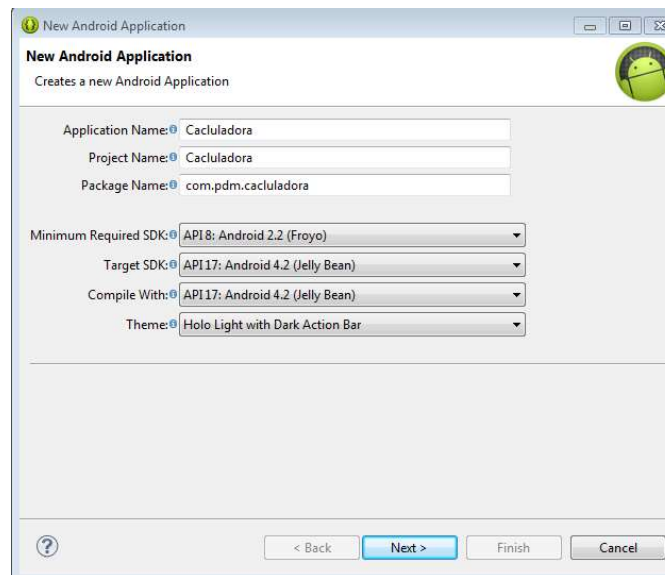


Hoje aprenderemos a fazer uma calculadora simples, com as quatro operações básicas. Crie um novo projeto com as informações conforme a imagem abaixo:

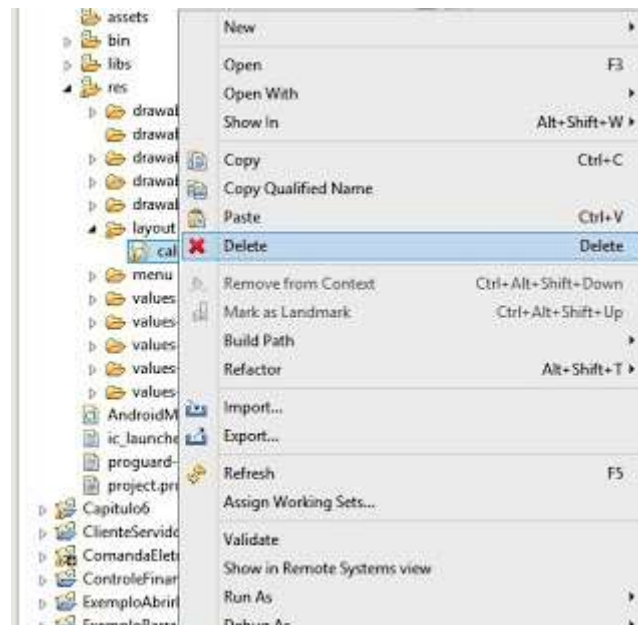


Interface

Para construir a interface gráfica utilizaremos o layout `TableLayout` e o `TableRow`. O `TableLayout` é um layout que organiza os componentes na forma de tabelas com linhas e colunas o componente `TableRow` representa as linhas dessa tabela.

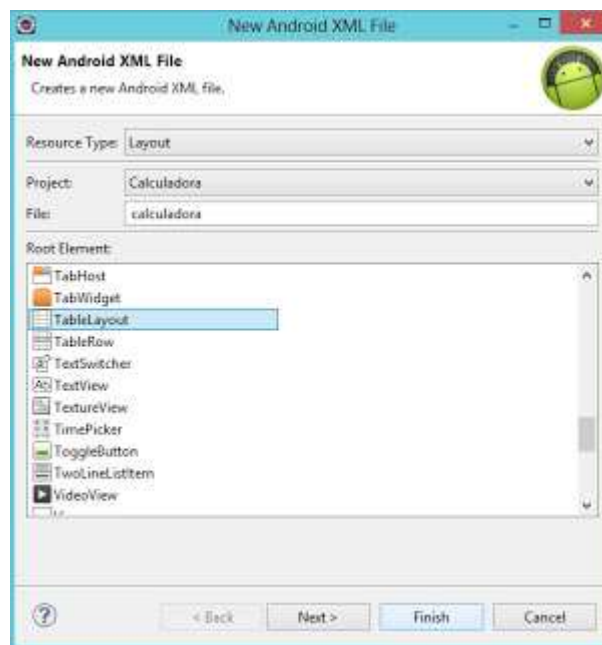
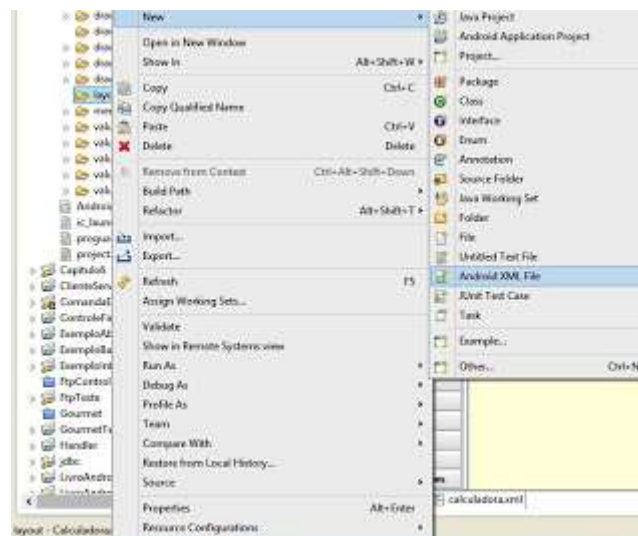
Exclua o arquivo xml da pasta layout da aplicação e crie-o novamente com o `TableLayout`:

Vá até a pasta `res` do projeto abra a pasta `layout`, clique com o botão direito do mouse no arquivo xml do projeto e clique em delete.



Criando arquivo xml

Clique com botão direito na pasta `layout` > new > Android XML File:



Depois de criado o arquivo de layout, clique com o botão direito do mouse no layout e depois em `addRow`, criamos a primeira linha do projeto `TableRow1`, repita esse processo mais 5 vezes. Configura as seguintes propriedades do `TableLayout`:

- Stretch Columns: 0,1,2,3;
- Padding: 5dp;

Na primeira linha coloque um componente `EditText` com os seguintes valores nos parâmetros `id` e `text`:

- Id: `txt_visor`;
- Texto: 0.0

Na segunda linha coloque quatro `Buttons` com os seguintes valores nos parâmetros `id` e `text`:

- Id: `bt_7`, `bt_8`, `bt_9` e `bt_divi`;
- Texto: 7, 8, 9, /;

Na terceira linha coloque quatro Buttons com os seguintes valores nos parâmetros id e text:

- Id: bt_4, bt_5, bt_6 e bt_mult;
- Text: 4, 5, 6 , *;

Na quarta linha coloque quatro Buttons com os seguintes valores nos parâmetros id e text:

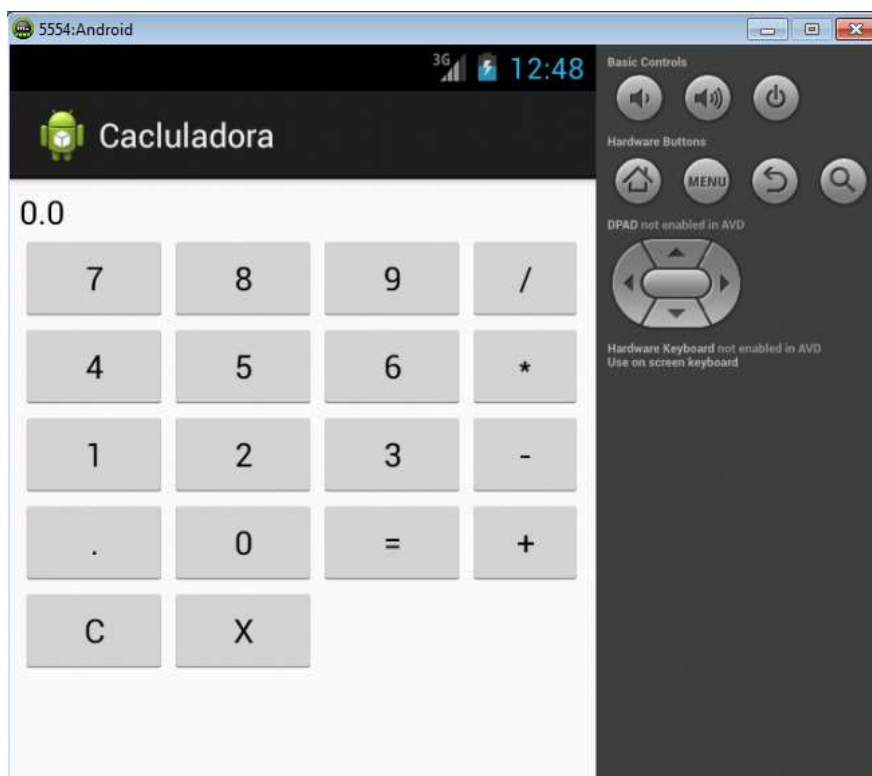
- Id: bt_1, bt_2, bt_3 e bt_subt;
- Text: 1, 2, 3 , -;

Na quinta linha coloque quatro Buttons com os seguintes valores nos parâmetros id e text:

- Id: bt_ponto, bt_0, bt_igual e bt_soma;
- Text: ., 0, = , +;

Acrescente uma linha e coloque dois Buttons com os seguintes valores nos parâmetros id e text:

- Id: bt_limpar, bt_sair;
- Text: C,X;



Implementando as funções da calculadora

Vamos agora criar as funções para efetuar operações na calculadora.

Variáveis a serem utilizadas na aplicação:

package com.pdm.cacluladora;

```

import android.os.Bundle;
import android.app.Activity;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;

import android.view.Menu;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener {
    /* Duas variáveis do tipo double para realizar operações */
    private double operador1, operador2;
    /* Variável do tipo string que armazena as operações */
    private String operadores;
    /* Variável do tipo EditText para o visor */
    private EditText txt visor;
    /* Componentes Buttons para o teclado numerico */
    private Button bt_0, bt_1, bt_2, bt_3, bt_4, bt_5, bt_6;
    private Button bt_7, bt_8, bt_9;
    /* Os outros Buttons do teclado */
    private Button bt_ponto, bt_soma, bt_multi, bt_sub, bt_divi;
    private Button bt_igual, bt_limpa, bt_sair;

    @Override
    protected void onCreate(Bundle savedInstanceState) { ....

```

Inicializa os componentes da tela:

```

private void Inicializar()
{
    txt_visor = (EditText) findViewById(R.id.txt_visor);
    bt_0 = (Button) findViewById(R.id.bt_zero);
    bt_1 = (Button) findViewById(R.id.bt_1);
    bt_2 = (Button) findViewById(R.id.bt_2);
    bt_3 = (Button) findViewById(R.id.bt_3);
    bt_4 = (Button) findViewById(R.id.bt_4);
    bt_5 = (Button) findViewById(R.id.bt_5);
    bt_6 = (Button) findViewById(R.id.bt_6);
    bt_7 = (Button) findViewById(R.id.bt_7);
    bt_8 = (Button) findViewById(R.id.bt_8);
    bt_9 = (Button) findViewById(R.id.bt_9);
    bt_ponto = (Button) findViewById(R.id.bt_ponto);
    bt_soma = (Button) findViewById(R.id.bt_soma);
    bt_multi = (Button) findViewById(R.id.bt_mult);
    bt_divi = (Button) findViewById(R.id.bt_div);
    bt_igual = (Button) findViewById(R.id.bt_igual);
    bt_limpa = (Button) findViewById(R.id.bt_limpar);
    bt_sair = (Button) findViewById(R.id.bt_sair);
}

```

Trata os eventos dos componentes na tela:

```

private void listener()
{
    bt_0.setOnClickListener((android.view.View.OnClickListener) this);

```

```

bt_1.setOnClickListener((android.view.View.OnClickListener) this);
bt_2.setOnClickListener((android.view.View.OnClickListener) this);
bt_3.setOnClickListener((android.view.View.OnClickListener) this);
bt_4.setOnClickListener((android.view.View.OnClickListener) this);
bt_5.setOnClickListener((android.view.View.OnClickListener) this);
bt_6.setOnClickListener((android.view.View.OnClickListener) this);
bt_7.setOnClickListener((android.view.View.OnClickListener) this);
bt_8.setOnClickListener((android.view.View.OnClickListener) this);
bt_9.setOnClickListener((android.view.View.OnClickListener) this);

bt_ponto.setOnClickListener((android.view.View.OnClickListener) this);
bt_soma.setOnClickListener((android.view.View.OnClickListener) this);
bt_multi.setOnClickListener((android.view.View.OnClickListener) this);
bt_divi.setOnClickListener((android.view.View.OnClickListener) this);
bt_igual.setOnClickListener((android.view.View.OnClickListener) this);
bt_limpa.setOnClickListener((android.view.View.OnClickListener) this);
bt_sair.setOnClickListener((android.view.View.OnClickListener) this);
}

```

Teclado numérico:

Para efetuarmos uma operação na calculadora primeiramente criaremos a função que faz o teclado numérico da calculadora funcionar.

```

/* função para o teclado numérico */
private void tecladoNumerico(int leNumero)
{
    if (txt_visor.getText().toString().trim().equals("0.0"))
    {
        limparVisor();
        txt_visor.setText(txt_visor.getText().toString()
            + String.valueOf(leNumero));
    } else {
        txt_visor.setText(txt_visor.getText().toString()
            + String.valueOf(leNumero));
    }
}

```

Inserir ponto, limpar visor e armazenar operações:

```

/* Função para inserção do ponto */
private void ponto(String ponto)
{
    txt_visor.setText(txt_visor.getText().toString()+ ponto);
}
/* Função para limpar o visor */
private void limparVisor()
{
    txt_visor.setText("");
}
/* Função que controla as operações */
private void operacoes(String operadores)
{
    this.operadores = operadores;

    if (operadores == "+")
    {

```

```

        operador1 =
        Double.parseDouble(
        txt_visor.getText().toString().trim());
        limparVisor();
    } else if (operadores == "-")
    {
        operador1 =
        Double.parseDouble(
        txt_visor.getText().toString().trim());
        limparVisor();
    } else if (operadores == "*")
    {
        operador1 =
        Double.parseDouble(
        txt_visor.getText().toString().trim());
        limparVisor();
    } else if (operadores == "/")
    {
        operador1 =
        Double.parseDouble(
        txt_visor.getText().toString().trim());
        limparVisor();
    }
}

```

Efetuar operações:

```

private void igual()
{
    if(!txt_visor.getText().toString().trim().equals(""))
    {
        if (operadores == "+")
        {
            operador2 = operador1 +
            Double.parseDouble(
            txt_visor.getText().toString().trim());
        } else if (operadores == "-")
        {
            operador2 = operador1 - Double.parseDouble(
            txt_visor.getText().toString().trim());
        } else if (operadores == "*")
        {
            operador2 = operador1 * Double.parseDouble(
            txt_visor.getText().toString().trim());
        } else if (operadores == "/")
        {
            if (operador1 == 0)
            {
                operador2 = 0;
            } else
            {
                operador2 = operador1 / Double.parseDouble(
                txt_visor.getText().toString().trim());
            }
        }
    } else
    {
        operador2 = 0;
    }
}

```

```

        txt_visor.setText(String.valueOf(operador2));
    }

```

Fechar aplicativo:

```

private void fecharAplicacao()
{
    AlertDialog.Builder dialog = new AlertDialog.Builder(this);
    dialog.setPositiveButton("Sim", new
        DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface arg0, int arg1) {
            Calculadora.this.finish();
        }
    });

    dialog.setNegativeButton("Não", new
        DialogInterface.OnClickListener() {

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    dialog.setTitle("Fechar");
    dialog.setMessage("Deseja realmente fechar a aplicação?");

    dialog.setIcon(android.R.drawable.ic_menu_close_clear_cancel);
    dialog.show();
}

```

Eventos dos componentes

Adicione um implements onClickListener a classe:

```

public class Calculadora extends Activity implements OnClickListener {

```

O nome da classe vai ficar sublinhado de vermelho, verifique o erro e note que ele pedirá para incluir métodos complementares, adicione esse método então aparecerá um novo método na classe chamado onClick implemente esse método da seguinte forma:

```

public void onClick(View v) {
    switch (v.getId()) {
        case id.bt_zero:
            tecladoNumerico(0);
            break;
        case id.bt_1:
            tecladoNumerico(1);
            break;
        case id.bt_2:
            tecladoNumerico(2);
            break;
        case id.bt_3:
            tecladoNumerico(3);
            break;
    }
}

```

```
    case id.bt_4:
        tecladoNumerico(4);
        break;
    case id.bt_5:
        tecladoNumerico(5);
        break;
    case id.bt_6:
        tecladoNumerico(6);
        break;
    case id.bt_7:
        tecladoNumerico(7);
        break;
    case id.bt_8:
        tecladoNumerico(8);
        break;
    case id.bt_9:
        tecladoNumerico(9);
        break;
    case id.bt_limpar:
        limparVisor();
        break;
    case id.bt_igual:
        igual();
        break;
    case id.bt_ponto:
        ponto(".");
        break;
    case id.bt_sair:
        fecharAplicacao();
        break;
    case id.bt_soma:
        operacoes("+");
        break;
    case id.bt_sub:
        operacoes("-");
        break;
    case id.bt_mult:
        operacoes("*");
        break;
    case id.bt_div:
        operacoes("/");
        break;
}
}
```