



Programação para Dispositivos Móveis

Aula 7 - Layouts e Icon

Ana Karina D. Salina de Oliveira

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Programação para Dispositivos Móveis
2018



Roteiro

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

- 1 Layouts
- 2 Layouts com um adaptador
- 3 ListView
- 4 Float Action Button
- 5 icon

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

<https://www.androidpro.com.br/blog/desenvolvimento-android/listviews/>

<https://developer.android.com/guide/topics/ui/layout/gridview#java>



Android

Layouts

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

- 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

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

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

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

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).

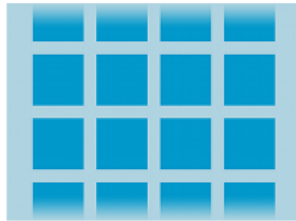
Alguns layouts comuns retornados por um adaptador:

Exibição List



Exibe uma lista de rolagem de coluna única.

Exibição Grid



Exibe uma grade de rolagem de colunas e linhas.



Preenchimento da exibição de adaptador com dados

- É possível preencher um AdapterView como ListView ou GridView vinculando a instância de AdapterView a um Adapter,
- O Adapter recupera dados de uma fonte externa e cria uma View que representa cada entrada de dados.
- Os dois adaptadores mais comuns são:
 - ArrayAdapter.
 - SimpleCursorAdapter.



ArrayAdapter

- Use esse adaptador quando a fonte de dados for uma matriz.
- O ArrayAdapter cria uma exibição para cada item de matriz chamando toString() em cada item e posicionando o conteúdo em uma TextView.
- Ex: Para exibir uma matriz de strings em uma ListView,
 - inicialize um novo ArrayAdapter especificando o layout e a matriz de strings (myString),
 - Em seguida, chame o setAdapter() na ListView.

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>  
    (this, android.R.layout.simple_list_item_1, myString);  
ListView listView = findViewById(R.id.listview);  
listView.setAdapter(adapter);
```



SimpleCursorAdapter

- Esse adaptador é utilizado quando os dados vierem de um Cursor.
- É necessário especificar um layout para cada linha no Cursor e que colunas no Cursor devem ser inseridas,
- Ex: Para criar uma lista de nome e número de telefone de pessoas,
 - pode-se executar uma consulta que retorna um Cursor que contém uma linha para cada pessoa e colunas para os nomes e números.
 - depois, crie uma matriz de strings que especifique quais colunas do Cursor estarão no layout para cada resultado
 - e uma matriz de números inteiros especificando as exibições correspondentes em que cada coluna deve ser colocada.



ListView

PROGRAMAÇÃO
PARA DISPOSITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

Visualização em lista

- A ListView é um grupo de exibições que exibe uma lista de itens roláveis.
- Os itens da lista são inseridos automaticamente na lista usando um Adapter
 - O Adapter obtém conteúdo de uma origem como uma matriz ou consulta de banco de dados
 - depois converte cada resultado de item em uma exibição.

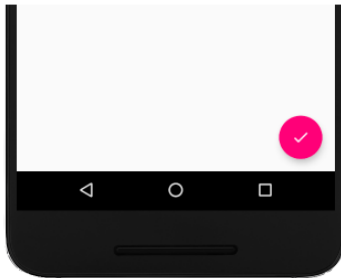
Processamento de eventos de clique

- Para responder a eventos de clique em cada item em um AdapterView, implemente a interface AdapterView.OnItemClickListener:

```
private OnItemClickListener Message =  
    new OnItemClickListener() {  
        public void onItemClick(AdapterView parent,  
            View v, int position, long id) {  
            // Do something in response to the click  
        }  
    };  
listView.setOnItemClickListener(Message);
```

Floating Action Button

- É um botão flutuante que é usado para executar uma ação.
- Tem forma de ícone circular.
- Apenas um botão é recomendado por tela para representar a ação mais comum.



Material Icon

- Ícones disponíveis para serem utilizados em projetos web, Android e iOS.





Android

PROGRAMAÇÃO
PARA DISPO-
SITIVOS
MÓVEIS

Ana Karina

Conteúdo da
Aula

Layouts

Tipos de
Layouts

List View

Floating
Action Button

Material Icon

Exemplo

Exemplo com:

- floatActionButton
- material icon
- layoutAdaptor
- listview