

Desenvolvimento de Aplicações para Dispositivos Móveis

Aula 05 - ListViews: Filtros, Busca e Intents

- ListViews (Revisão)
- Filtragem de Dados
- Envio de Dados Através de Intents

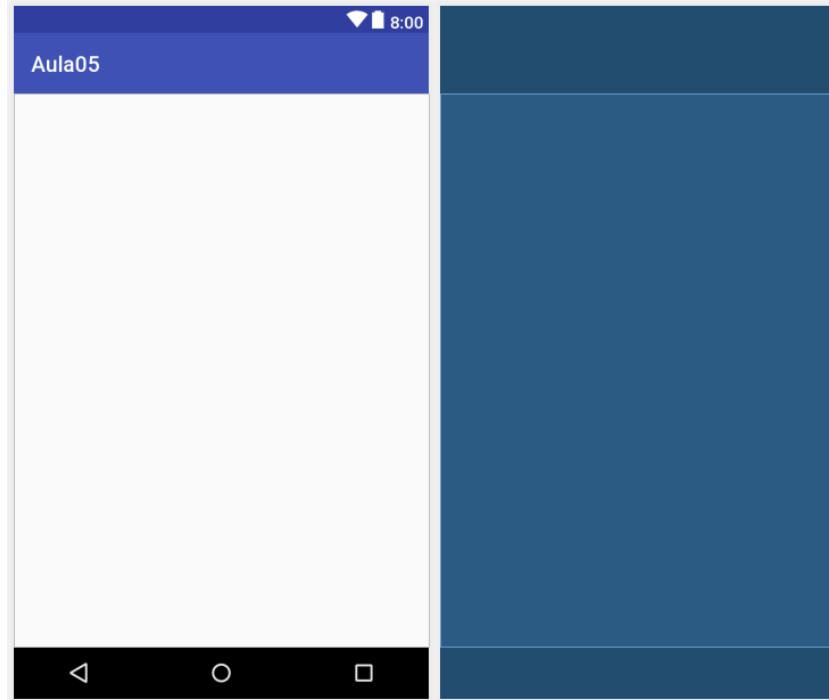
Prof. Fernando Gonçalves Abadia

ListsViews: Revisão

- ▶ O **ListView** é um componente visual utilizado para o Android onde sua finalidade é de armazenar uma ampla quantidade de informações e com o poderio de atualizar, receber e enviar eventos dinamicamente.
- ▶ Para esta aula, usaremos alguns recursos vistos, como o **ListView**, **Intents** e também transições de tela.

ListViews: Revisão

- ▶ Para iniciar o aplicativo desta aula, precisaremos iniciar um novo projeto, aula05 com uma atividade em branco.
- ▶ E, da mesma forma que na aula04, no XML da atividade principal, deve-se apagar o TextView do Hello World de Entrada.



ListViews: Revisão

- ▶ Precisaremos de um XML com um **RelativeLayout** e com um **ListView** com um ID, como por exemplo: “lista”.



The screenshot shows an IDE interface with two tabs: "activity_main.xml" and "MainActivity.java". The "activity_main.xml" tab is active, displaying the following XML code:

```
<?xml version="1.0" encoding="utf-8" ?>
<RelativeLayout 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"
    tools:context="com.example.arfso.aula05.MainActivity">

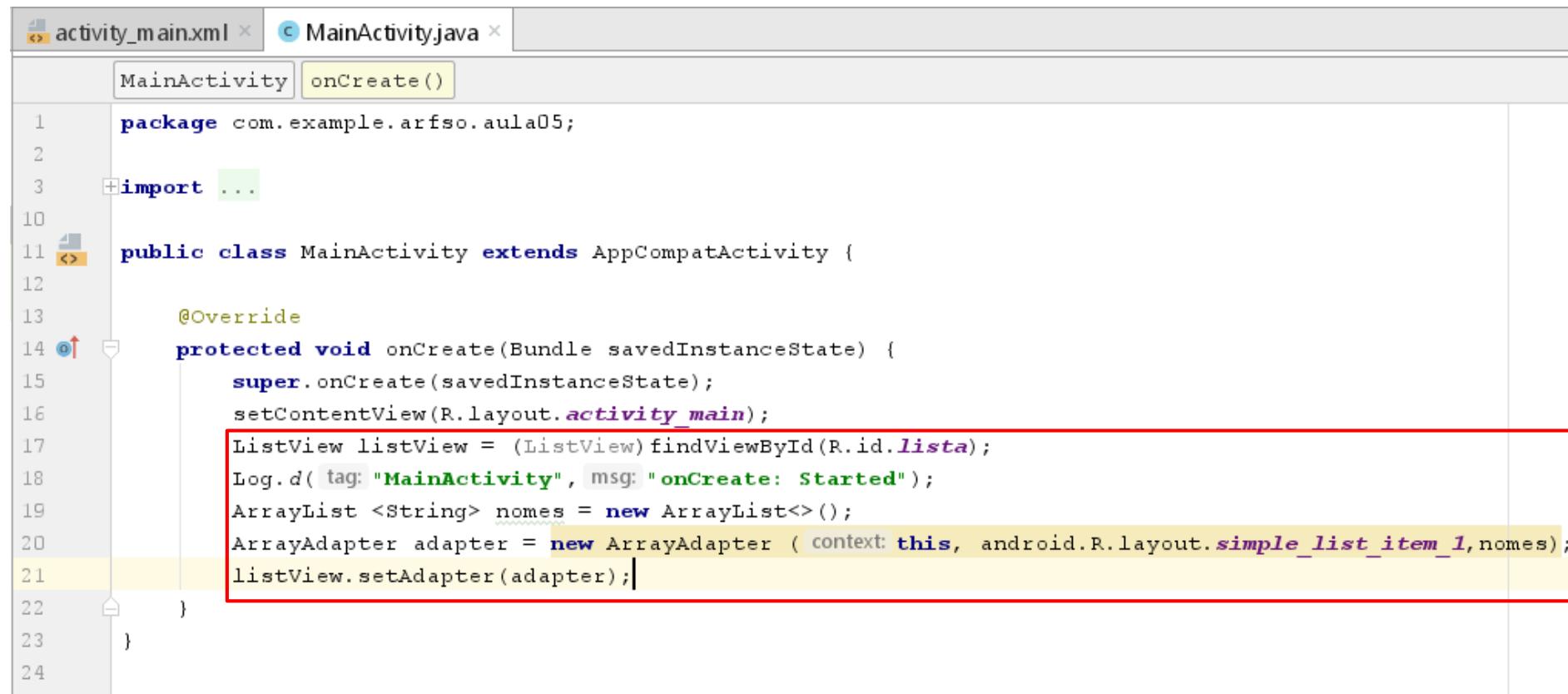
    <ListView
        android:id="@+id/lista"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true" />

</RelativeLayout>
```

The code defines a **RelativeLayout** containing a **ListView**. The **ListView** has an **id** attribute set to `@+id/lista`. The **RelativeLayout** also includes attributes for width, height, and alignment.

ListViews: Revisão

- ▶ No arquivo java principal, crie a listView através de findViewById, crie o Log.d, um arrayList e um arrayAdapter.



The screenshot shows the Android Studio interface with two tabs at the top: 'activity_main.xml' and 'MainActivity.java'. The 'MainActivity.java' tab is active, showing the Java code for the MainActivity. A red box highlights the section of code from line 17 to line 21, which is responsible for creating a ListView, setting its adapter, and displaying it.

```
activity_main.xml x MainActivity.java x
MainActivity onCreate()
1 package com.example.arfso.aula05;
2
3 import ...
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17         ListView listView = (ListView) findViewById(R.id.lista);
18         Log.d( tag: "MainActivity", msg: "onCreate: Started");
19         ArrayList <String> nomes = new ArrayList<>();
20         ArrayAdapter adapter = new ArrayAdapter ( context: this, android.R.layout.simple_list_item_1, nomes);
21         listView.setAdapter(adapter);
22     }
23 }
```

ListView: Revisão

- ▶ Não será necessário, mas caso queira adicionar nomes à ListView manualmente para eventuais testes, não há problemas.

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listView = (ListView) findViewById(R.id.lista);
        Log.d( tag: "MainActivity", msg: "onCreate: Started");
        ArrayList <String> nomes = new ArrayList<>();

        nomes.add("ALEXANDRE MARQUES DE OLIVEIRA");
        nomes.add("BRUNO PEREIRA RAMOS");
        nomes.add("CARLOS GEANINNE AQUINO SILVA");
        nomes.add("CECÍLIA DE BRITO PALHANO");
        nomes.add("DANRLEY CORREIA RAMOS");
        nomes.add("DIEGO VIANA DA SILVA");
        nomes.add("JONATAS TORRES VIEIRA");
        nomes.add("JULIO ALVES XAVIER");
        nomes.add("JÚLIO CESAR PEREIRA JÚNIOR");
        nomes.add("LUCAS DIOGO FRANÇA");
        nomes.add("LUCAS GABRIEL FERREIRA RODRIGUES");
        nomes.add("PAULO HENRIQUE TOMAZ DOS SANTOS");
        nomes.add("VINÍCIUS VIEIRA ABREU");
        nomes.add("VITOR PIMENTA MARQUES");
        nomes.add("WEVERTON ALMEIDA AMADOR");

        ArrayAdapter adapter = new ArrayAdapter ( context: this, android.R.layout.simple_list_item_1,nomes);
        listView.setAdapter(adapter);
    }
```

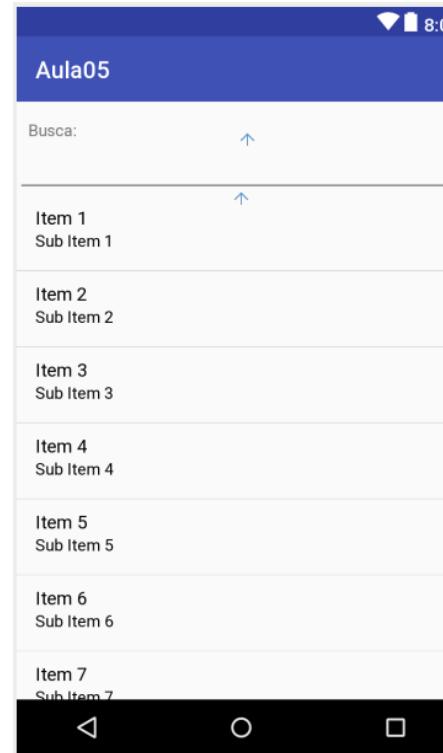
Semelhante a
aula passada

Teste a
aplicação

Filtragem de Dados

- ▶ Para trabalhar uma simples filtragem de dados nesta **ListView**, precisaremos adicionar um **EditText** ao XML, de modo que fique visível no Layout. Para este exemplo, foi colocado na parte superior do aplicativo. Adicione um id ao **EditText**.

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="15dp"  
    android:layout_marginStart="10dp"  
    android:layout_marginTop="15dp"  
    android:text="@string/busca" />  
  
<EditText  
    android:id="@+id/busca"  
    android:layout_width="match_parent"  
    android:layout_height="50dp"  
    android:layout_alignParentStart="true"  
    android:layout_below="@+id/textView"  
    android:inputType="text" />  
  
<ListView  
    android:id="@+id/lista"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_alignParentStart="true"  
    android:layout_below="@+id/busca" />
```



Filtragem de Dados

- ▶ De volta ao arquivo java principal, precisaremos declarar um objeto **EditText** e criar um link através de seu ID.



```
1 package com.example.arfso.aula05;
2
3 import ...
11
12 public class MainActivity extends AppCompatActivity {
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18         ListView listView = (ListView) findViewById(R.id.lista);
19         EditText filtro = (EditText) findViewById(R.id.busca); // Line 19 and 20 are highlighted with a red box
20         Log.d( tag: "MainActivity", msg: "onCreate: Started");
21         ArrayList <String> nomes = new ArrayList<>();
22     }
}
```

Filtragem de Dados

- ▶ Através de uma função chamada, `addTextChangedListener`, poderemos criar um filtro simplificado, adicionando a opção `new TextWatcher` em seu escopo.

```
nomes.add("WEVERTON ALMEIDA AMADOR");

ArrayAdapter adapter = new ArrayAdapter ( context: this, android.R.layout.simple_list_item_1,nomes);
listView.setAdapter(adapter);

filtro.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {

    }

    @Override
    public void afterTextChanged(Editable s) {

    }
});
```

Filtragem de Dados

- ▶ Para trabalhar com o filtro, precisaremos do ArrayAdapter como uma variável global, para isso, mudamos a forma de escrita do adapter.

```
public class MainActivity extends AppCompatActivity {

    ArrayAdapter adapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ListView listView = (ListView) findViewById(R.id.lista);
        EditText filtro = (EditText) findViewById(R.id.busca);
        Log.d(tag: "MainActivity", msg: "onCreate: Started");
        ArrayList <String> nomes = new ArrayList<>();

        nomes.add("ALEXANDRE MARQUES DE OLIVEIRA");
        nomes.add("BRUNO PEREIRA RAMOS");
        nomes.add("CARLOS GEANINNE AQUINO SILVA");
        nomes.add("CECÍLIA DE BRITO PALHANO");
        nomes.add("DANRLEY CORREIA RAMOS");
        nomes.add("DIEGO VIANA DA SILVA");
        nomes.add("JONATAS TORRES VIEIRA");
        nomes.add("JULIO ALVES XAVIER");
        nomes.add("JÚLIO CESAR PEREIRA JÚNIOR");
        nomes.add("LUCAS DIOGO FRANÇA");
        nomes.add("LUCAS GABRIEL FERREIRA RODRIGUES");
        nomes.add("PAULO HENRIQUE TOMAZ DOS SANTOS");
        nomes.add("VINÍCIUS VIEIRA ABREU");
        nomes.add("VITOR PIMENTA MARQUES");
        nomes.add("WEVERTON ALMEIDA AMADOR");

        adapter = new ArrayAdapter ( context: this, android.R.layout.simple_list_item_1,nomes);
        listView.setAdapter(adapter);
    }
}
```

Filtragem de Dados

- ▶ Na função onTextChanged, adicione um filtro através da função getFilter e passe a sequencia char descrita pelo parâmetro CharSequence.

```
adapter = new ArrayAdapter ( context: this, android.R.layout.simple_list_item_1,nomes);
listView.setAdapter(adapter);

filtro.addTextChangedListener(new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {

    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        (MainActivity.this).adapter.getFilter().filter(s);
    }

    @Override
    public void afterTextChanged(Editable s) {

    }
});
```



Teste a
aplicação

Filtragem de Dados

- ▶ Pode-se acrescentar um setOnItemClickListener na lista logo após o filtro.

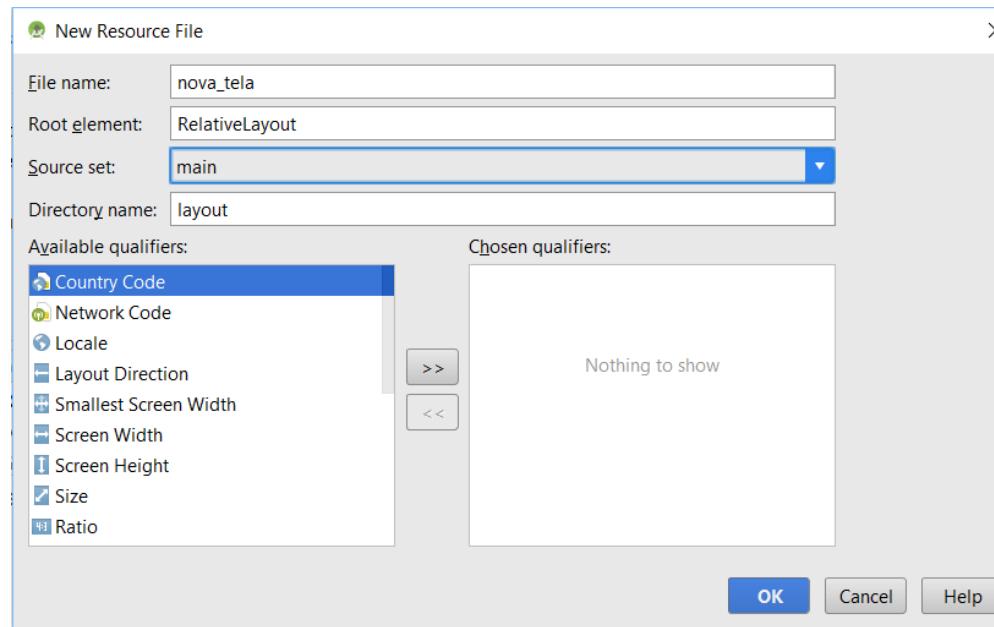
```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Log.d( tag: "MainActivity", msg: "onItemClick: name: " +nomes.get(position));
        Toast.makeText( context: MainActivity.this, text: "Clicou em: "+nomes.get(position),Toast.LENGTH_SHORT).show();
    }
});
```



Obs.: O ClickListener levará em consideração apenas a posição cadastrada dos itens da lista, logo não irá se adaptar às mudanças do filtro.

Trabalhando uma Segunda Activity

- ▶ Para passar dados de uma atividade para outro, precisaremos trabalhar com **intents** novamente.
- ▶ Inicialmente, precisaremos de um novo layout. Na pasta `res>layout`, podemos criar um novo XML `nova_tela` com um `RelativeLayout`



Trabalhando uma Segunda Activity

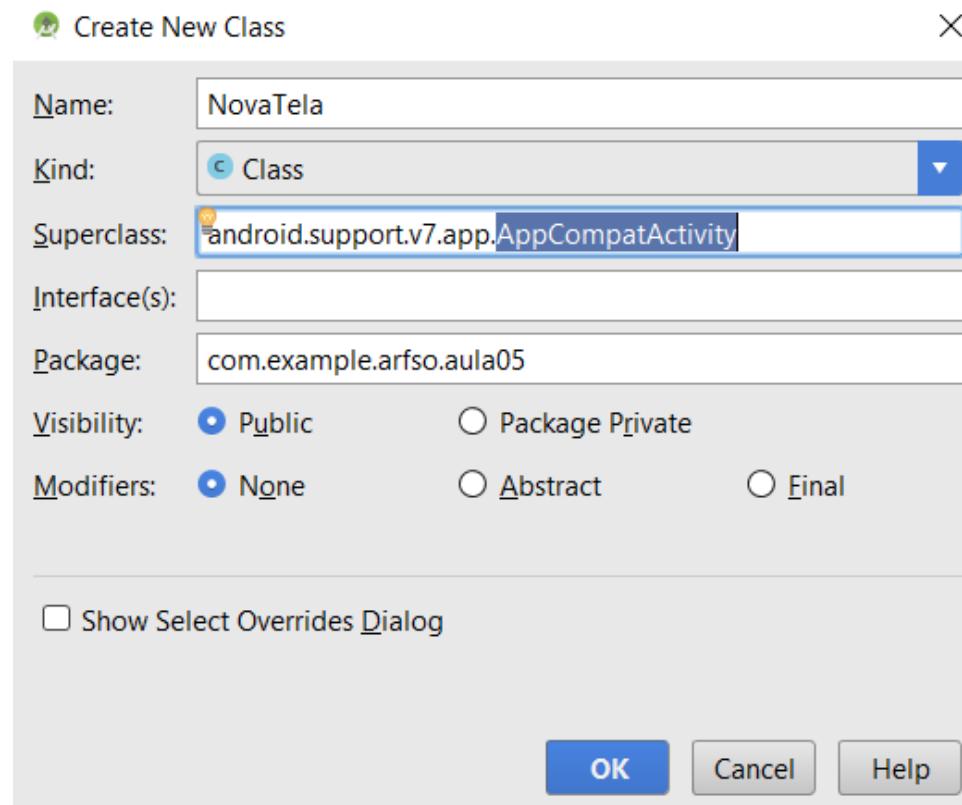
- ▶ Neste novo arquivo XML, será necessário adicionar um TextView com um id.



```
activity_main.xml × nova_tela.xml × MainActivity.java ×
RelativeLayout
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent">
5     <TextView
6         android:id="@+id/textoAlvo"
7         android:textStyle="bold"
8         android:textSize="35sp"
9         android:textAlignment="center"
10        android:textColor="@color/colorPrimaryDark"
11        android:layout_width="match_parent"
12        android:layout_height="wrap_content"
13        android:layout_alignParentTop="true"
14        android:layout_centerHorizontal="true"
15        android:layout_marginTop="80dp" />
16 </RelativeLayout>
```

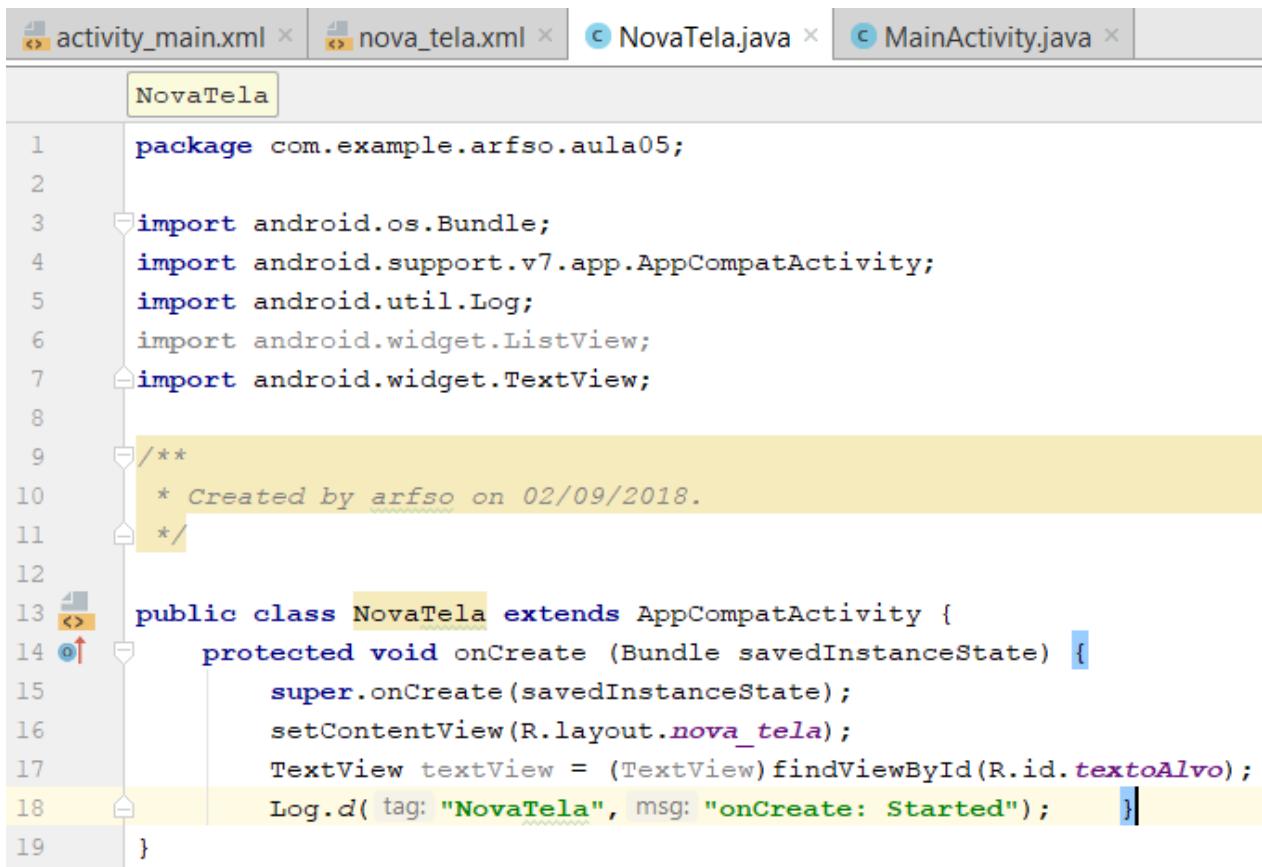
Trabalhando uma Segunda Activity

- ▶ Precisaremos criar uma nova **Activity**, NovaTela e linkar o XML criado a esta nova Atividade. Esta classe deverá herdar a **AppCompatActivity**



Trabalhando uma Segunda Activity

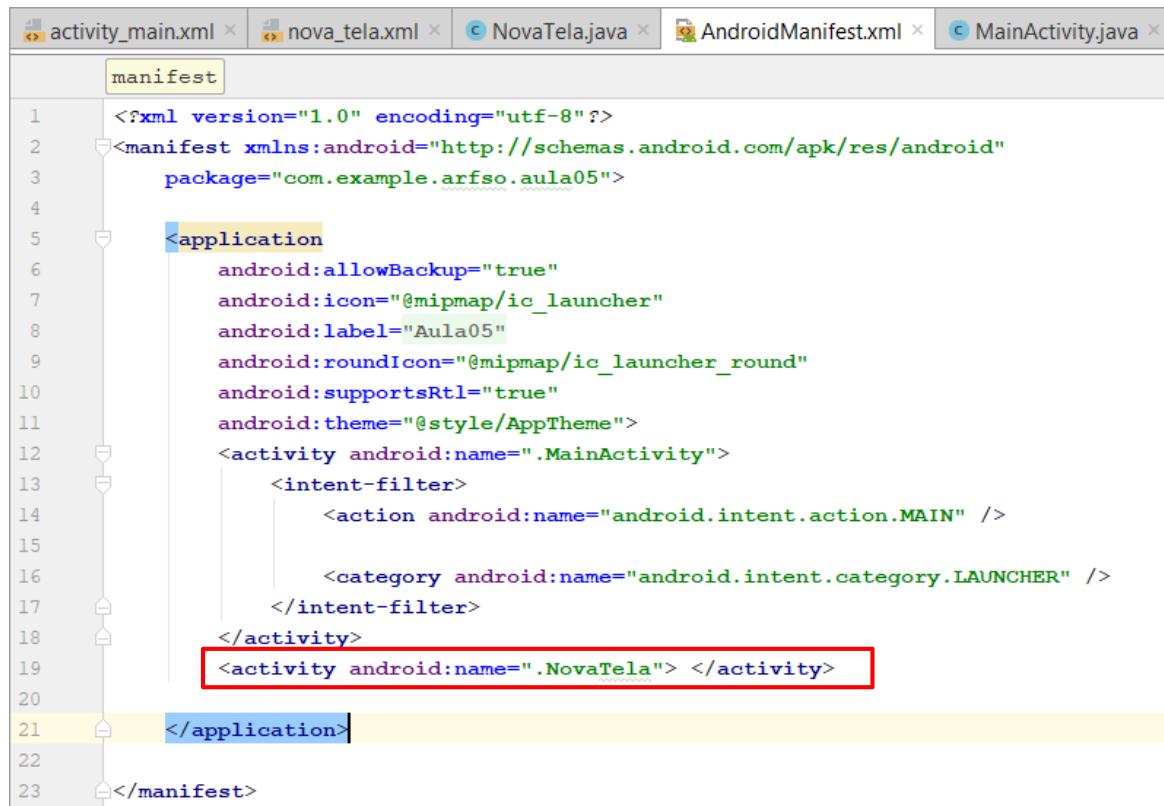
- ▶ Devemos agora acrescentar a função onCreate, setar a view, adicionar o textView e o Log.d para esta nova Atividade.



```
activity_main.xml × nova_tela.xml × NovaTela.java × MainActivity.java ×
NovaTela
1 package com.example.arfso.aula05;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.util.Log;
6 import android.widget.ListView;
7 import android.widget.TextView;
8
9 /**
10 * Created by arfso on 02/09/2018.
11 */
12
13 public class NovaTela extends AppCompatActivity {
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.nova_tela);
17         TextView textView = (TextView) findViewById(R.id.textoAlvo);
18         Log.d(tag: "NovaTela", msg: "onCreate: Started");
19     }
}
```

Trabalhando uma Segunda Activity

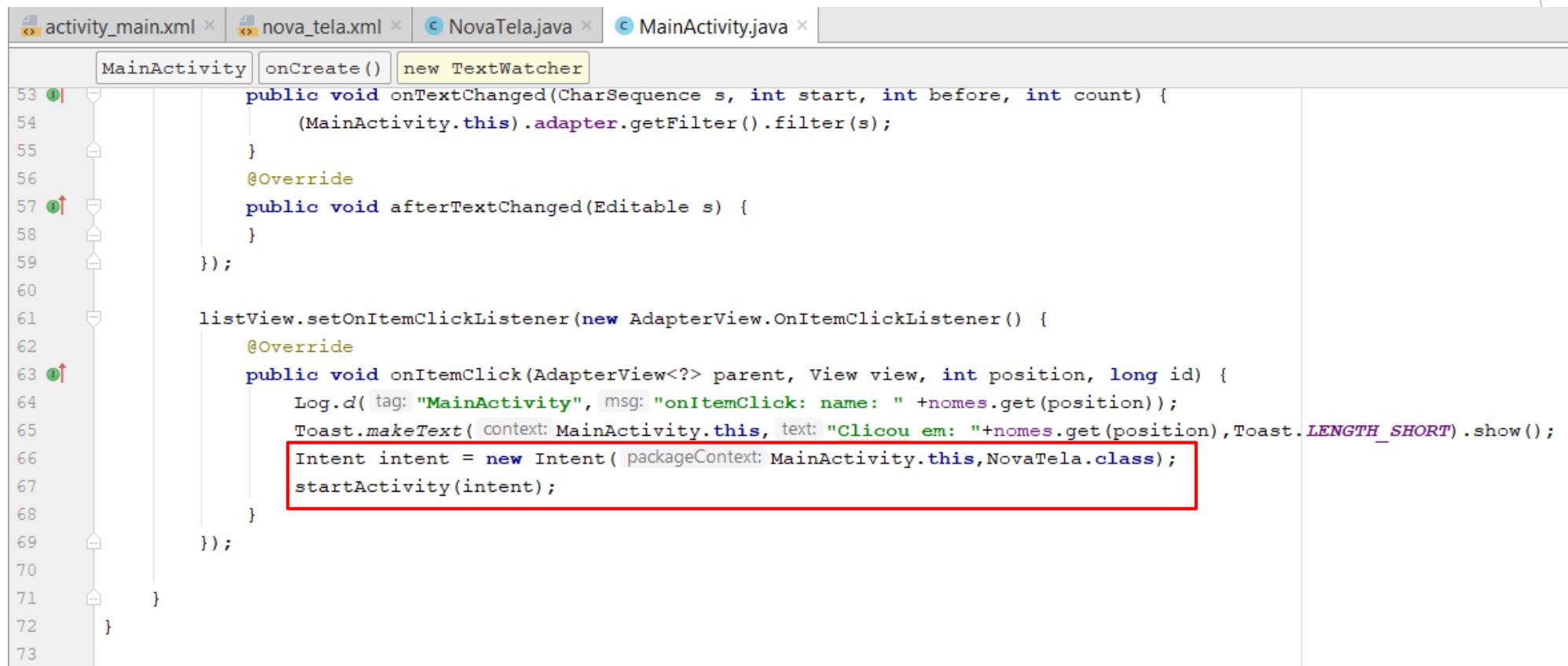
- ▶ Para que essa nova atividade funcione, precisamos registrar a nova atividade no **AndroidManifest**.



```
manifest
1 <?xml version="1.0" encoding="utf-8" ?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.arfso.aula05">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Aula05"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19         <activity android:name=".NovaTela"> </activity>
20
21     </application>
22
23 </manifest>
```

Envio de Dados Através de Intents

- ▶ Na atividade principal, podemos criar uma **intent** dentro da função **onClickListener** criada anteriormente.



```
activity_main.xml × nova_tela.xml × NovaTela.java × MainActivity.java ×
MainActivity onCreate() new TextWatcher
53     public void onTextChanged(CharSequence s, int start, int before, int count) {
54         (MainActivity.this).adapter.getFilter().filter(s);
55     }
56     @Override
57     public void afterTextChanged(Editable s) {
58     }
59 }
60
61 listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
62     @Override
63     public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
64         Log.d( tag: "MainActivity", msg: "onItemClick: name: " +nomes.get(position));
65         Toast.makeText( context: MainActivity.this, text: "Clicou em: "+nomes.get(position),Toast.LENGTH_SHORT).show();
66         Intent intent = new Intent( packageContext: MainActivity.this,NovaTela.class);
67         startActivity(intent);
68     }
69 });
70
71 }
72 }
73 }
```

The screenshot shows the Android Studio interface with the code editor open to the `MainActivity.java` file. The code is part of the `onItemClick` listener for a list view. A red box highlights the line of code where an intent is created and started:

```
Intent intent = new Intent( packageContext: MainActivity.this,NovaTela.class);
startActivity(intent);
```

Envio de Dados Através de Intents

- ▶ Para enviar dados de uma **Activity** para outra, precisaremos informar que existem dados extras a serem passados.
- ▶ Será necessário adicionar parâmetros extras com a função **putExtra** da **intent**.

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Log.d( tag: "MainActivity", msg: "onItemClick: name: " +nomes.get(position));
        Toast.makeText( context: MainActivity.this, text: "Clicou em: "+nomes.get(position),Toast.LENGTH_SHORT).show();
        Intent intent = new Intent( packageContext: MainActivity.this,NovaTela.class);
        intent.putExtra( name: "nome",nomes.get(position));
        startActivity(intent);
    }
});
```

Envio de Dados Através de Intents

- ▶ Agora na segunda atividade (NovaTela), precisaremos receber os dados recolhidos da intent. A primeira coisa será declarar um objeto intent fazendo-o receber toda a intent.

```
public class NovaTela extends AppCompatActivity {
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nova_tela);
        TextView textView = (TextView) findViewById(R.id.textoAlvo);
        Log.d( tag: "NovaTela", msg: "onCreate: Started");
        Intent intent = getIntent();
```

Envio de Dados Através de Intents

- ▶ Para receber os dados colocados pela chamada do `putExtra`, precisaremos receber os dados de alguma forma, chamando a função `get`. Como estamos trabalhando apenas com uma `String`, podemos então receber os dados através da `getStringExtra`.

```
public class NovaTela extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nova_tela);
        TextView textView = (TextView) findViewById(R.id.textoAlvo);
        Log.d(tag: "NovaTela", msg: "onCreate: Started");
        Intent intent = getIntent();
        String nome = intent.getStringExtra(name: "nome");
    }
}
```

O nome chave deverá ser o mesmo!

Envio de Dados Através de Intents

- ▶ Precisaremos agora registrar os dados recebidos em um log com o Log.d e setar o texto recebido no XML textoAlvo através do objeto textView criado anteriormente.

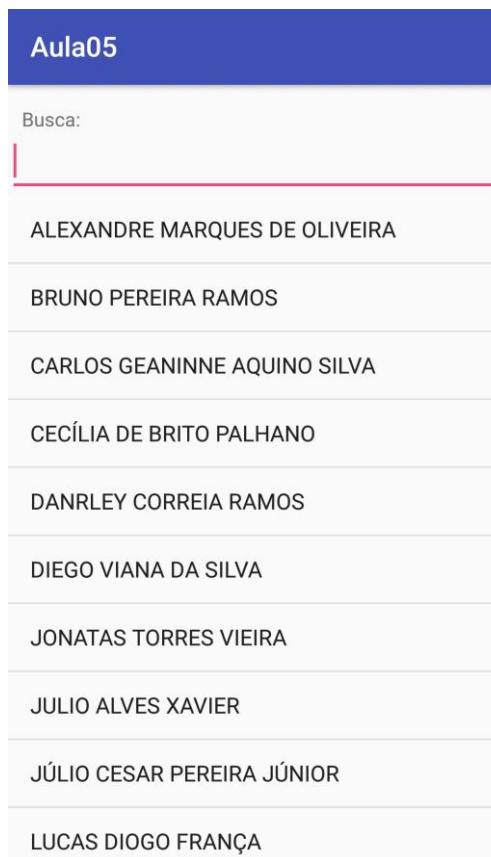
```
public class NovaTela extends AppCompatActivity {
    protected void onCreate (Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.nova_tela);
        TextView textView = (TextView) findViewById(R.id.textoAlvo);
        Log.d( tag: "NovaTela", msg: "onCreate: Started");
        Intent intent = getIntent();
        String nome = intent.getStringExtra( name: "nome");
        Log.d( tag: "NovaTela", msg: "Found incoming name: " + nome);
        textView.setText(nome);
    }
}
```



Teste a
aplicação

Envio de Dados Através de Intents

- ▶ Agora basta construir o aplicativo e verificar os resultados.



Exercícios

1. Crie uma forma de arrumar o filtro para que funcione também com o **clickListener**.
2. Crie um aplicativo Android que implemente uma forma de receber um nome e mostrá-lo em uma lista em uma nova janela.