



++

45697056

Análise e Desenvolvimento de Sistemas

Mobile Development and IOT - Android

45697056

■ ■ ■

Criando listagem de dados com ListView



Prof. Douglas Cabral <douglas.cabral@fiap.com.br>
<https://www.linkedin.com/in/douglascabral/>

Criando listagem de dados com ListView

Quando estamos desenvolvendo um aplicativo com muitos dados, provavelmente precisaremos listá-los em algum momento.

Um componente muito utilizado para isso é o **ListView**.

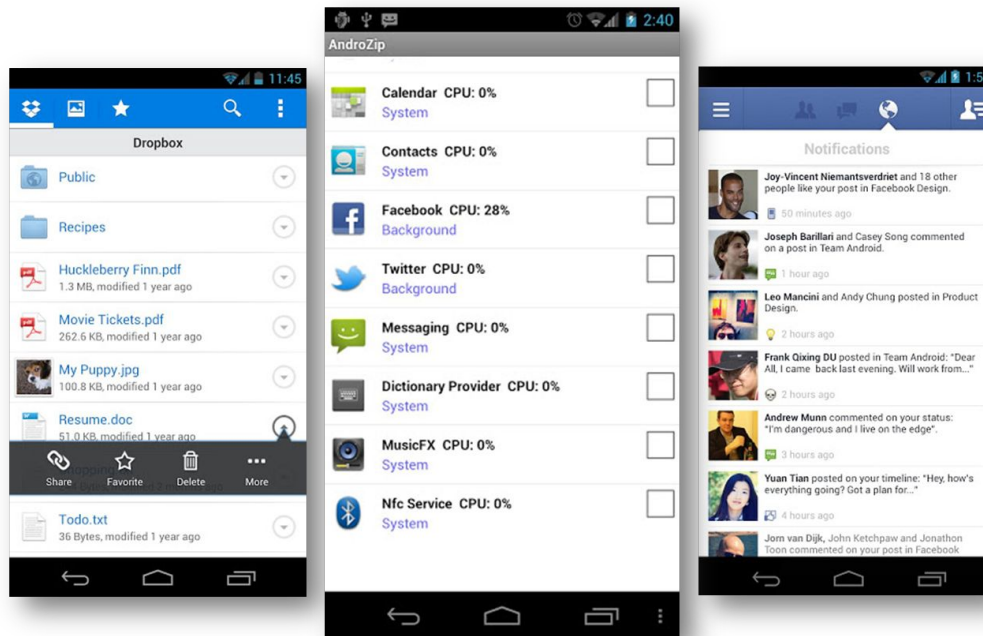


Imagem de Referência: <http://stacktips.com/tutorials/android/android-listview-tutorial>

Criando listagem de dados com ListView

Vamos inserir em nosso layout o componente ListView da seguinte maneira:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context="br.com.fiap.fiapzap.MainActivity">

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

</LinearLayout>
```




Criando listagem de dados com ListView

Vamos referenciar a ListView dentro de nossa Activity conforme a imagem abaixo:


```
package br.com.fiap.fiapzap;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ListView;

public class MainActivity extends AppCompatActivity {

    ListView lstContatos; 

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

        lstContatos = (ListView) findViewById(R.id.lstContatos); 
    }
}
```

Criando listagem de dados com ListView

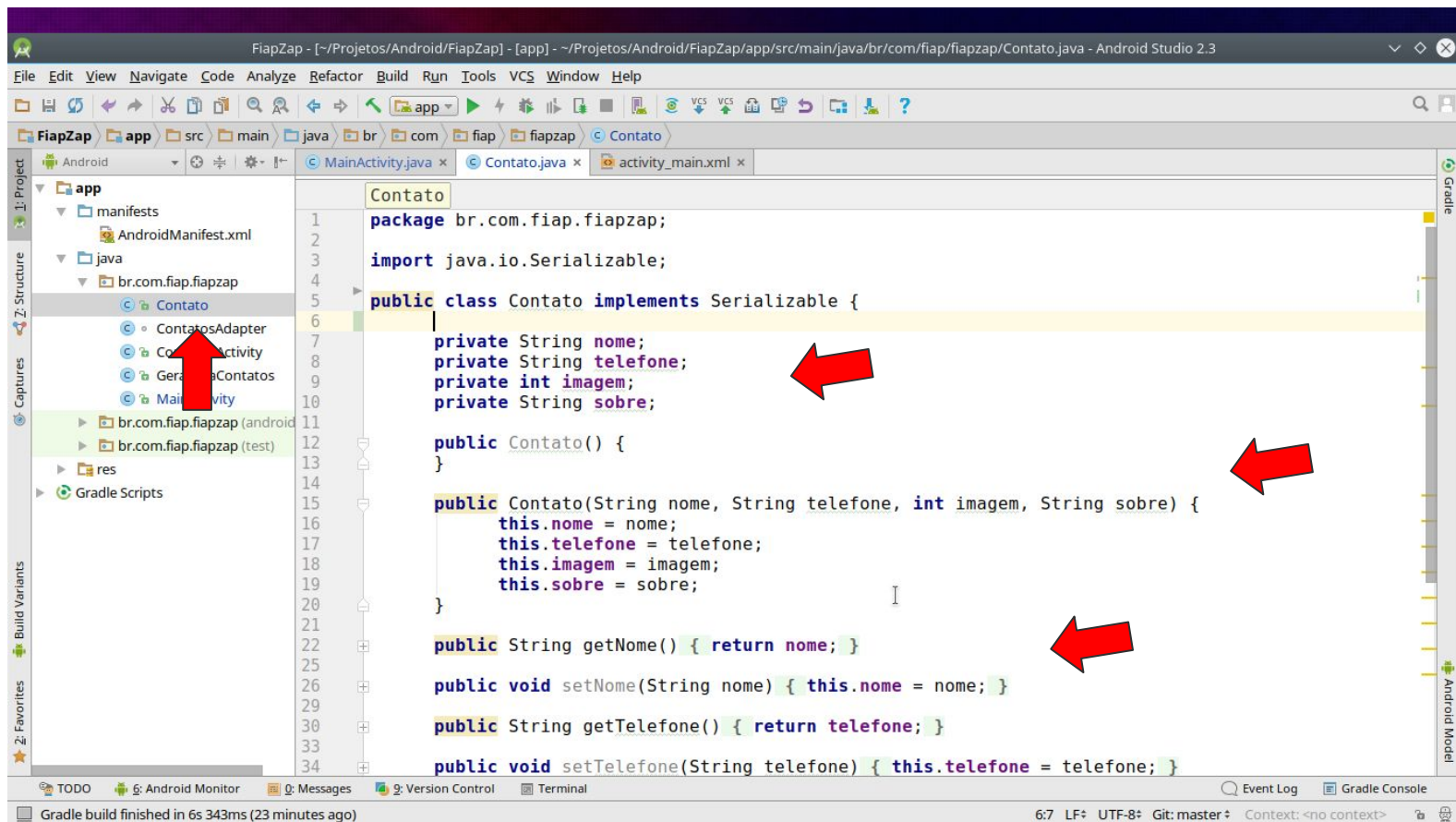


Criaremos uma classe representando um Contato contendo os seguintes dados:

- nome (String)
- telefone (String)
- sobre (String)
- imagem (int)

Fazer os métodos GET e SET de cada atributo e métodos construtores necessários.

Criando listagem de dados com ListView



```
1 package br.com.fiap.fiapzap;
2
3 import java.io.Serializable;
4
5 public class Contato implements Serializable {
6
7     private String nome;
8     private String telefone;
9     private int imagem;
10    private String sobre;
11
12    public Contato() {
13    }
14
15    public Contato(String nome, String telefone, int imagem, String sobre) {
16        this.nome = nome;
17        this.telefone = telefone;
18        this.imagem = imagem;
19        this.sobre = sobre;
20    }
21
22    public String getNome() { return nome; }
23
24    public void setNome(String nome) { this.nome = nome; }
25
26    public String getTelefone() { return telefone; }
27
28    public void setTelefone(String telefone) { this.telefone = telefone; }
```

Criando listagem de dados com ListView



Os dados poderiam vir de uma API, de um banco de dados ou de qualquer outro lugar, porém para facilitar o aprendizado vamos criar uma classe para gerar uma lista de contatos fake.

Crie uma classe chamado **GeraListaContatos**.

Esta classe terá uma método estático chamado **geraContatos** que retorna um **List<Contato>**.

Criando listagem de dados com ListView

```
package br.com.fiap.fiapzap;

import java.util.ArrayList;
import java.util.List;

public class GeraListaContatos {

    public static List<Contato> geraContatos() {
        List<Contato> contatos = new ArrayList<>();

        Contato contato1 = new Contato(
            "Maria José", "(11) 1111-1111", R.drawable.matureman2, "Hello! I'm in FIAP ZAP");
        Contato contato2 = new Contato(
            "Neusa Chamma", "(12) 1111-1112", R.drawable.female2, "Hello! I'm in FIAP ZAP");
        Contato contato3 = new Contato(
            "Joaquim Barbosa", "(13) 1111-1113", R.drawable.male2, "Hello! I'm in FIAP ZAP");

        contatos.add(contato1);
        contatos.add(contato2);
        contatos.add(contato3);

        return contatos;
    }
}
```

Busque na internet algumas imagens
para usar como foto do contato e salve na
pasta drawable



Criando listagem de dados com ListView

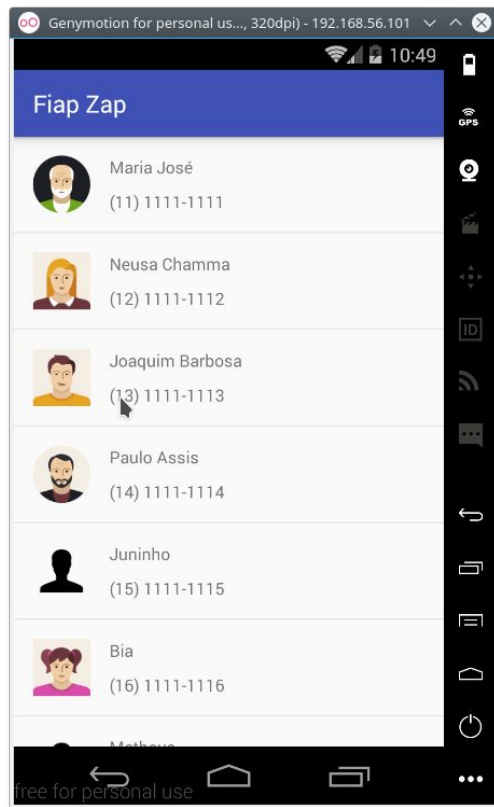
A aparência de nossa lista deverá ser como a imagem ao lado.

O **ListView** não sabe como queremos renderizar os dados dos contatos certo?

Para isso criaremos um **layout** com uma **ImageView** e dois **TextViews** que serão a foto do contato, nome e telefone.

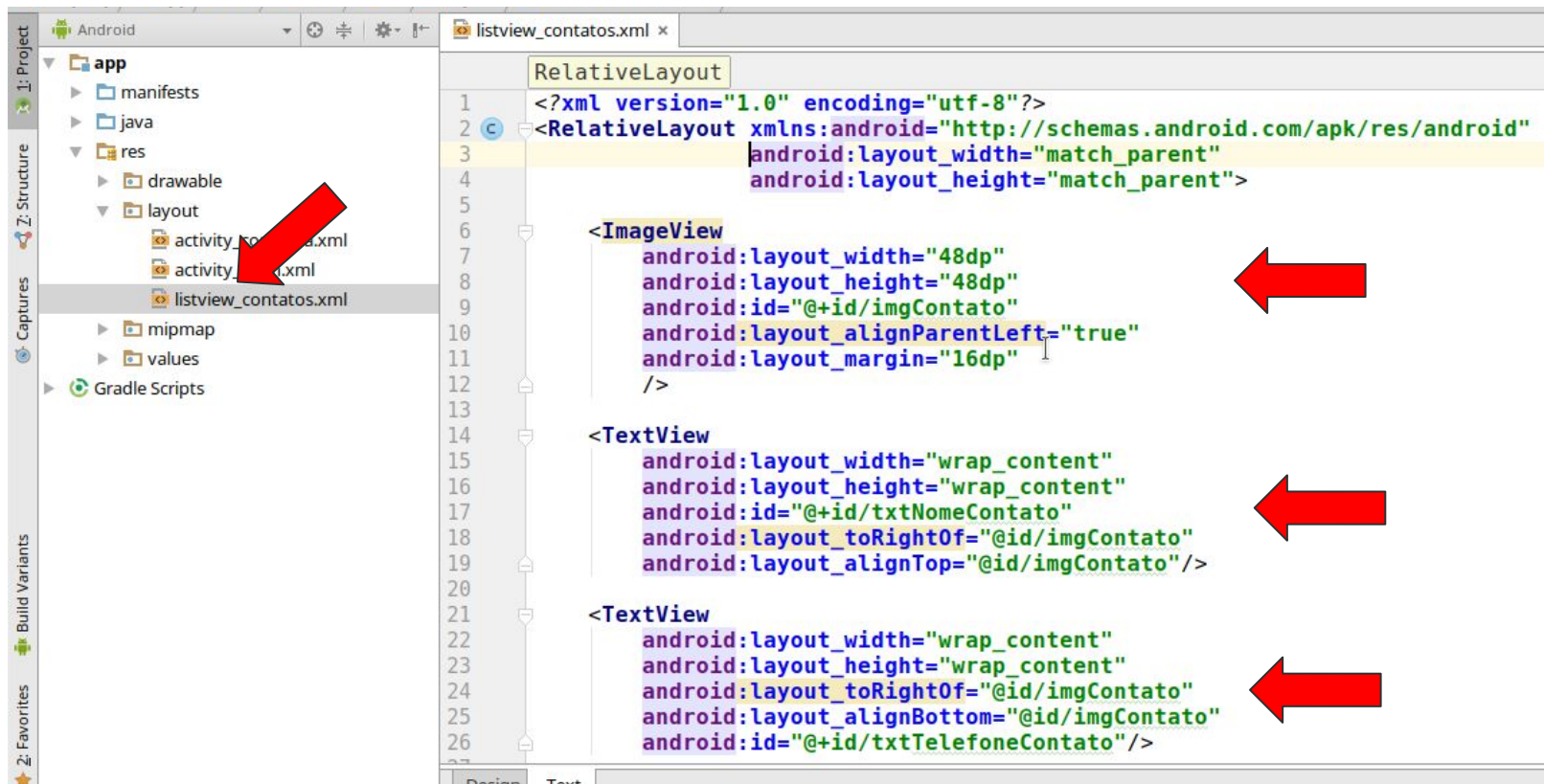
Cada item do **ListView** irá renderizar este **layout**.

O mapeamento de cada atributo da classe contato com cada View do layout será realizado posteriormente com um **Adapter**.



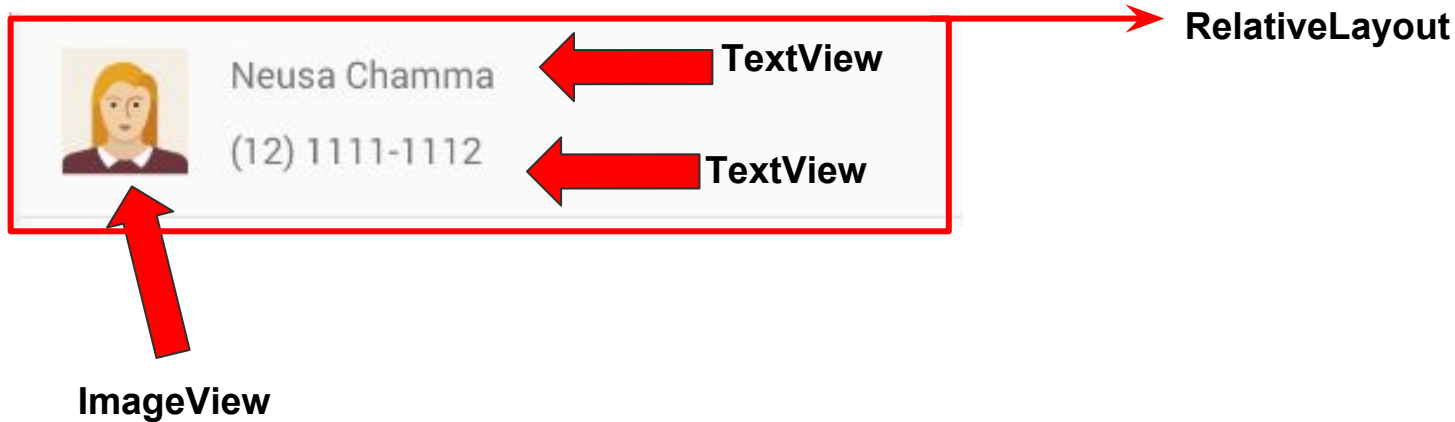
Criando listagem de dados com ListView

Crie uma nova layout chamada **listview_contato.xml** dentro da pasta **res/layout**



Criando listagem de dados com ListView

O layout criado irá renderizar os dados da seguinte forma quando preenchido com a foto, nome e telefone do contato:



Criando listagem de dados com ListView



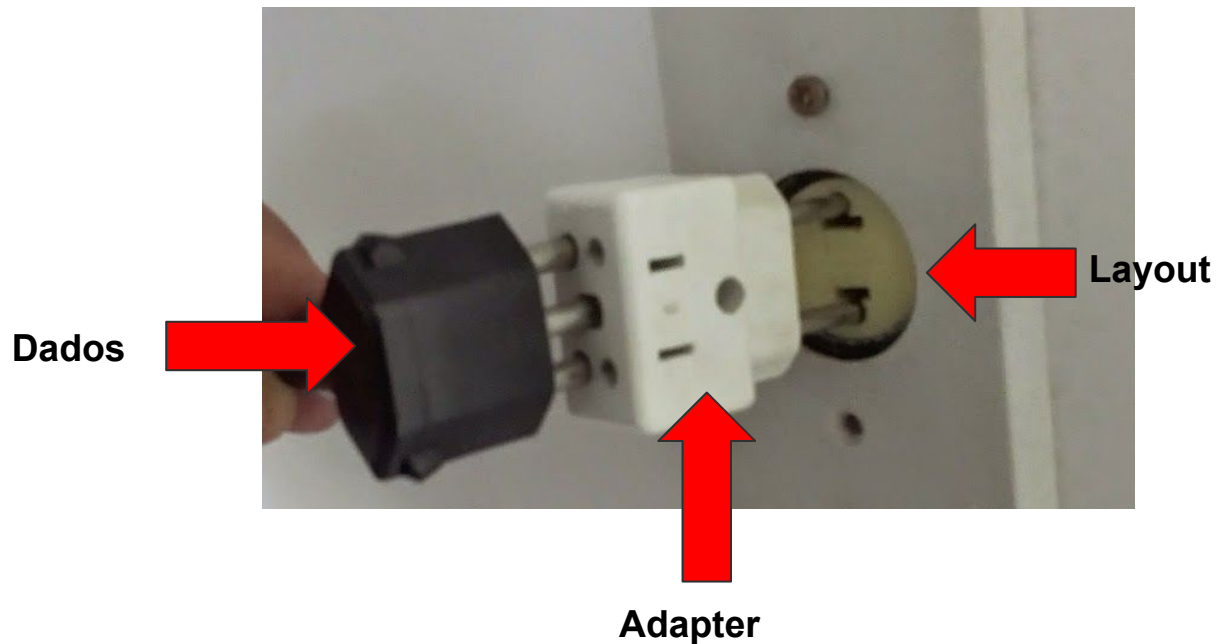
Já possuímos nosso **ListView**, nossa lista de contatos e o layout que precisamos para exibir os dados dos contatos dentro de cada item do **ListView**.

A questão agora é, como mapear cada atributo do Contato dentro de cada componente inserido na layout criada anteriormente.

Para isso, iremos utilizar um **Adapter** personalizado.

Criando listagem de dados com ListView

Analogia de um adapter



Criando listagem de dados com ListView

Criaremos uma classe chamada **ContatosAdapter** que estende a classe **BaseAdapter**.

```
package br.com.fiap.fiapzap;  
  
import android.widget.BaseAdapter;  
  
class ContatosAdapter extends BaseAdapter {  
  
}
```

Por herdar de **BaseAdapter**, somos obrigados a **implementar os seguintes métodos**:

- ❑ **getCount** - Total de registros para listar
- ❑ **getItem** - Retorna um item específico da lista
- ❑ **getItemId** - Retorna um identificador para um item da lista
- ❑ **getView** - Retorna uma view para renderizaar (Será nossa layout com os dados)

Criando listagem de dados com ListView

Criaremos dentro de nosso Adapter um construtor e dois atributos que irão armazenar o contexto onde o ListView está inserido e a lista de contatos.

```
|  
class ContatosAdapter extends BaseAdapter {
```

```
    Context context;
```

```
    List<Contato> contatos;
```



Criar os atributos do Adapter

```
    public ContatosAdapter(Context context, List<Contato> contatos) {  
        this.context = context;  
        this.contatos = contatos;  
    }
```



Criar um construtor para inicializar os atributos do Adapter

Criando listagem de dados com ListView

O método **getCount** deve retornar a quantidade de itens na lista que é o mesmo que o tamanho de nosso atributo **contatos**.

```
@Override
public int getCount() {
    return this.contatos.size();
}
```


Criando listagem de dados com ListView

O método **getItem** deve retornar um object de determinada posição da lista. Sua implementação fica assim:

```
@Override  
public Object getItem(int i) {  
    return (Object)this.contatos.get(i);  
}
```

Criando listagem de dados com ListView

O método **getItemId** deve conter alguma lógica para dar identificadores para cada item da lista.

Para facilitar, nosso identificador será a posição do item na lista.

```
@Override  
public long getItemId(int i) {  
    return i;  
}
```

Criando listagem de dados com ListView

O método **getView** será o responsável pela mágica do mapeamento dos atributos dos contatos de nossa **List<Contato>** para cada componente de nosso layout. Sua implementação será:

@Override

```
public View getView(int i, View view, ViewGroup viewGroup) {  
    LayoutInflater inflater = LayoutInflater.from(this.context);  
  
    Contato contato = this.contatos.get(i);  
  
    View v = inflater.inflate(R.layout.listview_contatos, null);  
    ImageView imgContato = (ImageView) v.findViewById(R.id.imgContato);  
    TextView txtNomeContato = (TextView) v.findViewById(R.id.txtNomeContato);  
    TextView txtTelefoneContato = (TextView) v.findViewById(R.id.txtTelefoneContato);  
  
    imgContato.setImageResource(contato.getImagem());  
    txtNomeContato.setText( contato.getNome() );  
    txtTelefoneContato.setText( contato.getTelefone() );  
  
    return v;  
}
```

Criando listagem de dados com ListView



Já possuímos nosso Adapter e agora iremos juntar as peças em nossa activity.

Temos nosso ListView criado e inicializado dentro da Activity, falta apenas instanciarmos o adapter com os dados de nosso gerador de dados fake.

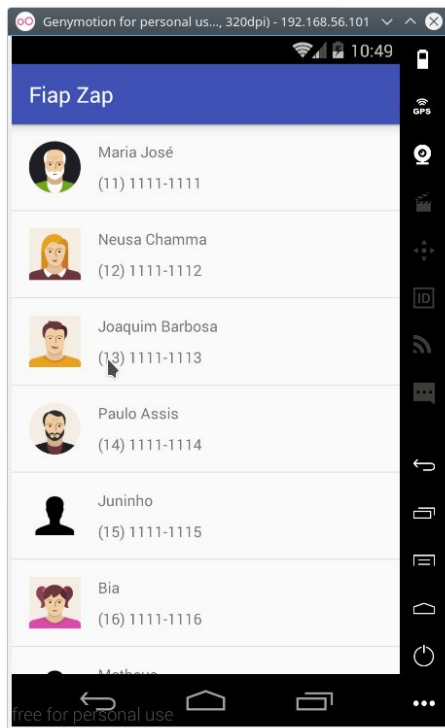
Criando listagem de dados com ListView

O código de nossa Activity ficará:

```
public class MainActivity extends AppCompatActivity {  
  
    List<Contato> contatos;  
    ListView lstContatos;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        contatos = GeraListaContatos.geraContatos();  
  
        lstContatos = (ListView) findViewById(R.id.lstContatos);  
  
        ContatosAdapter contatosAdapter = new ContatosAdapter(this, contatos);  
  
        lstContatos.setAdapter(contatosAdapter);  
    }  
}
```

Criando listagem de dados com ListView

Ao executar nossa aplicação o resultado final será como o esperado:



Criando listagem de dados com ListView

Caso queira criar o evento de clique para itens do ListView, basta implementar no ListView o método `setOnItemClickListener`. Ex:

```
lstContatos.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l)  
    }  
});
```

Dúvidas?



Copyright © 2018 Prof. Douglas Cabral <douglas.cabral@fiap.com.br> <https://www.linkedin.com/in/douglascabral/>

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).