.



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