

ListView customizada

Para criarmos uma list view customizada precisamos criar um layout próprio para nossa ListView, e inseri-lo no nosso adapter customizado.

Vamos criar o student_item.xml:

- *Ele deve conter um nome e a imagem do aluno, seguindo o exemplo abaixo:*

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="#FFFFFF"
    android:orientation="horizontal"
    android:padding="5dip" >

    <!-- Title Of Song -->

    <TextView
        android:id="@+id/name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text=""
        android:textColor="@android:color/black"
        android:textSize="15dip"
        android:textStyle="bold"
        android:typeface="sans" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_centerVertical="true"
        android:src="@drawable/abc_ic_search_api_mtrl_alpha" />

</RelativeLayout>
```

Agora, vamos criar um adapter customizado baseado no BaseAdapter:

No adapter temos que ter alguns pontos, comentados no código:

```
public class StudentAdapter extends BaseAdapter {

    private List<Student> students = null;
    private Context context = null;
    //Temos que passar nosso array e o contexto da activity que está nossa listView
    public StudentAdapter(List<Student> students, Context context) {
        this.students = students;
        this.context = context;
```

```

    }

    //Temos que implementar um metodo que retorna o tamanho do array
    @Override
    public int getCount() {
        return students.size();
    }

    //Temos que implementar um metodo que retorna o um item, passando a posição dele.
    @Override
    public Object getItem(int position) {
        return students.get(position);
    }

    //Temos que implementar um metodo que retorna o id do item, passando a posição dele.
    @Override
    public long getItemId(int position) {
        return students.get(position).getId();
    }

    //Temos que implementa a view e o layout que vai ser apresentado na list view.
    @Override
    public View getView(int position, View convertView, ViewGroup viewGroup) {

        //busca o item do item passado.
        Student c = students.get(position);

        //busca a estrutura para criarmos uma layout do resources.
        LayoutInflater inflater = (LayoutInflater)
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        //Seta o layout customizado que vem do Resources.
        View view = inflater.inflate(R.layout.student_item, null);

        //Da um findviewbyid nos components do layout customizado da listview.
        TextView type = (TextView) view.findViewById(R.id.name);

        //Seta o valor na view
        type.setText(String.valueOf(c.getName()));

        //Retorna o layout.
        return view;
    }

```

Agora, vamos criar um adapter customizado baseado no BaseAdapter:

No adapter temos que ter alguns pontos, comentados no código:

```
studentAdapter = new StudentAdapter(students, this);
studentListView.setAdapter(studentAdapter);
```

//Para criar um menu de contexto

```
registerForContextMenu(studentListView);
```

//Para criar um menu de context com click rapido

```
studentListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int
position, long id) {
        Student student = (Student)
        studentListView.getItemAtPosition(position);
        Log.i("click", student.getName());
    }
});
```

//Para criar um menu de context com click longo

```
studentListView.setOnItemLongClickListener(new
AdapterView.OnItemLongClickListener() {
    @Override
    public boolean onItemLongClick(AdapterView<?> adapterView, View
view, int position, long id) {
        Student value = (Student) studentListView.getItemAtPosition(position);
        Log.i("click", value.getName());
        Toast.makeText(StudentActivity.this, "Clicado no item - Long" +
position + "valor" + value, Toast.LENGTH_SHORT).show();
        return false;
    }
});
```