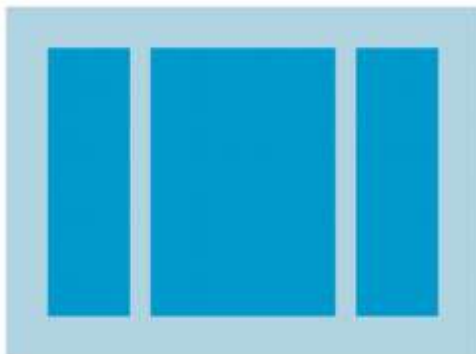


INDICE

<u>1.1.</u>	<u>INTRODUCCIÓN</u>	<u>1</u>
<u>1.2.</u>	<u>CASO PRÁCTICO</u>	<u>2</u>
<u>1.2.1.</u>	<u>LA PROPIEDAD ANDROID:ORIENTATION</u>	<u>2</u>
<u>1.2.2.</u>	<u>LA PROPIEDAD ANDROID:LAYOUT_WEIGHT</u>	<u>4</u>
<u>1.2.3.</u>	<u>LA PROPIEDAD ANDROID:GRAVITY Y ANDROID:MARGIN</u>	<u>5</u>
<u>1.2.4.</u>	<u>PADDING</u>	<u>6</u>

1.1. Introducción

- Es uno de los Layouts más comunes y más usados.
- Dispone las vistas (los elementos hijos) en una sola orientación: o en vertical o en horizontal. Uno a continuación de otro.
- Para esto tiene el atributo: **android:orientation**



- Este layout tiene una propiedad **android:layout_weight** para las vistas.
- Esta propiedad indica cuanto debe ocupar ese view del espacio restante de su contenedor, de forma proporcional al weight (peso) que se le asigne.
- Así, si tenemos dos cajas de texto y asignamos un weight (peso) de 2 e 1 respectivamente, estaremos estableciendo un ancho o doble de grande en la primera caja.
- Si tenemos 3 cajas y una de ellas le ponemos peso y a las otras no, la que tiene el peso asignado ocupará todo el espacio que pueda de su contenedor padre, quedando las otras dos ajustadas al tamaño indicado en el archivo xml.
- Si asignamos un layout_weight, normalmente se pone un layout_width="0dp" o un layout_height="0dp".

➤ Referencias:

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

1.2. Caso Práctico

- Comenzamos creando un nuevo proyecto: **U2_04_Linear**.

1.2.1. La propiedad android:orientation

- Comenzamos cambiando el XML del layout por este otro:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```

- Observar cómo se van orientar los elementos hijos (vistas) en el layout. En este caso en horizontal.
- En este ejemplo los elementos hijos (vistas) son botones.



➤ Ahora cambiamos a orientación en **vertical**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```



1.2.2. La propiedad android:layout_weight

➤ Añadimos/modificamos las propiedades marcadas en el siguiente fichero XML.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```

- Se pretende que si el espacio sobrante en el layout en vertical se puede dividir en 6 huecos (la suma de los pesos), pues que la primera view ocupe 1 hueco, la segunda view 3 huecos y la tercera view el restante, 2 huecos.
- Observar como da igual el alto que se indique en cualquiera de las tres vistas **android:height**.



1.2.3. La propiedad android:gravity y android:margin

- Modificamos el layout anterior para dejarlo como:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```

- Observar en la siguiente imagen la ubicación de los botones.



1.2.4. Padding

- Se va a realizar relleno en el Layout y alguno de los botones.
- En el siguiente XML se realiza padding en el propio Layout.
- Con lo cual, cada elemento del interior estará a 20dp del borde del layout.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    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"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```

- ¿A qué sabrías como substituir las líneas marcadas por una sola propiedad?
- Observar como los botones están separados dos bordes da izquierda y superior.



- Finalmente vamos a hacer padding de una vista.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    android:orientation="vertical"
    tools:context=".UI.Layouts.U2_04_Linear">

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

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

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

</LinearLayout>
```

- Observar las líneas marcadas y su efecto en el contenido de las vistas en la siguiente imagen. Apreciará la diferencia entre margen y relleno, entre los Botones Pulsar 2 y Pulsar 3.

