

## Objetivos:

- Introdução aos projetos em Android Studio.

## Regras de implementação:

- Criar aplicações móveis para Android.
- Implementar o código necessário e testar no fim de cada exercício.
- Use as convenções de codificação adoptadas habitualmente para a linguagem Java. Para os identificadores siga as convenções:
  - A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos.
  - A notação PascalCase para os nomes das classes.
  - Não utilize o símbolo '\_', nem abreviaturas nos identificadores.

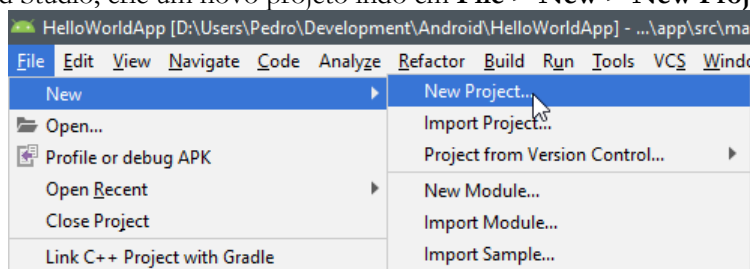
## Programa:

Construa várias apps móveis simples de exemplo.

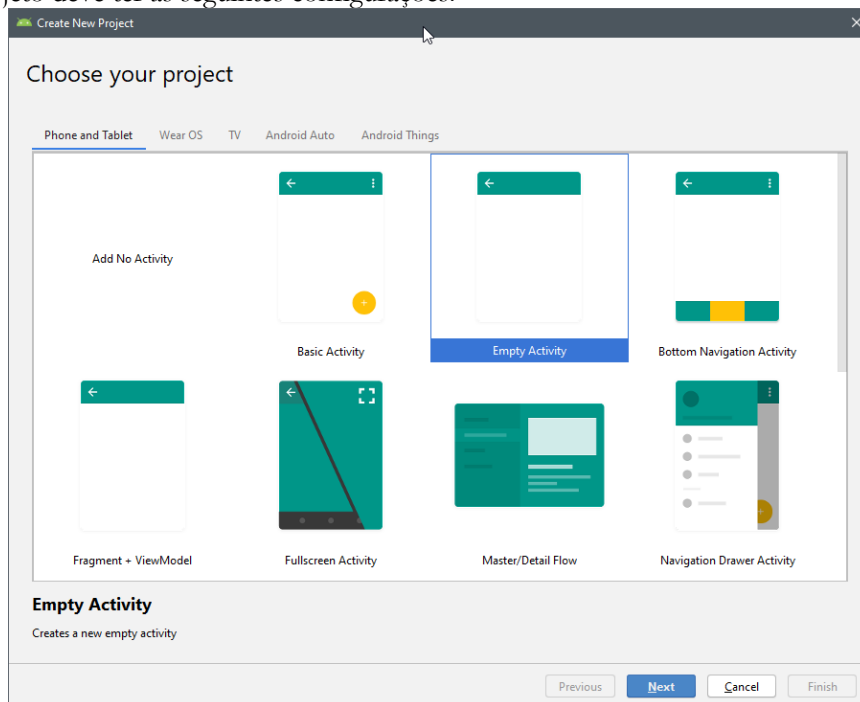
## Exercício 1:

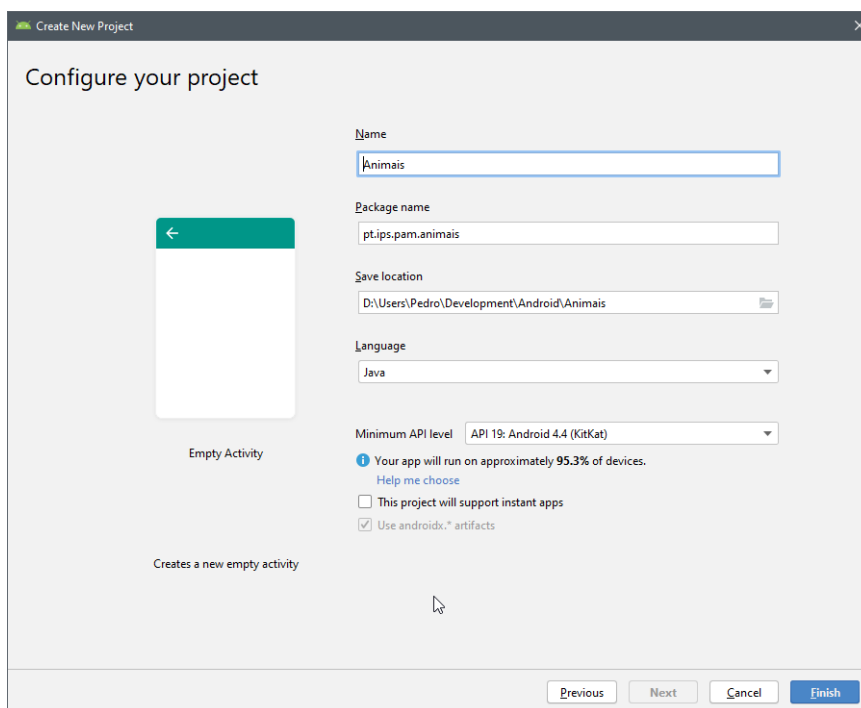
Implementação de uma pequena app que altera a imagem que nos é mostrada quando o utilizador toca nos botões com o nome de um animal.

1. No Android Studio, crie um novo projeto indo em **File > New > New Project...**

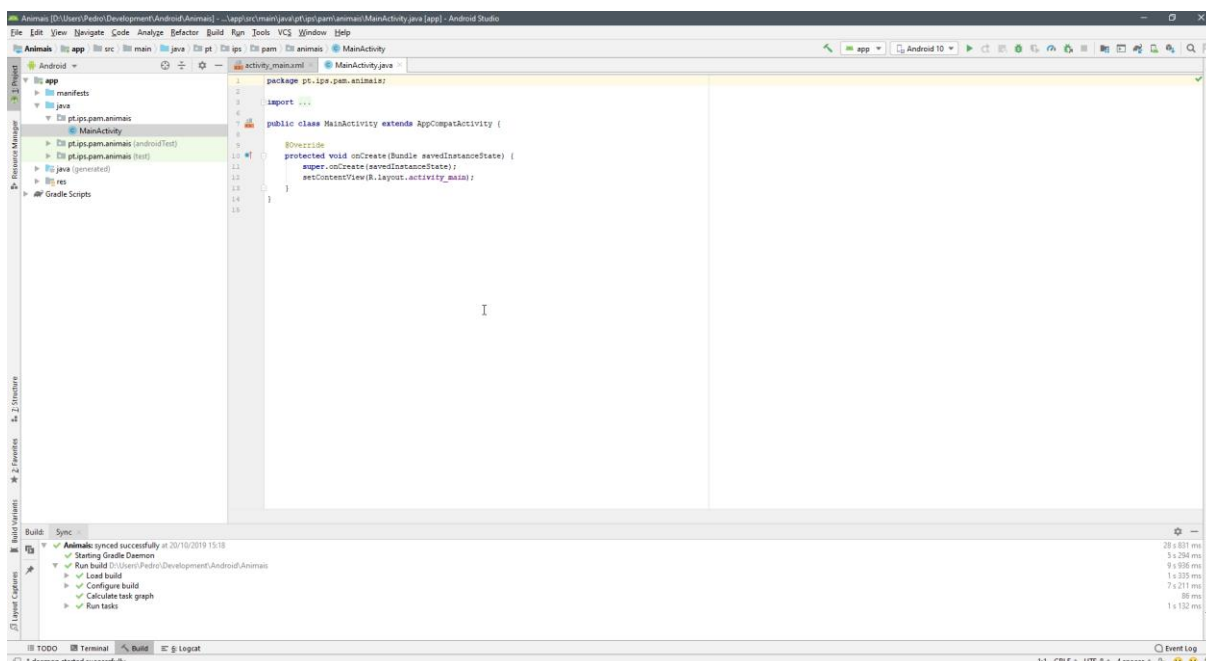


2. O projeto deve ter as seguintes configurações:

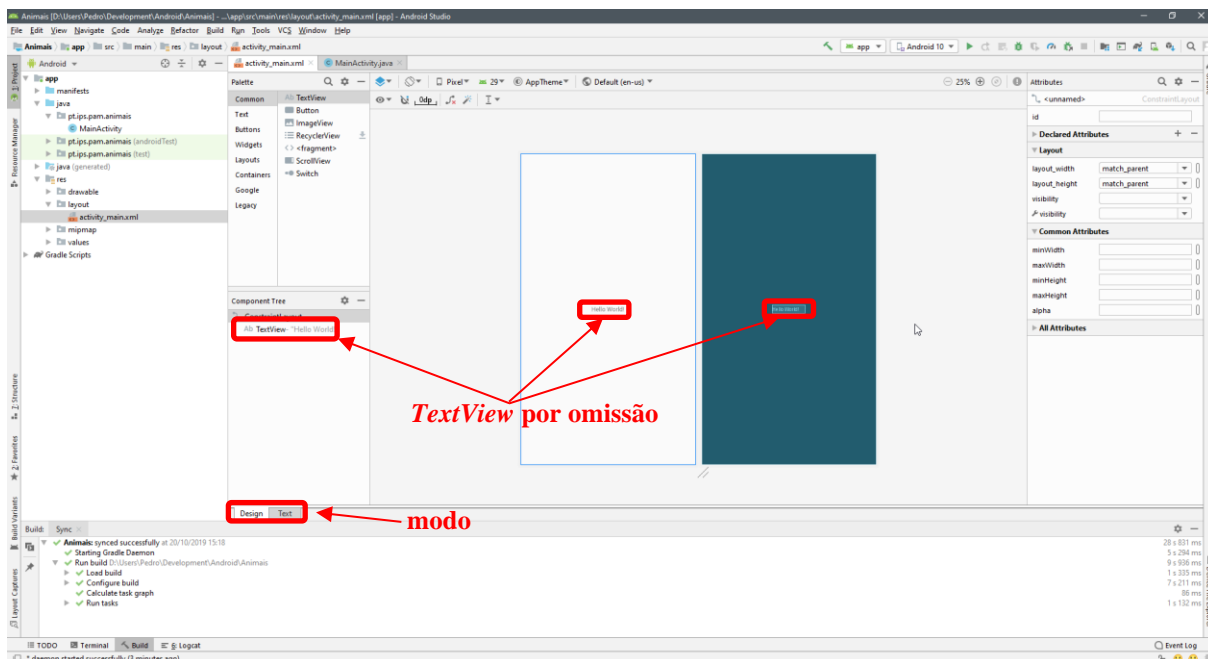




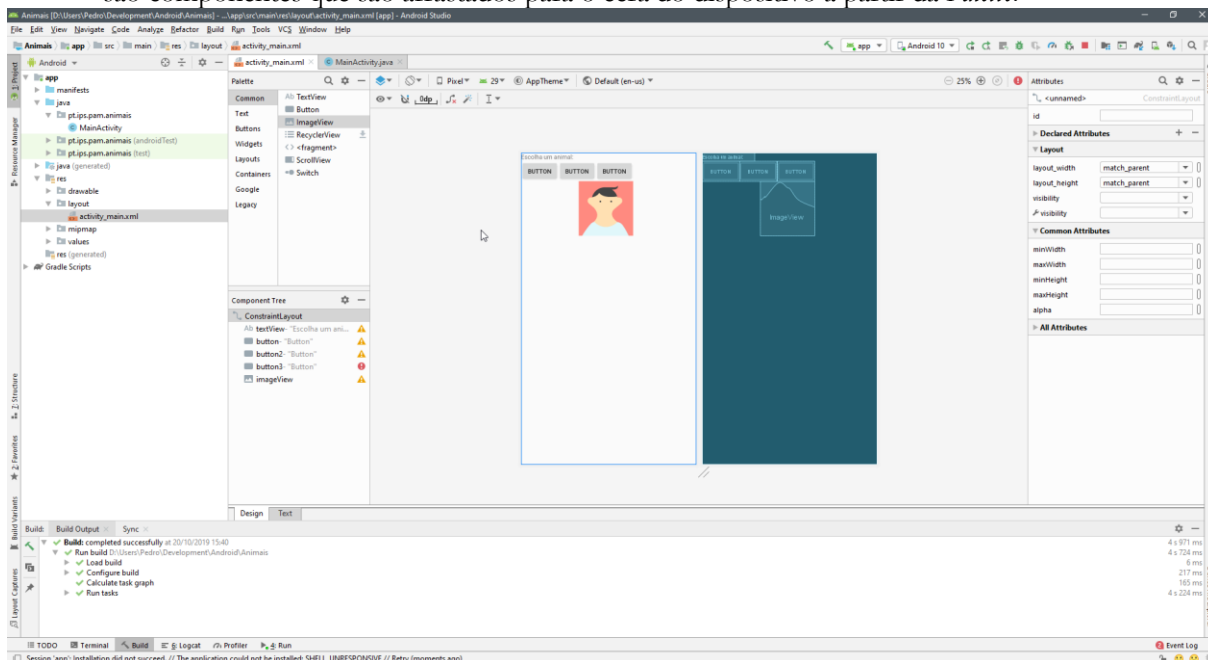
3. O projeto abre com a classe *MainActivity* no editor do lado direito. De seguida, abrir o ficheiro “activity\_main.xml” fazendo duplo clique sobre ele na árvore de projeto que fica no lado esquerdo.



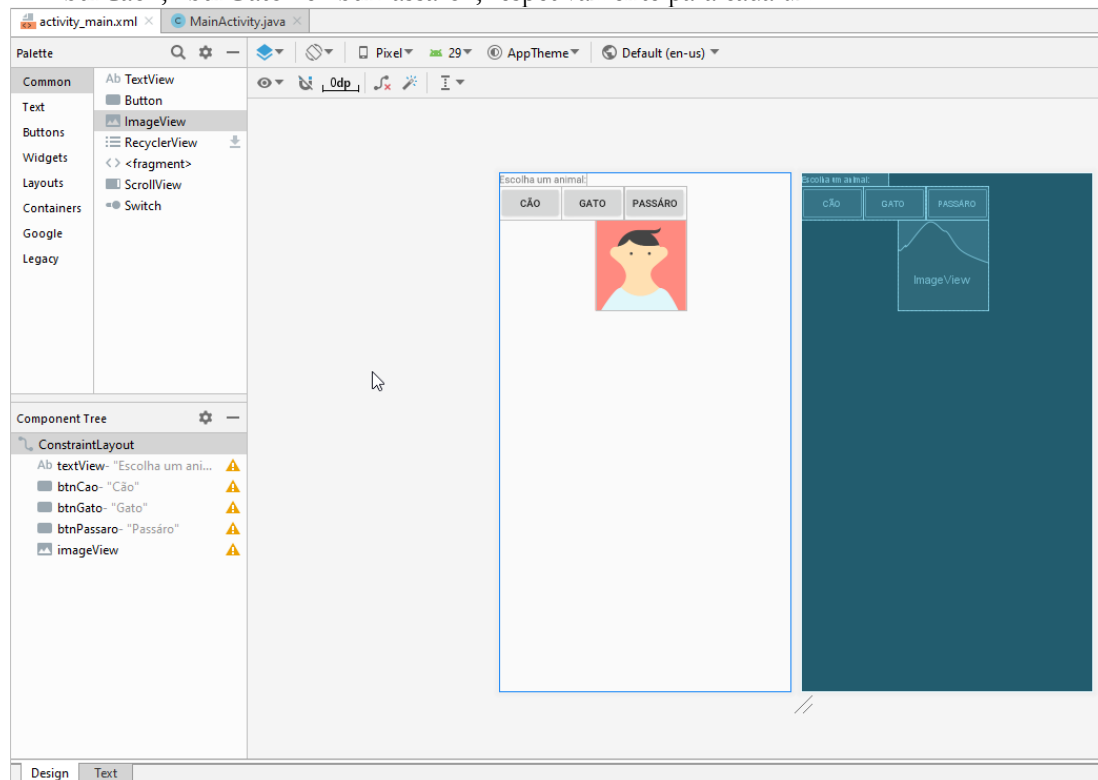
4. O ficheiro abre em modo *Design* (podemos ver também o *Text*) e com um Texto (numa *TextView*) “Hello World” por omissão num layout *Constraint*.



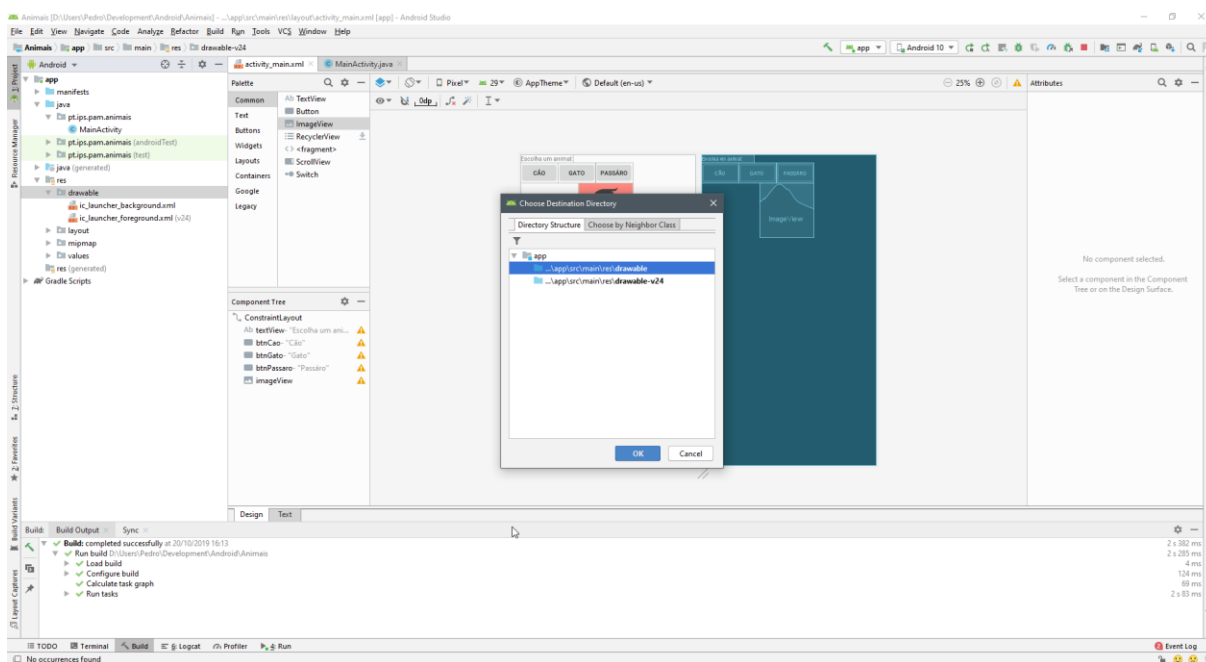
5. Mantendo o *ConstraintLayout*, alterar o texto da *TextView* por omissão para “Escolha um Animal:” e reposicione-a de acordo com a figura abaixo. Adicione 3 *Buttons* ao lado uns dos outros e por baixo do texto. Adicione também uma *ImageView* por baixo dos 3 botões. Os *Buttons* e a *ImageView* são componentes que são arrastados para o ecrã do dispositivo a partir da *Palette*.



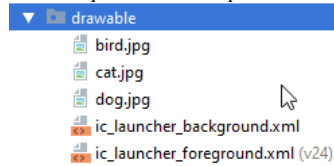
6. Clique 2 vezes sobre cada *Button* e, em *text*, escreva “Cão”, “Gato” e “Pássaro” e em *id* escreva “btnCao”, “btnGato” e “btnPassaro”, respetivamente para cada um.



7. Copiar os ficheiros fornecidos para os 3 tipos de animais (“dog.jpg”, “cat.jpg”, “bird.jpg”) no explorador do Windows e colá-los no Android Studio na pasta “res/drawable”.



8. Deverá então ter os 3 ficheiros disponíveis na pasta “res/drawable”:



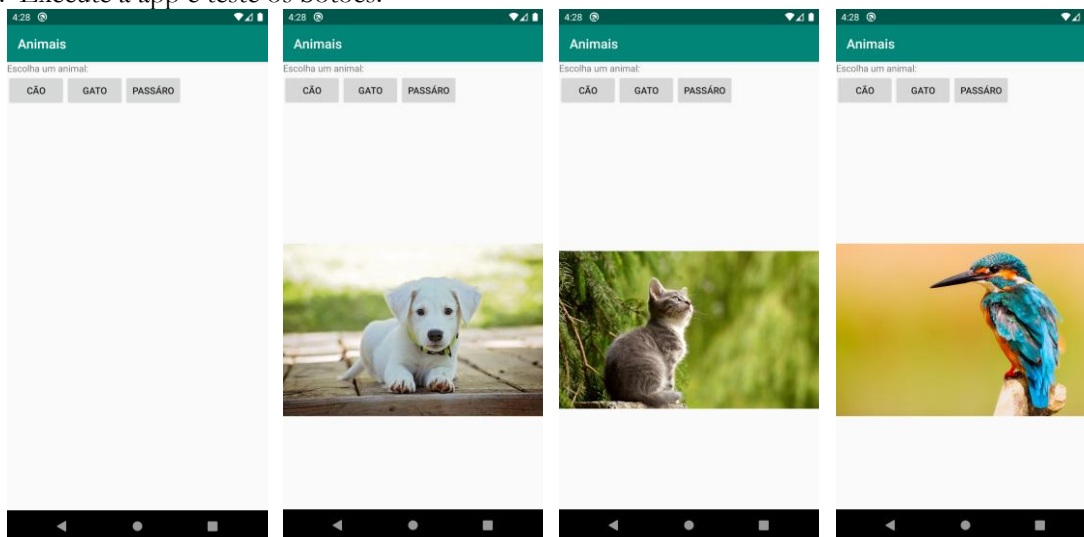
9. Escreva o seguinte código na classe MainActivity.

```

1  package pt.ips.pam.animais;
2
3  import ...
4
5
6
7
8
9
10 public class MainActivity extends AppCompatActivity {
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         Button btnCao = (Button) findViewById(R.id.btnCao);
17         Button btnGato = (Button) findViewById(R.id.btnGato);
18         Button btnPassaro = (Button) findViewById(R.id.btnPassaro);
19         final ImageView imageView = (ImageView) findViewById(R.id.imageView);
20
21         btnCao.setOnClickListener(new View.OnClickListener() {
22             @Override
23             public void onClick(View v) {
24                 imageView.setImageResource(R.drawable.dog);
25             }
26         });
27
28         btnGato.setOnClickListener(new View.OnClickListener() {
29             @Override
30             public void onClick(View v) {
31                 imageView.setImageResource(R.drawable.cat);
32             }
33         });
34
35         btnPassaro.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 imageView.setImageResource(R.drawable.bird);
39             }
40         });
41     }
42 }

```

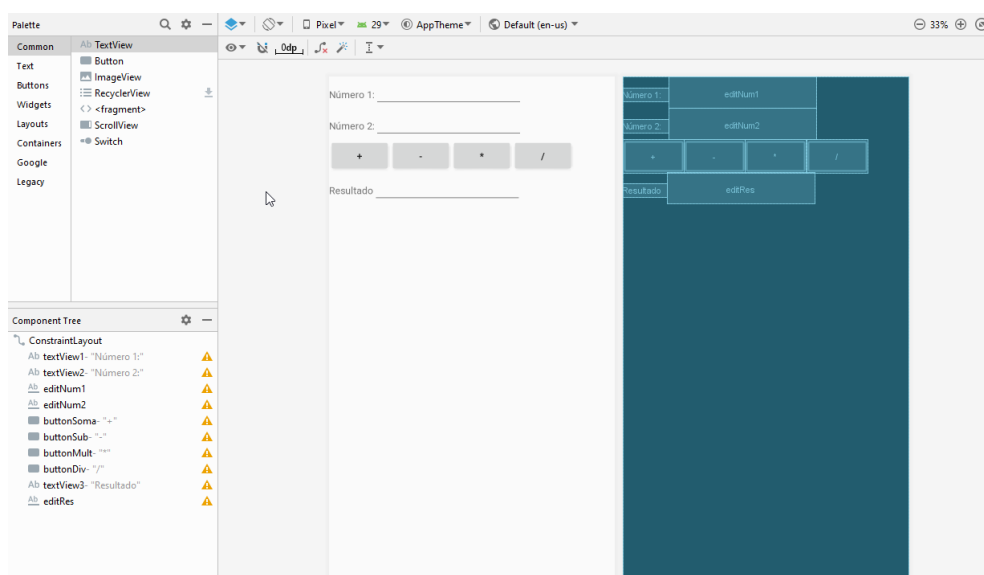
10. Execute a app e teste os botões.



## Exercício 2:

Implementação de uma calculadora básica usando componentes Button e EditText.

1. No Android Studio, crie um novo projeto...
2. Escolha as configurações de acordo com o exercício anterior, mas atribuindo o nome “CalculadoraBasica” à aplicação.
3. De seguida, abrir o ficheiro “activity\_main.xml” fazendo duplo clique sobre ele na árvore de projeto que fica no lado esquerdo e adicionar os componentes de acordo com a imagem seguinte.



4. Escreva o seguinte código na classe “MainActivity”.

```

10 public class MainActivity extends AppCompatActivity {
11     private EditText num1;
12     private EditText num2;
13     private EditText res;
14     private double valor1;
15     private double valor2;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         num1 = (EditText) findViewById(R.id.editNum1);
22         num2 = (EditText) findViewById(R.id.editNum2);
23         res = (EditText) findViewById(R.id.editRes);
24         res.setFocusable(false);
25         Button buttonSoma = (Button) findViewById(R.id.buttonSoma);
26         buttonSoma.setOnClickListener(new View.OnClickListener() {
27             @Override
28             public void onClick(View v) {
29                 if(MainActivity.this.updateValores()) {
30                     res.setText(String.valueOf(valor1+valor2));
31                 }
32             }
33         });
34         Button buttonSub = (Button) findViewById(R.id.buttonSub);
35         buttonSub.setOnClickListener(new View.OnClickListener() {
36             @Override
37             public void onClick(View v) {
38                 if(MainActivity.this.updateValores()) {
39                     res.setText(String.valueOf(valor1-valor2));
40                 }
41             }
42         });
43         Button buttonMult = (Button) findViewById(R.id.buttonMult);
44         buttonMult.setOnClickListener(new View.OnClickListener() {
45             @Override
46             public void onClick(View v) {
47                 if(MainActivity.this.updateValores()) {
48                     res.setText(String.valueOf(valor1*valor2));
49                 }
50             }
51         });
52         Button buttonDiv = (Button) findViewById(R.id.buttonDiv);
53         buttonDiv.setOnClickListener(new View.OnClickListener() {
54             @Override
55             public void onClick(View v) {
56                 if(MainActivity.this.updateValores()) {
57                     if(valor2 == 0.0) {
58                         res.setText("Erro! Divisão por zero!");
59                     } else {
60                         res.setText(String.valueOf(valor1/valor2));
61                     }
62                 }
63             }
64         });
65     }

```

5. E acrescente o seguinte método à classe “MainActivity”.

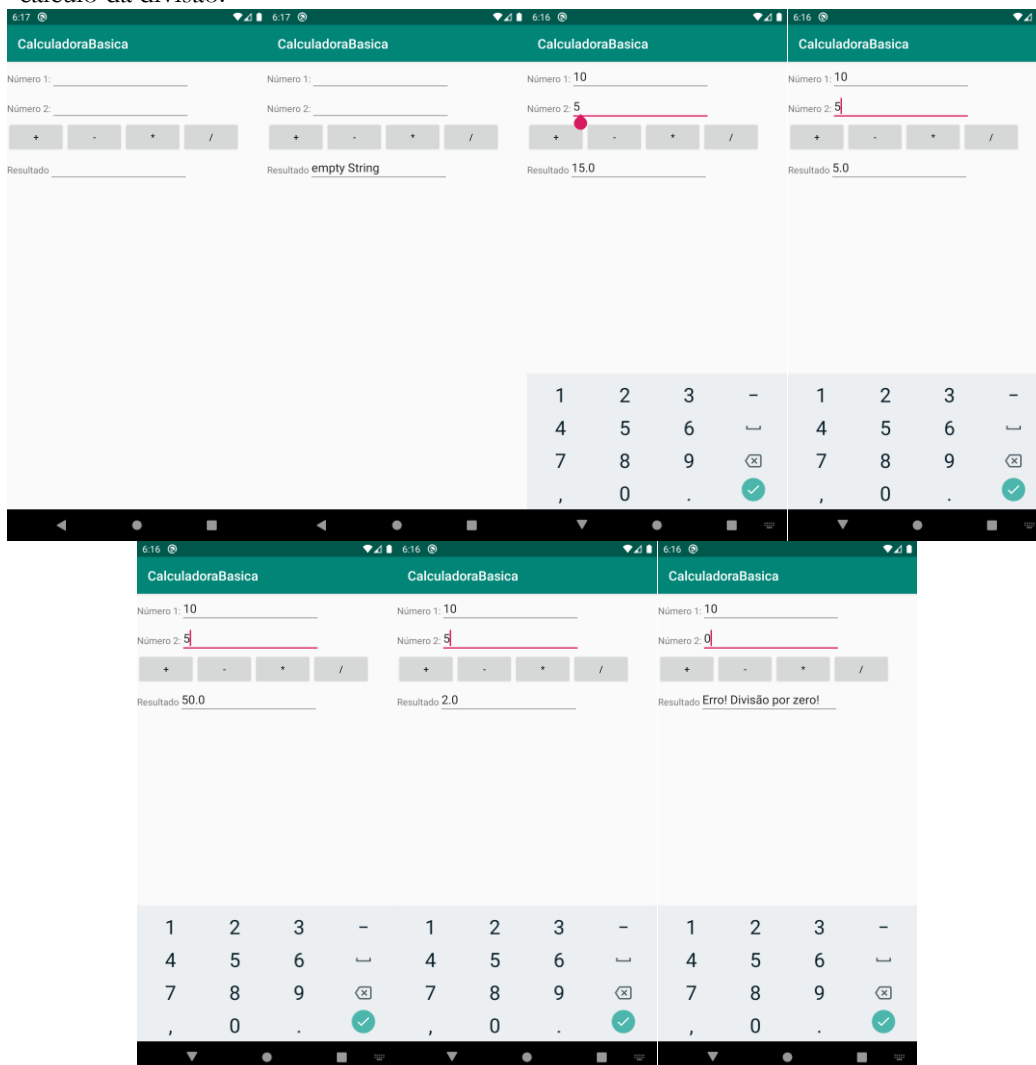
```

67 private boolean updateValores() {
68     try {
69         valor1 = Double.parseDouble(num1.getText().toString());
70         valor2 = Double.parseDouble(num2.getText().toString());
71         return true;
72     } catch (Exception ex) {
73         res.setText(ex.getMessage());
74         return false;
75     }
76 }
77 }

```



6. Execute a app e teste as várias operações, incluindo atribuir zero ao segundo número para o cálculo da divisão.



### Exercício 3:

Alteração da app do exercício anterior acrescentando um Toast para mostrar a mensagem de erro na divisão por zero e nos casos em que não seja possível decifrar o input do utilizador.

1. No projeto do exercício anterior, substituir a linha 58 pelas linhas 58 e 59 da seguinte figura.

```

53 Button buttonDiv = (Button) findViewById(R.id.buttonDiv);
54 buttonDiv.setOnClickListener(new View.OnClickListener() {
55     @Override
56     public void onClick(View v) {
57         if(MainActivity.this.updateValores()) {
58             if(valor2 == 0.0) {
59                 res.setText("");
60                 Toast.makeText(context, MainActivity.this, text: "Erro! Divisão por zero!", Toast.LENGTH_SHORT).show();
61             } else {
62                 res.setText(String.valueOf(valor1/valor2));
63             }
64         }
65     }
66 });

```

2. Substituir também a linha 73 pelas linhas 75 e 76 da seguinte figure.

```

69 private boolean updateValores() {
70     try {
71         valor1 = Double.parseDouble(num1.getText().toString());
72         valor2 = Double.parseDouble(num2.getText().toString());
73         return true;
74     } catch(Exception ex) {
75         res.setText("");
76         Toast.makeText(context, MainActivity.this, ex.getMessage(), Toast.LENGTH_SHORT).show();
77         return false;
78     }
79 }

```

3. Execute a app e teste a atribuição de zero ao segundo número para o cálculo da divisão, bem como deixar os campos em branco antes de tentar executar uma operação.

