



Relative Layout

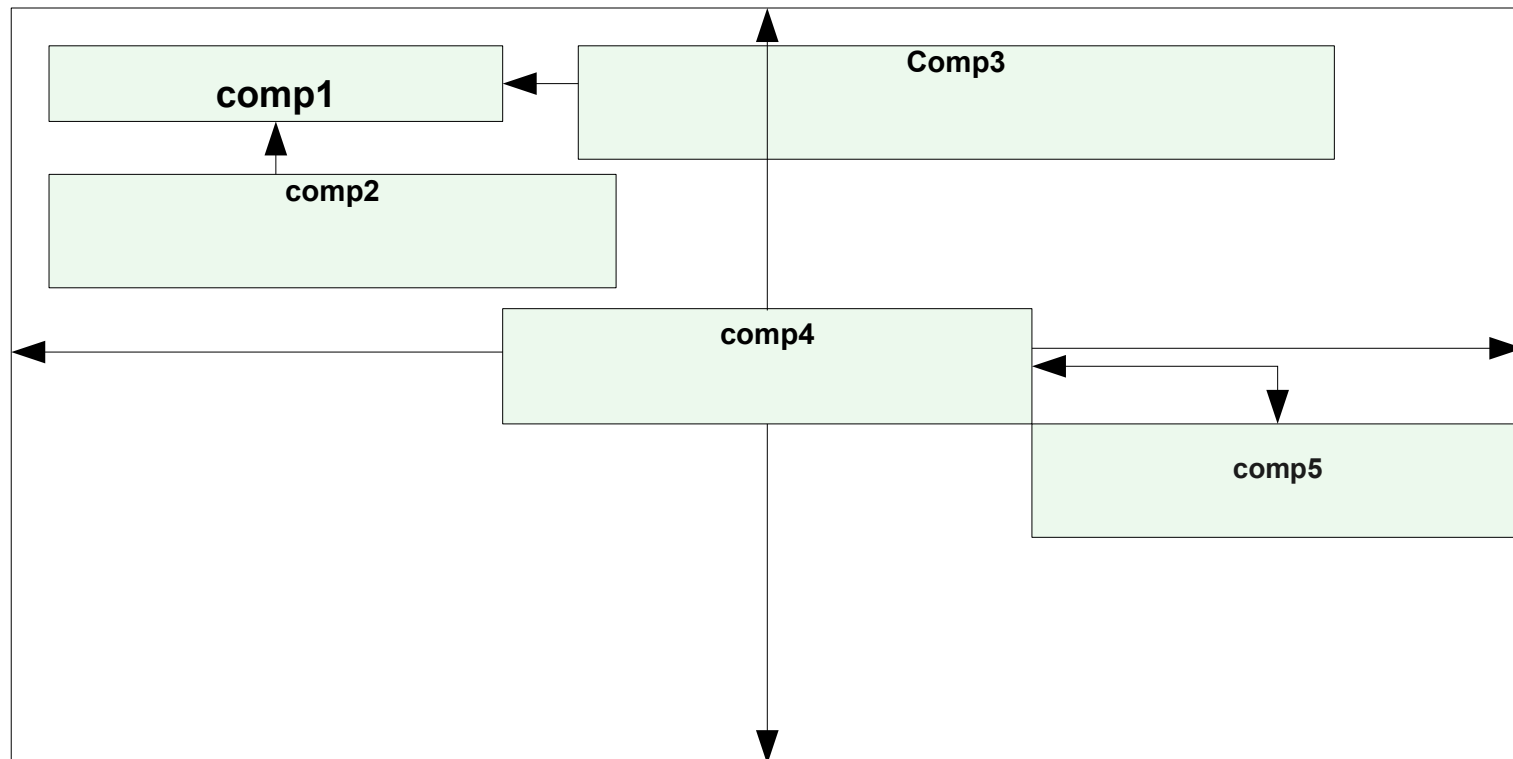
Professor Msc Romulo Beninca



Gerenciadores de *layout* *Relative Layout*

O *Relative layout* é uma alternativa ao posicionamento absoluto dos componentes da *view*, pois este possibilita que os *views* sejam definidos um em relação aos outros, ou em relação ao Pai (*RelativeLayout*).

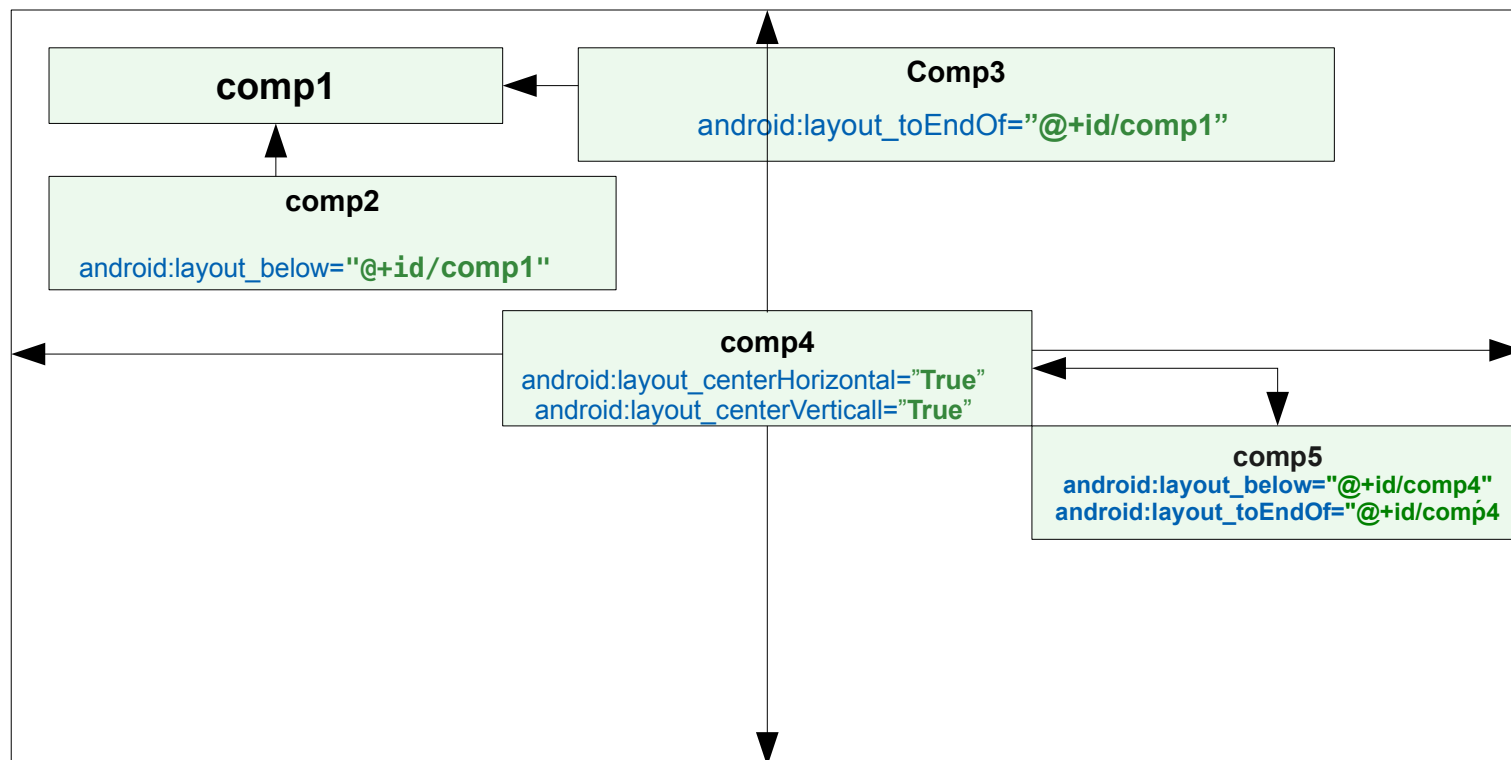
É como definir a posição de uma **aluno A** na sala em relação a outro **aluno B**.

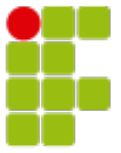


Gerenciadores de *layout*

Relative Layout

O *RelativeLayout* é um *ViewGroup* que exibe *views* em posições relativas. A posição de cada visualização pode ser especificada em relação as outros componentes vizinhos, a esquerda, direita, topo ou abaixo, ou ainda em relação a o pai que é o *RelativeLayout*, o que possibilita alinhar á parte inferior, esquerda ou central (DeveloperAndroid,2018).

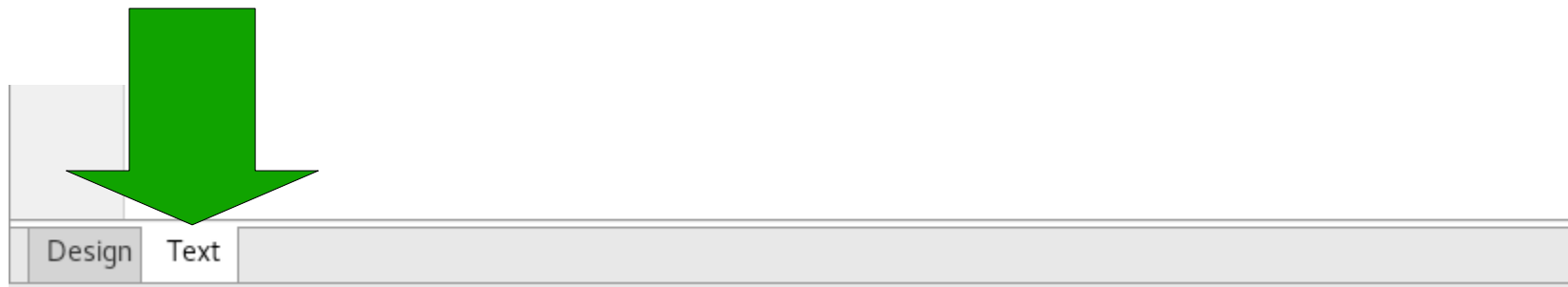




Gerenciadores de *layout* *Relative Layout*

Ao utilizar *RelativeLayout* é comum recorrermos ao XML e não a ferramenta *Design* para definir as propriedades, no entanto a ferramenta *Design* possibilita a pré visualização do layout de uma forma simplificada.

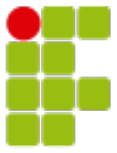
```
<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout  
  xmlns:android="http://schemas.android.com/apk/res/android"  
  xmlns:app="http://schemas.android.com/apk/res-auto"  
  xmlns:tools="http://schemas.android.com/tools"  
  android:layout_width="match_parent"  
  android:layout_height="match_parent"  
  tools:context=".IMCRelativeLayout">  
  
</RelativeLayout>
```



Gerenciadores de *layout*

Relative Layout

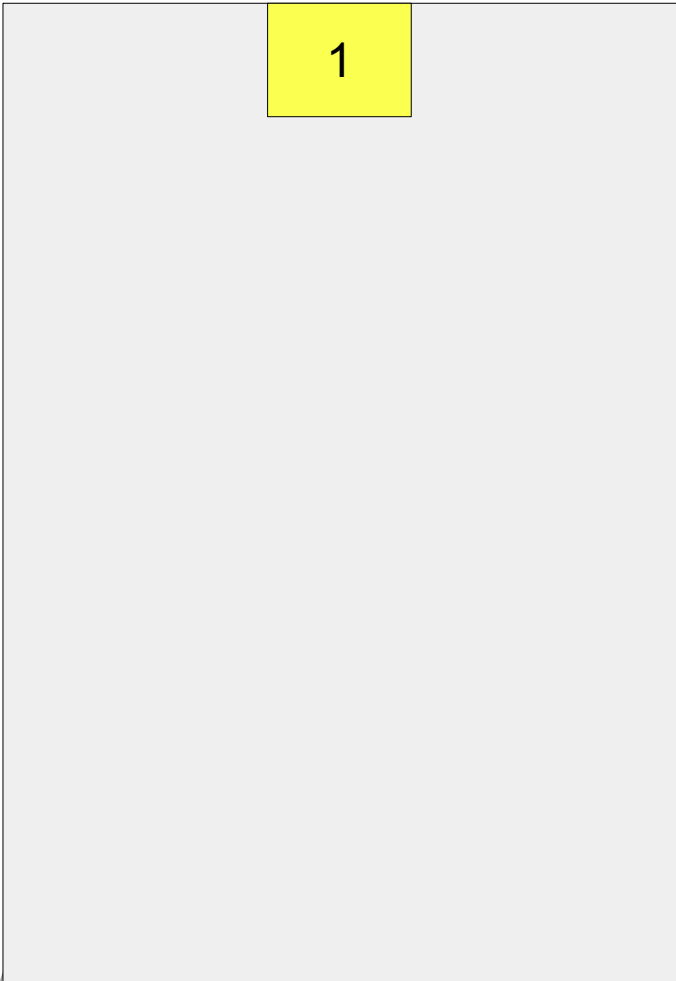
Existem, atualmente, 23 propriedades para posicionar as *views* utilizando *RelativeLayout*, algumas destas propriedades possibilitam posicionar a *view* em relação a componente pai, enquanto outras possibilitam posicioná-las em relação aos componentes vizinhos (DeveloperAndroid, 2018).



Gerenciadores de *layout*

Relative Layout - Parâmetros

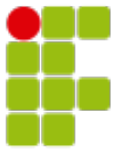
Parâmetros `layout_center{Horizontal, Vertical, InParent}` possibilita centralizar a a `view` em relação a horizontal vertical ou ambos do componente pai.(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        android:center_horizontal="true"
    />
```

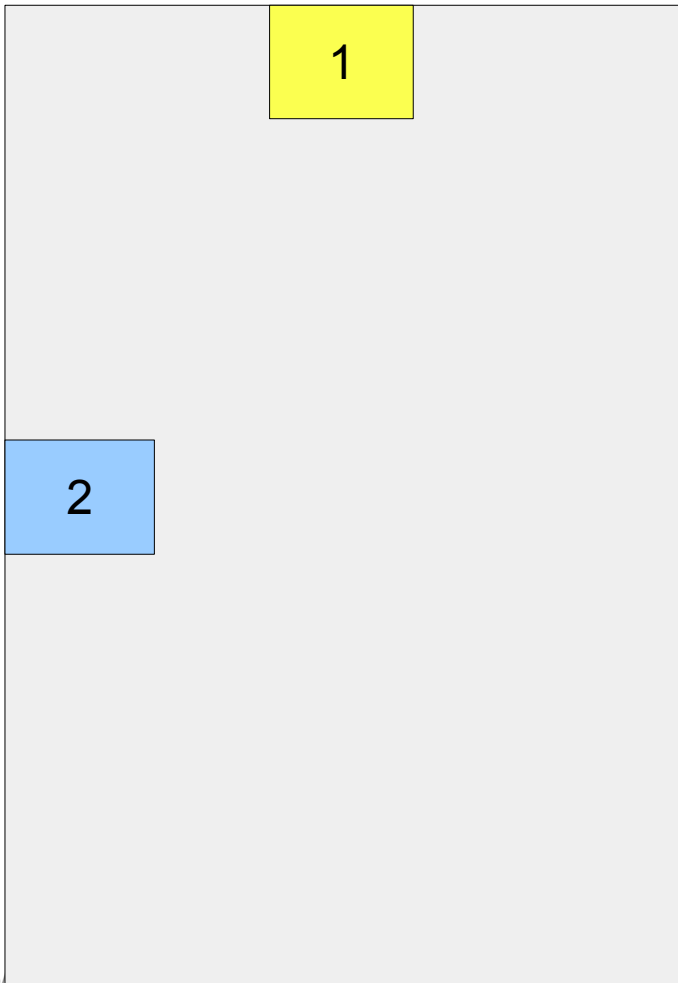
```
</RelativeLayout>
```



Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros `layout_center{Horizontal, Vertical, InParent}` possibilita centralizar a a *view* em relação a horizontal vertical ou ambos do componente pai.(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="1"
        android:center_horizontal="true"
    />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="2"
        android:center_vertical="true"
    />

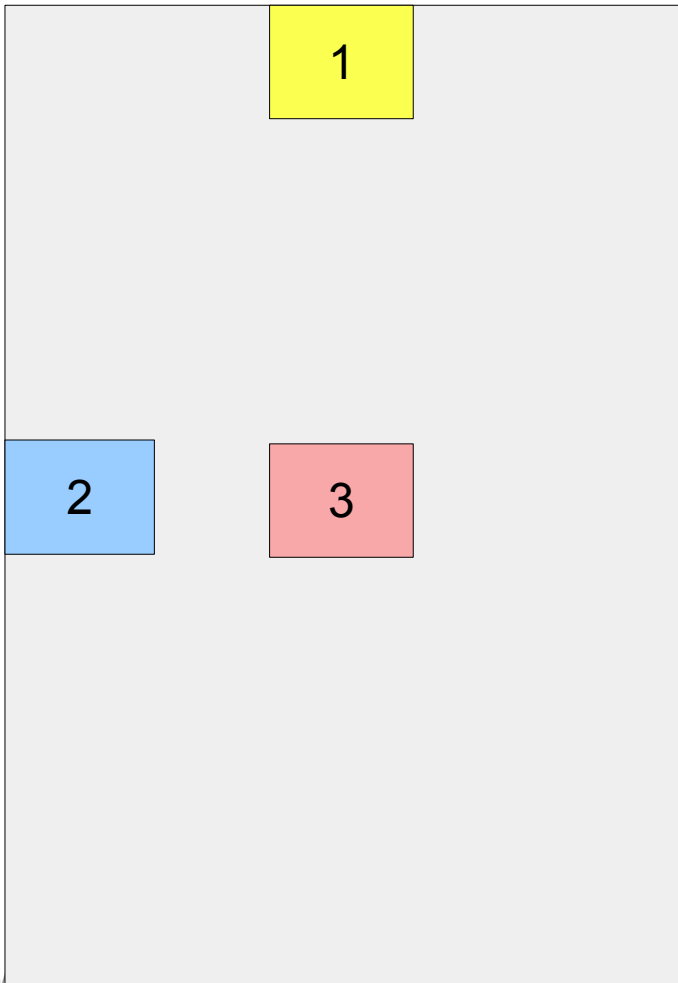
</RelativeLayout>
```



Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_center*{*Horizontal*, *Vertical*, *InParent*} possibilita centralizar a a *view* em relação a horizontal vertical ou ambos do componente pai.(AndroidDeveloper,2018)



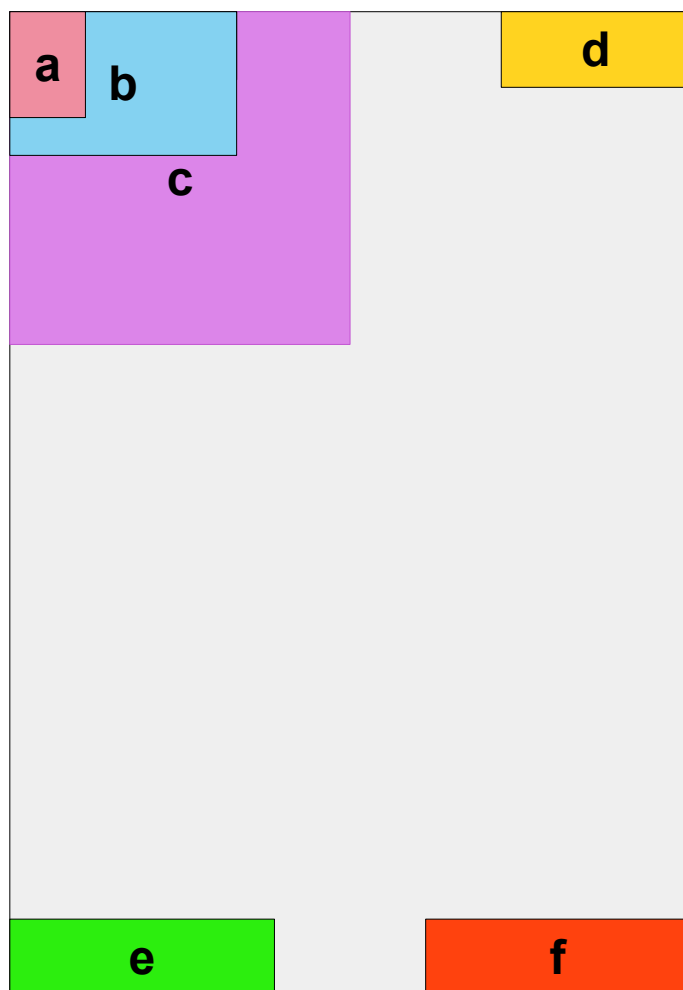
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".IMCRelativeLayout">
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="1"
    android:center_horizontal="true"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="2"
    android:center_vertical="true"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="3"
    android:center_InParent="true"
  />
</RelativeLayout>
```




Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_alignParent...* permitem definir a relação do alinhamento das bordas *view* com as bordas do componente pai (AndroidDeveloper,2018).



a: `android:layout_alignParentStart="true"`

b: `android:layout_alignParentTop="True"`

c: `android:layout_alignParentLeft="True"`

d: `android:layout_alignParentRight="true"`

e: `android:layout_alignParentBottom="true"`

f: `android:layout_alignParentEnd="true"`

Fonte:Autor (2018)



Gerenciadores de *layout* *Relative Layout*

O posicionamento relativo dos componentes do layout pode ser expresso com a definição de algumas propriedades na declaração das *Views*, como por exemplo a propriedade *layout_toEndOf* = "<identificador>" possibilita definir a posição de um elemento ao termino de outro. O que é útil, por exemplo, em um formulário com campo peso em que temos um *TextView* com valor *labelPeso*, seguido a direita por um componente *EditText*, para que o usuário informe o valor do peso direita pode ser declarada.



```
<RelativeLayout ...>
  <TextView
    android:id="@+id/labelPeso"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/editTextPeso"
    android:layout_alignParentStart="true"
    android:text="Peso:"
    android:textSize="18sp" />
  <EditText
    android:id="@+id/editTextPeso"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toEndOf="@+id/labelPeso"
    android:ems="10"
    android:inputType="textPersonName"
    android:text="Name" />
</RelativeLayout>
```

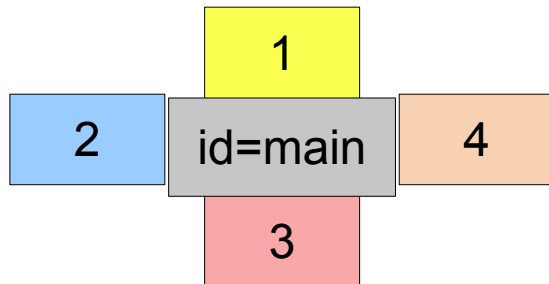




Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



1: `android:layout_above="@id/main"`

2: `android:layout_toLeftOf="@id/main"`

3: `android:layout_toRightOf="@id/main"`

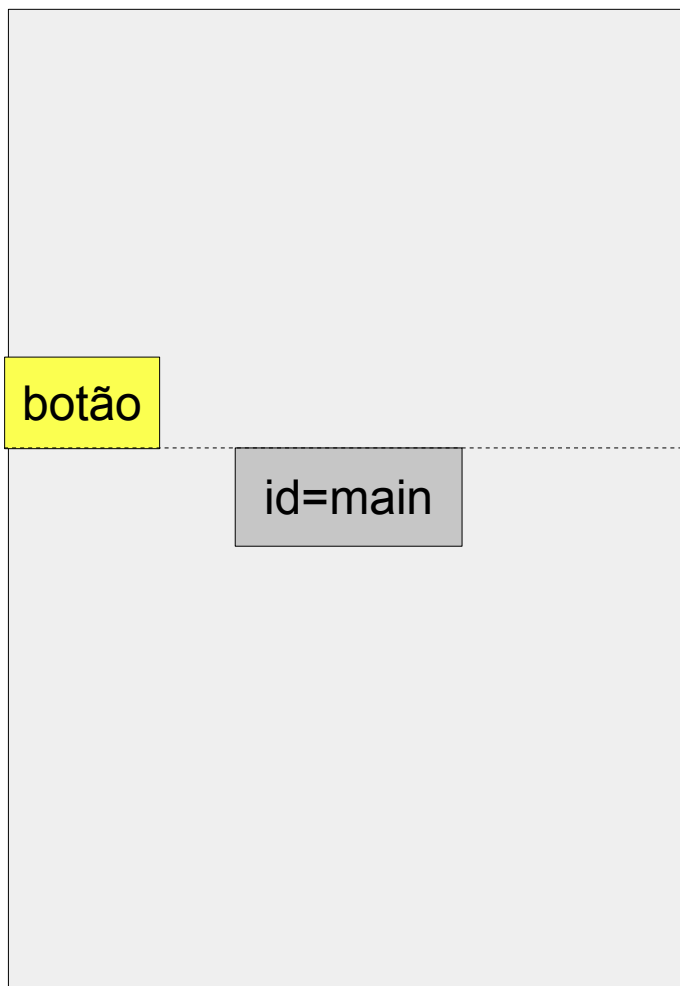
4: `android:layout_below="@id/main"`

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

  <Button
    android:id="@+id/main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:align_inParent="true"
    android:text="id=main:"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botão"
    android:layout_above="@id/main"
  />

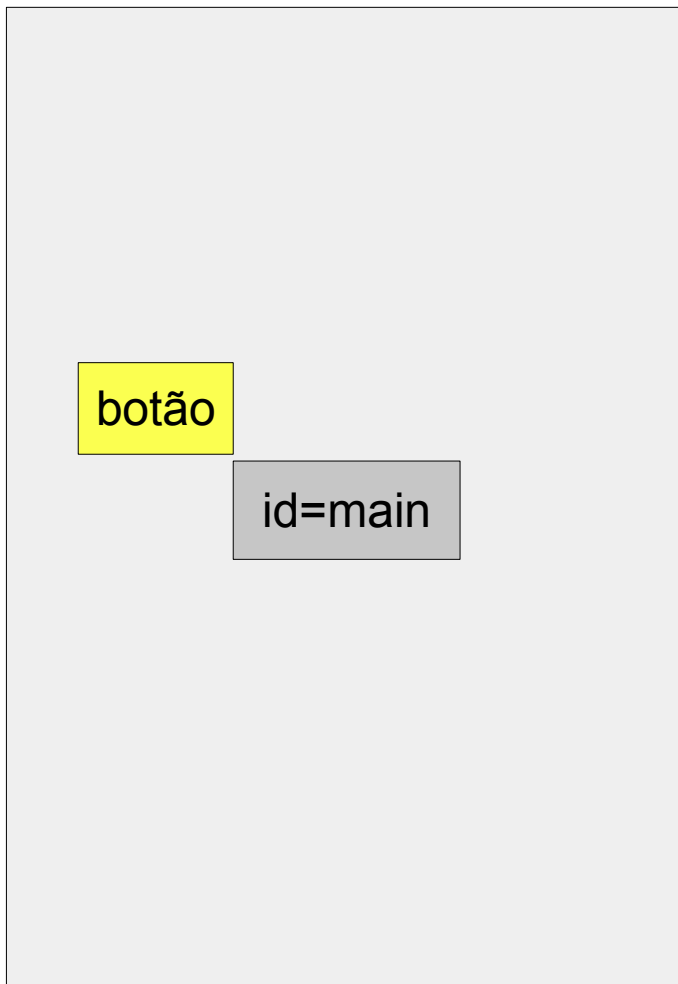
</RelativeLayout>
```

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

  <Button
    android:id="@+id/main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:align_inParent="true"
    android:text="id=main:"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botão"
    android:layout_above="@id/main"
    android:layout_toLeftOf="@id/main"
  />

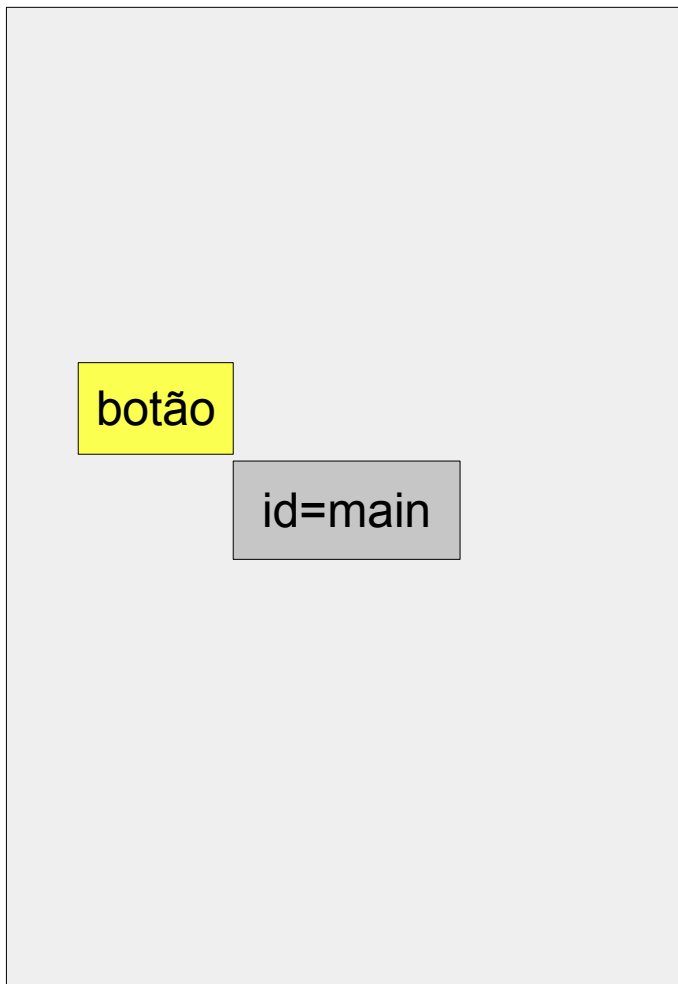
</RelativeLayout>
```

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

  <Button
    android:id="@+id/main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:align_inParent="true"
    android:text="id=main:"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botão"
    android:layout_above="@id/main"
    android:layout_toLeftOf="@id/main"
  />

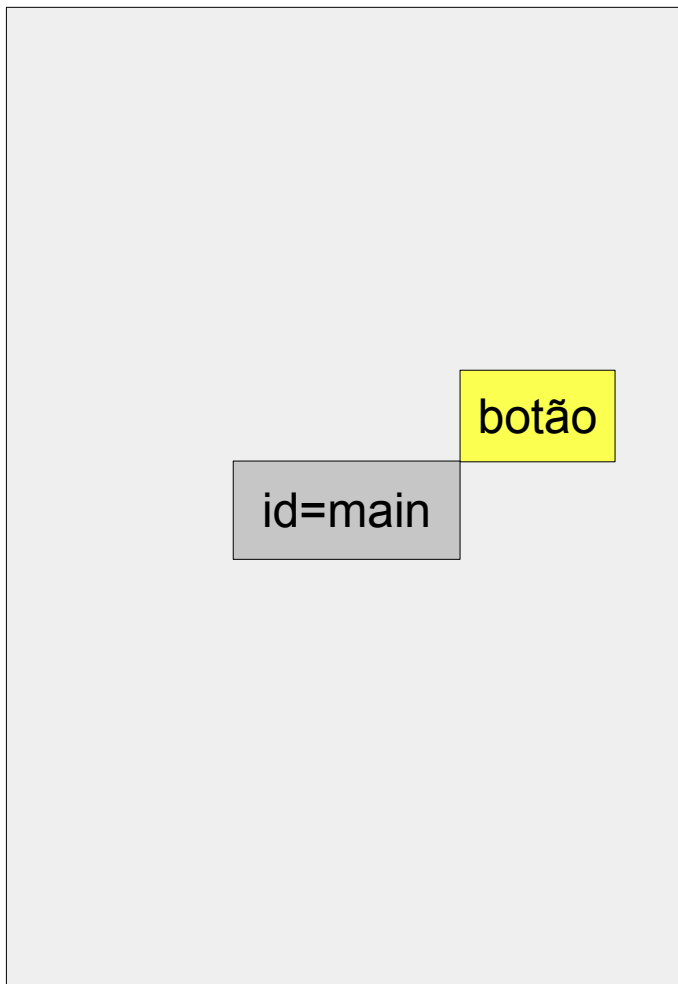
</RelativeLayout>
```

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

  <Button
    android:id="@+id/main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:align_inParent="true"
    android:text="id=main:"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botão"
    android:layout_above="@id/main"
    android:layout_toRightOf="@id/main"
  />

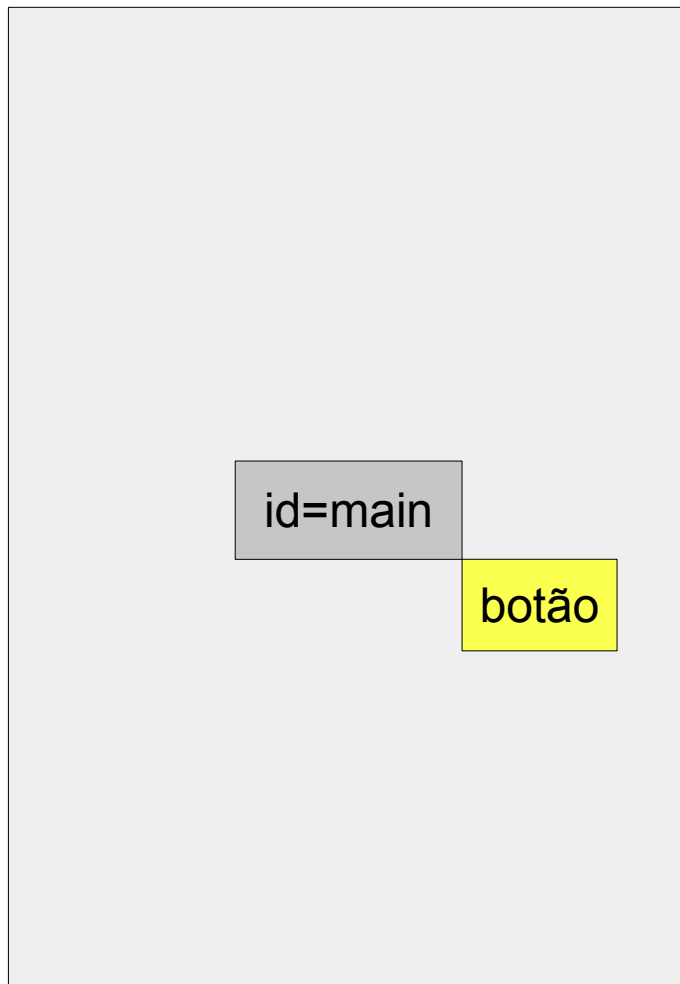
</RelativeLayout>
```

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

    <Button
        android:id="@+id/main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:align_inParent="true"
        android:text="id=main:"
    />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="botão"
        android:layout_below="@id/main"
        android:layout_toRightOf="@id/main"
    />

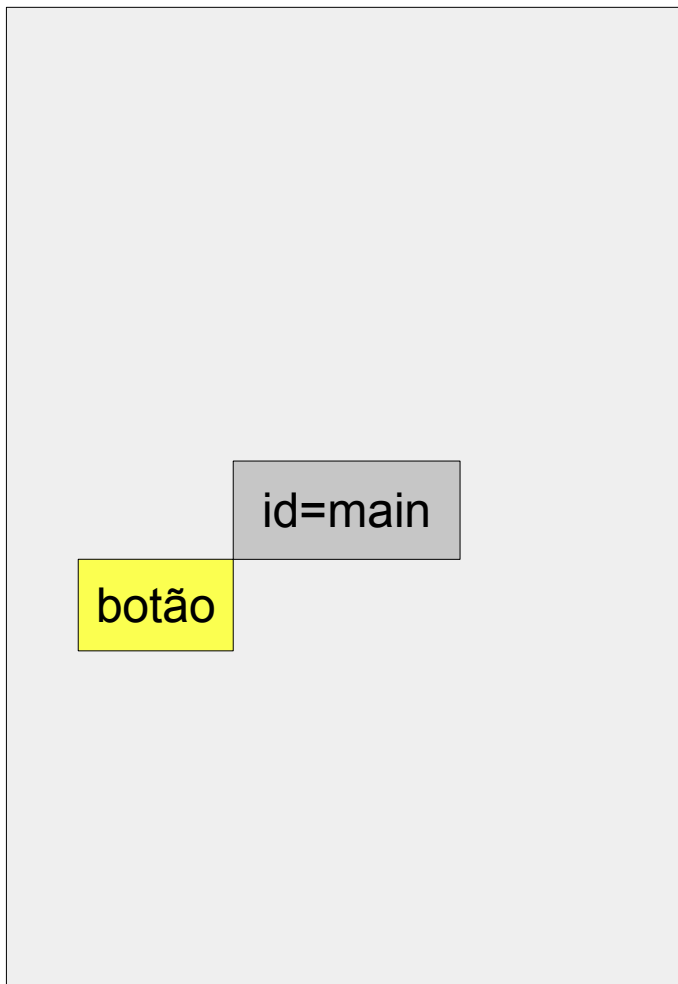
</RelativeLayout>
```


Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

  <Button
    android:id="@+id/main"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:align_inParent="true"
    android:text="id=main:"
  />
  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="botão"
    android:layout_above="@id/main"
    android:layout_toLeftOf="@id/main"
  />

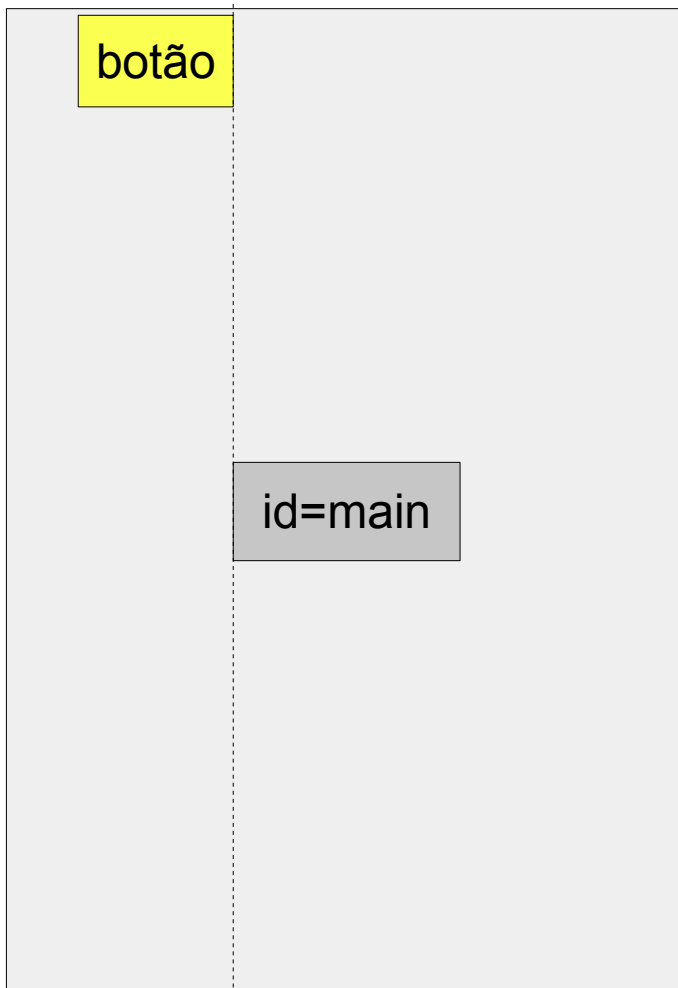
</RelativeLayout>
```

Gerenciadores de *layout*

Relative Layout - Parâmetros



Parâmetros *layout_{above, toLeftOf, toRightOf, below}* permite definir a posição da *view* usando como referencia uma outra *view* referenciada. .(AndroidDeveloper,2018)



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
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"
tools:context=".IMCRelativeLayout">

    <Button
        android:id="@+id/main"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:align_inParent="true"
        android:text="id=main:"
    />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="botão"

        android:layout_toLeftOf="@id/main"
    />

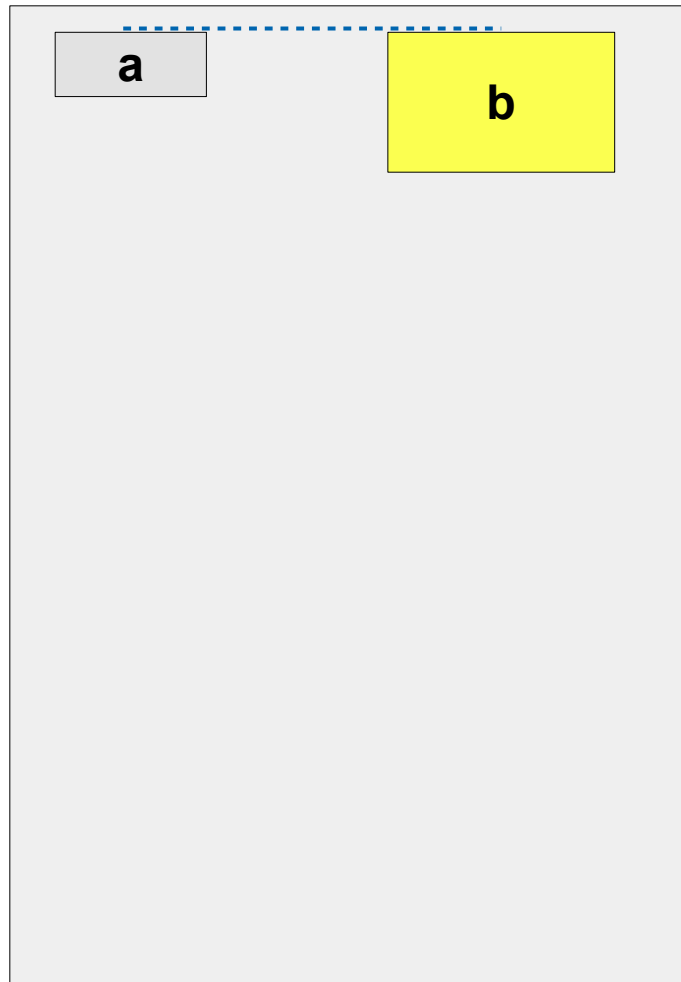
</RelativeLayout>
```



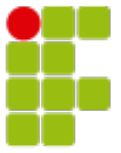
Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



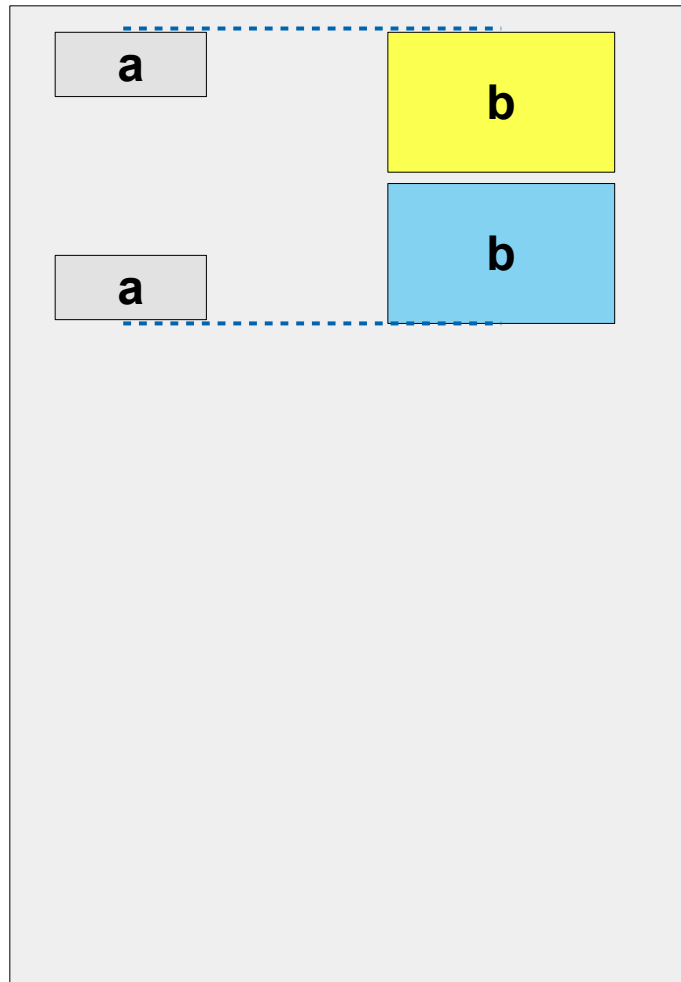
1: `Android:layout_alignTop="@id/a"`



Gerenciadores de *layout*

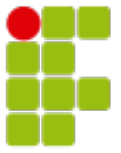
Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

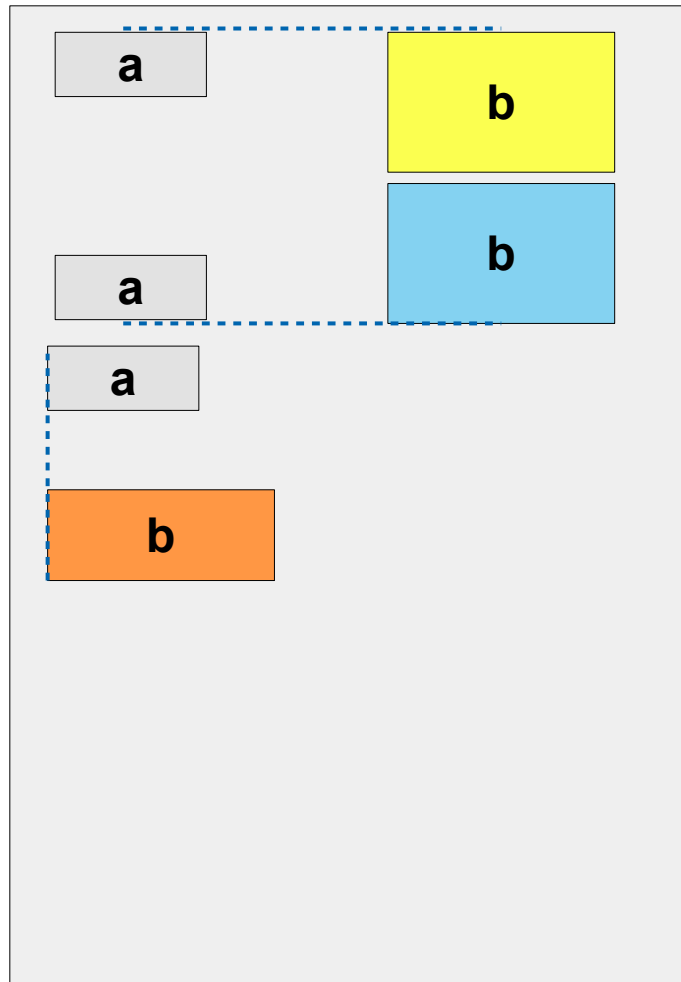
2: `Android:layout_alignBottom="@id/a"`



Gerenciadores de *layout*

Relative Layout - Parâmetros

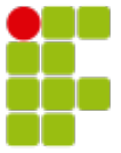
Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

2: `Android:layout_alignBottom="@id/a"`

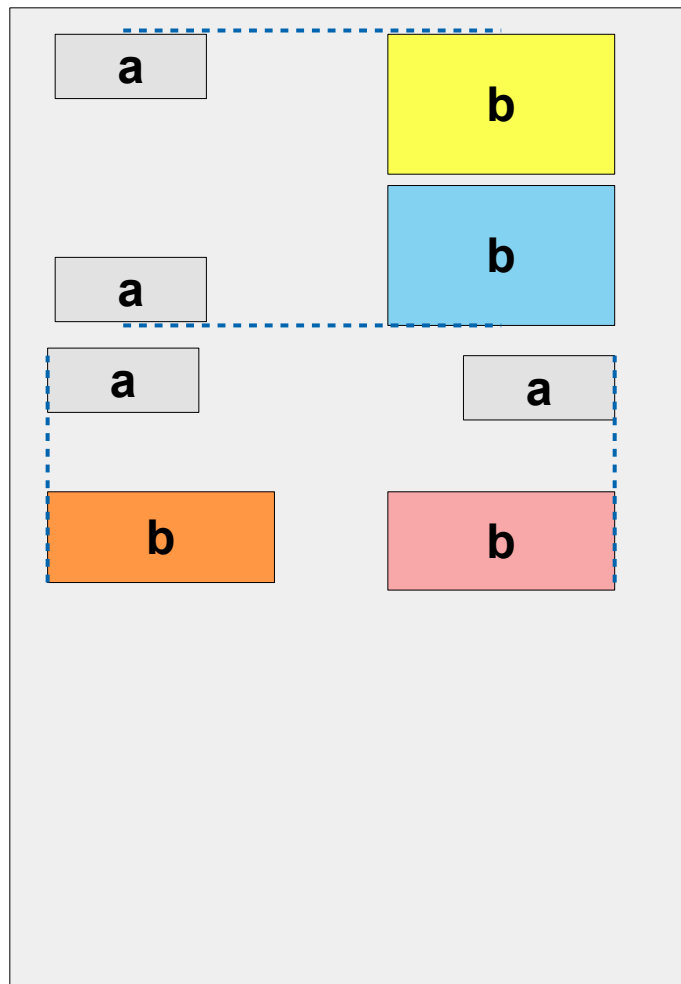
3: `Android:layout_alignLeft="@id/a"`



Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

2: `Android:layout_alignBottom="@id/a"`

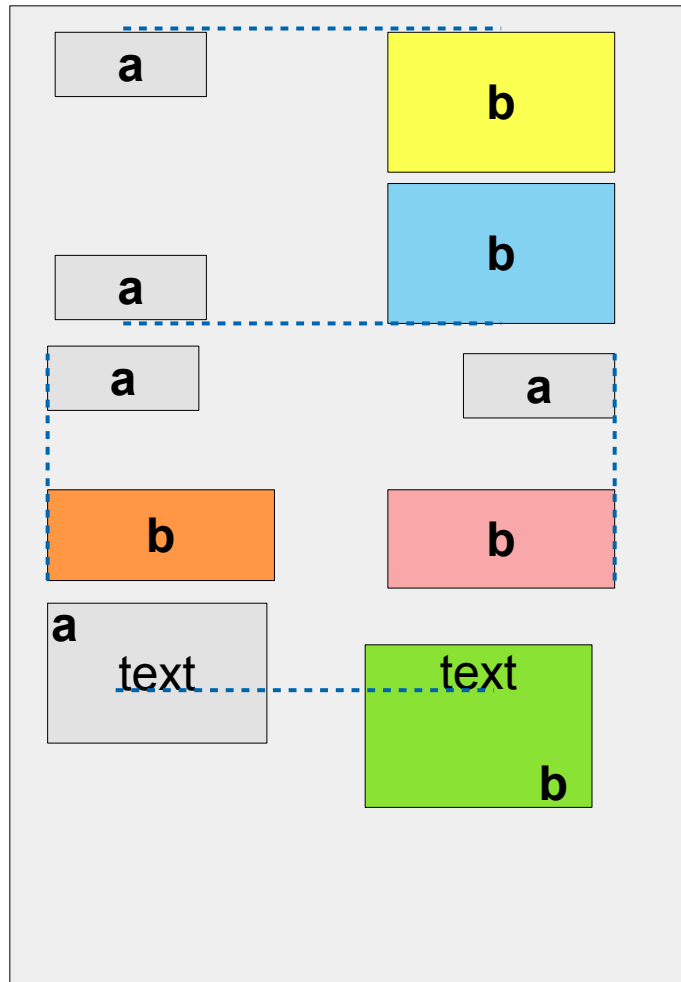
3: `Android:layout_alignLeft="@id/a"`

4: `Android:layout_alignRight="@id/a"`

Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

2: `Android:layout_alignBottom="@id/a"`

3: `Android:layout_alignLeft="@id/a"`

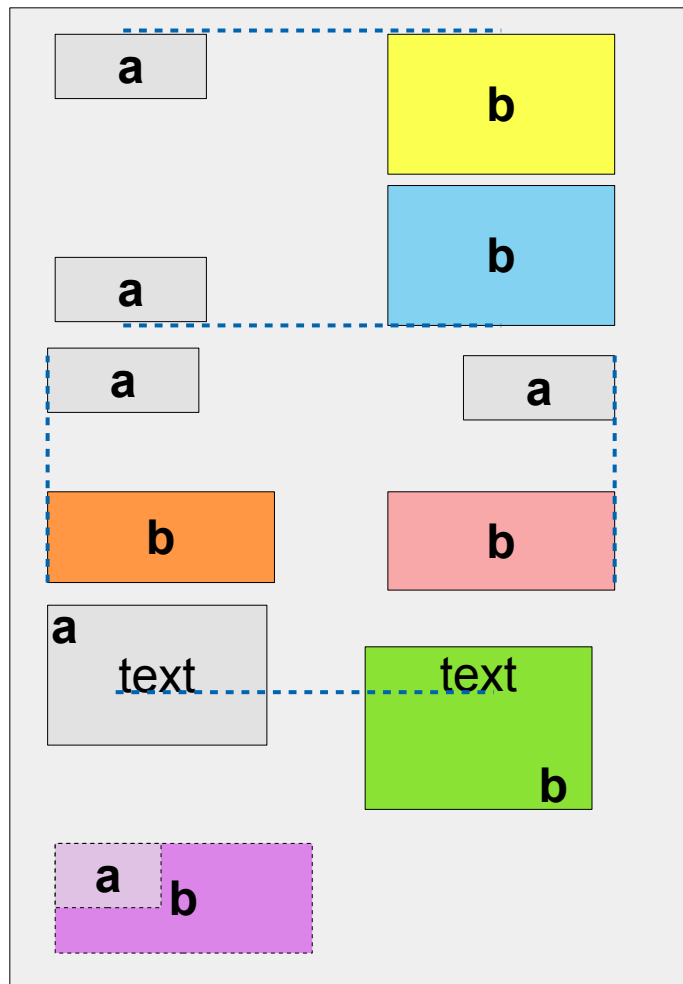
4: `Android:layout_alignRight="@id/a"`

5: `Android:layout_alignBaseline="@id/a"`

Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

2: `Android:layout_alignBottom="@id/a"`

3: `Android:layout_alignLeft="@id/a"`

4: `Android:layout_alignRight="@id/a"`

5: `Android:layout_alignBaseline="@id/a"`

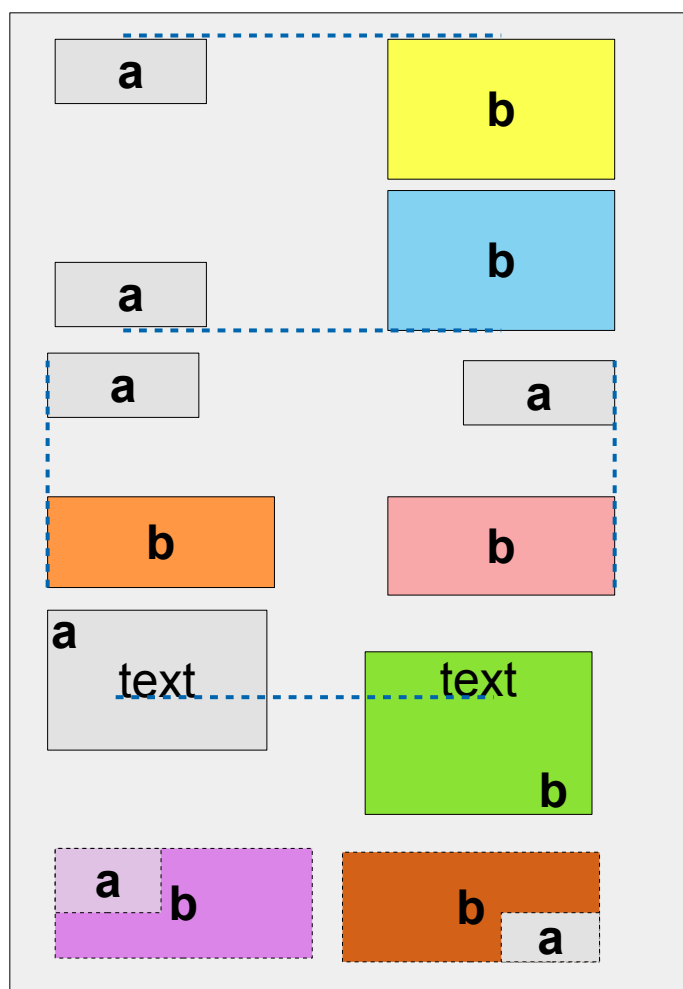
6: `Android:layout_alignStart="@id/a"`



Gerenciadores de *layout*

Relative Layout - Parâmetros

Parâmetros *layout_align{Top, Bottom, Left, Right, BaseLine}*... permite definir a relação do alinhamento, de uma das bordas da *view*, com as bordas da *view* referencia ou ainda com a linha base da *view* referenciada.(AndroidDeveloper,2018)



1: `Android:layout_alignTop="@id/a"`

2: `Android:layout_alignBottom="@id/a"`

3: `Android:layout_alignLeft="@id/a"`

4: `Android:layout_alignRight="@id/a"`

5: `Android:layout_alignBaseline="@id/a"`

6: `Android:layout_alignStart="@id/a"`

7: `Android:layout_alignEnd="@id/a"`

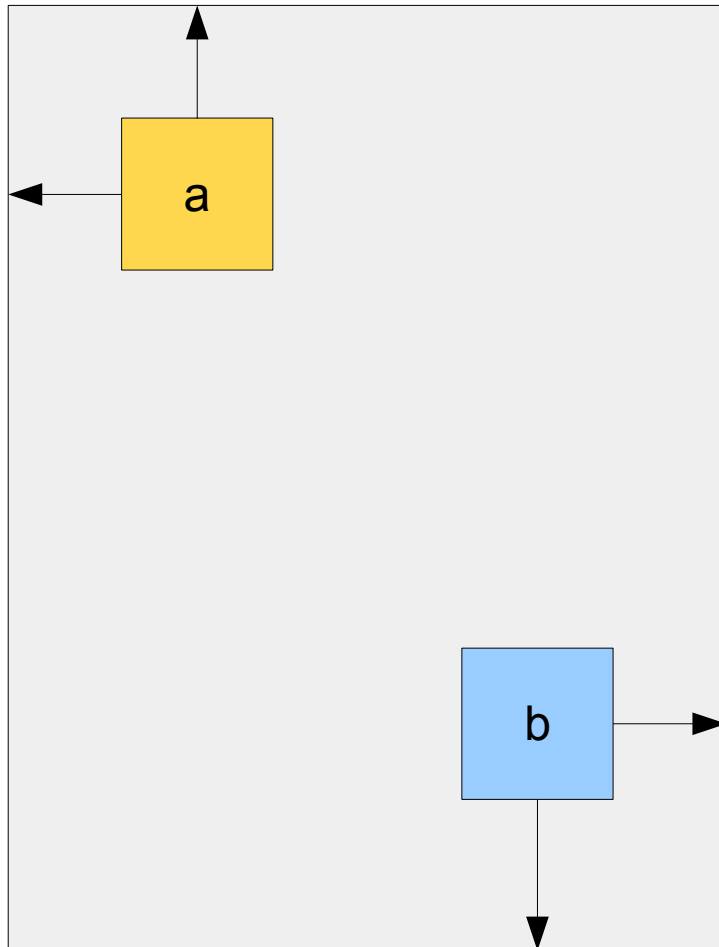
Fonte:Autor (2018).



Gerenciadores de *layout*

Relative Layout - Parâmetros

- O parâmetro ***layout_margin{Start, Top, Bottom, End}*** possibilita definir uma posição em relação a margem do componente pai.



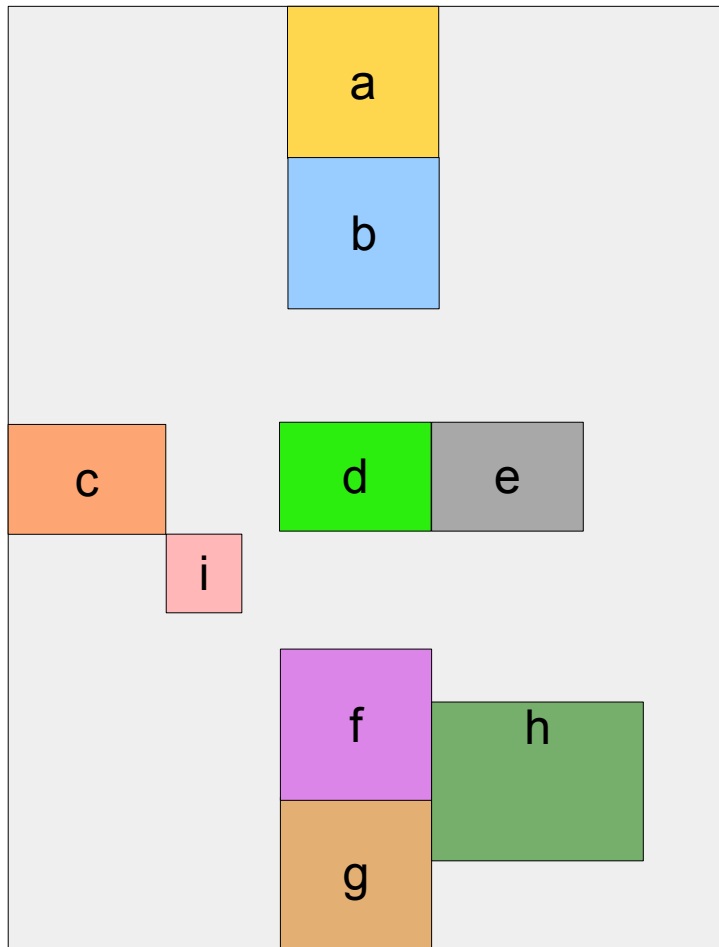
a: `android:layout_marginStart="30dp"`
`android:layout_marginTop="35dp"`

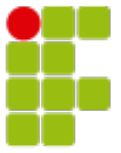
b: `android:layout_marginBottom="132dp"`
`android:layout_marginEnd="71dp"`

Gerenciadores de *layout*

Relative Layout - Atividade

(1) Utilizando *RelativeLayout* qual os parâmetro que necessitam ser informados na *view A* para que ela apresente o resultado da tela apresentada a seguir.

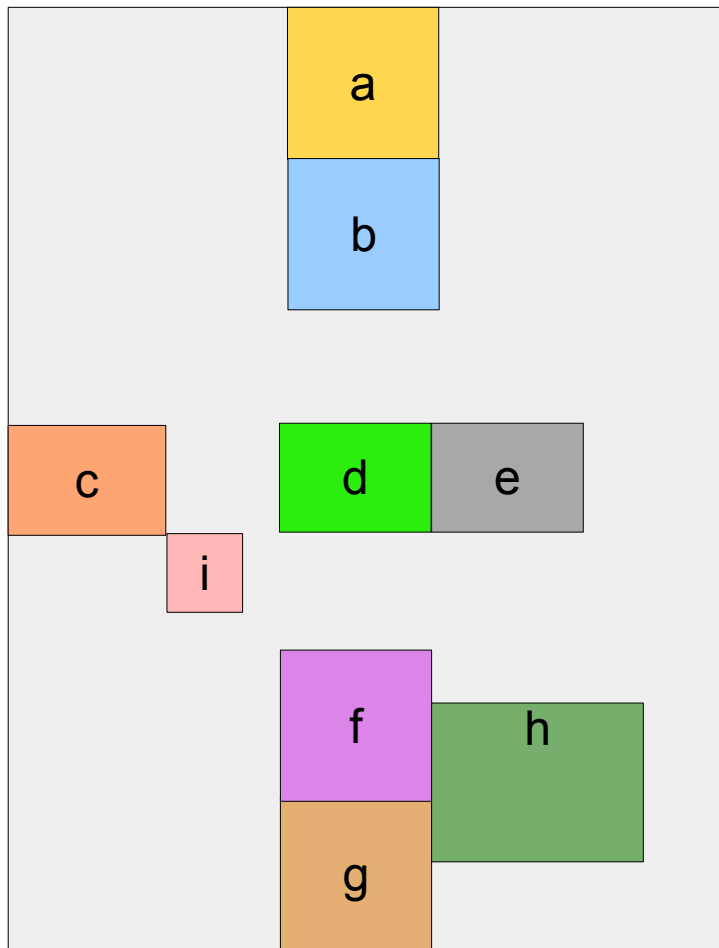




Gerenciadores de *layout*

Relative Layout - Atividade

(1) Utilizando *RelativeLayout* qual os parâmetro que necessitam ser informados na *view A* para que ela apresente o resultado da tela apresentada a seguir.



a: `android:layout_centerHorizontal="True"`

b: `android:layout_below="@id/a"`
`android:layout_centerHorizontal="@id/a"`

c: `android:layout_centerVertical="True"`

d: `android:layout_centerInParent="True"`

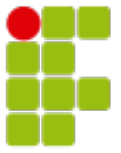
e: `android:layout_toRightOf="@id/d"`
`android:layout_centerVertical="True"`

f: `android:layout_above="@id/g"`
`android:layout_centerHorizontal="True"`

g: `Android:layout_centerHorizontal="True"`
`Android:layout_parentBottom="True"`

h: `android:layout_alignBaseline="@id/f"`
`android:layout_toRightOf="@id/f"`

i: `android:layout_abouve="@id/c"`
`android:layout_end="@id/c"`



Referências

- Lecheta, R. Google Android: Aprenda a criar aplicações para dispositivos móveis. 3a Ed. Novatec, 2013.
- Androd Developer. Android Disponível em:<<https://developer.android.com>>
- Androd Developer, 2018. Relative Parameters Disponível em:<<https://developer.android.com/reference/android/widget/RelativeLayout.LayoutParams>>