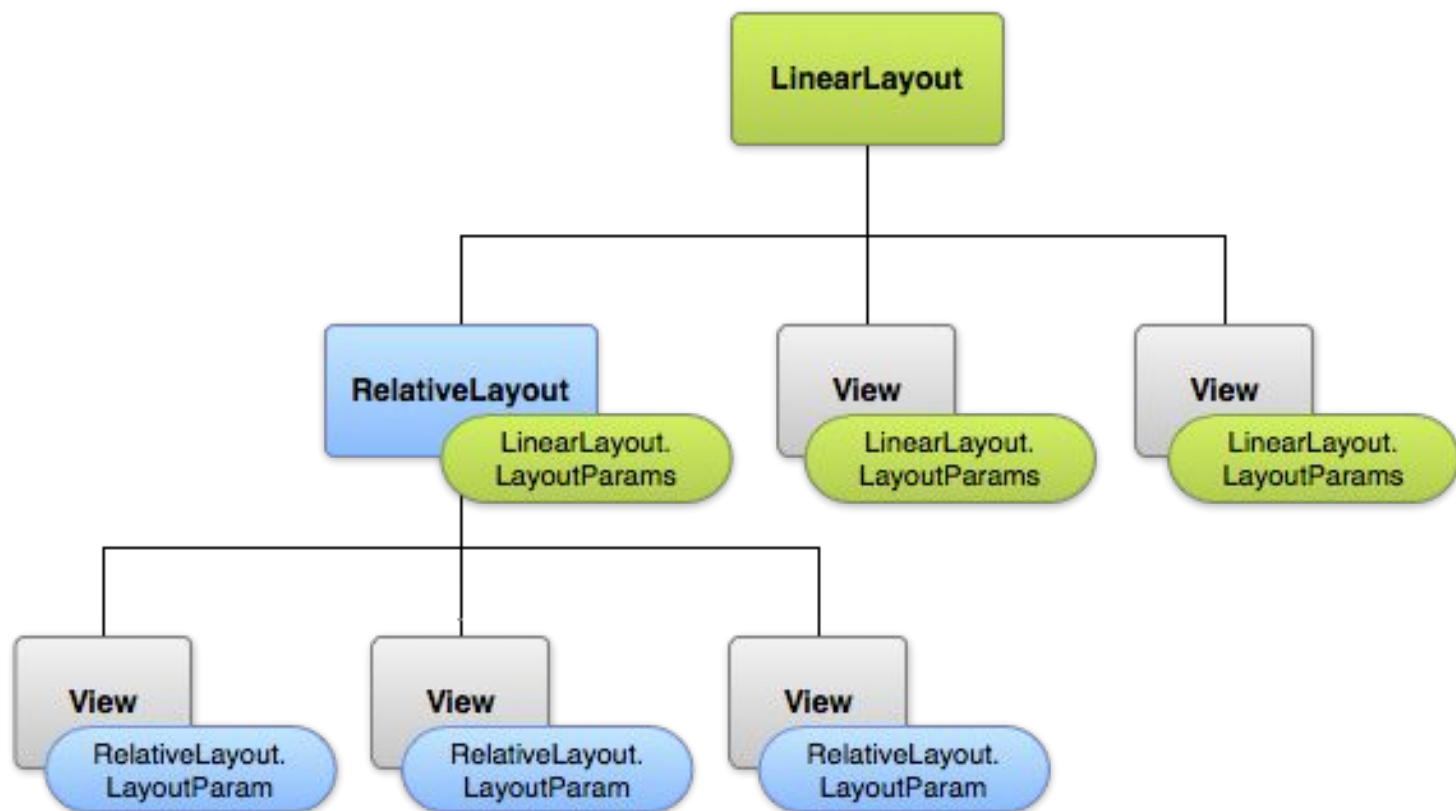


Android: Gerenciadore s de Layout



Gerenciadores de Layout

- Componentes que **organizam os elementos visuais** tanto no sentido horizontal quanto no vertical, bem como em linhas e colunas.
- Esses componentes **podem receber outros componentes dentro deles** e ajustar automaticamente sua aparência.
- São herdados diretamente da classe **android.view.ViewGroup**.

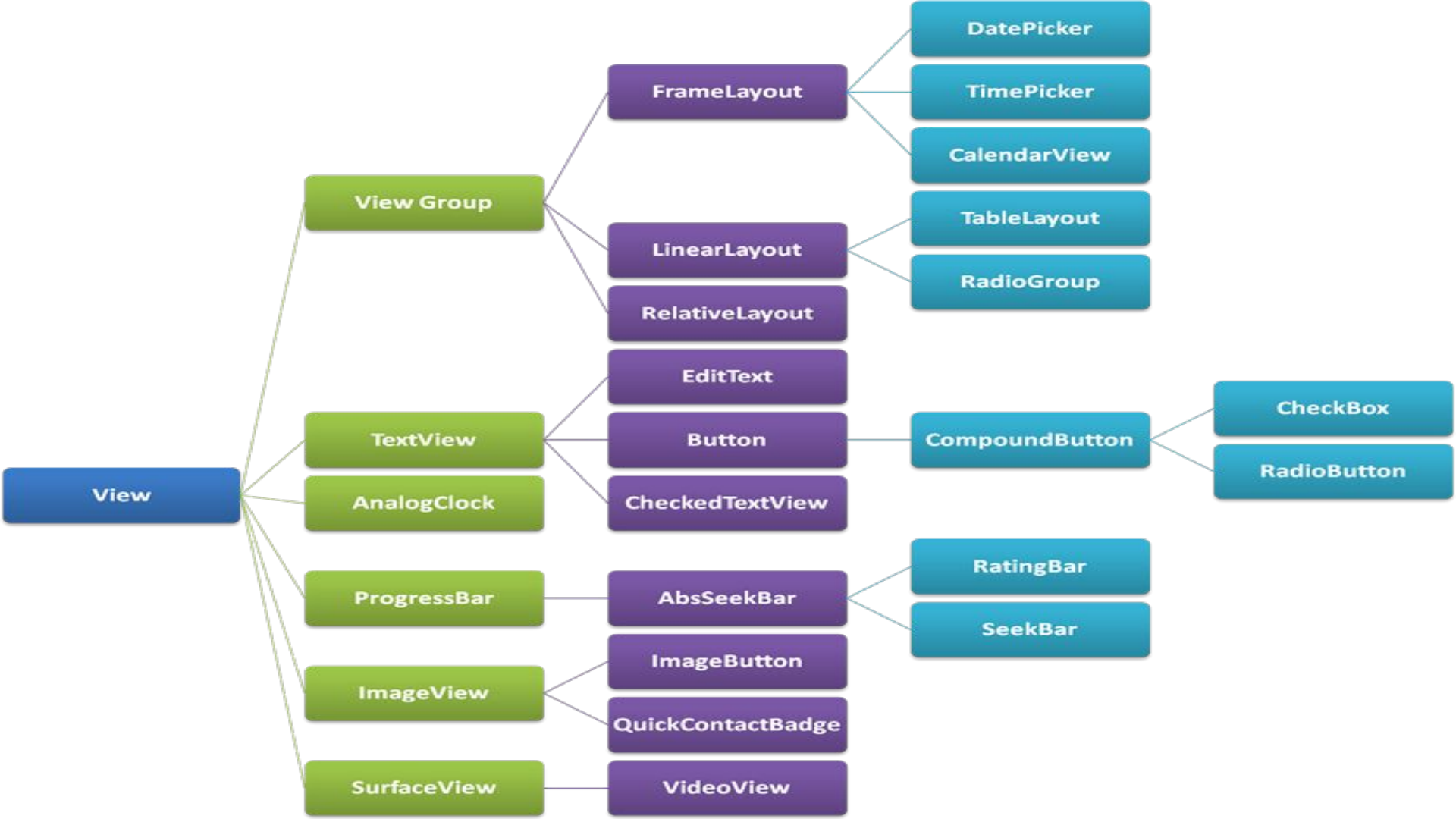


ViewGroup

- A classe `android.view.ViewGroup` é a classe mãe de todos os gerenciadores de componentes visuais do Android.

View

- A classe `android.view.View` é a classe mãe de todos os componentes visuais do Android.
- Cada subclasse de `View` precisa implementar o método `onDraw()` para desenhar o componente na tela.



Widgets

- Componentes simples que herdam diretamente da classe View.
- Exemplos: Button, TextView, ImageView e etc...

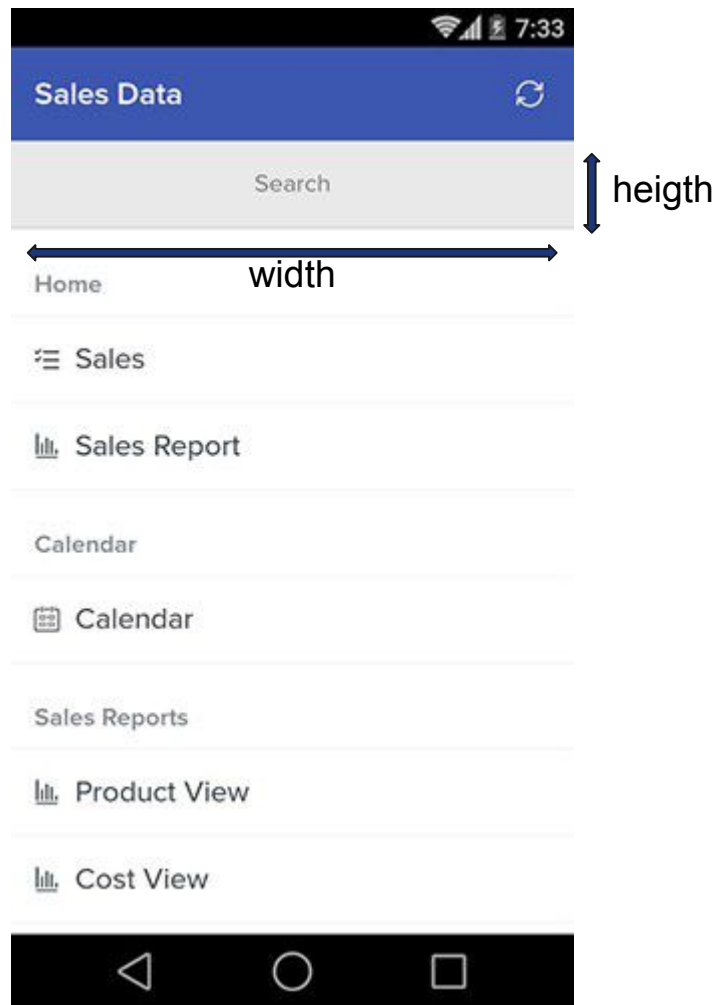
Layouts

- Componentes robustos que herdam diretamente da classe ViewGroup.
- Exemplos: LinearLayout, RelativeLayout, FrameLayout e etc...

Dimensões das Views

- Altura:
 - `android:layout_height`
- Largura:
 - `android:layout_width`

Largura e altura



Valores dos parâmetros

- Utilizando as constantes
 - *fill_parent* e *match_parent*: a view irá ocupar todo o tamanho definido por seu pai (layout)
 - *wrap_content*: a view irá ocupar apenas o tamanho necessário para exibir seu conteúdo na tela.
- Utilizando número (dp)
 - número inteiro na notação dp (density independent pixel) especificando o valor fixo do tamanho a ser ocupado na tela.
 - A notação dp faz a conversão correta de pixels conforme a densidade/resolução da tela do dispositivo.
 - Evitar utilizar a notação px.

```
android:layout_width = "100dp"  
android:layout_width = "100px"
```

Gravity

- A propriedade *android:gravity* define o alinhamento interno do componente.
- Já a propriedade *android:layout_gravity* alinha o componente dentro do gerenciador de layout.



```
<LinearLayout xmlns:android=
    "http://schemas.android.com/apk/res/
    android"
    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="Primeiro" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Segundo Botão" />

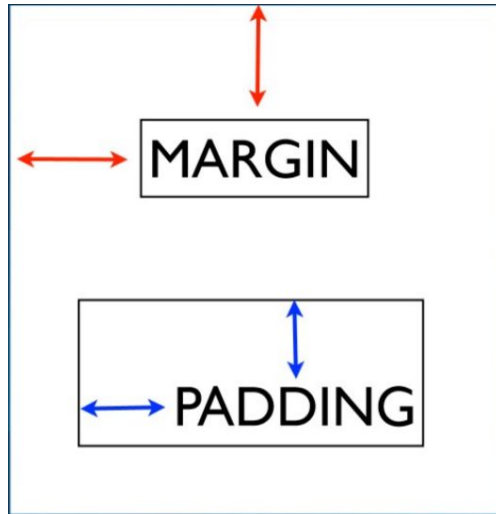
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Terceiro" />

    <Button
        android:layout_width="200dp"
        android:layout_height="match_parent"
        android:text="Quarto" />

</LinearLayout>
```

Padding e Marging

```
android:layout_marginLeft="10dp"  
android:layout_marginTop="10dp"  
android:layout_marginRight="10dp"  
android:layout_marginBottom="10dp"  
android:padding="10dp"  
android:paddingLeft="10dp"  
android:paddingTop="10dp"  
android:paddingRight="10dp"  
android:paddingBottom="10dp"
```



LinearLayout

- É utilizado para distribuir os componentes na vertical ou horizontal.
- A classe *android.widget.LinearLayout* organiza a orientação através do atributo *android:orientation*.
- O alinhamento dos componentes dentro dele é feito pelo atributo *android:layout_gravity*.
- O atributo *android:layout_weight* distribui os elementos na tela pelo peso. O mais pesado será maior.

Exercício 10

LinearLayout

To

Subject

Message

SEND



RelativeLayout

- A classe *android.widget.RelativeLayout* permite posicionar um componente relativo a outro, por exemplo, abaixo, acima ou ao lado de um componente já existente.
- Os componentes precisam ter um ID definido, pois o posicionamento de um depende do outro.

Exercício 11

RelativeLayout

Reminder

DONE

FrameLayout

- A classe *android.widget.FrameLayout* é o tipo mais comum e mais simples de layout.
- Funciona como uma pilha, onde uma *view* fica por cima da outra.
- A posição visual é configurada no atributo *android:layout_gravity*.

FrameLayout

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
    />

</FrameLayout>
```



Exercício 12

FrameLayout

Governo de Pernambuco



O lugar ideal para suas férias!



Desafio 4

RelativeLayout



Pode Me Chamar Que Eu Vou
Eddie

4:47

