



Programação para Dispositivos Móveis

Aula 4 - LAYOUTS

Ana Karina D. Salina de Oliveira

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Programação para Dispositivos Móveis



Android

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Android

Ambiente de Desenvolvimento





Roteiro

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

- 1 Layouts
- 2 Tipos de Layouts
- 3 Constraint Layout
- 4 Linear Layout
- 5 Exemplo com Linear Layout



Android

Layouts

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

- O layout define a estrutura visual para uma interface do usuário, como a IU de uma atividade.
- É possível criar layouts XML para diferentes orientações de tela e diferentes tamanhos de tela.
- Cada arquivo de layout deve conter exatamente um elemento raiz, que deve ser um objeto View ou ViewGroup.
 - Com o elemento raiz definido, é possível adicionar objetos.
- O arquivo layout deverá ser salvo com uma extensão `.xml` no diretório `res/layout/`



Android

Layouts

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Carregamento do recurso XML

- Ao compilar o aplicativo, cada arquivo de layout XML é compilado em um recurso View.
- Deve-se carregar o recurso de layout do código do aplicativo na implementação de retorno de chamada *Activity.onCreate()*.
 - Para isso, chame *setContentView()*, passando a referência para o recurso de layout na forma:
R.layout.layout_file_name.

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.main_layout);  
}
```



Parâmetros do layout

- Todos os grupos de exibições contêm largura e altura (*layout_width* e *layout_height*).
- É possível especificar largura e altura com medidas exatas,
- Porém é recomendado usar uma destas constantes:
 - *wrap_content* instrui a exibição a se redimensionar de acordo com as medidas exigidas pelo conteúdo.
 - *match_parent* instrui a exibição a assumir o maior tamanho permitido pelo grupo de exibições pais.

Posição do layout

- As exibições têm uma localização, expressa como um par de coordenadas esquerda e topo e duas dimensões, expressas como largura e altura.
- A unidade de localização e de dimensões é o pixel.
- É possível recuperar a localização de uma exibição chamando os métodos `getLeft()` e `getTop()`.
 - `getLeft()` retorna a coordenada esquerda, ou X, do retângulo que representa a exibição.
 - `getTop()` retorna a coordenada topo, ou Y, do retângulo que representa a exibição.
- Esses métodos retornam a localização da exibição em relação ao pai correspondente.

Tamanho, preenchimento e margens

- O tamanho de uma exibição é expresso por largura e altura, chamando `getWidth()` e `getHeight()`.
- As margens podem ser definidas como:

XML attributes	
<code>android:layout_margin</code>	Specifies extra space on the left, top, right and bottom sides of this view.
<code>android:layout_marginBottom</code>	Specifies extra space on the bottom side of this view.
<code>android:layout_marginEnd</code>	Specifies extra space on the end side of this view.
<code>android:layout_marginHorizontal</code>	Specifies extra space on the left and right sides of this view.
<code>android:layout_marginLeft</code>	Specifies extra space on the left side of this view.
<code>android:layout_marginRight</code>	Specifies extra space on the right side of this view.
<code>android:layout_marginStart</code>	Specifies extra space on the start side of this view.
<code>android:layout_marginTop</code>	Specifies extra space on the top side of this view.
<code>android:layout_marginVertical</code>	Specifies extra space on the top and bottom sides of this view.

Layouts comuns

- Cada subclasse da classe ViewGroup fornece um modo exclusivo de exibições aninhadas dentro dela.

Layout linear



Layout que organiza os filhos em uma única linha horizontal ou vertical. Ele cria uma barra de rolagem se o comprimento da janela excede o comprimento da tela.

Layout relativo



Permite especificar a localização de objetos filhos relativos entre si (filho A à esquerda do filho B) ou relativos aos pais (alinhados no topo do pai).

Exibição Web



Exibe páginas da Web.



Android

Layouts

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Criação de layouts com um adaptador

- Quando o conteúdo do layout é dinâmico, é possível usar um layout que torne AdapterView
 - uma subclasse para preencher o layout com exibições em tempo de execução.
- Uma subclasse da classe AdapterView usa um Adapter para agrupar dados ao seu layout.
- O Adapter se comporta como um intermediário entre a fonte dos dados e o layout do AdapterView
- O Adapter recupera os dados (de uma fonte como uma matriz ou uma consulta de banco de dados).

Alguns layouts comuns retornados por um adaptador:

Exibição List



Exibe uma lista de rolagem de coluna única.

Exibição Grid



Exibe uma grade de rolagem de colunas e linhas.



Android

Constraint Layout

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

ConstraintLayout

- permite que você crie layouts grandes e complexos com uma hierarquia no seu XML
 - sem a necessidade de ficar aninhando um monte de layouts uns dentro dos outros
- é similar ao Relative Layout, uma vez que todas as views se relacionam entre si e com o parent,
 - mas é mais flexível e mais fácil de usar
- também possui as melhores características do LinearLayout,
 - o que nos permite interfaces bem distribuídas e simples de construir.



Android

Constraint Layout

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

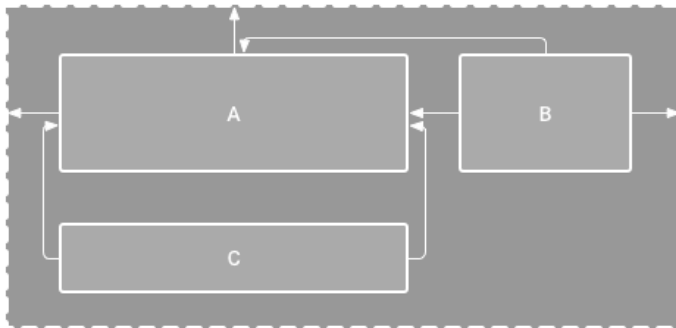
Linear Layout

Exemplo com
Linear Layout

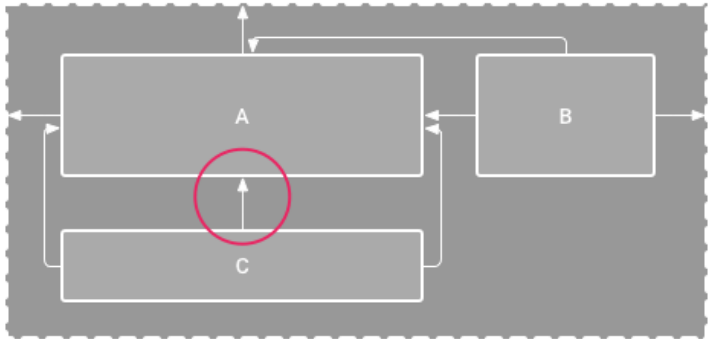
Constraint

- Constraints significa algo como “restrições” ou “limitações”,
- as restrições são o cerne por trás do funcionamento deste layout manager,
- Para definir a posição de uma view no ConstraintLayout, você deve adicionar ao menos:
 - uma constraint horizontal
 - uma vertical para a view.
- Cada constraint representa uma conexão ou alinhamento em relação à outra view.

A view C se alinhará horizontalmente com as faces esquerda e direita da view A, mas irá aparecer no topo da tela porque não há constraint vertical.



O correto seria adicionar uma constraint vertical entre as views A e C.



Android

Constraint

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

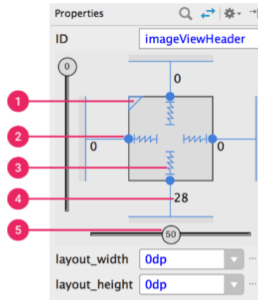
Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout



Esta janela inclui controles para 1) proporção do tamanho, 2) excluir constraints, 3) modo de altura e largura, 4) margens e 5) viés das constraints (tipo um offset).

LinearLayout

- é um grupo de visualizações que alinha todos os filhos em uma única direção **vertical** ou **horizontal**.
- pode-se especificar a direção do layout com o atributo *android:orientation*





Android

Linear Layout

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Ponderação do layout

- O `LinearLayout` também é compatível com a atribuição de uma ponderação a filhos individuais com o atributo *`android:layout_weight`*.
- Este atributo confere um valor de "importância" a uma visualização em relação à quantidade de espaço que ela deve ocupar na tela.
- Um valor de ponderação maior permite que ela se expanda para preencher qualquer espaço restante na visualização pai.
- A ponderação padrão é zero.



Android

Linear Layout

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Distribuição Desigual

- Se houver três campos de texto
- dois deles tiverem uma ponderação 1,
- o terceiro tiver uma ponderação 2 (em vez de 0),
- o último será declarado mais importante que os outros e ficará com metade do espaço total restante,
- os dois primeiros campos compartilharão o resto igualmente.



Android

Linear Layout

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

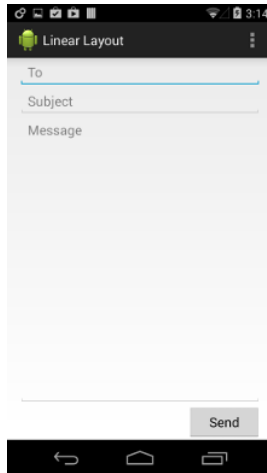
Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Exemplo de Distribuição Desigual





Android

Linear Layout

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Exemplo de Distribuição Desigual

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical" >
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/to" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/subject" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="10dp"
        android:layout_weight="1"
        android:gravity="top"
        android:hint="@string/message" />
    <Button
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="right"
        android:text="@string/send" />
</LinearLayout>
```

<https://developer.android.com/guide/topics/ui/layout/linear?hl=pt-br>



Android

Layouts

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Roteiro

Layouts

Tipos de
Layouts

Constraint
Layout

Linear Layout

Exemplo com
Linear Layout

Combinação de Layouts

- Outra prática interessante que pode ser feita é a combinação entre layouts,
 - onde em uma parte da tela você precise de um tipo de alinhamento e em outra parte um outro tipo.
- Pode-se utilizar, por exemplo, um Linear Layout dentro de um Relative Layout.

Exercício: Criar um aplicativo que calcula o IMC conforme as regras abaixo:

- O cálculo do IMC é feito dividindo o peso (em quilogramas) pela altura (em metros) ao quadrado.
- O aplicativo terá como entrada o peso e a altura.
- O aplicativo terá como saída o resultado do IMC, a classificação e os riscos.

Classificação	IMC	O que pode acontecer
Muito abaixo do peso	16 a 16,9 kg/m ²	Queda de cabelo, infertilidade, ausência menstrual
Abaixo do peso	17 a 18,4 kg/m ²	Fadiga, stress, ansiedade
Peso normal	18,5 a 24,9 kg/m²	Menor risco de doenças cardíacas e vasculares
Acima do peso	25 a 29,9 kg/m ²	Fadiga, má circulação, varizes
Obesidade Grau I	30 a 34,9 kg/m ²	Diabetes, angina, infarto, aterosclerose
Obesidade Grau II	35 a 40 kg/m ²	Apneia do sono, falta de ar
Obesidade Grau III	maior que 40 kg/m ²	Refluxo, dificuldade para se mover, escaras, diabetes, infarto, AVC