

Android
android-calculo-juros

SUMÁRIO

1.	Informativo	3
2.	Objetivo	4
3.	Pré-Requisitos.....	5
4.	Versão	6
4.1.	minSdkVersion	6
4.2.	targetSdkVersion	6
5.	Novo Projeto.....	7
6.	Construindo o Projeto	11
6.1.	Criando os layouts	11
6.2.	strings.xml.....	14
6.3.	Referências	16
6.4.	Ações	16
7.	Resumo	21
8.	Referências	22

1. Informativo

Autor(a): Helena Strada

Data de Criação: jan/2018

2. Objetivo

O objetivo desta apostila é apresentar o passo-a-passo para construir a aplicação do android-calculo-juros. Ela tem como objetivo mostrar como manipular os valores de entrada do usuário e os componentes de tela.

3. Pré-Requisitos

Java

Orientação a Objetos;
APIs;
Bibliotecas.

4. Versão

Android Studio 3.0.1

4.1. minSdkVersion

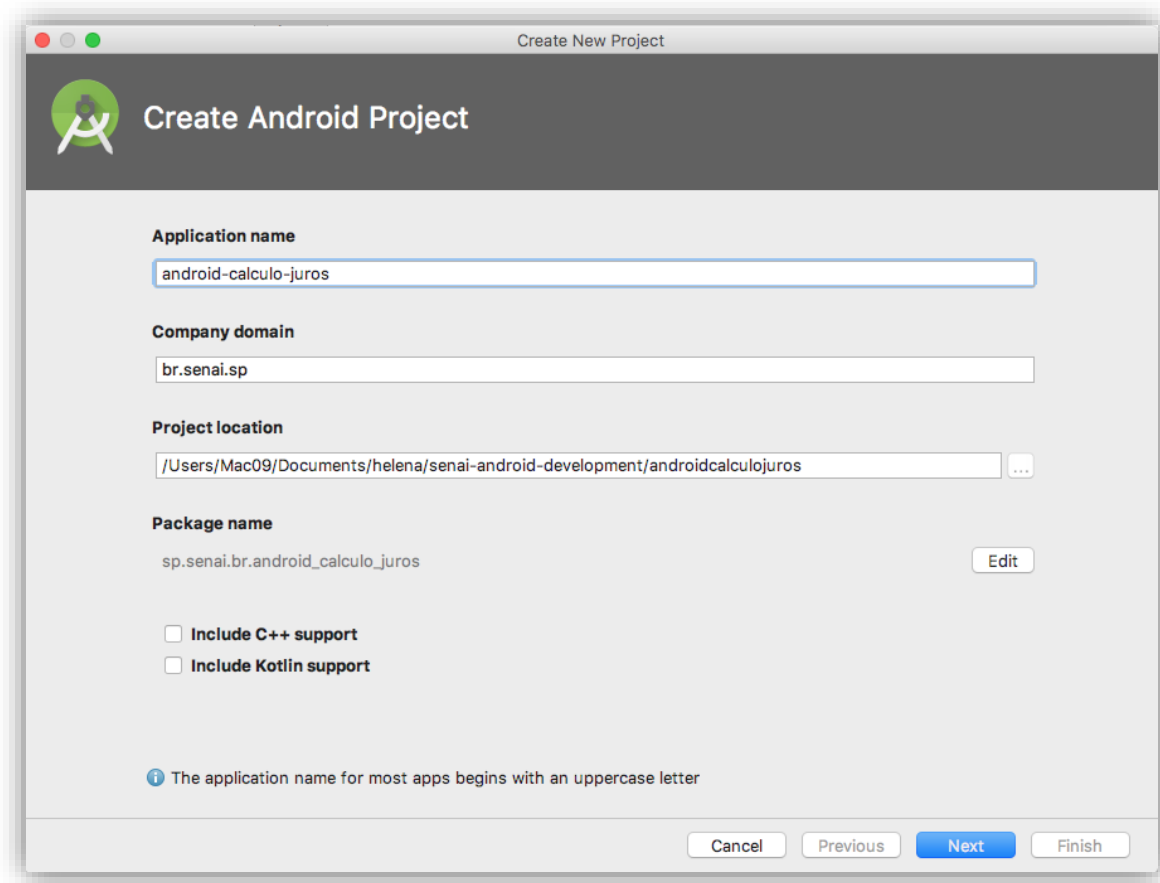
15

4.2. targetSdkVersion

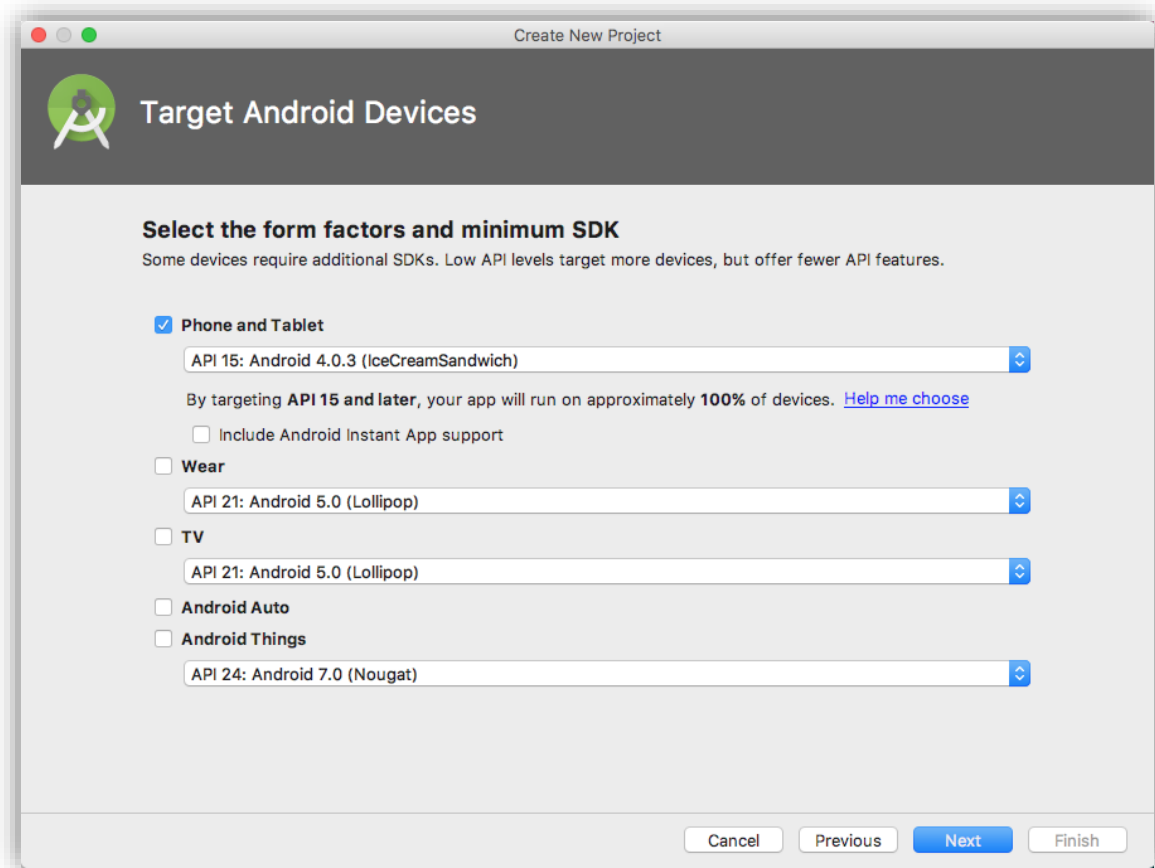
26

5. Novo Projeto

Vamos começar o nosso exercício criando um novo projeto chamado ‘android-calculadora-juros’. Para este projeto, utilizaremos uma *blank activity*.

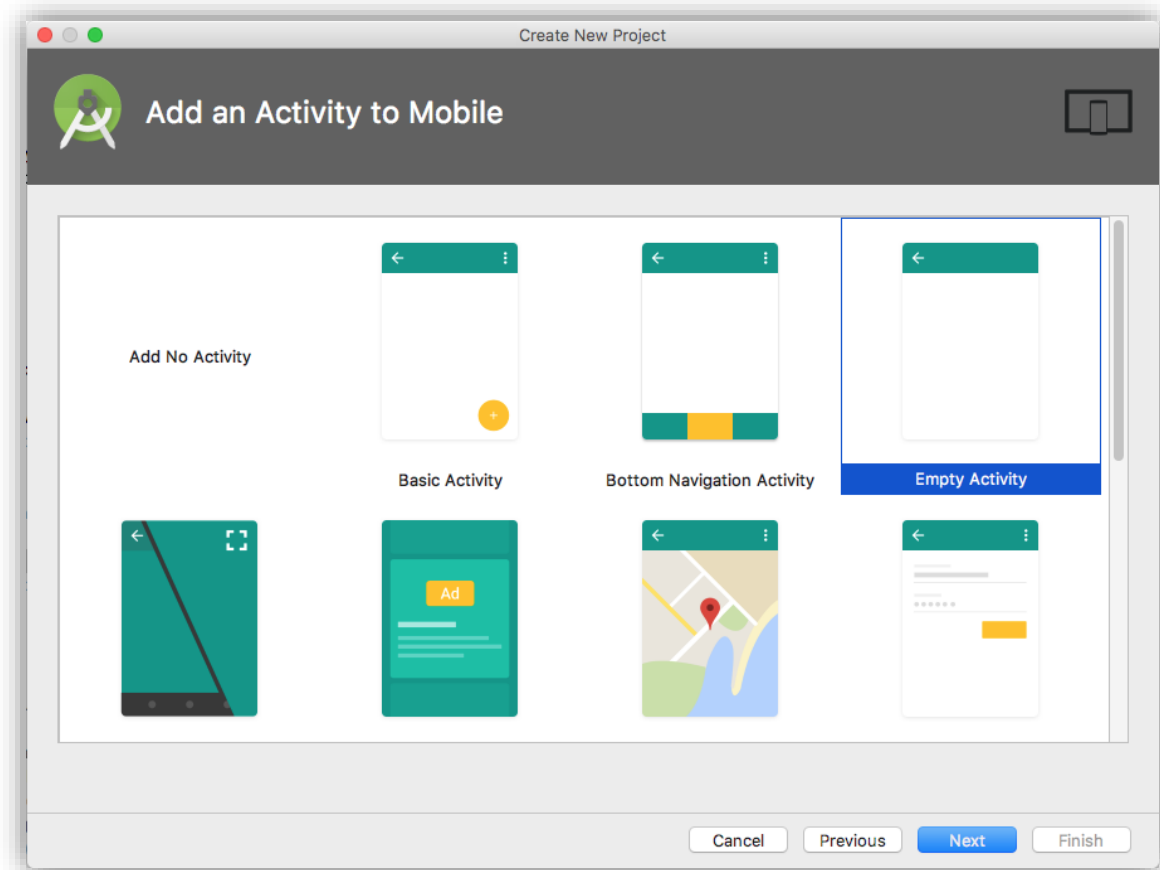


Clicar em “Next”.

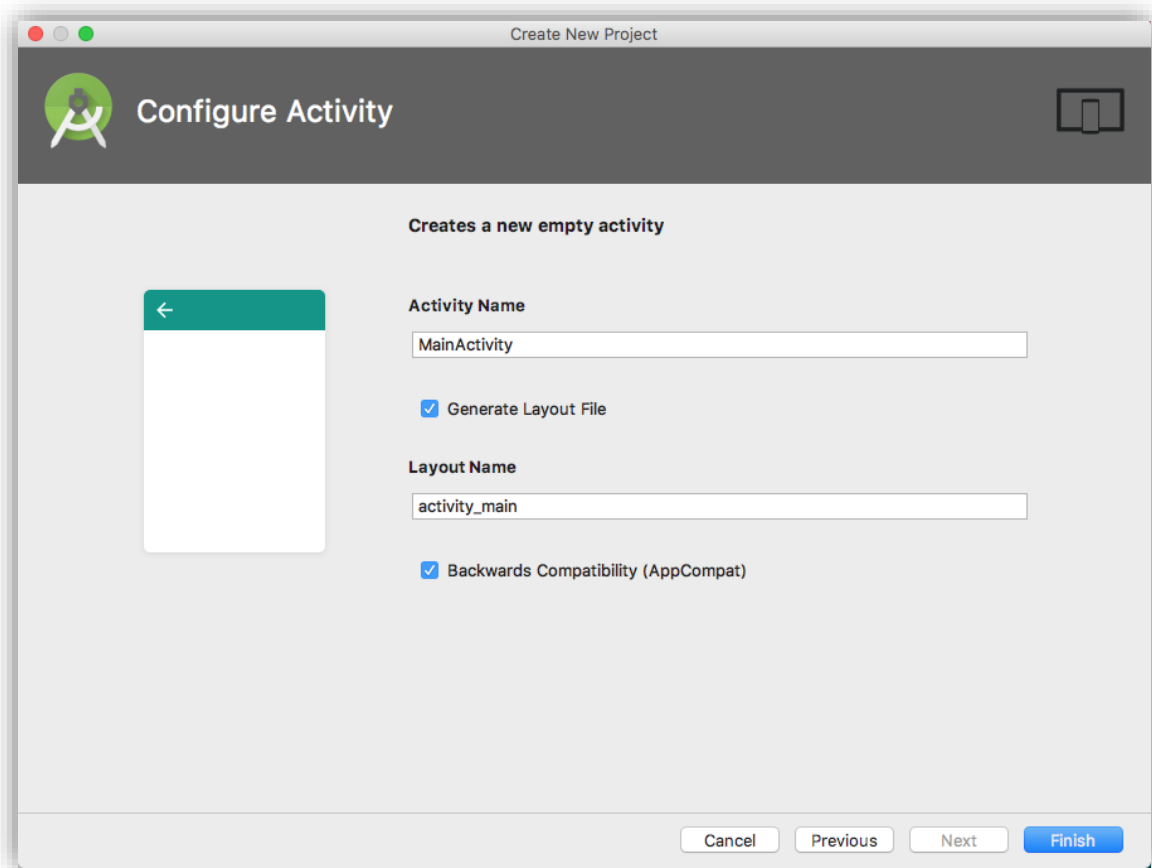


Clicar em “Next”.

Vamos começar com *Blank Activity*.



Clicar em “Next”.



Clicar em “Finish”.

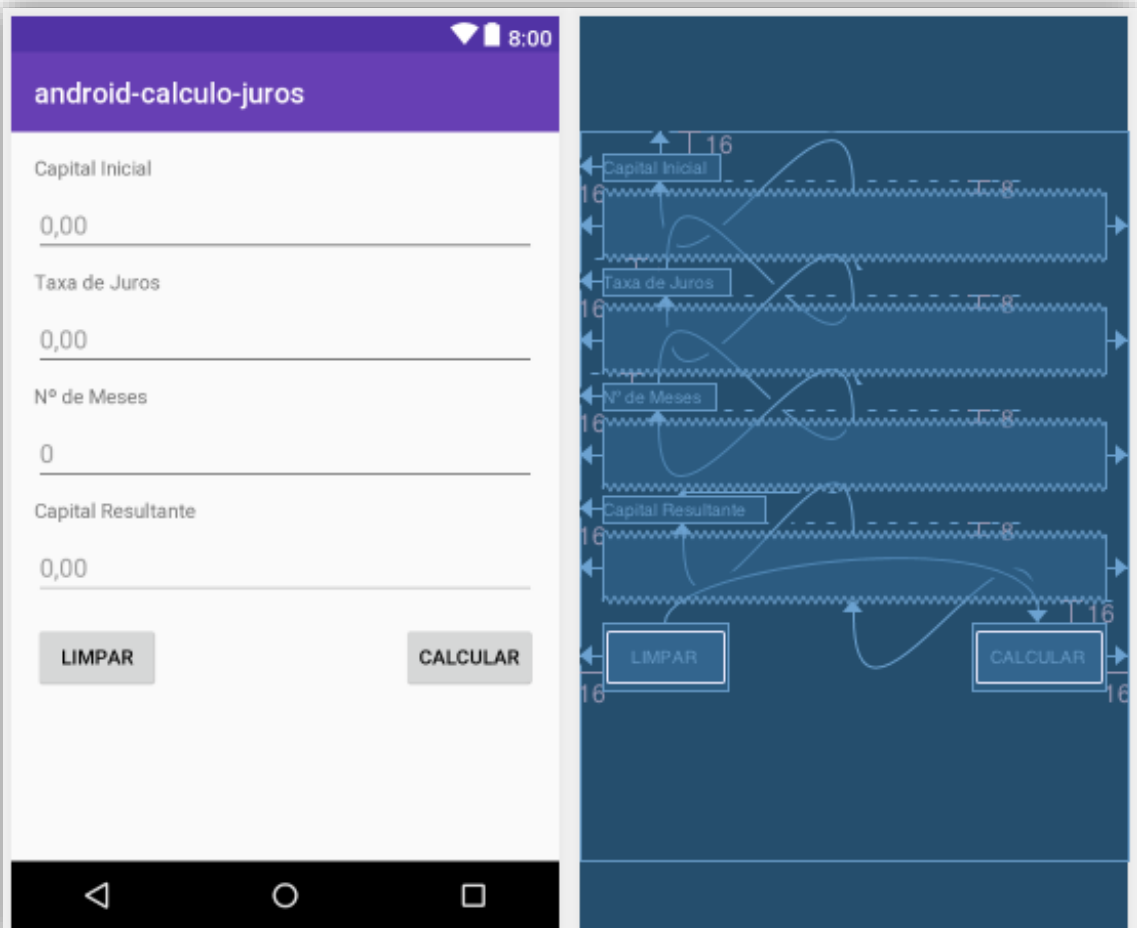
6. Construindo o Projeto

Para a construção do nosso projeto, iremos realizar os passos que seguem:

1. Criar os layouts
2. Associar e criar as strings que forem necessárias
3. Buscar as referências do que foi criado na nossa MainActivity
4. Realizar as ações que desejamos.

6.1. Criando os layouts

Na nossa `activity_main.xml`, iremos construir o layout para que ele fique conforme a imagem:



Para isto, iremos criar um *ConstraintLayout* e criar os campos que precisamos. São eles: capital inicial (o quanto será investido), taxa de juros (cobrado), número de meses (é o tempo que o dinheiro ficará investido) e o último é o nosso capital resultante (à partir do tempo que o nosso dinheiro será investido e a taxa de juros, podemos realizar o cálculo de quanto será nosso retorno).

```
<?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"
tools:ignore="RtlHardcoded"
tools:context="sp.senai.br.android_calculo_juros.MainActivity">
```

```
<TextView
    android:id="@+id/tvCapital"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:text="@string/capital_inicial"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
/>
```

```
<EditText
    android:id="@+id/edCapital"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginTop="8dp"
    android:ems="10"
    android:hint="@string/num_decimal"
    android:inputType="none"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/tvCapital" />
```

```
<TextView
    android:id="@+id/tvTaxa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="8dp"
    android:text="@string/taxa_juros"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edCapital" />
```

```
<EditText
    android:id="@+id/edTaxa"
    android:layout_width="0dp"
```

```
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginRight="16dp"
android:layout_marginTop="8dp"
android:ems="10"
android:hint="@string/num_decimal"
android:inputType="none"
app:layout_constraintHorizontal_bias="0.0"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toBottomOf="@+id/tvTaxa" />
```

<TextView

```
android:id="@+id/tvMeses"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginTop="8dp"
android:text="@string/num_meses"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toBottomOf="@+id/edTaxa" />
```

<EditText

```
android:id="@+id/edMeses"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginRight="16dp"
android:layout_marginTop="8dp"
android:ems="10"
android:hint="@string/num_inteiro"
android:inputType="number"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toBottomOf="@+id/tvMeses" />
```

<TextView

```
android:id="@+id/tvResultado"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginTop="8dp"
android:text="@string/capital_resultante"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintTop_toBottomOf="@+id/edMeses" />
```

<EditText

```
android:id="@+id/edResultado"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginLeft="16dp"
android:layout_marginRight="16dp"
```

```

        android:layout_marginTop="8dp"
        android:ems="10"
        android:enabled="false"
        android:hint="@string/num_decimal"
        android:inputType="numberDecimal"
        app:layout_constraintHorizontal_bias="0.503"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/tvResultado" />

<Button
    android:id="@+id/btnCalcular"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="16dp"
    android:layout_marginTop="16dp"
    android:text="@string/calcular"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edResultado" />

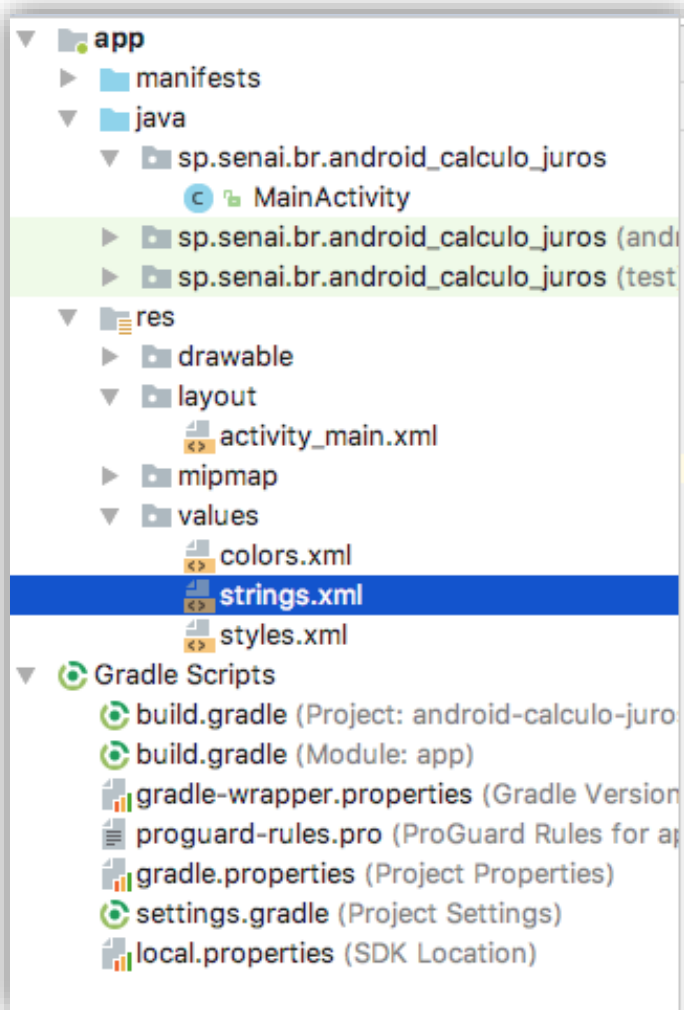
<Button
    android:id="@+id/btnLimpar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="0dp"
    android:text="@string/limpar"
    android:onClick="limparCampos"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btnCalcular" />

</android.support.constraint.ConstraintLayout>

```

6.2. strings.xml

Para realizarmos a alteração das strings para toda a aplicação, podemos criar strings que serão compartilhadas e podem ser utilizadas como recurso de toda a aplicação. Para isto, iremos alterar o arquivo strings.xml.



```
<resources>
  <string name="app_name">android-calculo-juros</string>
  <string name="capital_inicial">Capital Inicial</string>
  <string name="taxa_juros">Taxa de Juros</string>
  <string name="num_meses">Nº de Meses</string>
  <string name="capital_resultante">Capital Resultante</string>
  <string name="calcular">Calcular</string>
  <string name="limpar">Limpar</string>
  <string name="num_decimal">0,00</string>
  <string name="num_inteiro">0</string>
</resources>
```

Configuramos apenas o nosso layout.

6.3. Referências

Agora que criamos o nosso layout e estilizamos, vamos buscar as referências dos campos criados.

A finalidade é que precisamos receber como entrada os valores que o usuário digitou, realizar um cálculo com base nos dados que foram passados e, ao clicar no botão de calcular, gerar um resultado de saída para o usuário.

Podemos definir fora do onCreate os valores que desejamos para que tenhamos acesso em toda nossa classe.

```
private Button btnCalcular;  
private Button btnLimpar;  
  
private EditText edCapital;  
private EditText edMeses;  
private EditText edTaxa;  
private EditText edResultado;
```

E dentro do método onCreate, realizaremos os seguintes passos:

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    // Buscando as referencias dos botoes  
    btnCalcular = findViewById(R.id.btnCalcular);  
    btnLimpar = findViewById(R.id.btnLimpar);  
  
    // Buscando as referencias do campos de texto  
    edCapital = findViewById(R.id.edCapital);  
    edMeses = findViewById(R.id.edMeses);  
    edTaxa = findViewById(R.id.edTaxa);  
    edResultado = findViewById(R.id.edResultado);  
}
```

6.4. Ações

Agora que criamos o nosso layout e buscamos as referências através dos nossos ids, precisamos realizar as ações que desejamos.

Para realizarmos alguma ação com o botão, algumas delas.

A primeira é realizar a chamada diretamente no botão.

```
// Acao no proprio botao ao clicar  
btnCalcular.setOnClickListener(new View.OnClickListener() {  
    @Override
```



```

public void onClick(View view) {
    try {
        // Configurar o formatador de Números
        NumberFormat fmt = NumberFormat.getInstance();
        fmt.setMaximumFractionDigits(2);
        fmt.setMinimumFractionDigits(2);

        // lê e converte os valores informados na tela
        double capital = fmt.parse(edCapital.getText().toString()).doubleValue();
        int meses = Integer.parseInt(edMeses.getText().toString());
        double taxa = fmt.parse(edTaxa.getText().toString()).doubleValue();

        // efetua o calculo
        double resultado = capital * taxa / 100 * meses + capital;

        // apresenta o resultado
        edResultado.setText(String.format("%.2f", resultado));

    } catch (Exception ex) {
        // Caso contrário, informa que há erro nos valores digitados
        Toast.makeText(getApplicationContext(), "Número(s) inválido(s)!",
            Toast.LENGTH_LONG).show();
    }
}
});

```

Podemos criar um método e chamá-lo. A vantagem deste é que o seu método onCreate não conterá todos os métodos e você pode reaproveitá-lo para outras funções/métodos.

```

btnCalcular.setOnClickListener(chamandoUmMetodo);

```

```

private View.OnClickListener chamandoUmMetodo = new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        try {
            // Configurar o formatador de Números
            NumberFormat fmt = NumberFormat.getInstance();
            fmt.setMaximumFractionDigits(2);
            fmt.setMinimumFractionDigits(2);

            // lê e converte os valores informados na tela
            double capital = fmt.parse(edCapital.getText().toString()).doubleValue();
            int meses = Integer.parseInt(edMeses.getText().toString());
            double taxa = fmt.parse(edTaxa.getText().toString()).doubleValue();

            // efetua o calculo
            double resultado = capital * taxa / 100 * meses + capital;

```

```

        // apresenta o resultado
        edResultado.setText(String.format("%.2f", resultado));

    } catch (Exception ex) {
        // Caso contrário, informa que há erro nos valores digitados
        Toast.makeText(getApplicationContext(), "Número(s) inválido(s)!",
            Toast.LENGTH_LONG).show();
    }

}
};

```

Além disso, podemos criar uma classe para realizar a nossa ação, porém será necessário que essa classe implemente View.OnClickListener.

```

private View.OnClickListener chamandoUmMetodo = new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        try {
            // Configurar o formatador de Números
            NumberFormat fmt = NumberFormat.getInstance();
            fmt.setMaximumFractionDigits(2);
            fmt.setMinimumFractionDigits(2);

            // lê e converte os valores informados na tela
            double capital = fmt.parse(edCapital.getText().toString()).doubleValue();
            int meses = Integer.parseInt(edMeses.getText().toString());
            double taxa = fmt.parse(edTaxa.getText().toString()).doubleValue();

            // efetua o calculo
            double resultado = capital * taxa / 100 * meses + capital;

            // apresenta o resultado
            edResultado.setText(String.format("%.2f", resultado));

        } catch (Exception ex) {
            // Caso contrário, informa que há erro nos valores digitados
            Toast.makeText(getApplicationContext(), "Número(s) inválido(s)!",
                Toast.LENGTH_LONG).show();
        }

    }
};

btnCalcular.setOnClickListener(new ChamandoUmaClasse());

```

Há uma outra alternativa. Podemos na nossa MainActivity realizar a ação:

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener{
```

Ele pedirá para que seja implementado o método onClick. Ao realizarmos essa ação, podemos fazer um switch/case para pegarmos qual botão estamos clicando e realizarmos a ação que desejamos.

```
    btnCalcular.setOnClickListener(this);
```

```
    @Override
```

```
    public void onClick(View view) {
```

```
        int id = view.getId();
```

```
        switch (id) {
```

```
            case R.id.btnCalcular:
```

```
                try {
```

```
                    // Configurar o formatador de Números
```

```
                    NumberFormat fmt = NumberFormat.getNumberInstance();
```

```
                    fmt.setMaximumFractionDigits(2);
```

```
                    fmt.setMinimumFractionDigits(2);
```

```
                    // lê e converte os valores informados na tela
```

```
                    double capital = fmt.parse(edCapital.getText().toString()).doubleValue();
```

```
                    int meses = Integer.parseInt(edMeses.getText().toString());
```

```
                    double taxa = fmt.parse(edTaxa.getText().toString()).doubleValue();
```

```
                    // efetua o calculo
```

```
                    double resultado = capital * taxa / 100 * meses + capital;
```

```
                    // apresenta o resultado
```

```
                    edResultado.setText(String.format("%.2f", resultado));
```

```
                } catch (Exception ex) {
```

```
                    // Caso contrário, informa que há erro nos valores digitados
```

```
                    Toast.makeText(this, "Número(s) inválido(s)!", Toast.LENGTH_LONG).show();
```

```
                }
```

```
                break;
```

```
            case R.id.btnLimpar:
```

```
                edCapital.setText(getResources().getText(R.string.num_decimal));
```

```
                edMeses.setText(getResources().getText(R.string.num_inteiro));
```

```
                edTaxa.setText(getResources().getText(R.string.num_decimal));
```

```
                edResultado.setText(getResources().getText(R.string.num_decimal));
```

```
                break;
```

```
        }
```

```
    };
```

Há mais uma maneira que não é comum de ser utilizada, pelo fato de refatoração e referência direto na activity_main.xml.

Quando criamos o nosso layout lá em cima, no botão de limpar, vamos colocar uma ação diretamente nele.

```
<Button
    android:id="@+id/btnLimpar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="0dp"
    android:text="@string/limpar"
    android:onClick="limparCampos"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="@+id/btnCalcular" />
```

Como a nossa activity_main está relacionada a nossa MainActivity, o nosso método estará nela.

```
public void limparCampos(View view) {
    edCapital.setText(getResources().getText(R.string.num_decimal));
    edMeses.setText(getResources().getText(R.string.num_inteiro));
    edTaxa.setText(getResources().getText(R.string.num_decimal));
    edResultado.setText(getResources().getText(R.string.num_decimal));
    Toast.makeText(this, "Campos Limpos", Toast.LENGTH_LONG).show();
};
```

7. Resumo

Aplicação construída.

8. Referências

<https://developer.android.com/studio/index.html?hl=pt-br>
<http://andersonleite.com.br/2015/06/15/parcelable-android.html>
<https://stackoverflow.com/questions/3323074/android-difference-between-parcelable-and-serializable>
<https://www.3pillarglobal.com/insights/parcelable-vs-java-serialization-in-android-app-development>
<https://android.jlelse.eu/parcelable-vs-serializable-6a2556d51538>
<https://pt.stackoverflow.com/questions/38492/quando-e-como-implementar-o-parcelable-vs-serializable>