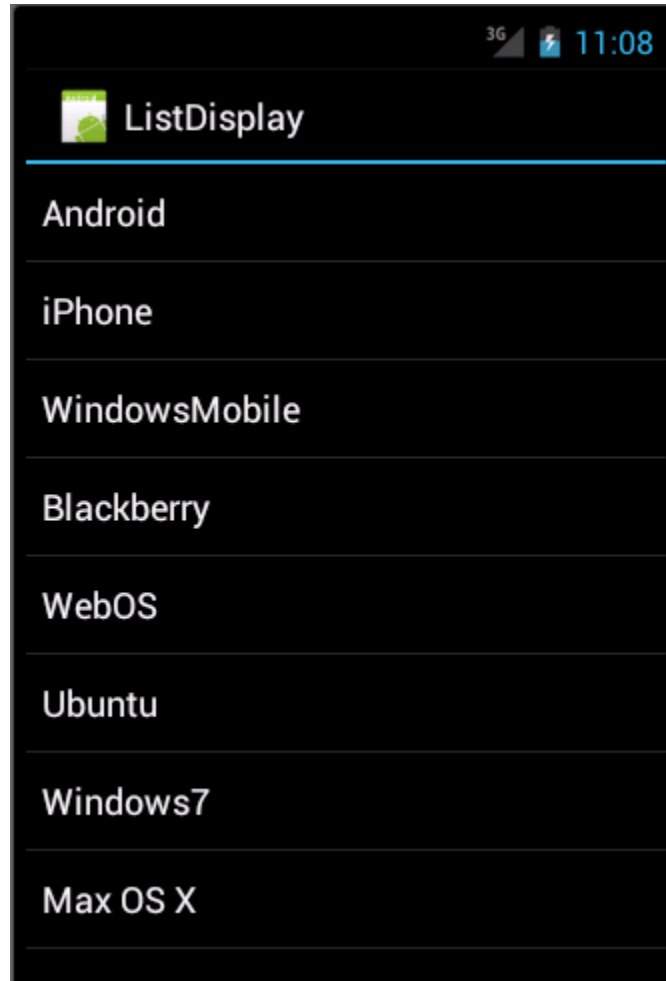


Programação para Dispositivos Móveis

05 – ListView



Possibilitam mostrar longas listas de dados.



Insira um controle do tipo ListView.

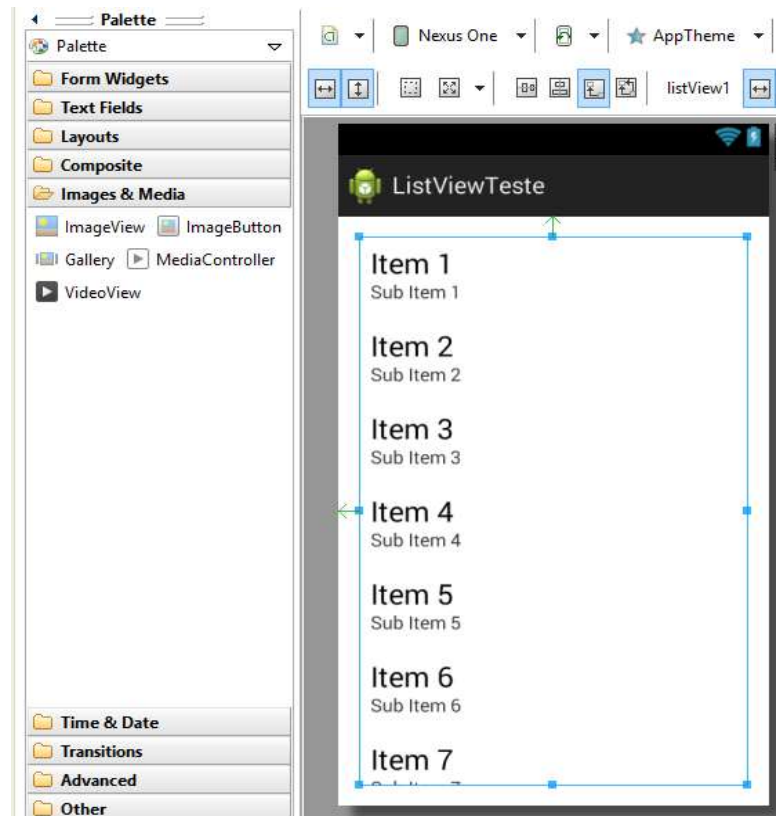
Algumas características que podemos mudar via xml;

`android:divider="@android:color/transparent"`

`android:dividerHeight="100dp"`

`android:dividerPadding="22dip"`

`android:showDividers="middle"`



Deveremos criar um xml contendo o item que será mostrado na ListView, neste primeiro exemplo utilizaremos uma listview básica.

```
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textoLinha"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="@dimen/tam_lv" />
```

```
public class MainActivity extends Activity {
    ListView lv;

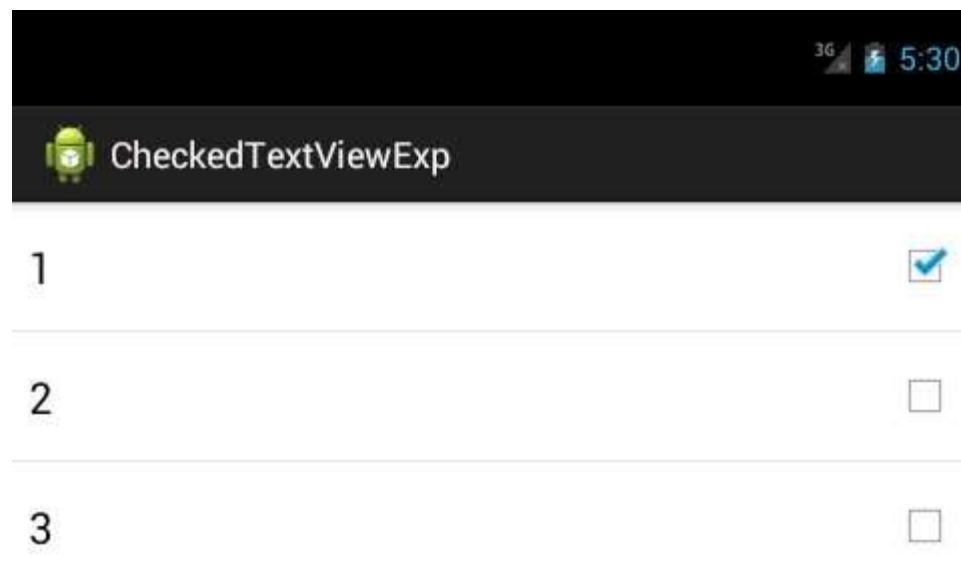
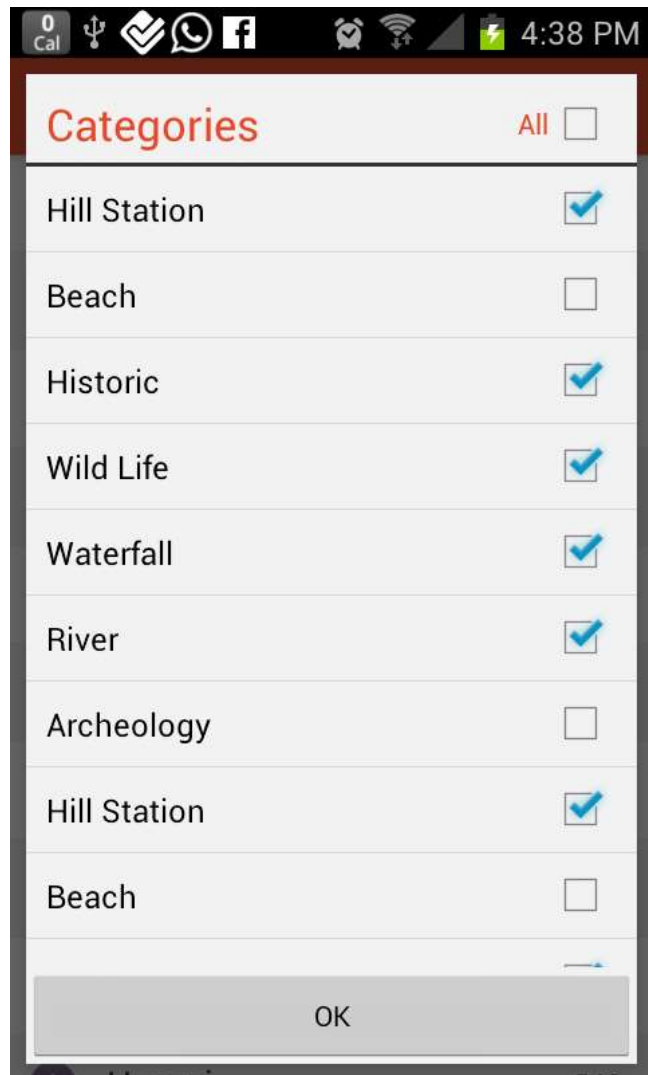
    String opcoes[] = new String[] { "Um", "Dois", "Três", "Quatro", "Cinco",
        "Seis", "Sete", "Oito", "Nove", "Dez" };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Instanciamos o objeto lv que é do tipo ListView
        lv = (ListView) findViewById(R.id.listView1);

        // Definimos o Adapter sendo um vetor de strings, utilizando o layout linha_lv,
        // com as opcoes definidas no vetor opcoes
        lv.setAdapter(new ArrayAdapter<String>(this, R.layout.linha_lv, opcoes));

        // Quando clicado um item, o que fazemos
        lv.setOnItemClickListener(new AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view, int posicao,
                long posicaoLong) {
                // Sua escolha é recebida através da posicao
                Toast.makeText(getApplicationContext(),
                    "Voce Escolheu -->" + posicao, Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```



Modificar o ListView colocando o tipo de escolha (choiceMode).

O que acontece se escolher singleChoice ao invés de multipleChoice ?

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >

    <ListView
        android:id="@+id/listView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:divider="@android:color/transparent"
        android:dividerHeight="40dp"
        android:choiceMode="multipleChoice"

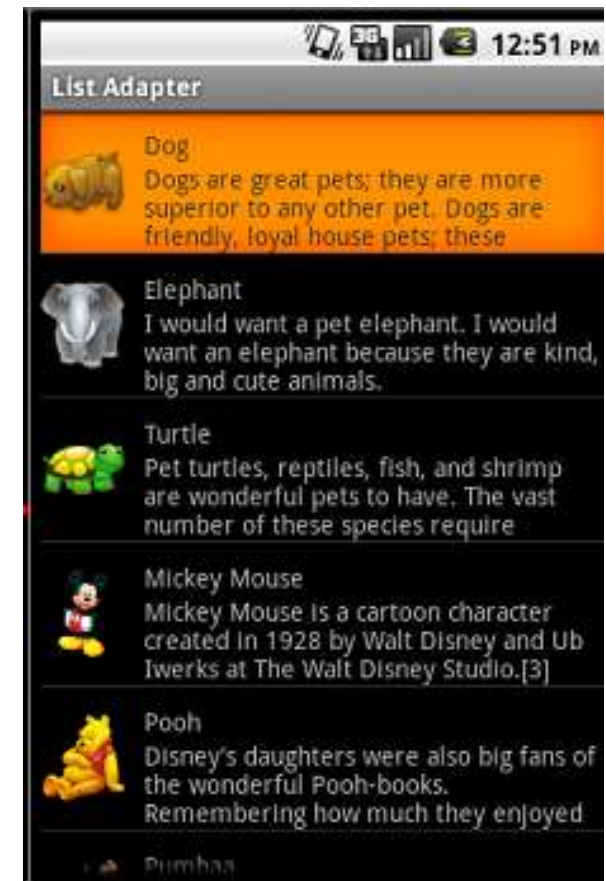
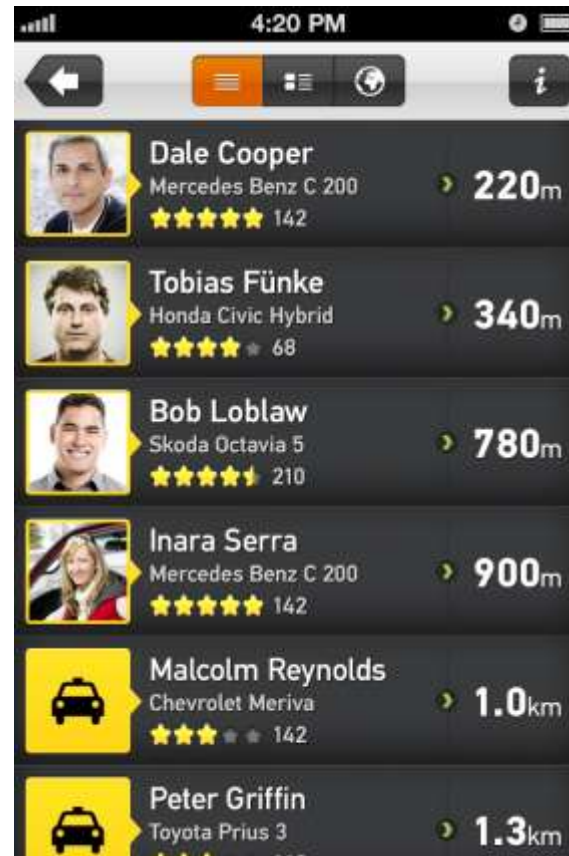
        >
    </ListView>

</RelativeLayout>
```

Modificar o xml da linha do item de TextView para CheckedtextView

```
<?xml version="1.0" encoding="utf-8"?>
<CheckedTextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textoLinha"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:textSize="@dimen/tam_lv"
    android:gravity="center_vertical"
    android:checkMark="?android:attr/listChoiceIndicatorMultiple"
    android:paddingLeft="6dp"
    android:paddingRight="6dp"

/>
```

Criar o xml do item da ListView customizado. No caso abaixo uma imagem com texto.

```
<?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"
    >

    <ImageView
        android:id="@+id/imageViewLinha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:src="@drawable/ic_launcher" />

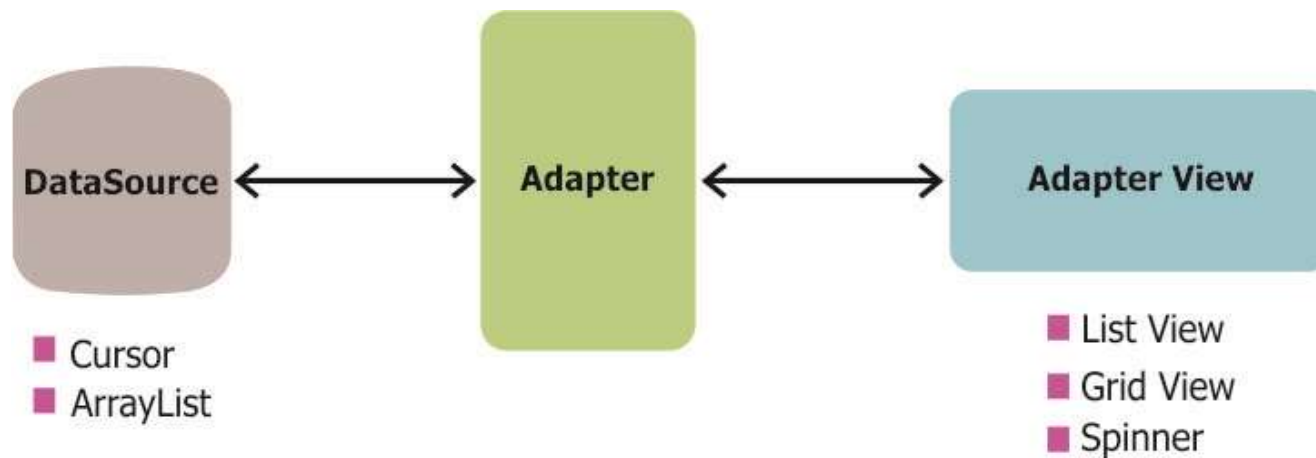
    <TextView
        android:id="@+id/textViewLinha"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center_vertical"
        android:text="Large Text"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

Criar a Classe que representa a linha do ListView. (Item). Criar os getters e setters

```
public class LinhaItem {  
    String texto;  
    int IDdaImagem;  
  
    public LinhaItem(String texto, int iDdaImagem) {  
        this.texto = texto;  
        IDdaImagem = iDdaImagem;  
    }  
  
    public String getTexto() {  
        return texto;  
    }  
  
    public void setTexto(String texto) {  
        this.texto = texto;  
    }  
  
    public int getIDdaImagem() {  
        return IDdaImagem;  
    }  
  
    public void setIDdaImagem(int iDdaImagem) {  
        IDdaImagem = iDdaImagem;  
    }  
}
```

Adapter são a ponte entre o Adapter View (Exemplo: ListView) e os dados mostrados na View.



Estender a classe ArrayAdapter para se adequar ao seu ListView customizado

```
public class MeuAdapter extends ArrayAdapter<LinhaItem> {
    Context context;

    public MeuAdapter(Context context, int resource, List<LinhaItem> itens) {
        super(context, resource, itens);
        this.context = context;
        // TODO Auto-generated constructor stub
    }

    public class Item {
        ImageView imageViewLogo;
        TextView textViewTitulo;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        Item item = null;
        LinhaItem linha = getItem(position);

        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Activity.LAYOUT_INFLATER_SERVICE);
        if (convertView == null) {
            convertView = inflater.inflate(R.layout.linha_lv_custom, null);
            item = new Item();
            item.imageViewLogo = (ImageView) convertView
                .findViewById(R.id.imageViewLinha);
            item.textViewTitulo = (TextView) convertView
                .findViewById(R.id.textViewLinha);
            convertView.setTag(item);
        } else {
            item = (Item) convertView.getTag();
        }
        item.textViewTitulo.setText(linha.getTexto());
        item.imageViewLogo.setImageResource(linha.getIDdaImagem());

        return convertView;
    }
}
```

```
public class CustomAdapter extends Activity {
    ListView lv;
    List<LinhaItem> itemLinha;

    String opcoes[] = new String[] { "FaceBook", "LinkedIn", "Twiteer",
        "Youtube" };

    int[] imagensID = { R.drawable.facebook, R.drawable.linkedin,
        R.drawable.twiteer, R.drawable.youtube };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_custom_adapter);

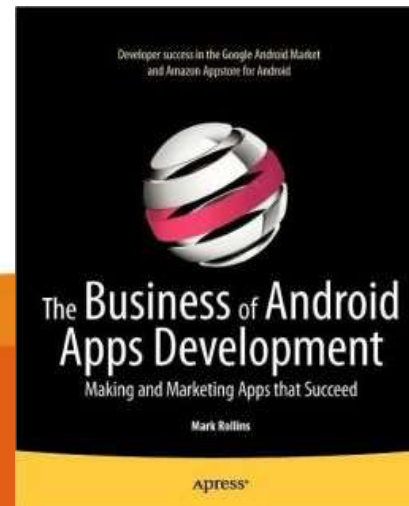
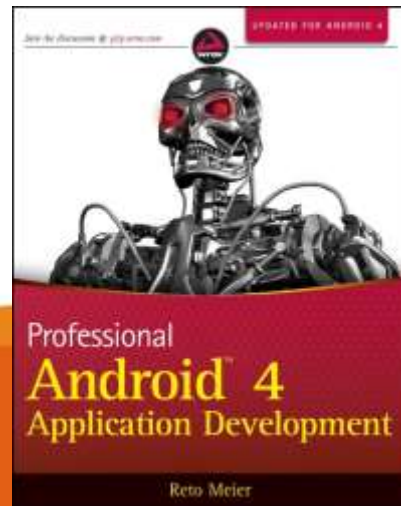
        itemLinha = new ArrayList<LinhaItem>();
        for (int i = 0; i < opcoes.length; i++) {
            LinhaItem item = new LinhaItem(opcoes[i], imagensID[i]);
            itemLinha.add(item);
        }

        lv = (ListView) findViewById(R.id.listView1);
        MeuAdapter adapter = new MeuAdapter(this, R.layout.linha_lv_custom,
            itemLinha);
        lv.setAdapter(adapter);
    }
}
```

Folha 05-Exercícios



Anhanguera



Lauren Dorsey
Shane Conder

