Desenvolvimento para Dispositivos Móveis

Prof. MSc. Alan Souza

alan.souza@unama.br

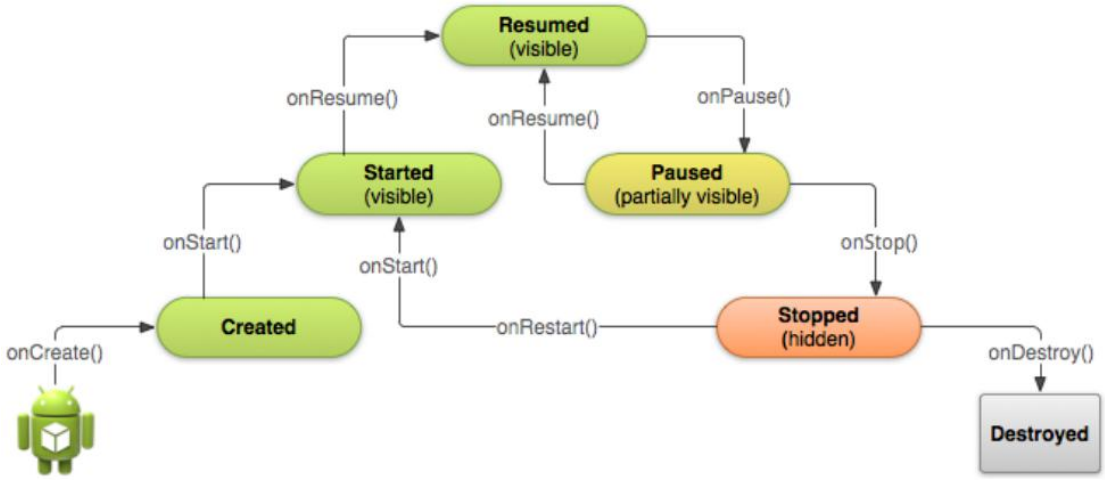
2019

2. ACTIVITY

- Corresponde a uma tela do app;
- Cada Activity é composta por um arquivo XML e outro JAVA.
  - XML para o layout (*drag and drop / arraste e solte*)
  - JAVA para a programação da tela (dinamismo)
- Classes que estendem Activity interagem tanto com usuários como com serviços ou intenções;
- O método **onCreate** é o primeiro a ser chamado quando uma Activity é executada.

2. ACTIVITY

Ciclo de Vida de uma Activity:



2. ACTIVITY

Ciclo de Vida de uma Activity (métodos Java):

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // A activity está sendo criada
}
```

2. ACTIVITY

Arquivo XML:

- Contém as informações estáticas da tela (layout);
- Estrutura os componentes de entrada e saída de dados;
- Cores, tamanhos, estilos e outros fatores são definidos no XML;

Arquivo Java:

- Onde se programa a parte dinâmica da tela;
- Realiza a interação com o usuário;
- Por exemplo: quando o usuário toca em um botão, um método no arquivo Java é chamado para executar operações lógicas, matemáticas, banco de dados, etc;

2. ACTIVITY

Arquivo strings.xml:

- Arquivo criado automaticamente quando se inicia o projeto;
- Local onde se declara o conteúdo textual das telas;
- Importante por causa da internacionalização do app:
  - Tradução para outros idiomas.
- É possível usar o mesmo texto em várias telas do app:
  - Reutilização de conteúdo.
- Organização do projeto:
  - Não misturar conteúdo textual com código-fonte da programação.

2. ACTIVITY

Classe R:

•Liga o arquivo XML ao arquivo JAVA e vice-versa;

•A ligação é feita através de identificadores únicos, que são definidos pelo programador na hora do desenvolvimento do app [arquivo XML / android:id];

•No Java, deve-se utilizar o método findViewById( ) para realizar a integração.

•Essa classe é criada automaticamente pelo Android Studio;

Esquema resumido dos arquivos de duas telas (activities)

Classe R:

Liga o Java e o XML!!!

Exemplos: **R.id.txtNome / findViewById(R.id.txtNome)**

Criada automaticamente pelo Android Studio

Desenvolvimento para Dispositivos Móveis

Prof. MSc. Alan Souza

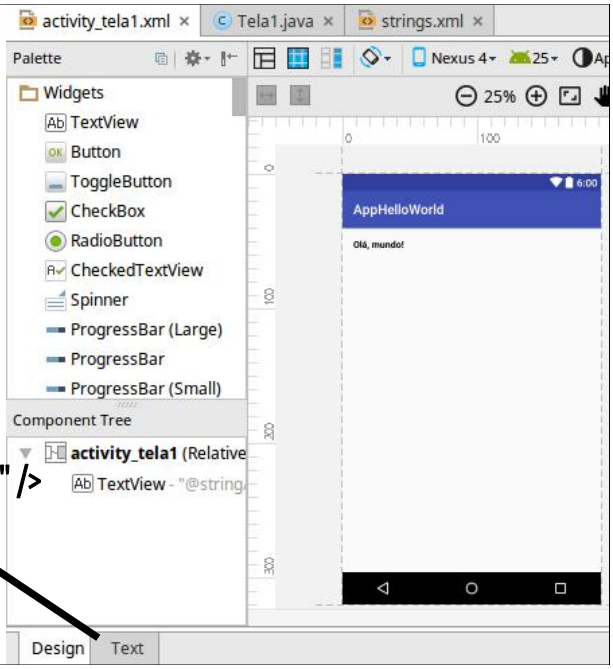
alan.souza@unama.br

2019

1º app didático – Hello World

XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout  ...  >
  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/msg_ola_mundo" />
</LinearLayout>
```



1º app didático – Hello World – JAVA

```
public class Tela1 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView( R.layout.activity_tela1 );
    }

}
```

1º app didático – Hello World – strings.xml

```
<resources>
    <string name="app_name">AppHelloWorld</string>
    <string name="msg_ola_mundo">Olá, mundo!</string>
</resources>
```

Arquivo XML de layout

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout   ...   >
    <TextView   ...
        android:text="@string/msg_ola_mundo" />
</RelativeLayout>
```

Desenvolvimento para Dispositivos Móveis

Prof. MSc. Alan Souza

alan.souza@unama.br

2019

2º app didático

Efeito dinâmico:
Requer programação Java!!!

2º app didático

Novas funcionalidades:

1. Tipo de layout: **LinearLayout** (android:orientation="vertical");
2. Incluir um texto simples "Valor da Conta";
3. Incluir um campo de texto (*widget* EditText);
4. Incluir um botão (*widget* Button);
5. Incluir um outro texto simples vazio ("") para mostrar o resultado da conta quando o botão for tocado;

O atributo **android:id** é importante para posterior referência na classe Java.

2º app didático

4. É importante criar as strings que serão utilizadas como títulos, rótulos de botão e saudações -> Arquivo **app/res/values/strings.xml**

```
<resources>
    <string name="txt_conta">Valor da conta:</string>
    <string name="bt_calculo">Calcular com 10%</string>
    <string name="txt_resultado">Conta com 10%:</string>
</resources>
```

2º app didático

5. Na classe **MainActivity**, criar o método para responder ao evento do ***onClick*** do botão chamado **surpreenderUsuario( )**

```
public class MainActivity extends Activity {
    private EditText campoConta;
    private TextView txtResultado;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void calcTotal(View v) { }
}
```

2º app didático

6. Instanciar as referências para os objetos dos componentes do layout no método **onCreate( )** através do método **findViewById( )**. Exemplo:

```
campoConta = (EditText) findViewById(R.id.edit_conta);
```

7. Implementar o método **calcCom10Porcento( ) ->** pegar o que foi digitado no campo de texto, transformar para double e calcular os 10% e concatenar com a mensagem de resultado (strings.xml):

```
String contaString = campoConta.getText( ).toString( );
double valorConta = Double.parseDouble( contaString );
double resultado = valorConta * 1.1;
String resultadoTxt = getResources( ).getString(R.string.txt_resultado);
String msg = resultadoTxt + " " + resultado;
txtResultado.setText( msg );
```

8. Executar e testar a aplicação.