

DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS

Parte III

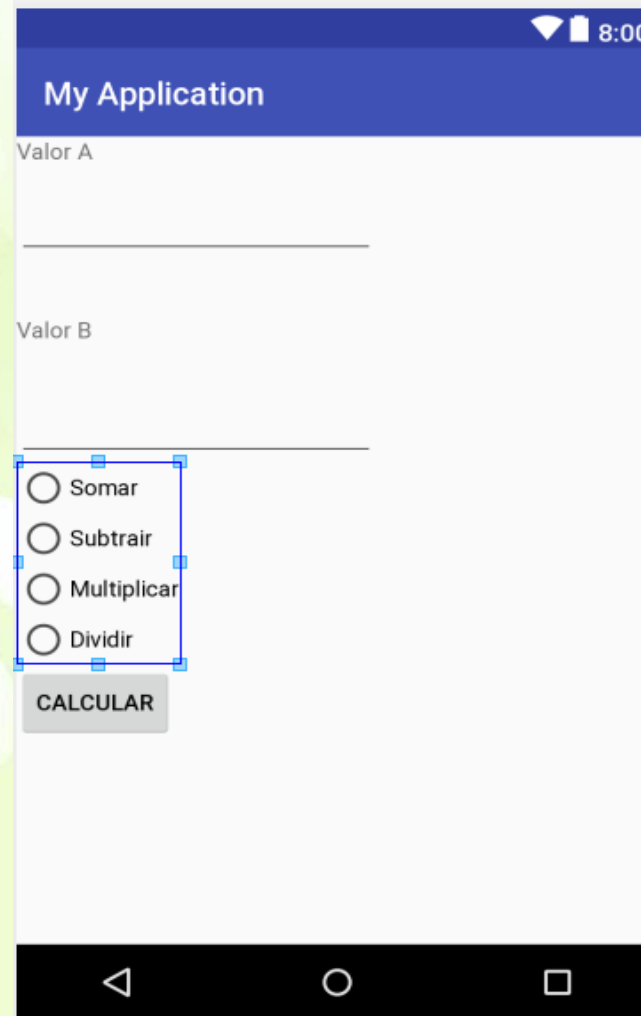
PROF. MSC. FELIPE DINIZ DALLILO



RadioGroup e RadioButton

Para inserir uma opção onde o usuário pode selecionar um registro, é utilizado o RadioButton.

Caso seja decidido agrupar entre opções (conjunto de RadioButtons) e ao selecionar uma destas, as demais não podem estar selecionadas, é utilizado o componente RadioGroup.



```
<RadioGroup
    android:id="@+id/radioGroup"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="false"
    android:layout_below="@+id/txtValorB">

    <RadioButton
        android:id="@+id/rbSomar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Somar"
        tools:layout_editor_absoluteX="16dp" />

    <RadioButton
        android:id="@+id/rbSubtrair"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Subtrair"
        tools:layout_editor_absoluteX="16dp" />

    <RadioButton
        android:id="@+id/rbMultiplicar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Multiplicar"
        tools:layout_editor_absoluteX="16dp" />

    <RadioButton
        android:id="@+id/rbDividir"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Dividir"
        tools:layout_editor_absoluteX="16dp" />

</RadioGroup>
```

RadioButton

O Código abaixo possui a finalidade de coletar o valor do componente e avaliar se o mesmo está selecionado:

```
RadioButton rbSomar = (RadioButton) findViewById(R.id.rbSomar);  
if(rbSomar.isChecked()) {  
  
}  
}
```

Calculadora com RadioButton

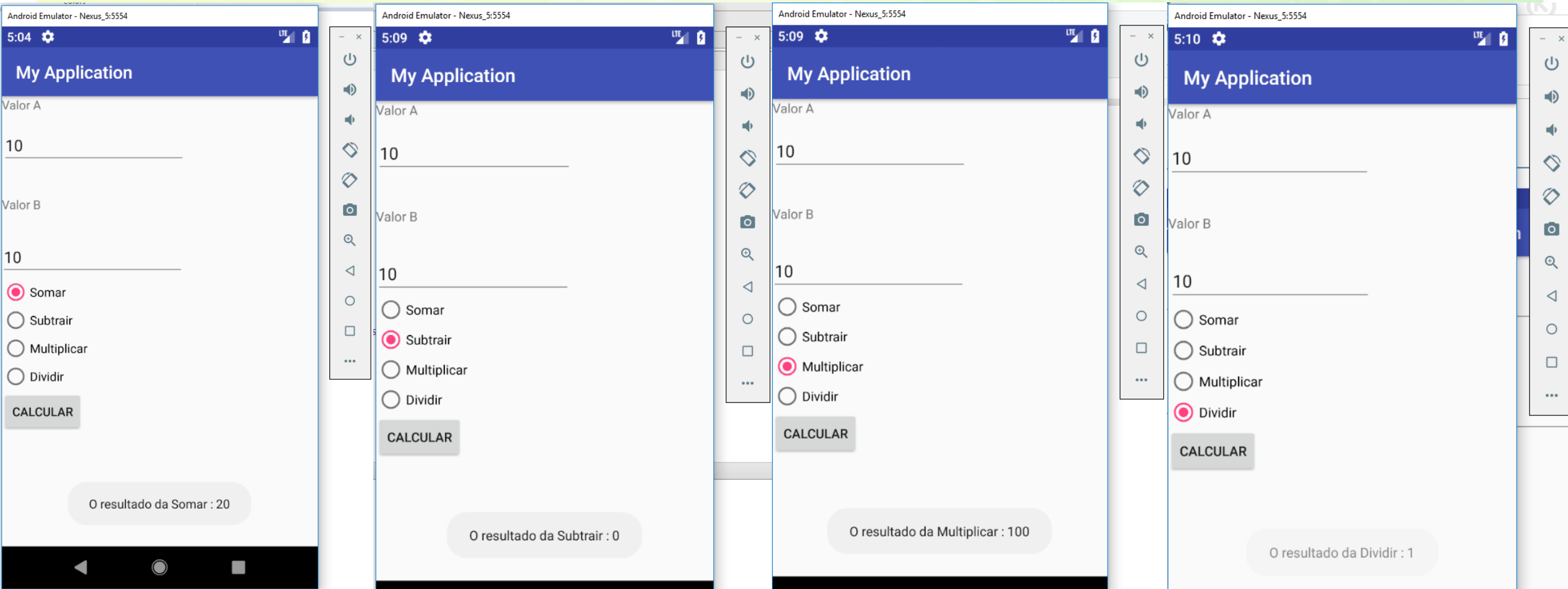
```
RadioButton rbSomar = (RadioButton) findViewById(R.id.rbSomar);
RadioButton rbSubtrair = (RadioButton) findViewById(R.id.rbSubtrair);
RadioButton rbMultiplicar = (RadioButton) findViewById(R.id.rbMultiplicar);
RadioButton rbDividir = (RadioButton) findViewById(R.id.rbDividir);

int resultado = 0;
String operacao = "";

if(rbSomar.isChecked()) {
    resultado = intValA + intValB;
    operacao = rbSomar.getText().toString();
}
if(rbSubtrair.isChecked()) {
    resultado = intValA - intValB;
    operacao = rbSubtrair.getText().toString();
}
if(rbMultiplicar.isChecked()) {
    resultado = intValA * intValB;
    operacao = rbMultiplicar.getText().toString();
}
if(rbDividir.isChecked()) {
    resultado = intValA / intValB;
    operacao = rbDividir.getText().toString();
}

Toast.makeText(getApplicationContext(), text: "O resultado da " + operacao + " : " + resultado, Toast.LENGTH_SHORT).show();
```

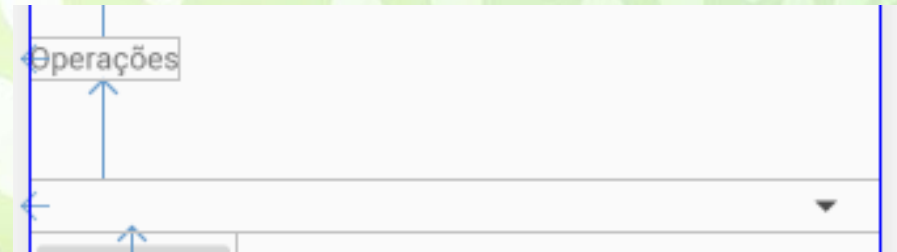
EXECUÇÃO – Radio button



Spinner

- Conhecido em outras linguagens de programação como Select ou Combobox, possui a finalidade de fornecer um conjunto de informações.

```
<Spinner  
    android:id="@+id/spOperacao"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentStart="true"  
    android:layout_below="@+id/lblOpcao"  
    android:layout_marginTop="45dp" />
```



Carregar Spinner

- O código abaixo é responsável por carregar o Spinner, o mesmo é ideal quando recebe valores dinâmicos provenientes de um banco, arquivo ou serviço.

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

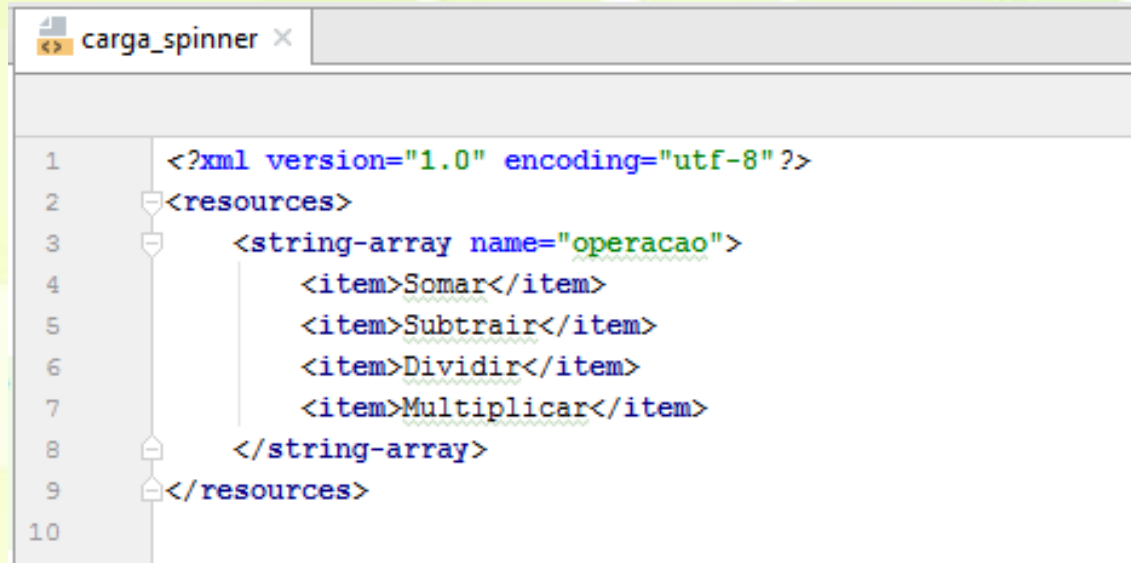
    carregaSpinner();
    addBotaoCalcularClickListener();
}

public void carregaSpinner() {
    Spinner spOperacao = (Spinner) findViewById(R.id.spOperacao);
    List<String> list = new ArrayList<String>();
    list.add("Somar");
    list.add("Subtrair");
    list.add("Multiplicar");
    list.add("Dividir");

    ArrayAdapter<String> dataAdapter = new ArrayAdapter<String>(context: this,
        android.R.layout.simple_spinner_item, list);
    dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
    spOperacao.setAdapter(dataAdapter);
}
```

Carregar Spinner

- Criar um arquivo com extensão .xml na pasta res -> values.
- Definir um nome para a tag string-array e seus respectivos itens:



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <string-array name="operacao">
4          <item>Somar</item>
5          <item>Subtrair</item>
6          <item>Dividir</item>
7          <item>Multiplicar</item>
8      </string-array>
9  </resources>
10
```

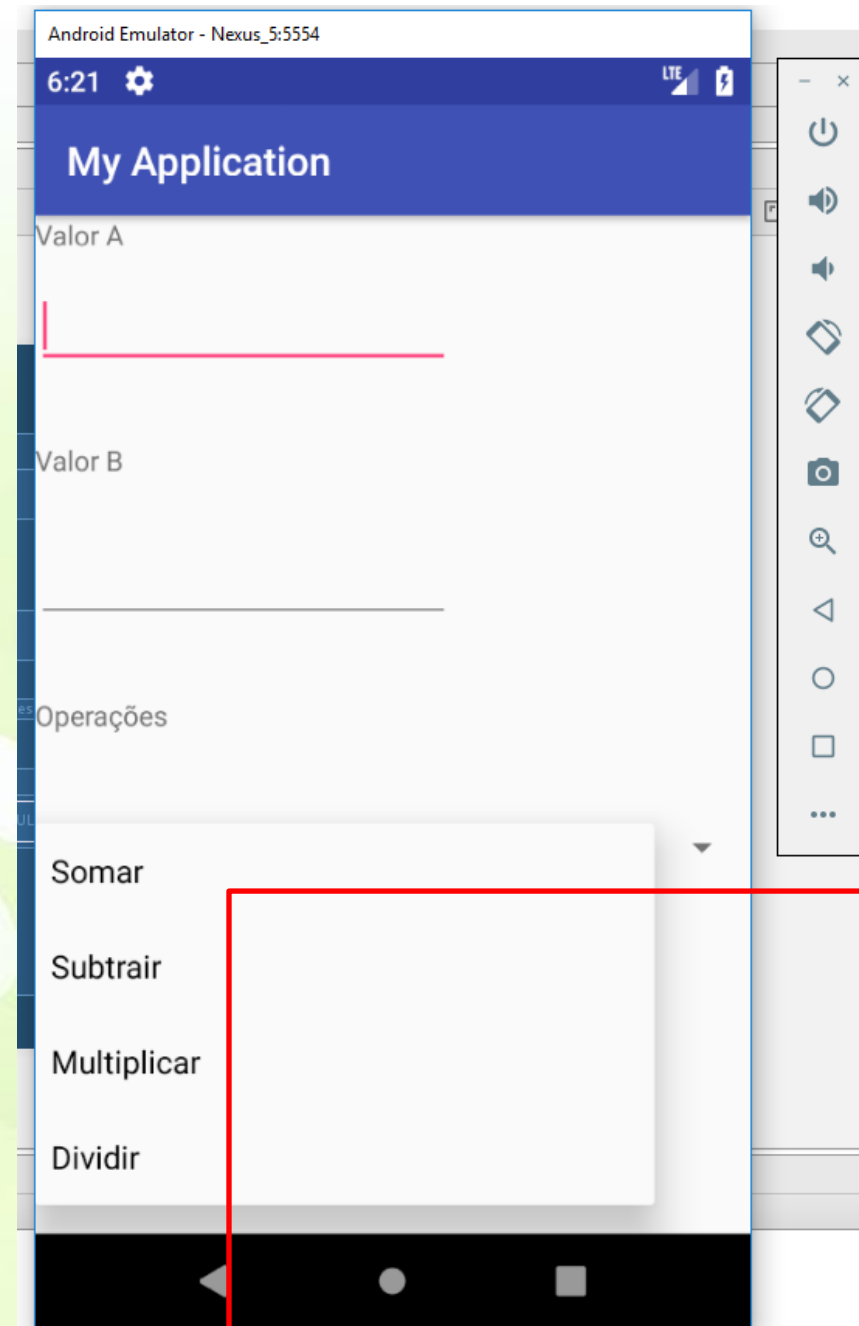

Carregar Spinner

- No Java, definir um método que recebe o componente Spinner e um adapter para receber os valores a partir do recurso criado.

```
public void carregaSpinner2() {  
    Spinner spOperacao = (Spinner) findViewById(R.id.spOperacao);  
  
    ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource( context: this,  
        R.array.operacao, android.R.layout.simple_spinner_item);  
  
    adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);  
  
    spOperacao.setAdapter(adapter);  
}
```

Spinner

- Ao selecionar o componente Spinner, um conjunto de opções é fornecido.

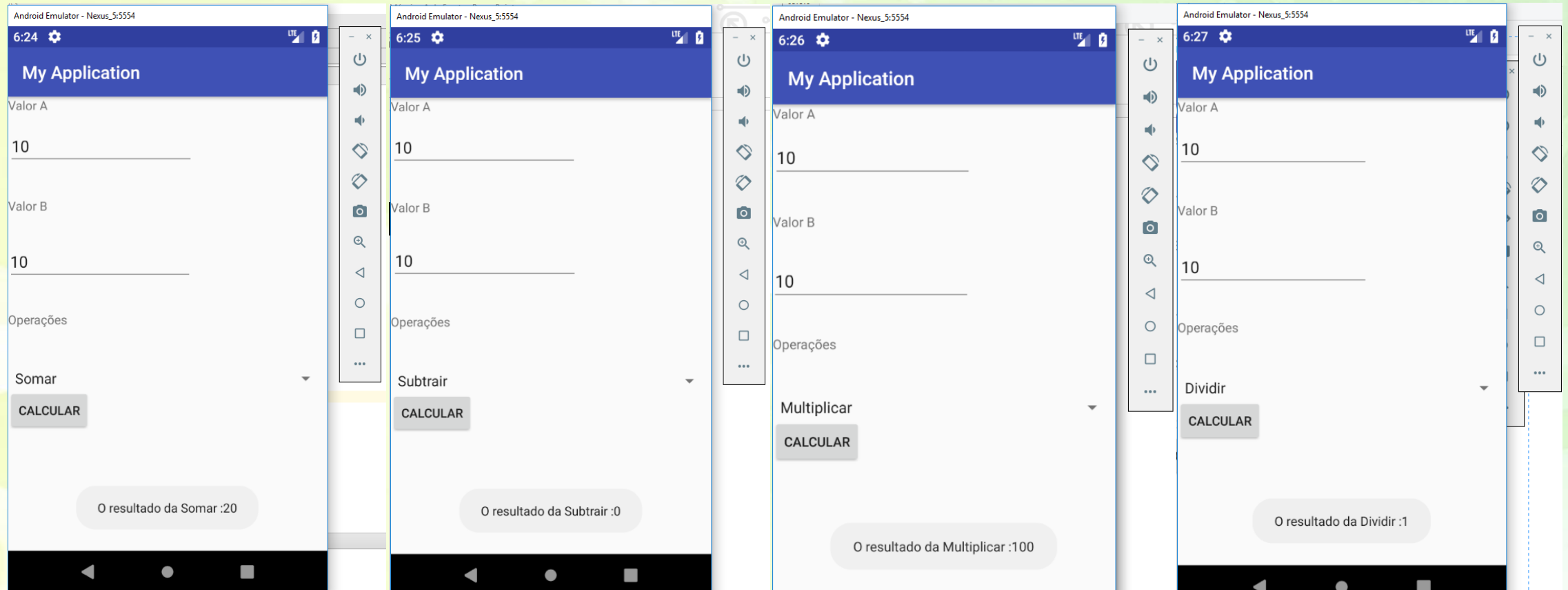


Spinner - Recuperar valor

- Para recuperar um valor selecionado no Spinner, no Java, é necessário executar as instruções abaixo:

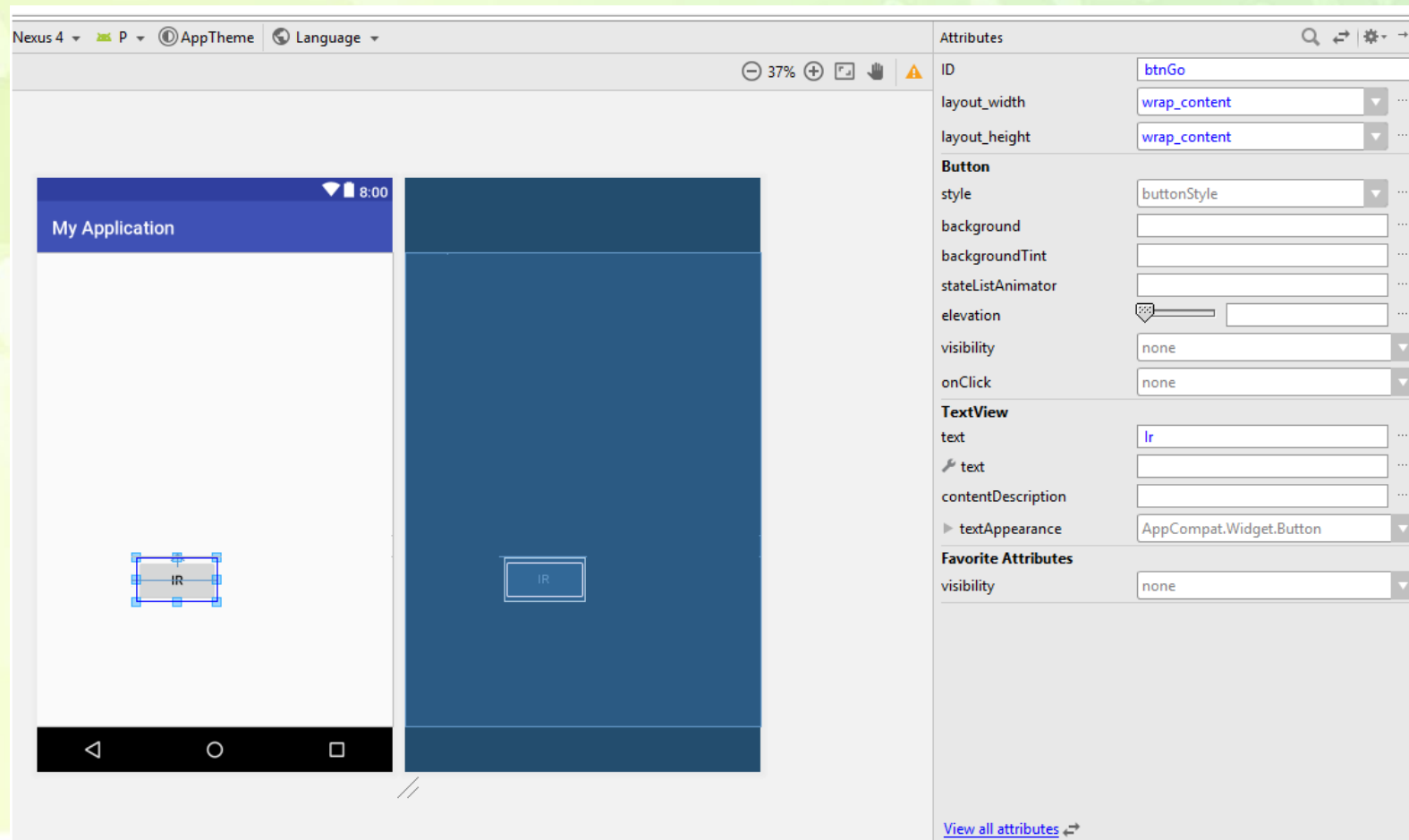
```
Spinner spOperacao = (Spinner) findViewById(R.id.spOperacao);  
String opcao = spOperacao.getItemAtPosition(spOperacao.getSelectedItemPosition()).toString();
```

Execução - Spinner

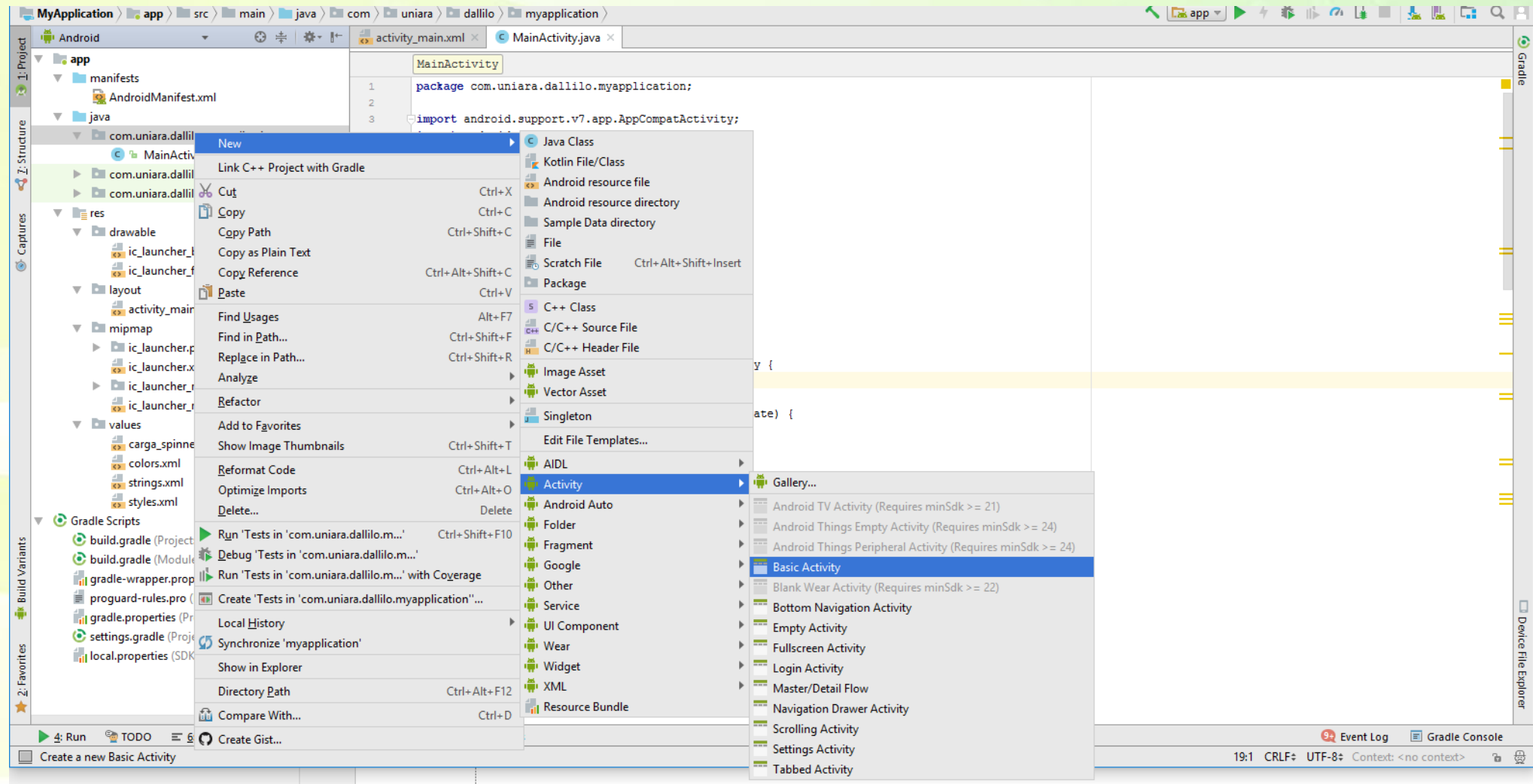


Navegar entre atividades

Crie um botão no arquivo xml da atividade inicial.

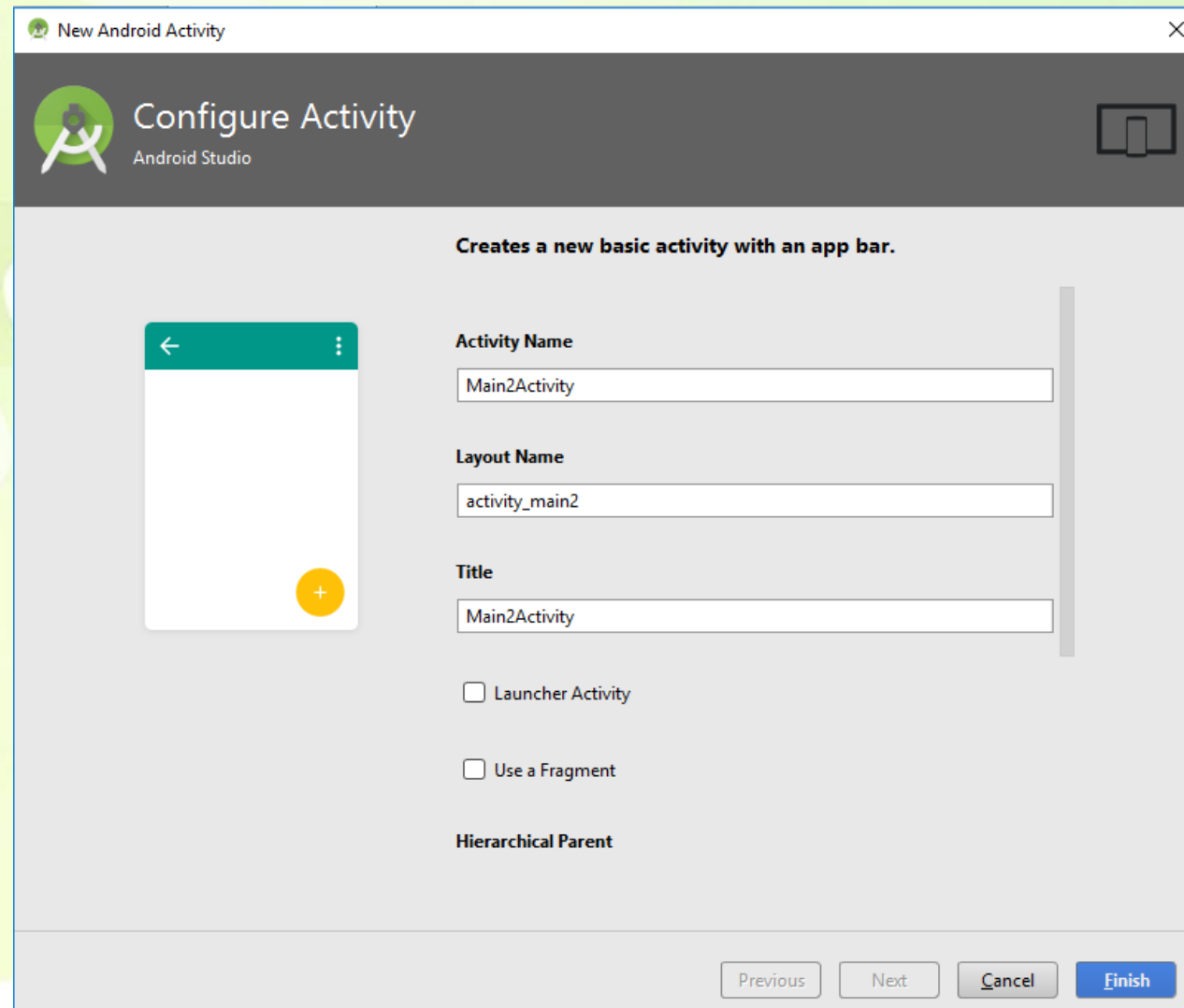


Crie uma Nova Atividade



Crie uma Nova Atividade

Defina o nome da Atividade Java, XML de Layout e o Título da Atividade. Aperte Finish.



New Android Activity

Configure Activity
Android Studio

Creates a new basic activity with an app bar.

Activity Name
Main2Activity

Layout Name
activity_main2

Title
Main2Activity

☐ Launcher Activity

☐ Use a Fragment

Hierarchical Parent

Previous Next Cancel Finish

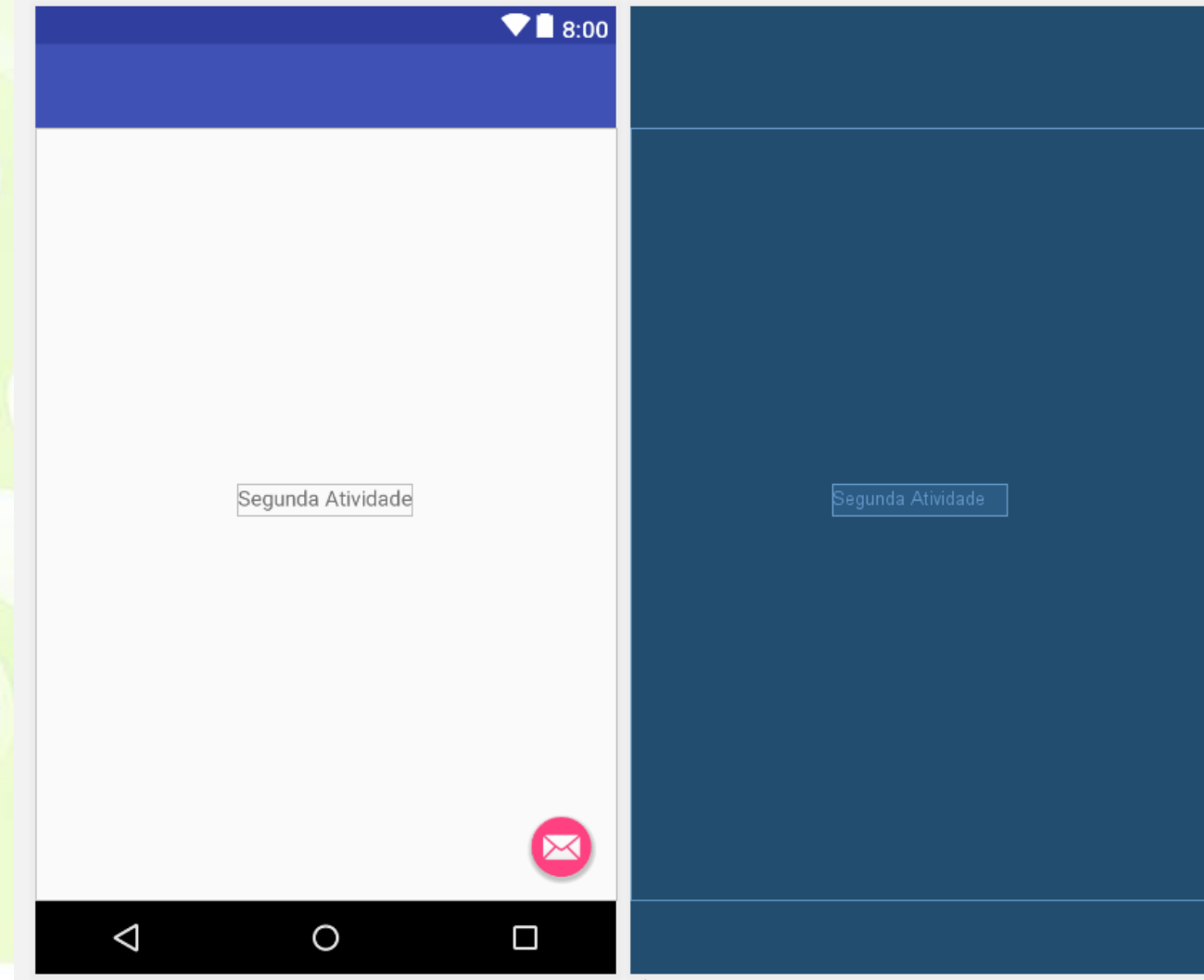
Nova Atividade

Segunda atividade definida apenas com um TextView.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout 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"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.uniara.dallilo.myapplication.Main2Activity"
    tools:showIn="@layout/activity_main2">

    <TextView
        android:id="@+id/txtInfo"
        android:layout_width="wrap_content"
        android:layout_height="21dp"
        android:text="Segunda Atividade"
        tools:layout_editor_absoluteX="134dp"
        tools:layout_editor_absoluteY="235dp" />

</android.support.constraint.ConstraintLayout>
```



Abrir Atividade

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

public void addBotaoListener(){
    Button btnGo = (Button) findViewById(R.id.btnGo);
    btnGo.setOnClickListener(new View.OnClickListener() {

        public void onClick(View v) {
            abrirAtividade2();
        }
    });
}

public void abrirAtividade2(){
    Intent intent = new Intent( packageContext: this, Main2Activity.class);
    startActivity(intent);
}
```



My Application

IR



Main2Activity

Segunda Atividade



Enviar parâmetros entre Atividades

- Utilizar a chave como pacote da atividade + nome da mensagem.

```
public static final String EXTRA_MESSAGE = "com.uniara.dallilo.myapplication.MESSAGE";

public void abrirAtividade2() {
    Intent intent = new Intent( packageContext: this, Main2Activity.class);

    EditText valorA = (EditText) findViewById(R.id.txtValorA);
    String message = valorA.getText().toString();
    intent.putExtra(EXTRA_MESSAGE, message);

    startActivity(intent);
}
```

Enviar parâmetros entre Atividades

```
public class Main2Activity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main2);  
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
  
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);  
        fab.setOnClickListener((view) -> {  
            Snackbar.make(view, text: "Replace with your own action", Snackbar.LENGTH_LONG)  
                .setAction(text: "Action", listener: null).show();  
        });  
  
        Intent intent = getIntent();  
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);  
    }  
}
```

Enviar parâmetros entre Atividades

- O método `putExtra` da classe `Intent` é sobrecarregado para receber outros tipos de dados como `int`, `float`, `boolean`, `object`, entre outros.

```
//Para passar um valor da minha atividade do tipo object:  
intent.putExtra("MyClass", obj);
```

```
// Para receber um valor de object em outra atividade.  
Object ob = getIntent().getSerializableExtra("MyClass");
```

Atribuir Valor em um Componente

```
public class Main2Activity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main2);  
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);  
        setSupportActionBar(toolbar);  
  
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);  
        fab.setOnClickListener((view) -> {  
            Snackbar.make(view, text: "Replace with your own action", Snackbar.LENGTH_LONG)  
                .setAction(text: "Action", listener: null).show();  
        });  
  
        Intent intent = getIntent();  
        String message = intent.getStringExtra(MainActivity.EXTRA_MESSAGE);  
  
        TextView textView = findViewById(R.id.textView);  
        textView.setText(message);  
    }  
}
```

Referências

- SILVA, D. **Desenvolvimento para dispositivos móveis**. São Paulo: Pearson Education do Brasil, 2016.
- LEE, V., SCHNEIDER, H., SCHELL, R. **Aplicações móveis: arquitetura, projeto e desenvolvimento**. São Paulo: Pearson Education do Brasil, 2005.
- MEDNIEKS, Z; et al. **Programando o Android**. 2. ed. São Paulo: Novatec, 2012.
- DEITEL, P; DEITEL, H. **Java: como programar**. 8. ed. São Paulo: Pearson, 2010.
- DUARTE, W. **Delphi para Android e IOS Desenvolvendo Aplicações Móveis**. Rio de Janeiro: Brasport Livros e Multimídia, 2015.
- ABLESON, W. Frank; et al. **Android em ação**. 3. ed. Rio de Janeiro: Elsevier, 2012.
- LECHETA, R. R. **Google Android para Tablets. Aprenda a desenvolver aplicações para o Android**. São Paulo: Novatec, 2012.
- LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 2. ed. São Paulo: Novatec, 2010.
- ASCENCIO, A. F. G.; CAMPOS, E. A. V. de. **Fundamentos da programação de computadores: algoritmos, pascal, C/C ++ e Java**. 2. ed. São Paulo: Pearson, 2007