INSTITUTO FEDERAL
PARANÁ

DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

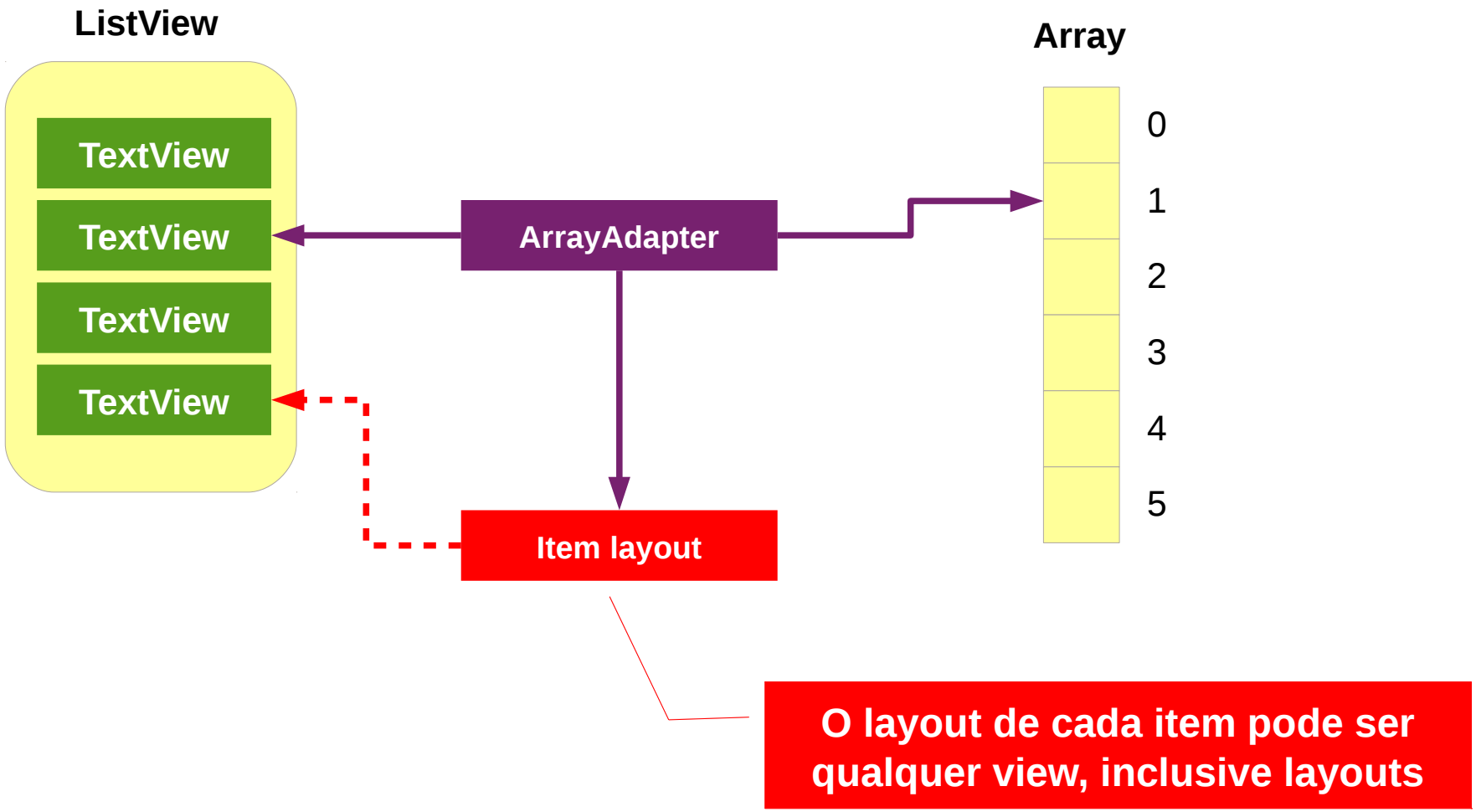
PROFª. M.Sc. JULIANA H Q BENACCHIO

Adapters

INSTITUTO FEDERAL
PARANÁ

- Os adapters fazem uma "ponte" entre os dados que você quer exibir na tela e as **`views`**
- Diversas **`views`** fazem o uso de adapters para exibir dados
  - **`ListView`**
  - **`Spinner`**
  - **`GridView`**
  - **`AutoCompleteTextView`**

Adapters


ListView
TextView
TextView
TextView
TextView
ArrayAdapter
Item layout
Array
0
1
2
3
4
5
O layout de cada item pode ser qualquer view, inclusive layouts

Adapters do Android

INSTITUTO FEDERAL
PARANÁ

- O Android possui alguns adapters nativos
- **`ArrayAdapter`**
  - Um **`array`** é usado como fonte de dados para a **`view`**
  - O tipo destes dados pode ser parametrizado via **`generics`**
- **`SimpleCursorAdapter`**
  - Um **`cursor`** é usado como fonte de dados para a **`view`**
  - Usado quando os dados vêm de um banco de dados ou de um **`content provider`**

Adapters do Android

INSTITUTO FEDERAL
PARANÁ

- Você pode criar seus próprios adapters, quando os existentes não atendem a sua necessidade
  - Herança de **`BaseAdapter`**
  - Herança de um adapter existente
- O método **`getView()`** deve ser sobrescrito

```
public View getView(int position, View convertView, ViewGroup parent)
{
    //...
}
```

Retorna a view de cada item do adapter

ListView

INSTITUTO FEDERAL
PARANÁ

- Lista itens na vertical e permite a seleção de itens
- Quando uma **`activity`** é composta apenas pela lista, basta herdá-la de **`ListActivity`**
- **`setListAdapter()`** é usado na definição do adapter

Criando ListViews

INSTITUTO FEDERAL
PARANÁ

```
public class MyActivity extends ListActivity {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        ArrayAdapter<CharSequence> adapter = ArrayAdapter
            .createFromResource(this, R.array.eletronicos,
            android.R.layout.simple_list_item_1);

        setListAdapter(adapter);
    }
}
```

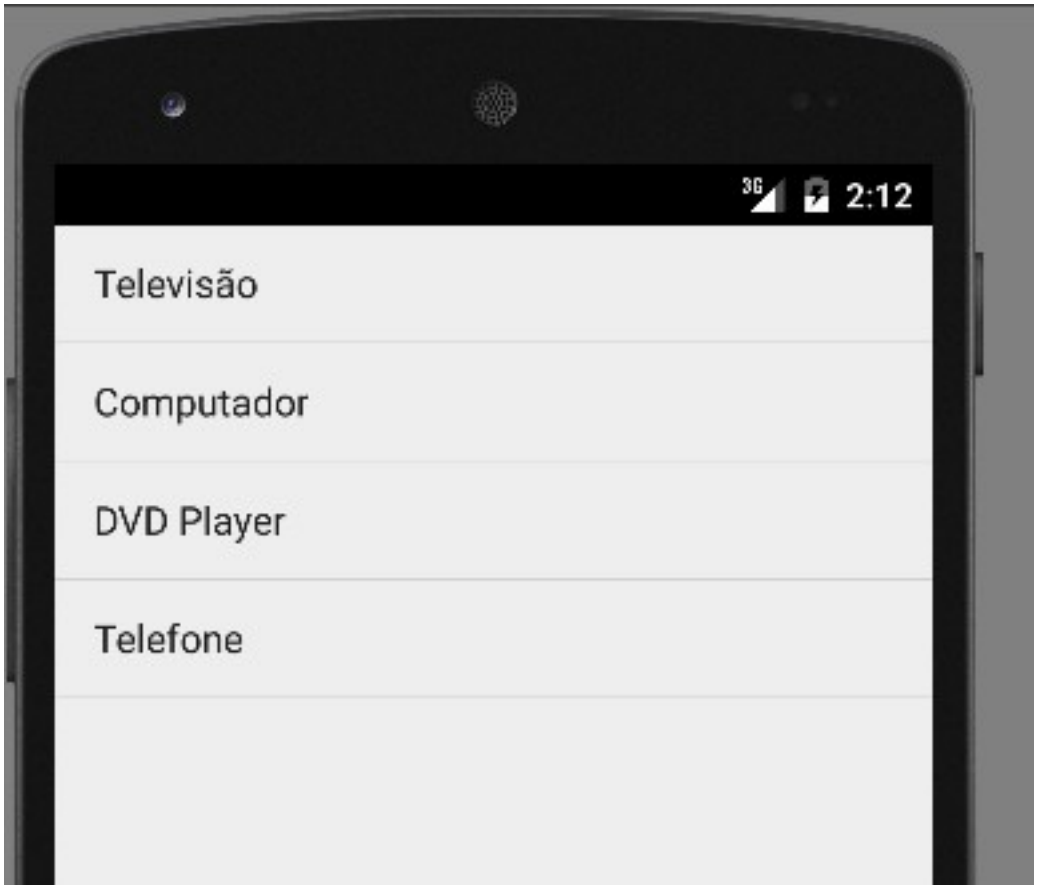
/res/values/arrays.xml

```
<resources>
    <string-array name="eletronicos">
        <item>Televisão</item>
        <item>Computador</item>
        <item>DVD Player</item>
        <item>Telefone</item>
    </string-array>
</resources>
```

Criando ListViews

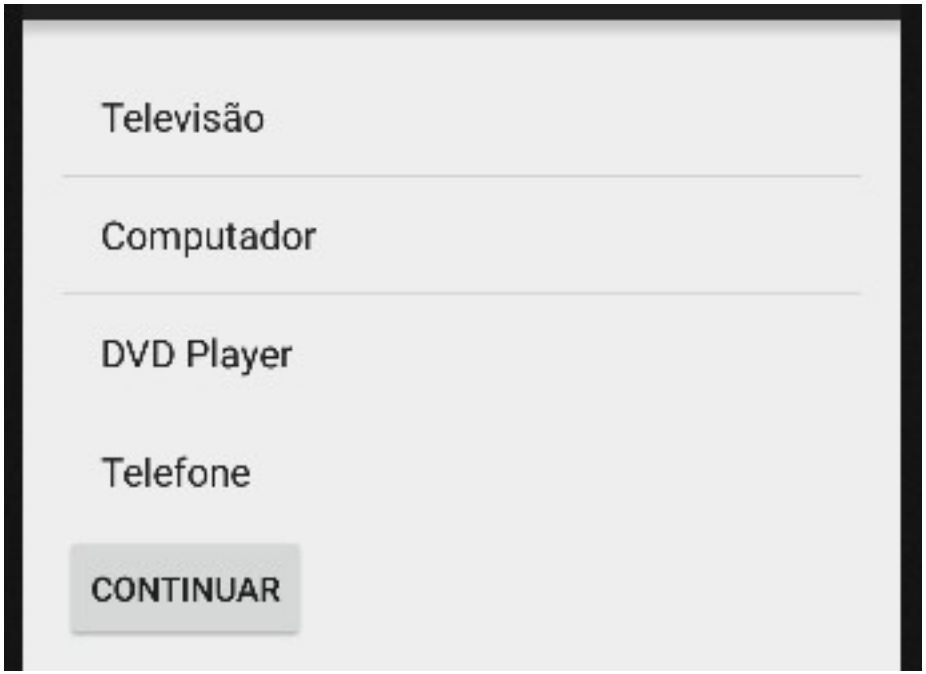
INSTITUTO FEDERAL
PARANÁ

- Um adapter pode ser utilizado para customizar a forma como cada item da lista será mostrado



Criando ListViews

- Se a **`activity`** não for composta apenas pela **`ListView`**, é possível declarar a **`view`** num arquivo de layout, juntamente com os outros elementos necessários



Criando ListViews

INSTITUTO FEDERAL
PARANÁ

/res/layout/activity_main.xml

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

  <ListView
       android:id="@+id/lista"
       android:layout_width="match_parent"
       android:layout_height="wrap_content" />

  <Button
       android:id="@+id/btnContinuar"
       android:text="@string/txt_continuar"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content" />

</LinearLayout>
```

Criando ListViews

INSTITUTO FEDERAL
PARANÁ

```
public class MyActivity extends AppCompatActivity {

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        ListView lista = (ListView) findViewById(R.id.lista);

        ArrayAdapter<CharSequence> adapter = ArrayAdapter
            .createFromResource(this, R.array.eletronicos,
            android.R.layout.simple_list_item_1);

    list.setAdapter(adapter);
}
```

Escolhendo Itens da ListView

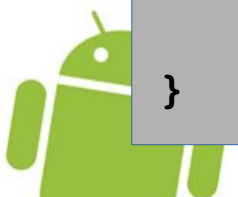
INSTITUTO FEDERAL
PARANÁ

- É possível registrar um **`listener`** na **`ListView`** para ser chamado quando determinado elemento é clicado

```
public class MyActivity extends Activity
    implements OnItemClickListener {

    public void onCreate(Bundle savedInstanceState) {
        //...
        list.setOnItemClickListener(this);
    }

    public void onItemClick(AdapterView<?> parent, View view,
        int position, long id) {
        //...
    }
}
```



Escolhendo Itens da ListView

INSTITUTO FEDERAL
PARANÁ

- No caso da activity herdar de **`ListActivity`**, o método **`onListItemClick()`** pode ser sobrescrito

```
public void onListItemClick(ListView l, View v, int position,
    long id) {

    //...
}
```

A Classe ListFragment

INSTITUTO FEDERAL
PARANÁ

- Com o surgimento dos fragments, foi criada a classe **`ListFragment`**
- Seu funcionamento é igual ao **`ListActivity`**, mas é usada em fragments

```
public class MyFragment extends ListFragment {
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        ArrayAdapter<CharSequence> adapter = ArrayAdapter
            .createFromResource(this, R.array.eletronicos,
            android.R.layout.simple_list_item_1);

        setListAdapter(adapter);
    }
}
```

A Classe ListFragment

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- Desde a primeira versão do Android, o **`ListView`** foi a view padrão para criar listas
- Com o surgimento do **Material Design**, também foi criado o **`RecyclerView`**, que é o novo **`ListView`** do Android, e a partir de agora é a view recomendada para criar listas segundo as boas práticas de interface

Criando RecyclerView

INSTITUTO FEDERAL
PARANÁ

- O **`RecyclerView`** tem uma configuração diferente, que é o gerenciador de layout, chamado de **`LayoutManager`**
- Outra melhoria é no suporte as animações, através da subclasse **`ItemAnimator`**

```
RecyclerView rv = (RecyclerView) view.findViewById(R.id.lista);
RecyclerView.LayoutManager layout = new LinearLayoutManager(getActivity());
rv.setLayoutManager(layout);
rv.setItemAnimator(new DefaultItemAnimator());
```

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- O **`RecyclerView`** também utiliza o conceito de **adapters** para preencher o conteúdo da lista, sendo que um adapter deve ser uma subclasse de **`RecyclerView.Adapter`**
- Basicamente o Google separou as responsabilidades de quem faz o controle do reaproveitamento das views e de quem organiza o layout

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- O **`RecyclerView`** na verdade é um componente especializado no reaproveitamento das views, ou seja, ele recicla as views e implementa automaticamente o padrão **`ViewHolder`**, para garantir um bom desempenho ao fazer rolagem com uma grande quantidade de itens.

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- Mas o **`RecyclerView`** não sabe desenhar nada na tela, e para isso ele precisa de alguma subclasse de **`LayoutManager`**, a qual é responsável por desenhar e organizar a disposição das views. Entre elas:
  - **`LinearLayoutManager`**
  - **`GridLayoutManager`**
  - **`StaggeredGridLayoutManager`**

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- A classe **`DefaultItemAnimator`** é filha de **`RecyclerView.ItemAnimator`** e implementa as animações básicas quando um item da lista é adicionado, removido ou movido de posição.
- Depois de inserir alguma informação na lista que é a fonte do conteúdo do adapter é possível chamar o método **`notifyItemInserted(idx)`** para informar que um item foi adicionado ou o método **`notifyItemRemoved(idx)`** para informar que um item foi removido da posição indicada.

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- Primeiramente, para utilizar o **`RecyclerView`** é preciso declarar a seguinte dependência no arquivo **`build.gradle`**

```
Dependencies{
...
   Compile 'com.android.support:recyclerview-v7:21.0.+'
}
```

RecyclerView

INSTITUTO FEDERAL
PARANÁ

- Um adapter do **`RecyclerView`** utiliza o conceito de Generics do Java (tipos genéricos) portanto os métodos **`onCreateViewHolder()`** e **`onBindViewHolder()`** recebem o tipo genérico declarado na classe.