

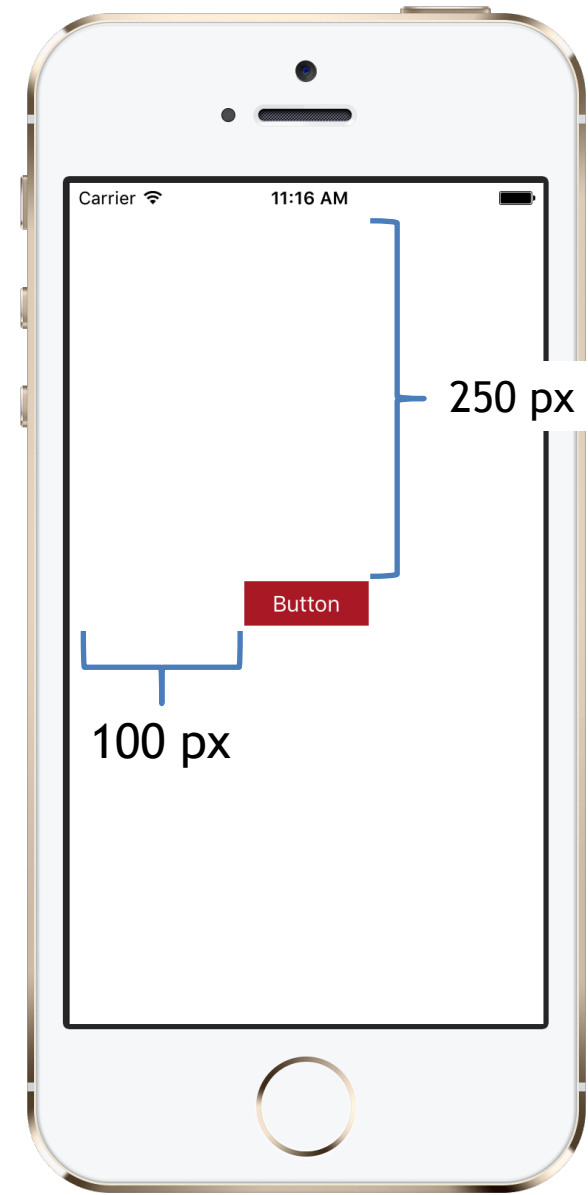
# Layout

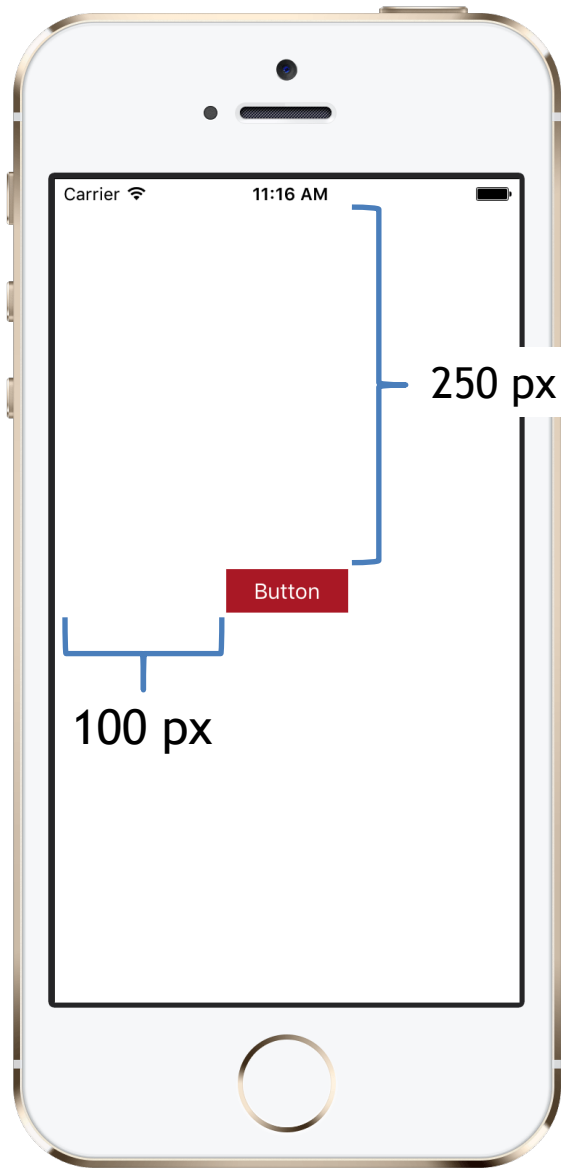
# Layout

**Problema: Como posicionar os elementos gráficos no écran?**

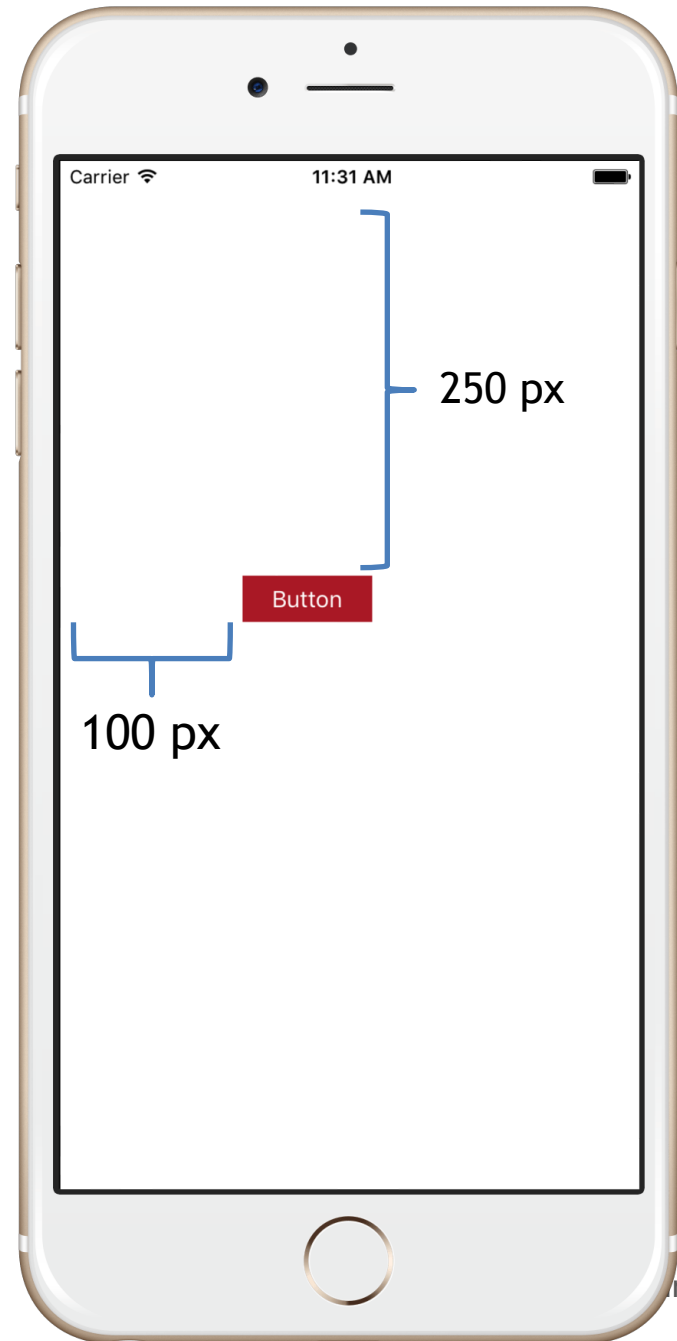
**Hipótese: Indico as coordenadas de cada elemento, no exemplo: (100, 250)**

A isto chama-se posicionamento absoluto



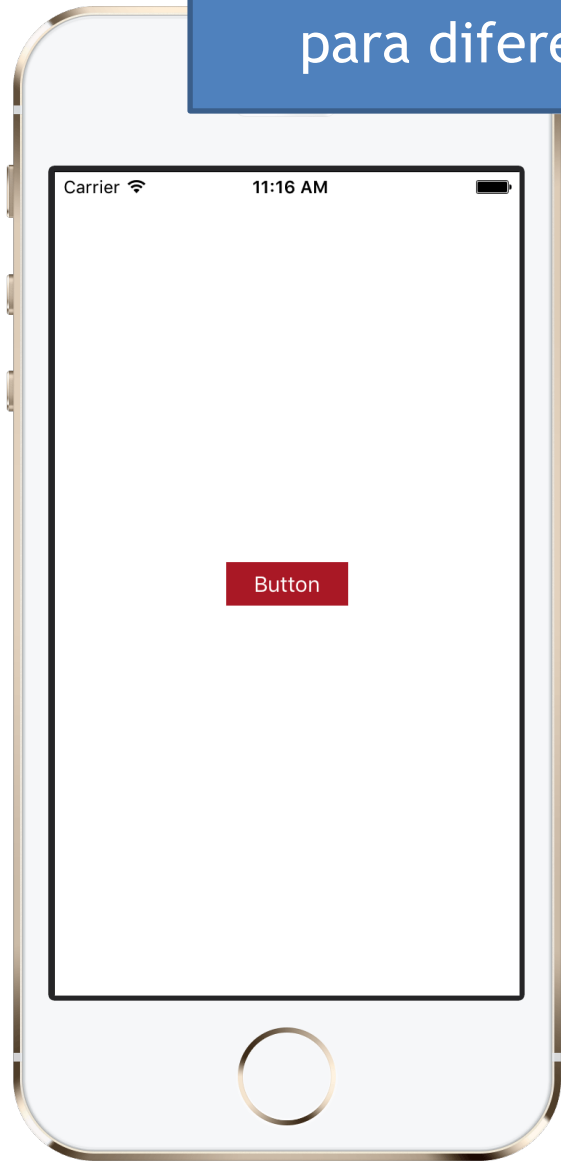


iPhone 5

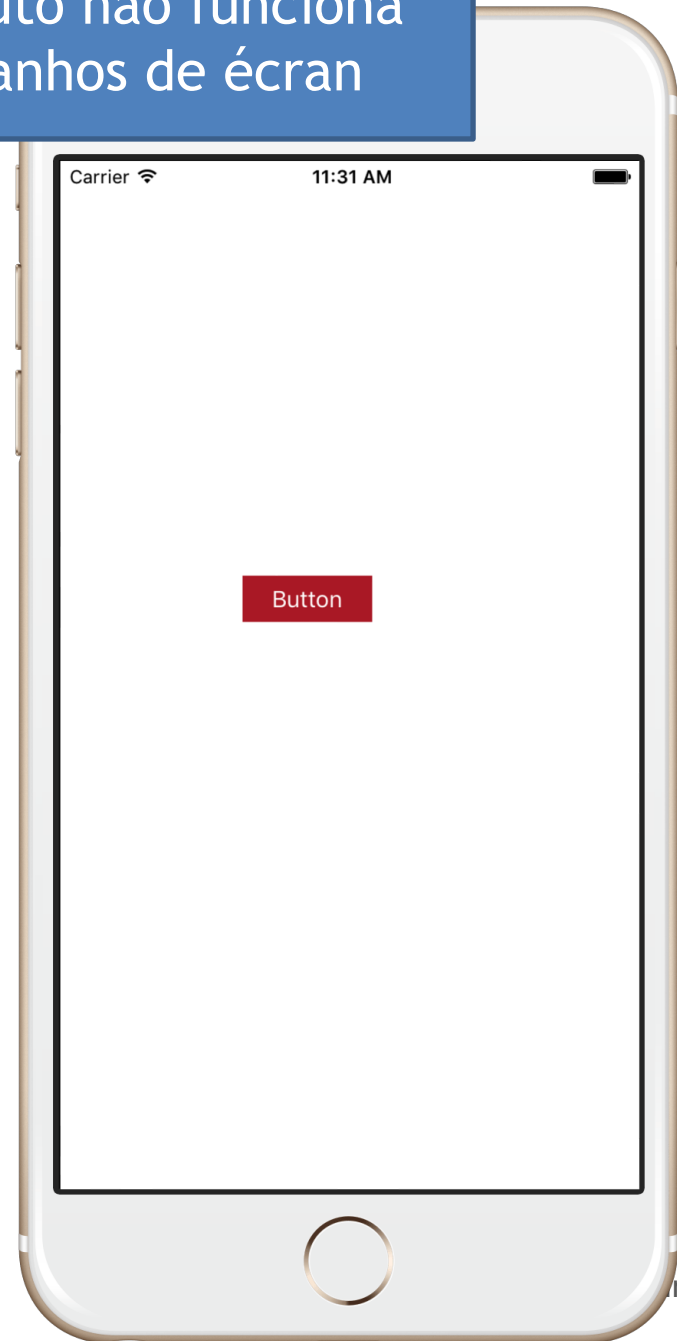


iPhone 6

Posicionamento absoluto não funciona  
para diferentes tamanhos de écran



iPhone 5



iPhone 6

# Layout

**Problema: Como posicionar os elementos gráficos no écran?**

Hipótese 2: Indico as posições relativas dos elementos:

- Relativas às margens do écran
- Relativas a outros elementos

**Exemplos:**

O botão A deve estar a 10 px para direita do botão B

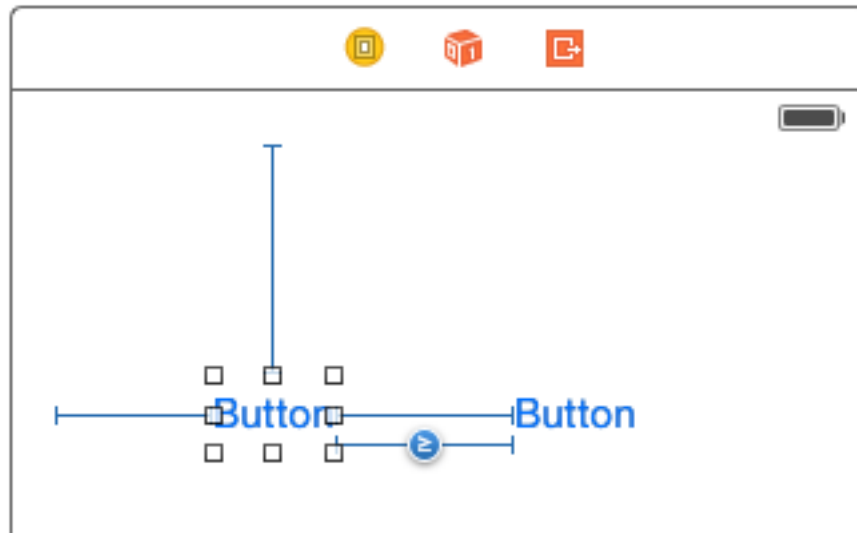
O botão A deve estar centrado verticalmente no écran

O botão A e o botão B devem estar alinhados horizontalmente

A isto chama-se posicionamento relativo

# Layout

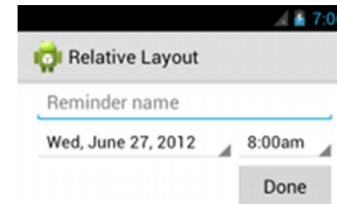
Aplicações iOS utilizam posicionamento relativo.  
Esse posicionamento é feito normalmente através do editor gráfico do XCode mas também pode ser feito programaticamente.



# Layout

Aplicações Android utilizam posicionamento relativo. Esse posicionamento é feito normalmente no ficheiro de layout em XML mas também pode ser feito programaticamente.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="16dp"
    android:paddingRight="16dp" >
    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/reminder" />
    <Spinner
        android:id="@+id/dates"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentLeft="true"
        android:layout_toLeftOf="@+id/times" />
    <Spinner
        android:id="@+id/times"
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/name"
        android:layout_alignParentRight="true" />
    <Button
        android:layout_width="96dp"
        android:layout_height="wrap_content"
        android:layout_below="@id/times"
        android:layout_alignParentRight="true"
        android:text="@string/done" />
</RelativeLayout>
```

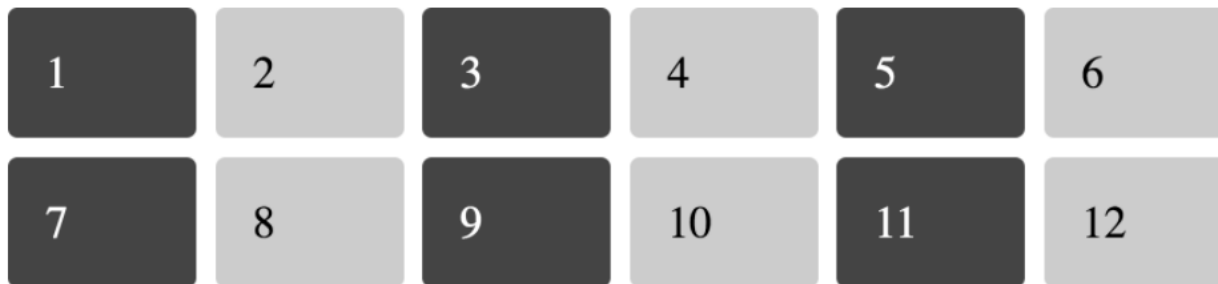


o Spinner dates fica por baixo do EditText name

# Layout

**Problema: Como posicionar os elementos gráficos no écran?**

Hipótese 3: Crio uma grelha “virtual” e distribuo os elementos por essa grelha



A isto chama-se posicionamento em grelha

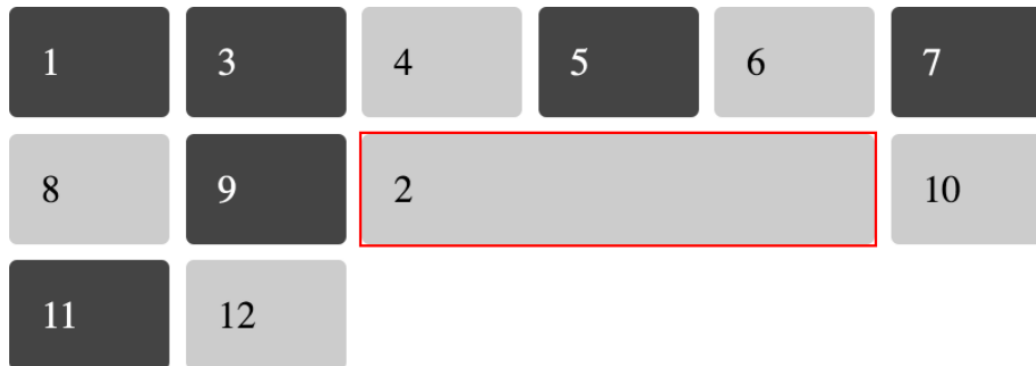


# Layout

## Posicionamento em grelha

O primeiro passo é decidir quantas colunas tem a grelha  
Por exemplo, o Bootstrap define uma grelha com 12 colunas.

Para cada elemento podemos indicar quantas colunas esse elemento pode ocupar.



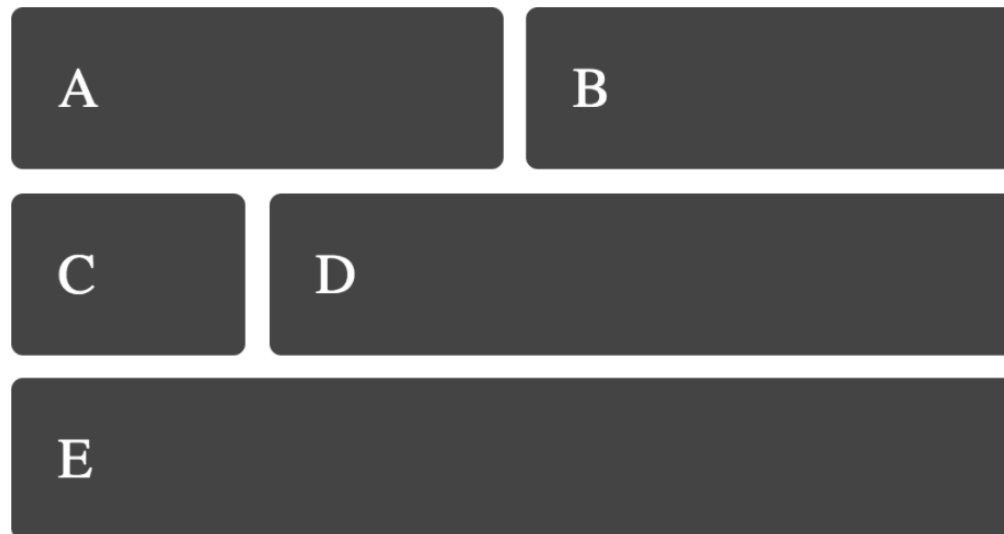
O elemento 2  
ocupa 3 colunas

Esta grelha tem 7  
colunas

# Layout

## Posicionamento em grelha

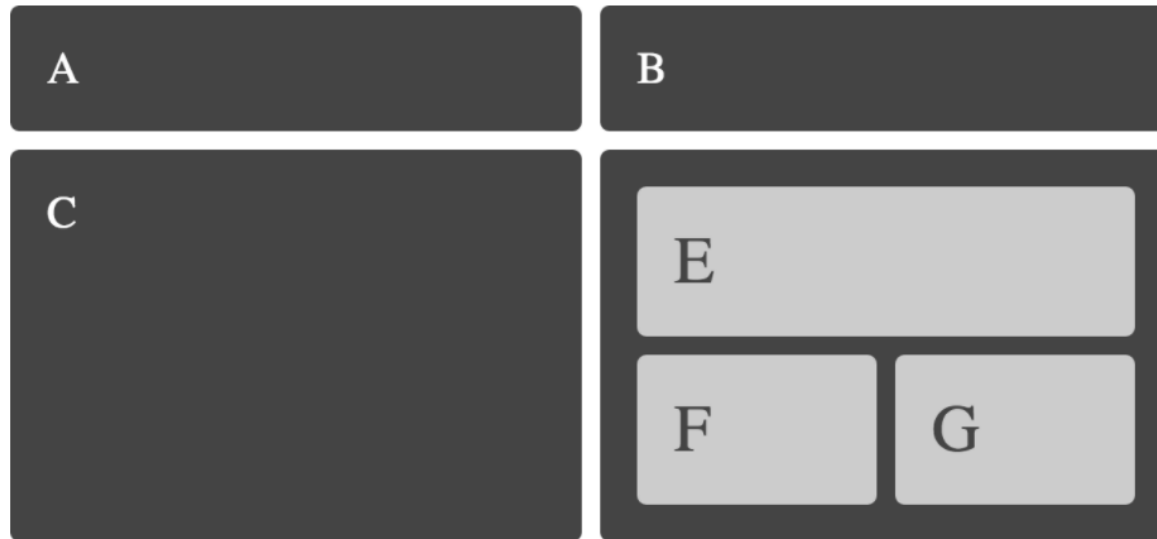
Combinando diferentes larguras consegue-se definir uma grande diversidade de écrans



# Layout

## Posicionamento em grelha

Por vezes é necessário colocar layouts dentro de layouts (nested grids). Nestes casos, o elemento passa a conter outros elementos



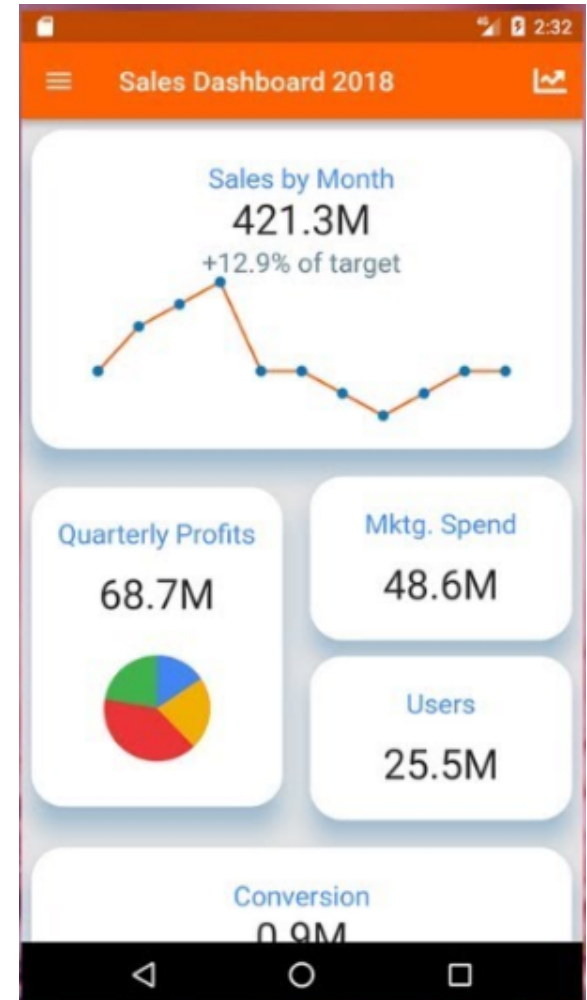
# Layout

## Posicionamento em grelha

O Flutter, através da combinação dos widgets Row e Column, favorece o posicionamento em grelha

Column

- SalesByMonth
- Row
  - QuarterlyProfits
  - Column
    - MktSpend
    - Users
- Conversion



# Layout

## Posicionamento em grelha

Este posicionamento permite também maximizar a utilização do espaço disponível, facilitando a interação em touch-screens

Ex: “Metro” interface do Windows 10



# Layout

Posicionamento	Usado quando...	Tecnologia
Absoluto	A app vai ser usado num tamanho de écran único ou queremos controlar ao pixel o posicionamento dos elementos (ex: jogos)	Todas
Relativo	Queremos distribuir os elementos sempre da mesma forma em vários tamanhos de écran	iOS, Android
Grelha	Queremos maximizar a utilização do espaço disponível e facilitar a interação em touch screens	HTML5, Flutter, Windows 10