

Ficha do Aplicativo

App 02 – Aplicativo Registro de Mensagens

Resumo

Neste aplicativo será possível.

Objetivos de Aprendizagem

1. Segundo aplicativo a ser desenvolvido. Repetição dos passos iniciais para criação do aplicativo consolidando a compreensão dos parâmetros envolvidos em sua criação.
2. Praticar os elementos essenciais de uma Activity:
 - a. Layout, que é o desenho da tela do aplicativo
 - b. Código Java, que garante a funcionalidade do aplicativo
3. Praticar a configuração de Layout, explorando novos elementos.
4. Voltar a usar os primeiros componentes TextView e Button.
5. Usar dois novos componentes:
 - a. EditText
 - b. Guideline
6. Implementar e testar o código necessário ao funcionamento do clique no botão.
7. Usar a Classe ArrayList, já vista nas aulas de Java básico, no Android.
8. Tornar um TextView dotado da capacidade de rolagem de texto (scrolling).

Dinâmica do Aplicativo

Neste aplicativo os elementos ativos serão: uma caixa de texto editável (EditText), um botão (Button) e uma caixa de texto (TextView).

O usuário do aplicativo irá digitar texto no EditText e, ao clicar em um botão, o texto digitado será transferido para um ArrayList a partir do qual um componente TextView será preenchido com múltiplas linhas. Cada texto digitado será inserido no ArrayList e escreveremos um método para carregar o TextView com os Strings carregados nesse ArrayList. Este TextView será configurado para permitir a rolagem do texto (scrolling).

Os componentes principais relativos ao funcionamento do aplicativo serão declarados na classe MainActivity, construídos no método onCreate e utilizados onde for necessário dentro da classe.

Lista de Activities do Aplicativo

Nome do arquivo de código	Nome do arquivo de Layout
MainActivity.java	activity_main.xml

Resources

strings.xml

```
<resources>
  <string name="app_name">Registro de Mensagens</string>
  <string name="lbl1">Digite uma mensagem</string>
  <string name="lblRegistra">Registra a mensagem</string>
</resources>
```

colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="purple_200">#FFB86FC</color>
  <color name="purple_500">#FF620EE</color>
  <color name="purple_700">#FF370B3</color>
  <color name="teal_200">#FF03DAC5</color>
  <color name="teal_700">#FF018786</color>
  <color name="black">#FF000000</color>
  <color name="white">#FFFFFFFF</color>
  <color name="corFundo">#CEF9FF</color>
  <color name="corFrente">#FBFBFB</color>
  <color name="corLbl">#FDE393</color>
</resources>
```

Detalhamento das Activities

Layout: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas
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:background="@color/corFundo"
tools:context=".MainActivity">

  <TextView
    android:id="@+id/txtRegistros"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="8dp"
    android:background="@color/corFrente"
    android:scrollbarStyle="outsideInset"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toStartOf="@+id/guidelineDir"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/guidelineEsq"
    app:layout_constraintTop_toBottomOf="@+id/btnRegistra"
    app:layout_constraintVertical_bias="1.0" />

  <androidx.constraintlayout.widget.Guideline
    android:id="@+id/guidelineEsq"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.05" />

  <androidx.constraintlayout.widget.Guideline
    android:id="@+id/guidelineDir"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    app:layout_constraintGuide_percent="0.95" />

  <TextView
    android:id="@+id/textView"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@color/corLbl"
    android:gravity="center"
    android:text="@string/lbl1"
    android:textSize="18sp"
```



```

app:layout_constraintEnd_toStartOf="@+id/guidelineDir"
app:layout_constraintStart_toStartOf="@+id/guidelineEsq"
app:layout_constraintTop_toTopOf="parent" />

<EditText
    android:id="@+id/edtMensagem"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="@color/colorPrimary"
    android:ems="10"
    android:hint="Mensagem"
    android:inputType="text"
    app:layout_constraintEnd_toStartOf="@+id/guidelineDir"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="@+id/guidelineEsq"
    app:layout_constraintTop_toBottomOf="@+id/textView" />

<Button
    android:id="@+id/btnRegistra"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:onClick="btnRegistraOnClick"
    android:text="@string/lblRegistra"
    app:layout_constraintEnd_toStartOf="@+id/guidelineDir"
    app:layout_constraintStart_toStartOf="@+id/guidelineEsq"
    app:layout_constraintTop_toBottomOf="@+id/edtMensagem" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

Código: MainActivity.java

```

public class MainActivity extends AppCompatActivity {
    private ArrayList<String> strRegistro;
    private EditText edtMensagem;
    private TextView txtRegistros;

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

        strRegistro = new ArrayList<>();
        edtMensagem = findViewById(R.id.edtMensagem);
        txtRegistros = findViewById(R.id.txtRegistros);

        txtRegistros.setMovementMethod(new ScrollingMovementMethod());
    }

    public void btnRegistraOnClick(View view) {
        String msg;
        msg = edtMensagem.getText().toString();
        // if(msg != null && !msg.trim().isEmpty()) {
        if(msg != null && !msg.isEmpty()) {
            strRegistro.add(msg);
            edtMensagem.setText("");
        }
        exibeRegistro();
    }

    private void exibeRegistro() {
        String s = "";
        for(int i = 0; i < strRegistro.size(); i++) {
            if (i < strRegistro.size()-1)
                s = s + (i+1) + ": " + strRegistro.get(i).toString() + "\n";
            else
                s = s + (i+1) + ": " + strRegistro.get(i).toString();
        }
        txtRegistros.setText(s);
    }
}

```

Notas Técnicas

Guidelines

Guideline é um componente invisível durante a execução do App e ele existe para ser um auxiliar no alinhamento de Views baseadas em ConstraintLayout. É importante ressaltar que guidelines só funcionam se estiverem contidos em um ConstraintLayout. Desse modo dizemos que um guideline está contido em um ConstraintLayout e que este último é o seu container ou parent.

São possíveis dois tipos de guidelines: verticais que terão largura zero e a altura do container; e horizontais cuja altura será zero e a largura será do container.

A posição do guideline vertical será especificada de três possíveis formas:

- Distância absoluta em relação ao lado esquerdo do container;
- Distância absoluta em relação ao lado direito do container;
- Distância relativa (%) em relação à largura de seu container;

A posição do guideline horizontal será especificada de três possíveis formas:

- Distância absoluta em relação ao topo do container;
- Distância absoluta em relação da base do container;
- Distância relativa (%) em relação à largura de seu container;

As distâncias absolutas mencionadas acima serão medidas na unidade density pixel (DP).

Uma vez posicionado um guideline, pode-se vincular a ele outros componentes visuais – View – do layout através da criação de constraint que parte da View e chega ao guideline. Caso o programador tenha interesse em alterar o id de um guideline, convém fazer isso antes de vincular outras views ao mesmo.

Propriedades privadas criadas na classe MainActivity

No código da página 3 pode-se ver que foram declaradas três propriedades (objetos) privadas na classe MainActivity: strRegistro, edtMensagem e txtRegistros. Além disso, verifique que essas propriedades foram construídas dentro do método onCreate().

Este método e sua importância serão detalhados e explicados convenientemente em um aplicativo futuro. Aqui é preciso saber apenas que esse método é executado uma única vez, assim que a Activity é criada. Desse modo trata-se de local adequado para a inicialização de objetos como esses três, que passarão a existir a partir de sua construção e ficarão disponíveis para serem usados em todos os métodos da classe.

Verificando o código na página 3 percebe-se que eles são usados dentro dos dois outros métodos: btnRegistraOnClick() e exibeRegistro(), sem a necessidade de serem contruídos.

Rolagem de texto em um TextView

Neste aplicativo queremos que o TextView txtRegistros seja dotado da capacidade de rolagem de texto. Essa rolagem acontecerá quando a quantidade de texto dentro do componente ocupar uma altura maior que sua área visível. Isso é especialmente importante quando o aparelho estiver virado na horizontal e a caixa de texto tiver uma altura consideravelmente menor. Para conseguir esse efeito é preciso tomar duas providências:

- Configurar a propriedade scrollbarStyle do TextView no layout; e
- Configurar o método de movimento do TextView: isso é feito no código Java através do uso do setter setMovementMethod(), ao qual devemos passar um parâmetro da classe ScrollingMovementMethod. Essa é uma tarefa comum para um TextView, portanto não é necessário que precisemos implementar elementos específicos dessa classe. Basta instanciar um objeto anônimo na própria chamada do setter. Por isso usamos a construção mostrada na linha a seguir

```
txtRegistros.setMovementMethod(new ScrollingMovementMethod());
```

na qual usamos new ScrollingMovementMethod para instanciar um objeto ao qual não demos um nome (por isso objeto anônimo – algo muito comum em Java) e imediatamente passá-lo como parâmetro ao setter setMovementMethod().