

Linguagem e Técnica de Programação Mobile

**AULA 5 – Trabalhando com mais de
uma tela em uma aplicação**

Prof. João Paulo Pimentel
joao.pimentel@projecao.br



Roteiro da Aula



- Trabalhando com mais de uma tela em uma aplicação
 - Criando o APP Troca de Telas como exemplo
 - Executando a aplicação Troca de Telas
- Construindo nossas aplicações no Android
 - Desenvolvendo um Sistema de Cadastro





Trabalhando com mais de uma tela em uma aplicação





Trabalhando com mais de uma tela



- Até agora as aplicações que desenvolvemos tinham somente uma única tela, mas, sabemos que a maioria das aplicações possuem normalmente mais de uma tela.
 - A partir de agora iremos aprender como inserir e gerenciar várias telas em uma aplicação **Android** através dos exemplos que serão demonstrados nesta aula e nos exercícios.
-

Trabalhando com mais de uma tela

- Para começarmos, vamos criar um novo projeto Android com os seguintes dados abaixo:

Application Name: Troca de Telas

Company Domain : app.usuario

Project location : (Fica a sua escolha)

Activity Name: PrincipalActivity

Layout Name : tela_principal



Trabalhando com mais de uma tela

- Altere a estrutura de layout da sua aplicação para o **LinearLayout** e em seguida altere o componente **TextView** de acordo com a tabela abaixo:

TextView

Propriedade	Valor
text	Você está na tela principal

Trabalhando com mais de uma tela

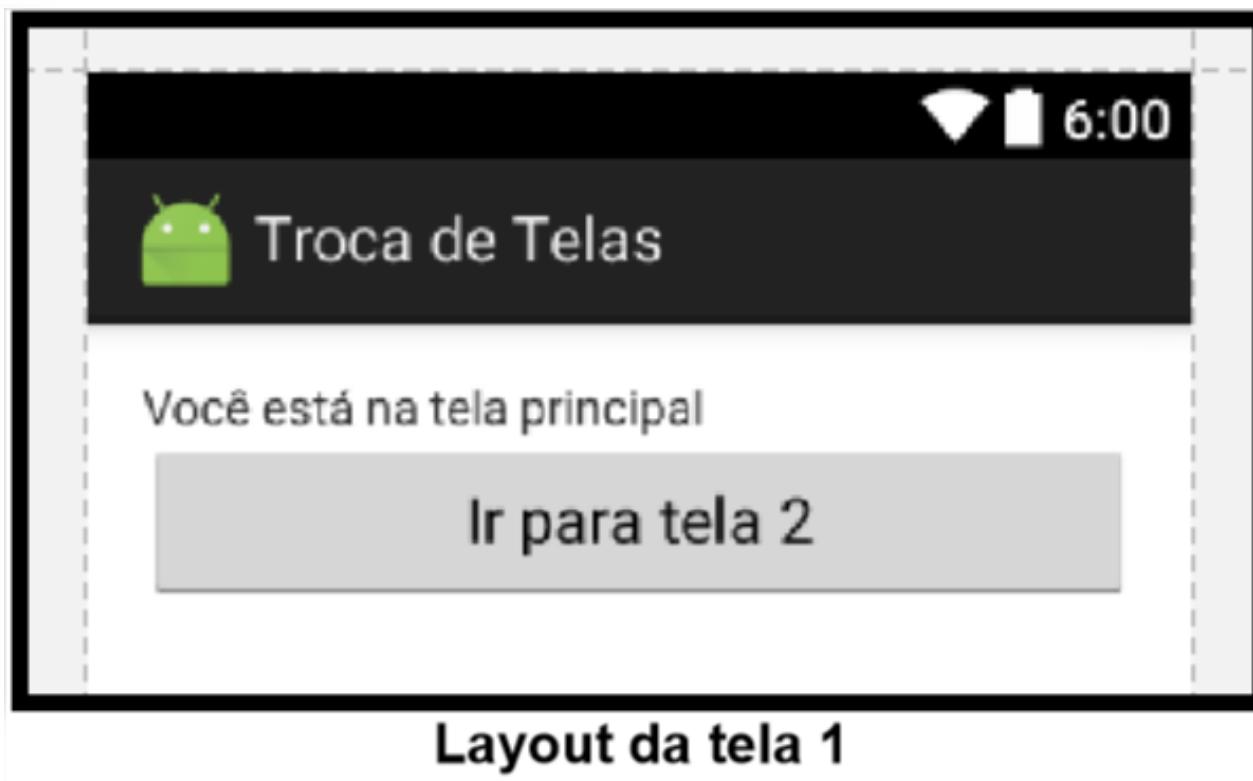
- Agora adicione um componente **Button** e modifique as seguintes propriedades:

Button

Propriedade	Valor
id	bttela2
layout:width	match_parent
text	Ir para tela 2

Trabalhando com mais de uma tela

- Seguindo os passos dos slides anteriores, a aplicação deve estar de acordo com a figura abaixo:



Trabalhando com mais de uma tela

- Vejamos agora nos próximos slides o código XML da tela da nossa aplicação:

```
<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"  
  
    tools:context=".TrocaTelasActivity">  
  
    <TextView android:text="Você está na tela principal"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:id="@+id/textView" />
```

Trabalhando com mais de uma tela

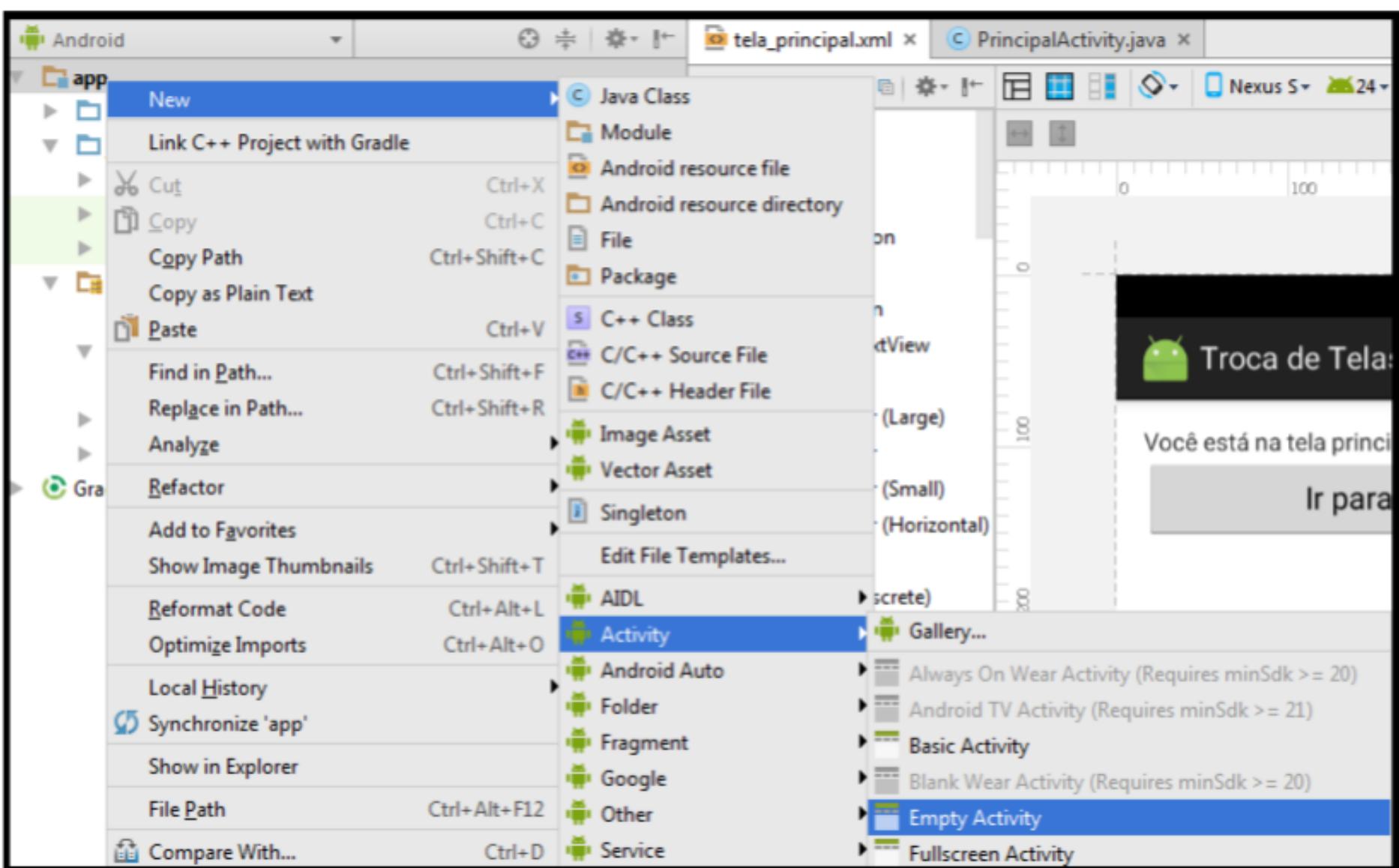
```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Ir para tela 2"  
    android:id="@+id/bttela2"  
    android:layout_below="@+id/textView"  
    android:layout_alignParentLeft="true" />  
  
</RelativeLayout>
```

Trabalhando com mais de uma tela

- Nossa primeira tela está pronta, muito bem! Agora vamos criar uma nova tela para a nossa aplicação. Essa nova tela será gerenciada por uma outra **Activity**. Para criarmos uma nova **Activity**, siga os passos que serão mostrados a seguir.
- Clique com o botão direito sobre a pasta “app” e em seguida selecione “**New**” / “**Activity**” / “**Empty Activity**” conforme mostrado na figura no próximo slide:



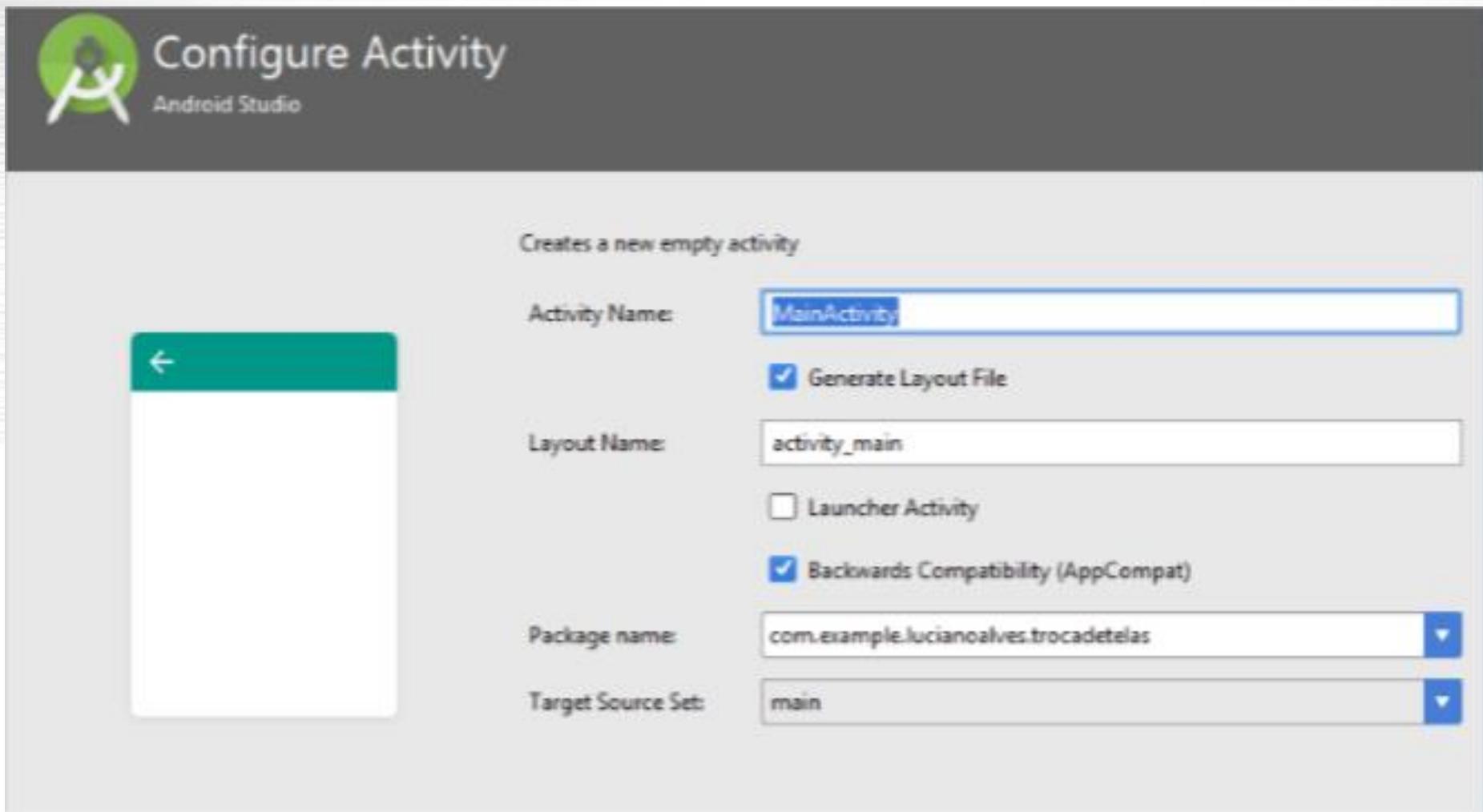
Trabalhando com mais de uma tela



Criando uma nova Activity

Trabalhando com mais de uma tela

- Feito isso surgira um formulário a seguir:



Trabalhando com mais de uma tela

- No campo “**Activity Name**” vamos informar o nome de nossa **Activity**, que irá se chamar **Tela2Activity**.
- No campo “**Layout Name**” vamos informar o nome do nosso arquivo **XML** que irá representar nova tela. Nessa campo vamos digitar “**tela2**”. Vejamos como ficou abaixo:



Trabalhando com mais de uma tela

- Vejamos agora o código XML da tela da nossa aplicação:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent" android:layout_height="match_parent"

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Você está na tela 2"
    android:id="@+id/textView2"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true" />
```

Trabalhando com mais de uma tela

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Ir para tela principal"  
    android:id="@+id/bttelaprincipal"  
    android:layout_below="@+id/textView2"  
    android:layout_alignLeft="@+id/textView2" />  
</RelativeLayout>
```

Trabalhando com mais de uma tela

- Agora vamos no arquivo “**PrincipalActivity.java**” para digitarmos o código em seguida:

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.*;
import android.view.*;

public class PrincipalActivity extends Activity {

    Button bt tela2;

    Intent iTela2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

Trabalhando com mais de uma tela

// continuação do código .java

```
setContentView(R.layout.tela_principal);
bt tela2 = (Button) findViewById(R.id.bttela2);

bt tela2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        iTela2 = new Intent(PrincipalActivity.this,
        Tela2Activity.class);
        startActivity(iTela2);
    }
});
```

Trabalhando com mais de uma tela

- O código para chamar a outra **Activity** é bastante simples. Observe que no código declaramos uma variável (objeto) chamado **iTela2** do tipo **Intent** (classe que utilizaremos para efetuar a chamada da outra tela da aplicação representada pela classe **Tela2Activity**).
- Dentro do evento **onClick** do botão, temos o seguinte trecho de código:

```
iTela2 = new Intent(PrincipalActivity.this,  
Tela2Activity.class);  
  
startActivity(iTela2);
```

Trabalhando com mais de uma tela

- Na primeira linha criamos a instância da classe **Intent**, passando como parâmetros a instância da classe atual (aplicação) e a **Activity** a ser chamada (**Tela2Activity**).
 - Na segunda linha, efetuamos a chamada da outra **Activity**, pelo método **startActitivy** (da classe **Activity**), cujo parâmetro de chamada está armazenado dentro do objeto **iTela2**.
 - Vamos abrir agora o arquivo **“Tela2Activity.java”** para digitarmos o seguinte código em seguida apresentado nos próximos slides:
-

Trabalhando com mais de uma tela

- Tela2Activity.java

```
import android.app.Activity;
import android.os.Bundle;
import android.widget.*;
import android.view.*;

public class Tela2Activity extends Activity {

    Button bttelaprincipal;

    @Override
```

Trabalhando com mais de uma tela

- //continuação do código .java

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.tela2);  
    btTELaprincipal = (Button) findViewById(R.id.bttelaprincipal);  
  
    btTELaprincipal.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
            Tela2Activity.this.finish();  
        }  
    });  
}
```

Trabalhando com mais de uma tela

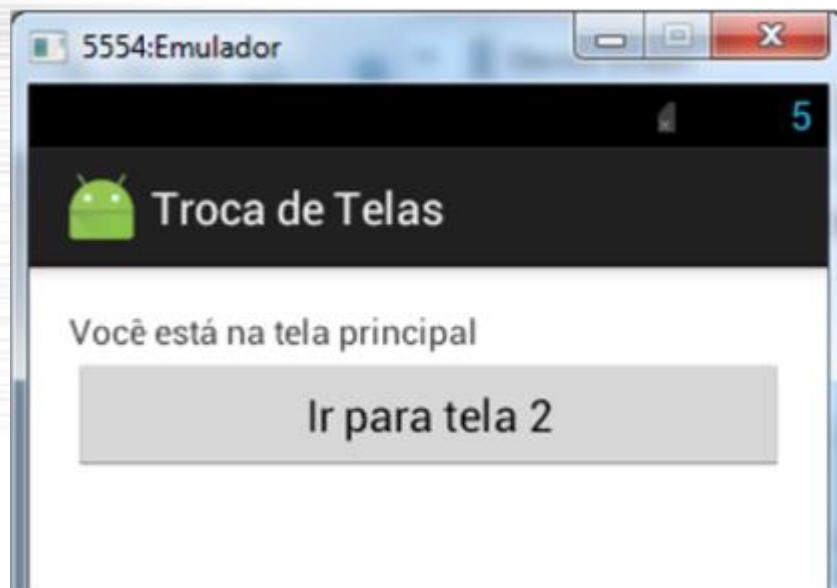
- Vamos analisar agora o código que colocamos dentro do evento do botão da segunda tela responsável por chamar a tela principal.

```
Tela2Activity.this.finish();
```

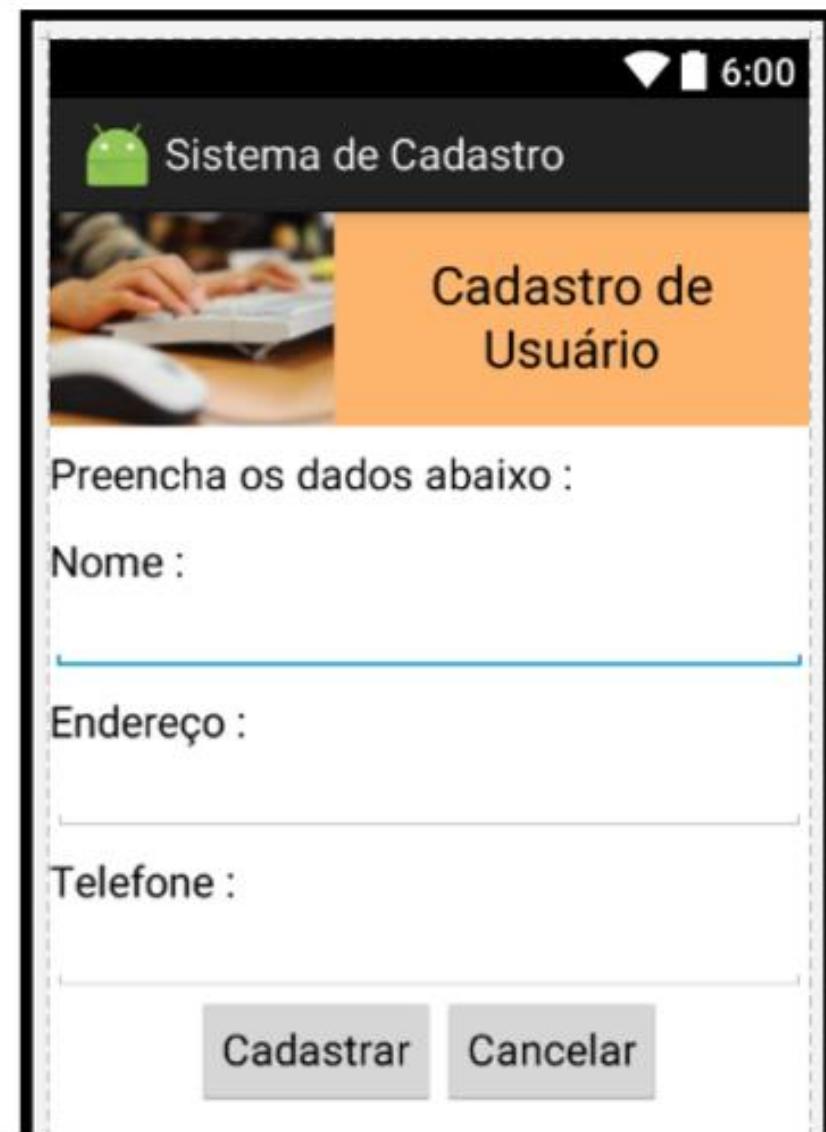
- O código acima é responsável por encerrar a execução da segunda tela, através do método **finish**. Como a tela principal já havia sido carregada anteriormente, NÃO PRECISA carrega-la novamente, basta encerrar a execução da tela atual.

Trabalhando com mais de uma tela

- Vejamos nossa aplicação em execução, conforme mostra a figura abaixo:



Desenvolvendo um Sistema de Cadastro





Desenvolvendo um Sistema de Cadastro



- Vamos agora criar um sistema de cadastro de usuários, onde através dele iremos cadastrar as seguintes informações: **nome, telefone e endereço**.
 - Essa aplicação vai possuir três telas, cada uma com as seguintes funcionalidades:
 - **Tela principal (tela com opções)**: nessa tela da aplicação teremos um menu que dará acesso ao cadastro do usuário e a visualização dos usuários cadastrados.
-



Desenvolvendo um Sistema de Cadastro



- **Tela de cadastro:** nessa tela o usuário irá preencher os campos solicitados pela aplicação e em seguida o mesmo poderá cadastrar para que as informações sejam registradas.
 - **Tela de visualização de dados:** nessa tela poderão ser visualizados os dados (**usuários**) cadastrados. Se nada foi cadastrado na aplicação, será exibida uma mensagem informando essa situação.
-

Desenvolvendo um Sistema de Cadastro

- Crie um novo projeto com os seguintes dados abaixo:

Application Name: Sistema de Cadastro

Company Domain : app.usuario

Project location : (Fica a sua escolha)

Activity Name: TelaPrincipalActivity

Layout Name : tela_principal



Desenvolvendo um Sistema de Cadastro

- Dentro da pasta “**drawable**” vamos colocar uma imagem que está disponibilizada no Blog dos Alunos chamada “**icone_programa.png**”.



**icone_programa.
png**

Desenvolvendo um Sistema de Cadastro

- Depois de colocar a imagem solicitada, abra o código XML do arquivo “**tela_principal.xml**” para adicionarmos o seguinte código XML abaixo e nos próximos slides:

```
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity" >
    <LinearLayout
        android:id="@+id/Layouttopo_tp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#fdb46d"
        android:orientation="horizontal" >
```

Desenvolvendo um Sistema de Cadastro

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/icone_programa" />  
  
<LinearLayout  
    android:id="@+id/Layout_titulo_programa_tp"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_gravity="center"  
    android:gravity="center_vertical"  
    android:orientation="vertical" >
```

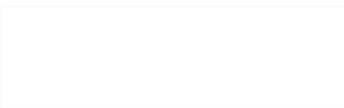
Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="Sistema de"  
    android:textAppearance="?android:attr/textAppearanceLarge" />  
  
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="Cadastro"  
    android:textAppearance="?android:attr/textAppearanceLarge" />  
  
</LinearLayout>  
  
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingBottom="15dp"  
    android:paddingTop="15dp"  
    android:text="Bem vindo ao Sistema de Cadastro. Esse é um pequeno  
    software de cadastro de usuários. Escolha uma das opções abaixo :"  
    android:textSize="18sp" />
```



Desenvolvendo um Sistema de Cadastro

```
<Button
```

```
    android:id="@+id/bt_cadastrar_usuario"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Cadastrar Usuário" />
```

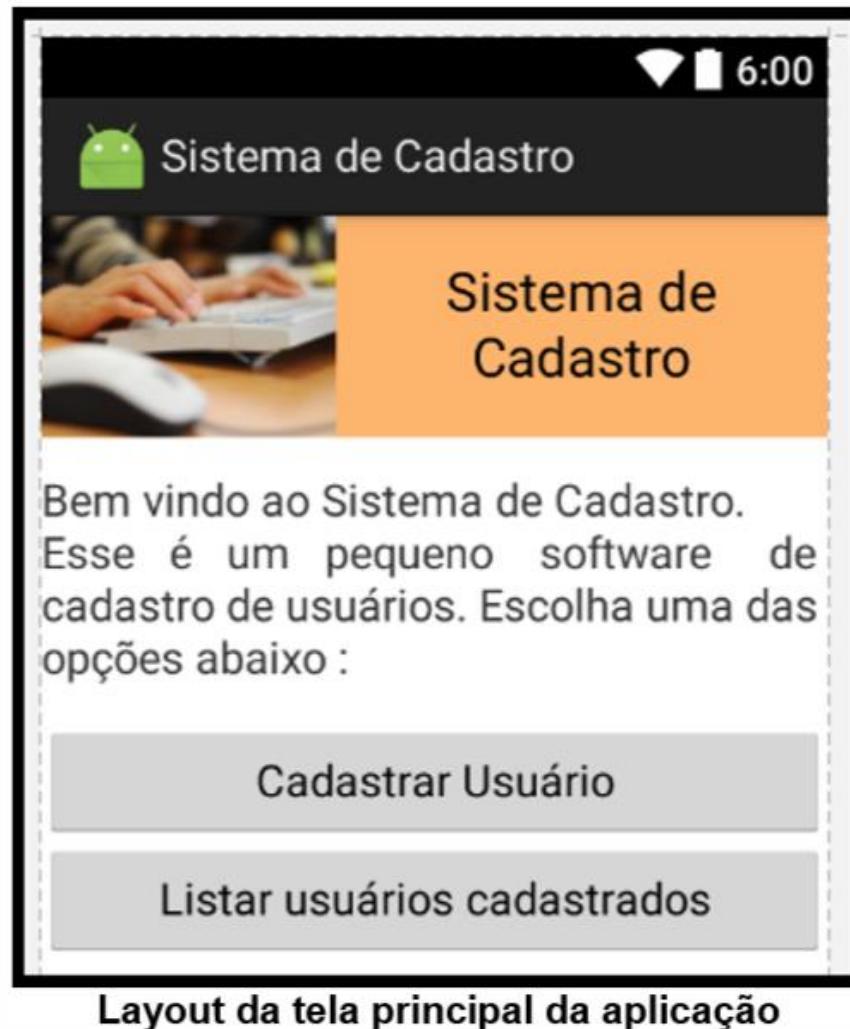
```
<Button
```

```
    android:id="@+id/bt_listar_usuarios_cadastrados"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Listar usuários cadastrados" />
```

```
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

- Depois de colocarmos o código XML solicitado, teremos o seguinte resultado abaixo :



Desenvolvendo um Sistema de Cadastro

- Agora vamos criar mais uma tela (que seja gerenciada por uma outra **Activity**). Crie uma nova **Activity**, de acordo com os seguintes dados abaixo:

Activitiy Name: CadastroActivity

Layout Name: tela_cadastro

Generate Layout File: DEIXAR MARCADO

Backwards Compatibility: DESMARCAR

Desenvolvendo um Sistema de Cadastro

- Depois de criado o arquivo vamos abrir o “**tela_cadastro.xml**” através do editor de XML para digitarmos o seguinte código referente a tela de cadastro de usuários:

```
<?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"
    android:orientation="vertical" >
    <LinearLayout
        android:id="@+id/Layouttopo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#fdb46d"
        android:orientation="horizontal" >
```

Desenvolvendo um Sistema de Cadastro

```
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/icone_programa" />  
  
<LinearLayout  
    android:id="@+id/Layout_titulo_programa"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_gravity="center"  
    android:gravity="center_vertical"  
    android:orientation="vertical" >  
  
<TextView  
  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="Cadastro de"  
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="Usuário"  
    android:textAppearance="?android:attr/textAppearanceLarge" />  
  
</LinearLayout>  
  
</LinearLayout>  
  
<TextView  
    android:id="@+id/txtnome"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="10dp"  
    android:text="Preencha os dados abaixo :"  
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="15dp"  
    android:text="Nome :"  
    android:textAppearance="?android:attr/textAppearanceMedium" />  
<EditText  
    android:id="@+id/ednome"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" >  
    <requestFocus />  
</EditText>
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:id="@+id/textView3"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="10dp"  
    android:text="Endereço :"  
    android:textAppearance="?android:attr/textAppearanceMedium" />  
<EditText  
    android:id="@+id/edendereco"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

Desenvolvendo um Sistema de Cadastro

```
<TextView
```

```
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingTop="10dp"
    android:text="Telefone :"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText
```

```
    android:id="@+id/edtelefone"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
<LinearLayout
```

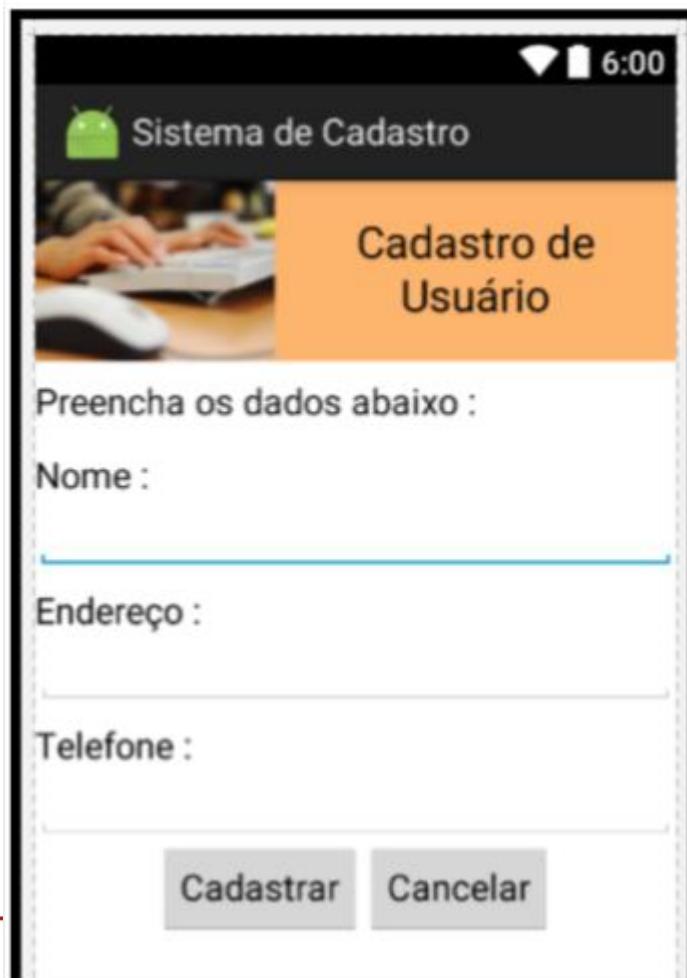
```
    android:id="@+id/Layout_botoes_cadastro"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center" >
```

Desenvolvendo um Sistema de Cadastro

```
<Button  
    android:id="@+id/btcadastrar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Cadastrar" />  
  
<Button  
    android:id="@+id/btcancelar_cadastro"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Cancelar" />  
  
</LinearLayout>  
  
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

- Depois de digitarmos o código solicitado, teremos a seguinte aparência abaixo :



Desenvolvendo um Sistema de Cadastro

- Agora vamos criar mais uma **Activity** (conforme já foi mostrado), de acordo com os seguintes dados abaixo:

Activitiy Name: ListagemActivity

Layout Name: tela_listagem

Generate Layout File: DEIXAR MARCADO

Backwards Compatibility: DESMARCAR

Desenvolvendo um Sistema de Cadastro

- Após criar a Activity, abra o arquivo “**tela_listagem.xml**” e com o editor de XML, digite o seguinte código a seguir:

```
<?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"
    android:orientation="vertical" >

    <LinearLayout
        android:id="@+id/Layouttopo_luc"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#fdb46d"
        android:orientation="horizontal" >
```

Desenvolvendo um Sistema de Cadastro

```
<ImageView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:src="@drawable/icone_programa" />
```

```
<LinearLayout
```

```
    android:id="@+id/Layout_titulo_programa_Luc"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:layout_gravity="center"  
    android:gravity="center_vertical"  
    android:orientation="vertical" >
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="Listagem de "  
    android:textAppearance="?android:attr/textAppearanceLarge" />  
  
<TextView  
    android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:text="Usuários"  
        android:textAppearance="?android:attr/textAppearanceLarge" />  
  
</LinearLayout>  
  
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

```
<LinearLayout
    android:id="@+id/Layout_status_Luc"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="right"
    android:orientation="horizontal" >

    <TextView
        android:id="@+id/txtstatus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:paddingRight="10dp"
        android:text="Registros : 1/10"
        android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nome :"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/txtnome"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingTop="5dp"
    android:text="[NOME]"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:textColor="#fb7600" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:paddingTop="15dp"
    android:text="Endereço :"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:id="@+id/txtendereco"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="5dp"  
    android:text="[ENDERECO]"  
  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:textColor="#fb7600" />  
  
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="15dp"  
    android:text="Telefone :"  
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

Desenvolvendo um Sistema de Cadastro

```
<TextView  
    android:id="@+id/txttelefone"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:paddingTop="5dp"  
    android:text="[TELEFONE]"  
    android:textAppearance="?android:attr/textAppearanceMedium"  
    android:textColor="#fb7600" />  
  
<LinearLayout  
    android:id="@+id/Layout_botoes_Luc"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:orientation="horizontal"  
    android:paddingTop="10dp" >
```

Desenvolvendo um Sistema de Cadastro

```
<Button  
    android:id="@+id/btanterior"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Anterior" />  
  
<Button  
    android:id="@+id/btproximo"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Próximo" />  
  
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

```
<LinearLayout  
    android:id="@+id/Layout_botao_fechar"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:orientation="vertical" >  
  
<Button  
    android:id="@+id/btfechar"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Fechar" />  
  
</LinearLayout>  
</LinearLayout>
```

Desenvolvendo um Sistema de Cadastro

- Depois de digitarmos o código XML de nossa tela, teremos o seguinte resultado, conforme mostra a figura a seguir:





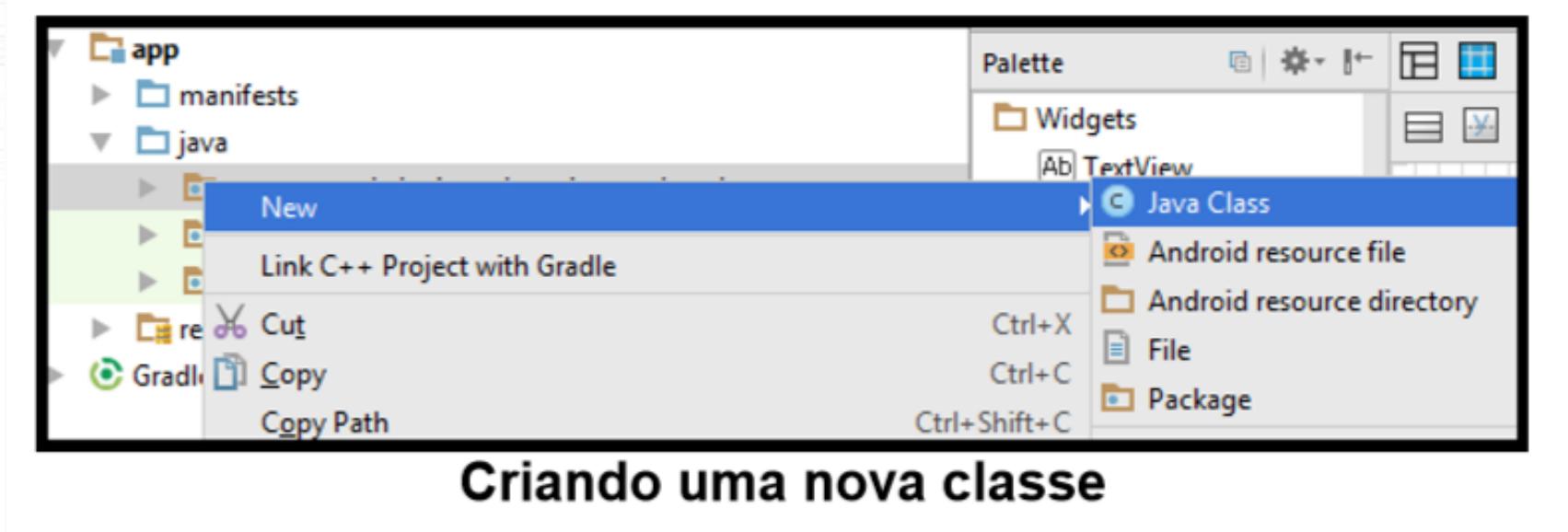
Desenvolvendo um Sistema de Cadastro



- Todos os registros dos usuários cadastrados estarão dentro de um objeto do tipo **ArrayList**. Para isso, iremos criar agora uma classe que irá armazenar (e retornar) os dados referentes ao usuário (**nome, telefone e endereço**) chamada **Registro**.
 - Essa classe vai estar dentro do mesmo pacote onde se encontra o arquivo **“TelaPrincipalActivity.java”** (neste caso, o primeiro pacote). Vamos criar agora uma classe chamada **Registro**, e em seguida vamos escrever o seu código.
-

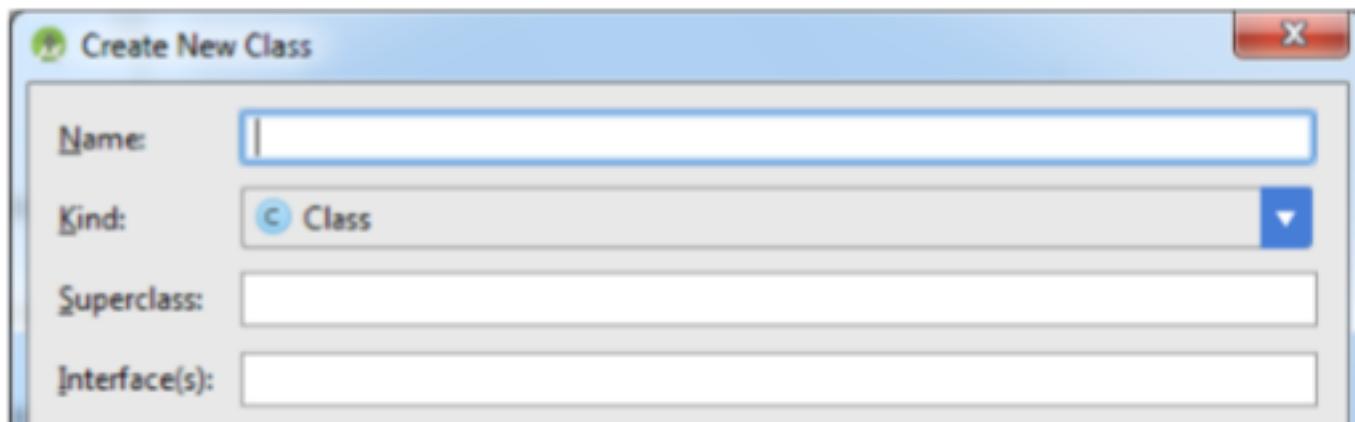
Desenvolvendo um Sistema de Cadastro

- Clique com o botão direito em cima do nome do nosso pacote e em seguida selecione “New” / “Java Class”:



Desenvolvendo um Sistema de Cadastro

- Feito isso será aberta a seguinte caixa de diálogo a seguir:



- Vamos criar uma classe chamada **Registro**, simplesmente digitando o seu nome no campo “**Name**” e em seguida clicando em “**OK**”. Feito isso, vamos definir seus métodos e atributos conforme mostrado nos próximos slides.

Desenvolvendo um Sistema de Cadastro

- Classe Registro

```
public class Registro {  
  
    private String nome;  
    private String endereco;  
    private String telefone;  
  
    public Registro(String nome, String endereco, String telefone)  
    {  
        this.nome = nome;  
        this.endereco = endereco;
```

Desenvolvendo um Sistema de Cadastro

//continuação da Classe Registro

```
        this.telefone = telefone;  
    }  
  
    public String getNome() { return nome; }  
    public String getTelefone() { return telefone; }  
    public String getEndereco() { return endereco; }  
}
```

Desenvolvendo um Sistema de Cadastro

- Agora, como já foi mostrado, vamos criar uma classe chamada **RegistrosManager**, que irá possuir o seguinte código abaixo :

```
package usuario.app.sistemadecadastro;

import java.util.ArrayList;

public class RegistrosManager {

    private static ArrayList<Registro> aRegistros = null;

    public static ArrayList<Registro> getRegistros()
    {
        if(aRegistros == null)
            aRegistros = new ArrayList<Registro>();

        return aRegistros;
    }

}
```

Desenvolvendo um Sistema de Cadastro

- Agora na classe (**Activity**) TelaPrincipalActivity vamos digitar o seguinte código abaixo :

```
package usuario.app.sistemadecadastro;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.Intent;
```



Desenvolvendo um Sistema de Cadastro

```
public class TelaPrincipalActivity extends Activity {  
  
    Button btcadastrarusuario;  
    Button btlistagemusuarios;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.tela_principal);  
        btcadastrarusuario = (Button)  
            findViewById(R.id.btcadastrar_usuario);  
        btlistagemusuarios = (Button)  
            findViewById(R.id.bt_listar_usuarios_cadastrados);  
    }  
}
```

Desenvolvendo um Sistema de Cadastro

```
btcadastrarusuario.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View view) {
        Intent intentTelaCadastro = new
        Intent(TelaPrincipalActivity.this,
        CadastroActivity.class);
        startActivity(intentTelaCadastro);
    }
});
```

Desenvolvendo um Sistema de Cadastro

```
btlistagemusuarios.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(RegistrosManager.getRegistros().size() == 0)
        {
            new AlertDialog.Builder
                (TelaPrincipalActivity.this)
                .setTitle("Aviso")
                .setMessage("Não existe nenhum registro
                cadastrado.")
                .setNeutralButton("OK", null)
                .show();
        }
        return;
    }
}
```

Desenvolvendo um Sistema de Cadastro

//continuação do método btlistagemusuarios

```
Intent intentTelaListagem = new  
Intent(TelaPrincipalActivity.this,  
ListagemActivity.class);
```

```
startActivity(intentTelaListagem);
```

```
}
```

```
});
```

```
}
```

```
}
```



Desenvolvendo um Sistema de Cadastro

- Agora dentro da classe (Activity) **CadastroActivity** vamos digitar o seguinte código a seguir :

```
package usuario.app.sistemadecadastro;

import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
```

Desenvolvendo um Sistema de Cadastro

```
public class CadastroActivity extends Activity {  
  
    EditText ednome, edendereco, edtelefone;  
    Button btcadastrar, btcancelar_cadastro;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.tela_cadastro);  
  
        ednome = (EditText) findViewById(R.id.ednome);  
        edtelefone = (EditText) findViewById(R.id.edtelefone);  
        edendereco = (EditText) findViewById(R.id.edendereco);  
        btcadastrar = (Button) findViewById(R.id.btcadastrar);  
        btcancelar_cadastro = (Button)  
            findViewById(R.id.btcancelar_cadastro);  
    }  
}
```

Desenvolvendo um Sistema de Cadastro

```
btcadastrar.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View view) {  
  
        AlertDialog.Builder dialogo = new  
        AlertDialog.Builder(CadastroActivity.this);  
        dialogo.setTitle("Aviso");  
        dialogo.setMessage("Cadastrar usuário ?");  
        dialogo.setNegativeButton("Não", null);  
        dialogo.setPositiveButton("Sim", new  
        DialogInterface.OnClickListener() {  
  
            @Override  
            public void onClick(DialogInterface dialog,  
                int which) {  
  
                String nome =  
                ednome.getText().toString();  
            }  
        });  
        dialogo.show();  
    }  
});
```

Desenvolvendo um Sistema de Cadastro

```
String telefone = edtelefone.getText().toString();
String endereco = edendereco.getText().toString();
RegistrosManager.getRegistros().add(new

Registro(nome,telefone,endereco));

(new AlertDialog.Builder(CadastroActivity.this))
.setTitle("Aviso")
.setMessage("Cadastro efetuado com sucesso")
.setNeutralButton("OK", new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {

        CadastroActivity.this.finish();
    }
})
.show();
```

Desenvolvendo um Sistema de Cadastro

```
    }

});  
dialogo.show();
}  
});  
  
btcancelar_cadastro.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View view) {  
  
        CadastroActivity.this.finish();  
  
    }
});  
}  
}
```

Desenvolvendo um Sistema de Cadastro

- Agora dentro da classe (Activity) **ListagemActivity** vamos digitar o seguinte código a seguir :

```
package usuario.app.sistemadecadastro;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.app.Activity;
```



Desenvolvendo um Sistema de Cadastro

```
public class ListagemActivity extends Activity {  
  
    Button btanterior, btproximo, btfechar;  
  
    TextView txtnome, txttelefone, txtendereco, txtstatus;  
  
    int index;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.tela_listagem);  
    }  
}
```

Desenvolvendo um Sistema de Cadastro

```
btanterior = (Button) findViewById(R.id.btanterior);
btproximo = (Button) findViewById(R.id.btproximo);
btfechar = (Button) findViewById(R.id.btfechar);

txtnome = (TextView) findViewById(R.id.txt nome);
txtendereco = (TextView) findViewById(R.id.txtendereco);
txttelefone = (TextView) findViewById(R.id.txttelefone);
txtstatus = (TextView) findViewById(R.id.txtstatus);

PreencheCampos(index);
AtualizaStatus(index);
```

Desenvolvendo um Sistema de Cadastro

```
btanterior.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View view) {  
  
        if(index > 0)  
        {  
            index--;  
            PreencheCampos(index);  
            AtualizaStatus(index);  
        }  
    }  
});
```

Desenvolvendo um Sistema de Cadastro

```
btproximo.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View view) {  
  
        if(index < RegistrosManager.getRegistros().size() - 1)  
        {  
            index++;  
            PreencheCampos(index);  
            AtualizaStatus(index);  
        }  
  
    }  
});
```

Desenvolvendo um Sistema de Cadastro

```
btfechar.setOnClickListener(new View.OnClickListener() {  
  
    @Override  
    public void onClick(View view) {  
  
        ListagemActivity.this.finish();  
    }  
});  
}  
}
```

Desenvolvendo um Sistema de Cadastro

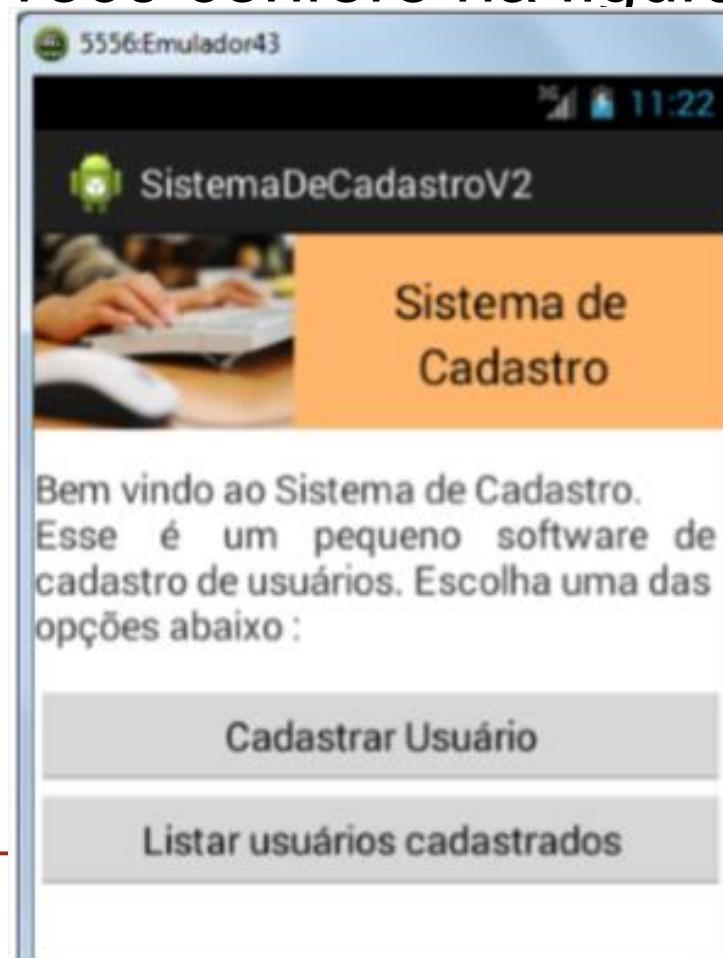
```
private void PreencheCampos(int idx)
{
    txtnome.setText( RegistrosManager.getRegistros().get(idx).getNome());
    txttelefone.setText(RegistrosManager.getRegistros()
        .get(idx).getTelefone());
    txtendereco.setText( RegistrosManager.getRegistros()
        .get(idx).getEndereco());
}

private void AtualizaStatus(int idx)
{
    int total =  RegistrosManager.getRegistros().size();
    txtstatus.setText("Registros : " + (idx+1) + "/" + total);
}

}
```

Desenvolvendo um Sistema de Cadastro

- Depois de digitar todo o código das classes citadas vamos executar a nossa aplicação. O resultado você confere na figura a seguir :



Desenvolvendo um Sistema de Cadastro

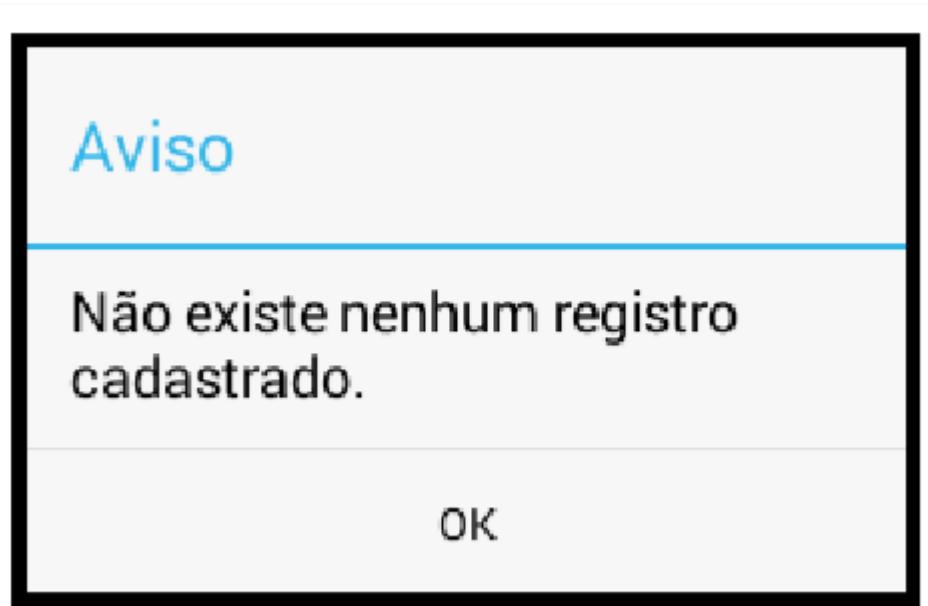
- Quando executamos a nossa aplicação, o que visualizamos é a tela principal da aplicação. Quando clicamos no botão “**Listar usuários cadastrados**”, é executado o seguinte trecho de código dentro da classe **TelaListagemUsuarios** :

```
if(RegistrosManager.getRegistros().size() == 0)
{
    (new AlertDialog.Builder(TelaPrincipalActivity.this))
        .setTitle("Aviso")
        .setMessage("Não existe nenhum registro cadastrado.")
        .setNeutralButton("OK", null)
        .show();

    return;
}
```

Desenvolvendo um Sistema de Cadastro

- Que verifica se o total de registros armazenados na aplicação é igual a “0” (ou seja, se não existe nenhum registro cadastrado). Se a condição for verdadeira (e será verdadeira), será visualizada a seguinte mensagem :



Desenvolvendo um Sistema de Cadastro

- Agora vamos clicar no botão “**Cadastrar Usuário**”, para visualizarmos a seguinte tela abaixo :

The screenshot shows a mobile application interface for user registration. At the top, there is a header bar with a red background containing a white Android icon on the left and the text "SistemaDeCadastro" on the right. Below the header, there is a small thumbnail image of a person's hands typing on a keyboard. To the right of the thumbnail, the text "Cadastro de Usuário" is displayed. The main content area has a white background and contains the following fields:

- A label "Preencha os dados abaixo :" followed by three input fields:
- A "Nome :" label with an associated text input field.
- An "Endereço :" label with an associated text input field.
- A "Telefone :" label with an associated text input field.

At the bottom of the screen, there are two buttons: "Cadastrar" on the left and "Cancelar" on the right. A horizontal red line runs across the bottom of the screen, separating the form from the footer area.

SistemaDeCadastro

Cadastro de
Usuário

Preencha os dados abaixo :

Nome :

Endereço :

Telefone :

Cadastrar Cancelar

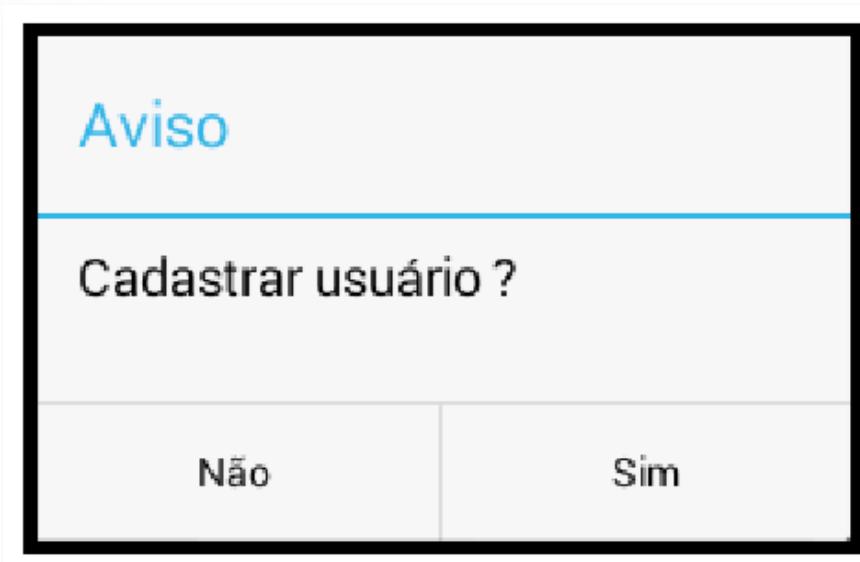
Desenvolvendo um Sistema de Cadastro

- Vamos preencher na tela de cadastro alguns dados e quando clicamos no botão “Cadastrar” é executado o seguinte trecho de código presente dentro da classe **TelaCadastradoUsuario**:

```
AlertDialog.Builder dialogo = new AlertDialog.Builder(act);
dialogo.setTitle("Aviso");
dialogo.setMessage("Cadastrar usuário ?");
dialogo.setNegativeButton("Não", null);
dialogo.setPositiveButton("Sim", new DialogInterface.OnClickListener()
{ ... });
```

Desenvolvendo um Sistema de Cadastro

- Que será responsável por exibir a seguinte mensagem abaixo:



- Perguntando se você deseja confirmar o cadastro do usuário.

Desenvolvendo um Sistema de Cadastro

- Se clicarmos no botão “Sim”, será executado o seguinte trecho de código abaixo (presente dentro do evento **OnClickListener** presente dentro do método **setPositiveButton** (responsável por mostrar o título “Sim” do botão), da classe **AlertDialog.Builder** :

```
String nome = ednome.getText().toString();
```

```
String telefone = edtelefone.getText().toString();
String endereco = edendereco.getText().toString();
```

```
RegistrosManager.getRegistros().add(new Registro(nome, telefone, endereco));
```

Desenvolvendo um Sistema de Cadastro

- As três primeiras linhas obtém dos campos os valores digitados (nome , telefone e endereço) , guardando cada valor em suas respectivas variáveis (nome, telefone e endereço).
- Na linha seguinte:

```
RegistrosManager.getRegistros().add(new Registro(nome, telefone, endereco));
```

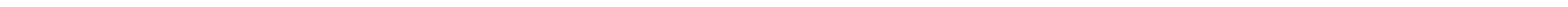
- É efetuado o registro do usuário, onde as suas informações são armazenadas dentro de uma instância da classe **Registro**.

Desenvolvendo um Sistema de Cadastro

- A instância da classe **Registro** após registrar as informações passadas como parâmetro no método construtor, a mesma é armazenada dentro do array de registros (retornado pelo método **getRegistros**), gerenciado pela classe **RegistrosManager**.
 - Após o registro é mostrado uma mensagem informando que o registro foi feito com sucesso e em seguida carregada a tela principal do programa.
-

Desenvolvendo um Sistema de Cadastro

- Na tela principal do programa, vamos clicar no botão “**Listar usuários cadastrados**” que serão mostrados todos os usuários cadastrados no nosso APP.





Por hoje é só!!!!

Até a próxima aula...

