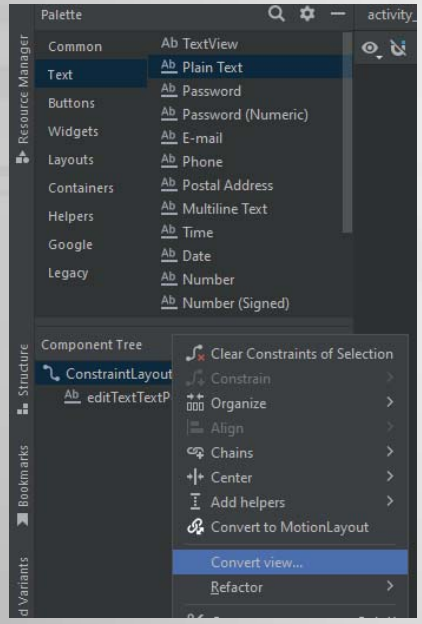
Componentes de Interface

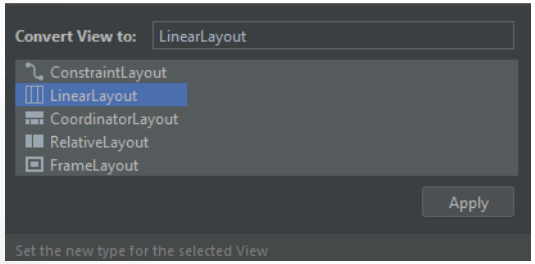
Professor: Hércules Santhus

Configurando Layout

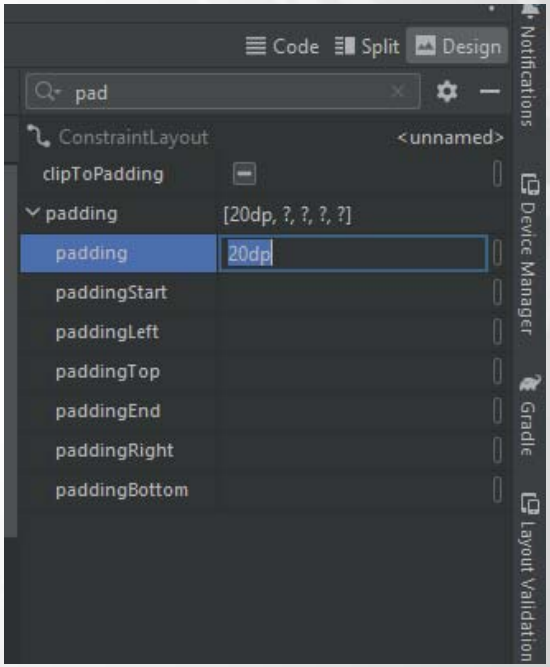
Antes de add qualquer conponente



1. Click em **ContrintLayout** com botão direito e escolha **Convert view**

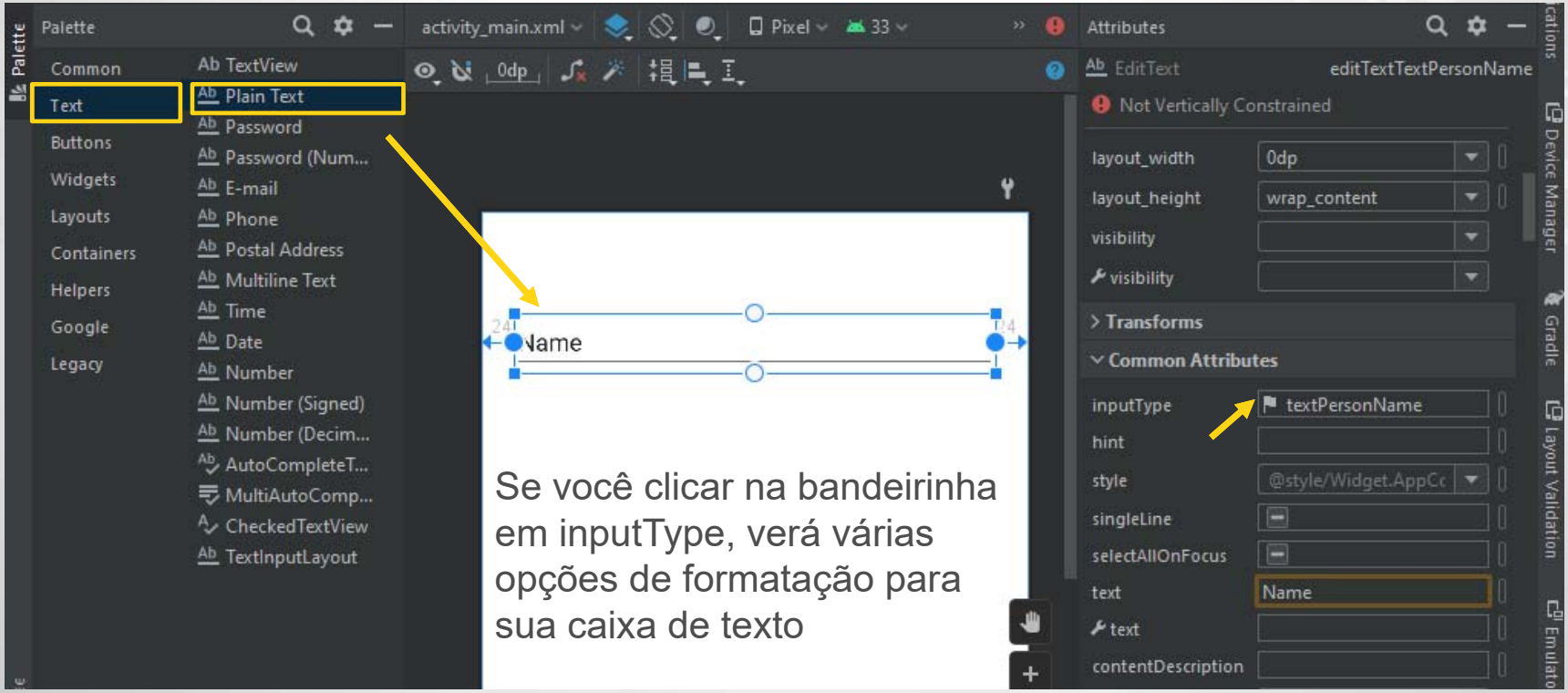


2. Escolha a opção **LinearLayout**

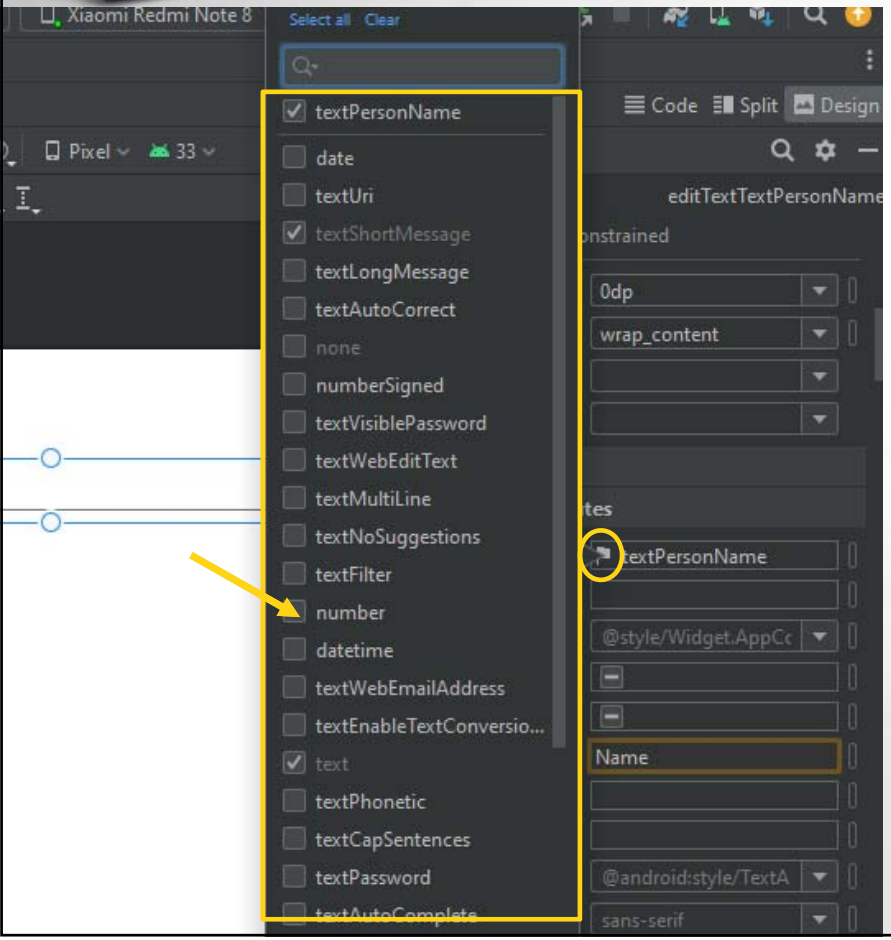


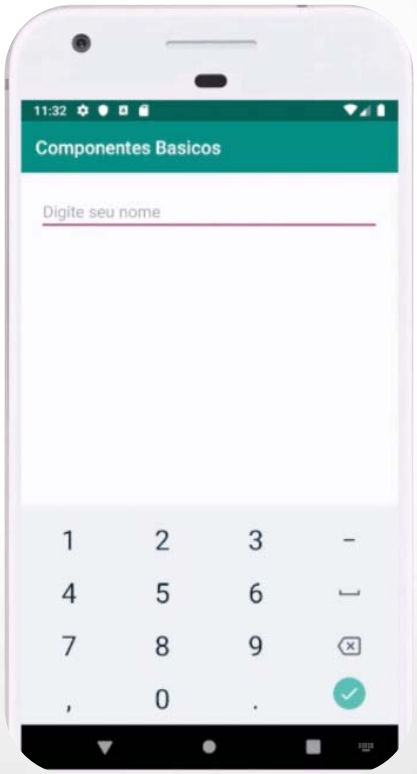
3. Configure um espaçamento nas laterais (**padding**) de **20dp**

Caixas de texto



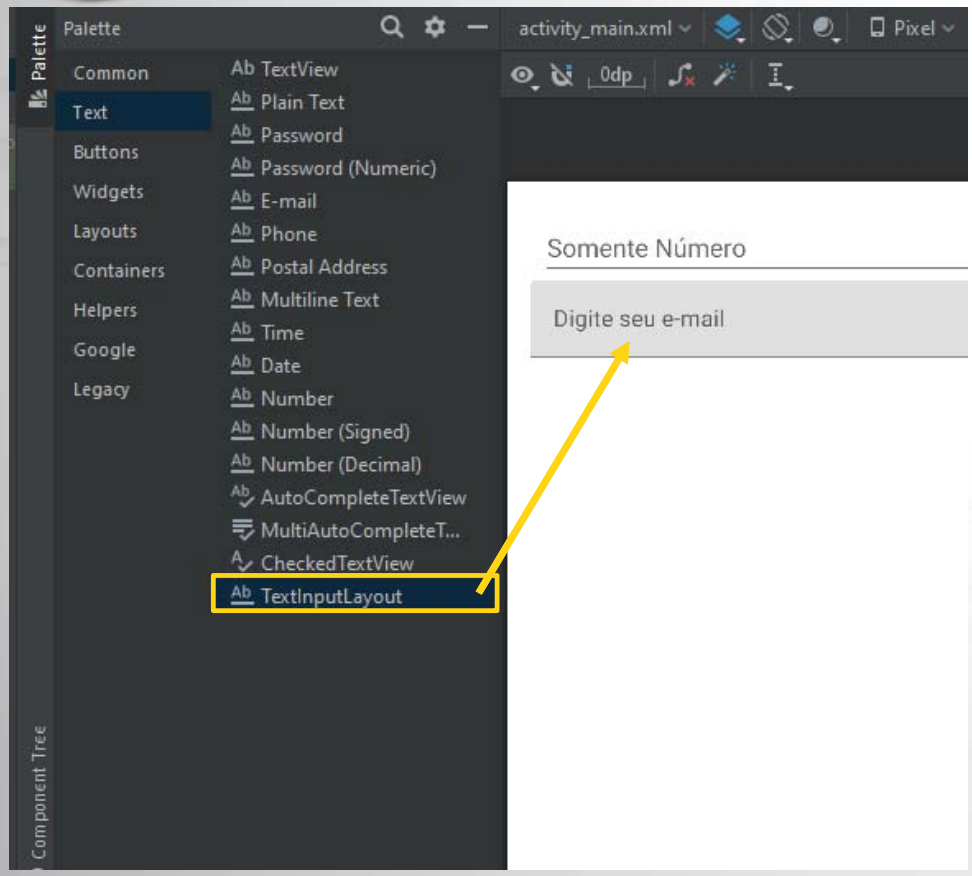
Tipos de caixas

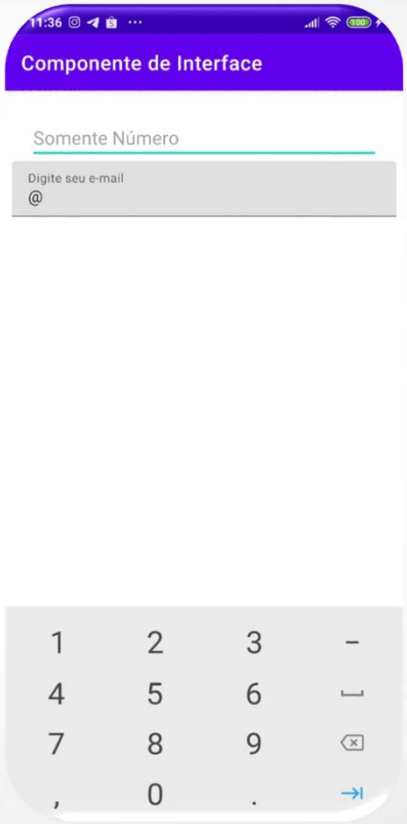


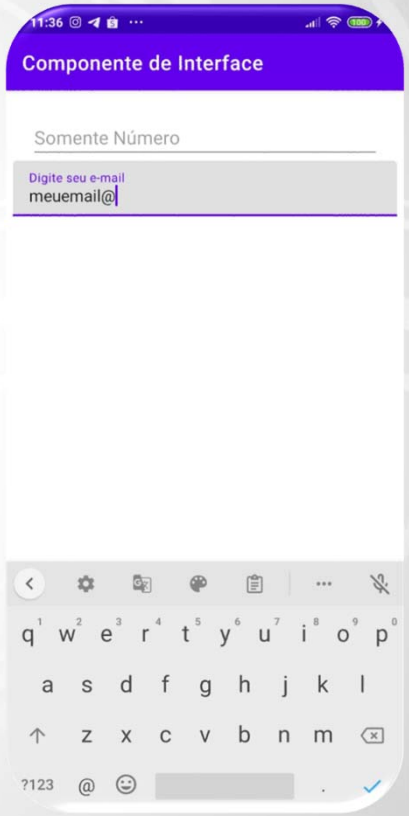


Se você escolher por exemplo **number**, a sua caixa de texto só permitrá números.
Ou seja, o teclado se adapta ao formato da caixa de texto.

TextInputLayout







Veja que o teclado se adapta à caixa de texto. No caso do TextInputLayout, o init permanece logo acima da caixa

Capturando valores dos componentes

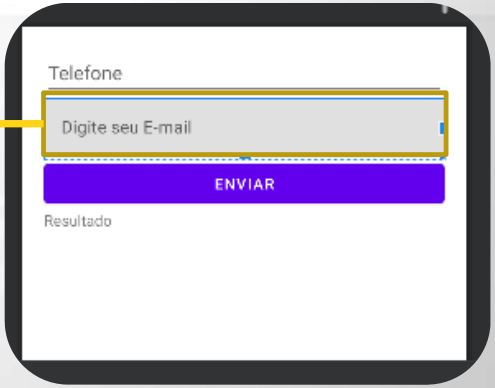
Tipo do componente

Nome do objeto

id do componente

```
EditText campoEmail = findViewById(R.id.editTextEmail);

String email = campoEmail.getText().toString();
```

Variável String que recebe o valor do componente



Setando valores ao componente

Tipo do componente

Nome do objeto

id do componente

```
TextView resultado = findViewById(R.id.textViewResultado);

resultado.setText(email);
```

Variável String que recebeu o valor do componente

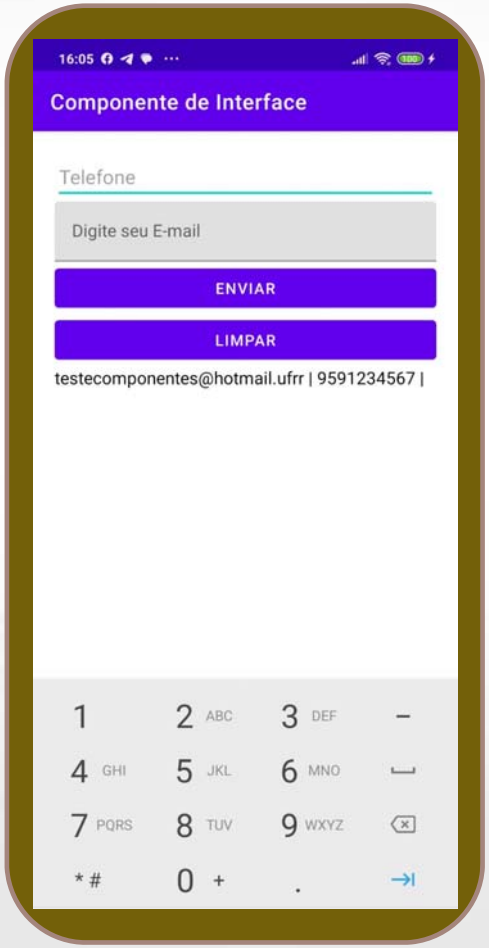
Método que inseri o valor ao componente

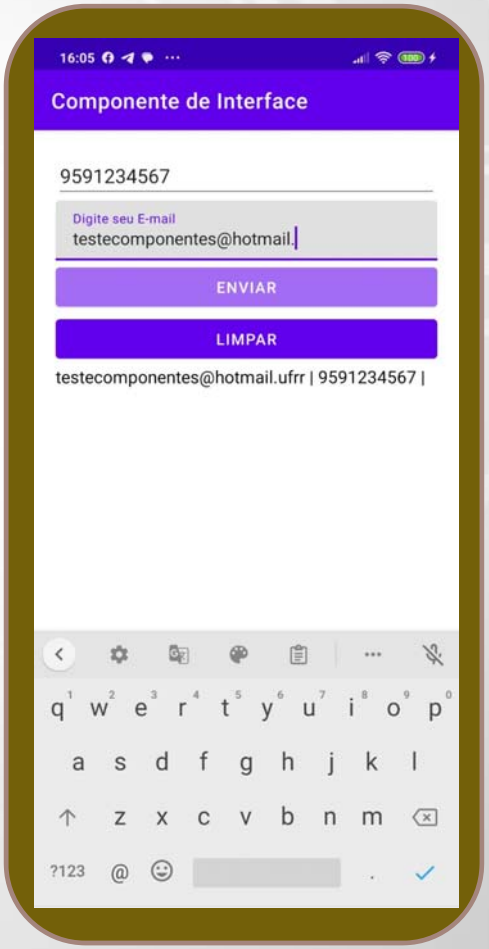
Atividade

Crie uma interface que receba o número de telefone e email.

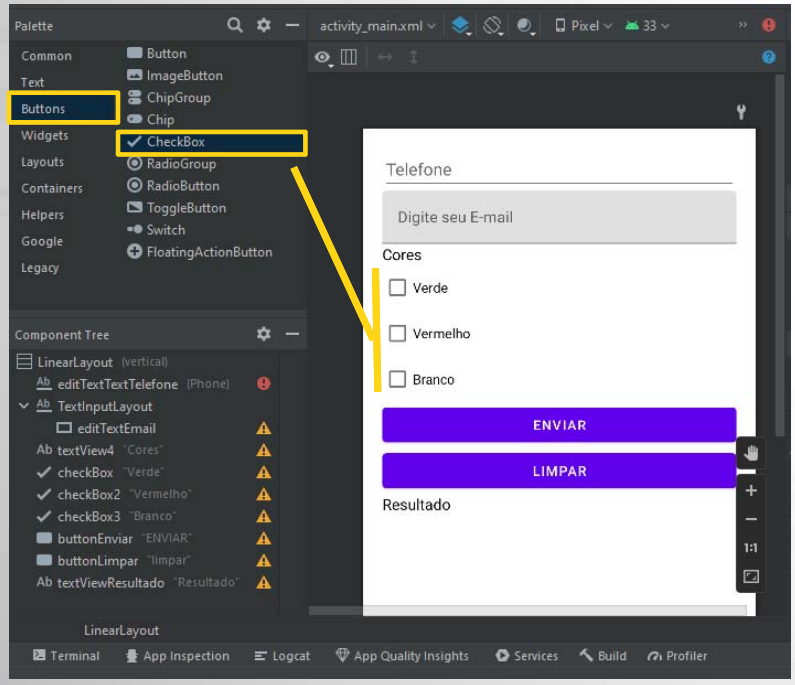
Ao clicar no botão enviar os valores da caixa de texto devem ser carregados no textView a baixo dos botões, conforme a imagem.

Obs: As caixas de texto devem está configurada para receber os tipos de dados específicos





CheckBox



Para verificar se um **CheckBox** foi selecionado, você deve usar o método **isChecked**( ).
Ele irá retornar um valor Boleano. **True** ou **False**

```
CheckBox checkVerde = findViewById(R.id.checkBoxVerde);
```

```
if (checkVerde.isChecked()){

    String corSelecionada = checkVerde.getText().toString();
    resultado.setText(corSelecionada);

}
```

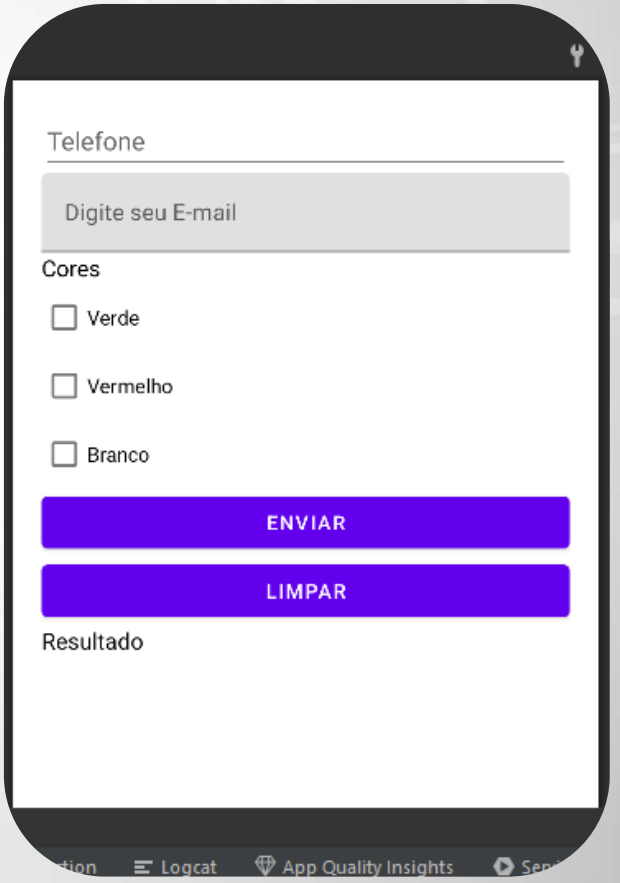
O **CheckBox** permite que você marque mais de uma opção.

Pratica

```
private CheckBox checkVerde, getCheckVermelho, getCheckBranco;
```

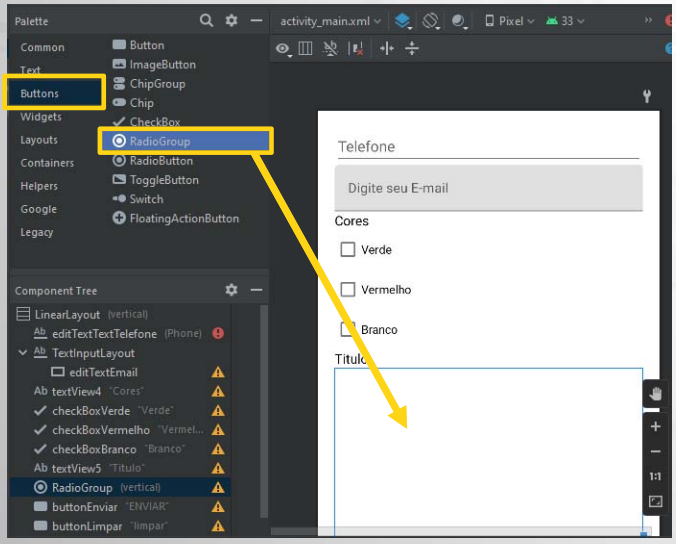
```
checkVerde = findViewById(R.id.checkBoxVerde);
getCheckVermelho = findViewById(R.id.checkBoxVermelho);
getCheckBranco = findViewById(R.id.checkBoxBranco);
```

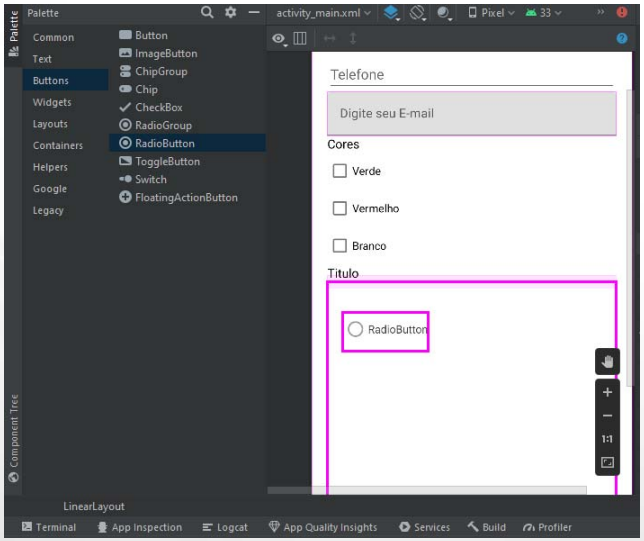
```
public void verificar_check(){

    String texto = "";

    if (checkVerde.isChecked()){
        texto = "Verde Selecionado";
    }
    if (getCheckVermelho.isChecked()){
        texto = texto + "Vermelho Selecionado"; //concatena caso outro checkBox seja selecionado
    }
    if (getCheckBranco.isChecked()){
        texto = texto + "Branco Selecionado";
    }
    resultado.setText(texto);
}
```

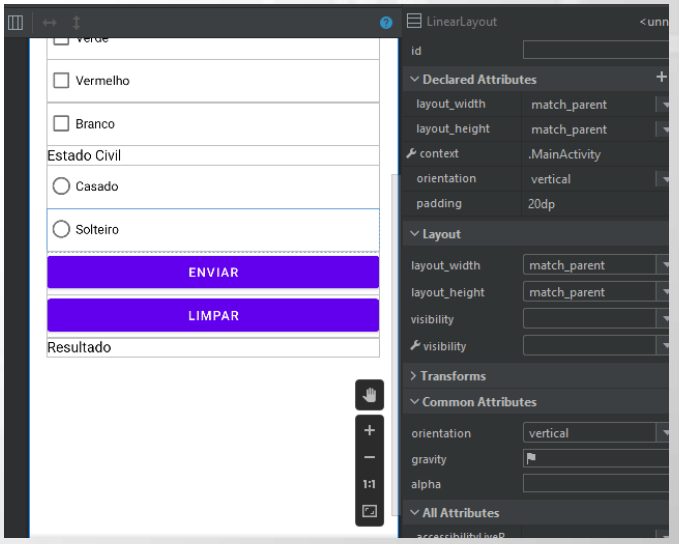


RadioGroup e RadioButton

Diferente do **CheckBox**, o **RadioButton** permite apenas um item selecionado. Para adicionar mais de um **RadioButton**, é necessário adicionar primeiramente o **RadioGroup**, em seguida adicionar as opções necessárias





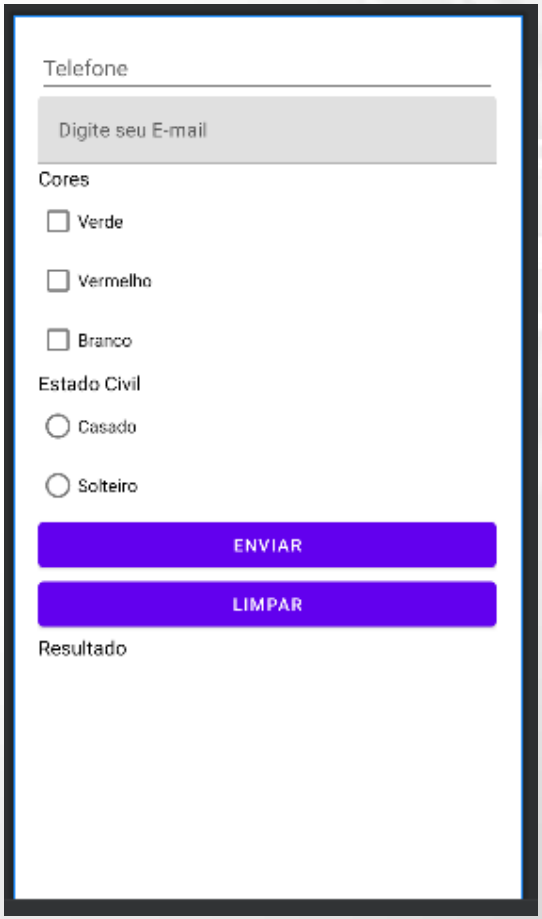


Prática

```
private RadioButton RadioCasado, RadioSolteiro;
```

```
RadioSolteiro = findViewById(R.id.radioButtonSolteiro);
RadioCasado = findViewById(R.id.radioButtonCasado);
```

```
public void radioEstadoCivil (){

    if(RadioCasado.isChecked()){
        resultado.setText("Casado");
    }else if (RadioSolteiro.isChecked()){
        resultado.setText("Solteiro");
    }

}
```



Adicionando ChangeListener

```
private RadioGroup opcaoEstadoCivil;
```

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

    radioEstadoCivil();
```

Podemos utilizar o **RadioGroup** para saber qual opção o usuário digitou, sem precisar usar o **RadioButton.**

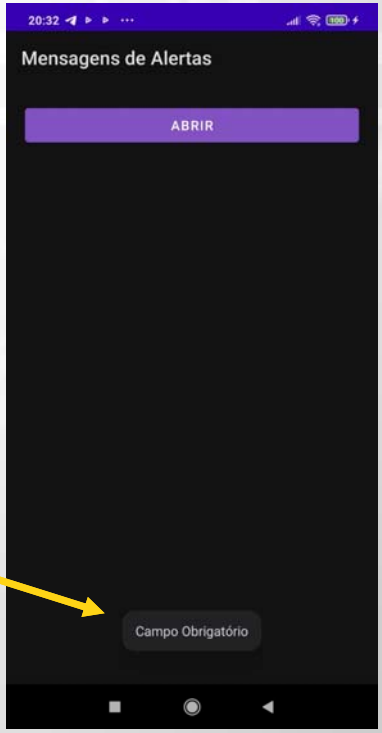
Dessa forma basta criarmos ouvinte para que o App detecte automaticamente ao usuário clicar na opção, sem precisar usar o botão ENVIAR. Para isso precisamos chamar o método dentro do **onCreate** para que seja carregado antes dda interface ser construída para o usuário.

```
    public void radioEstadoCivil() {
        opcaoEstadoCivil.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(RadioGroup radioGroup, int i) {
                if (i == R.id.radioButtonCasado) {
                    resultado.setText("Casado");
                } else if (i == R.id.radioButtonSolteiro) {
                    resultado.setText("Solteiro");
                }
            }
        });
}
```

Toast

Exibe uma mensagem, quando o usuário faz uma ação e depois de um tempo a mensagem desaparece automaticamente

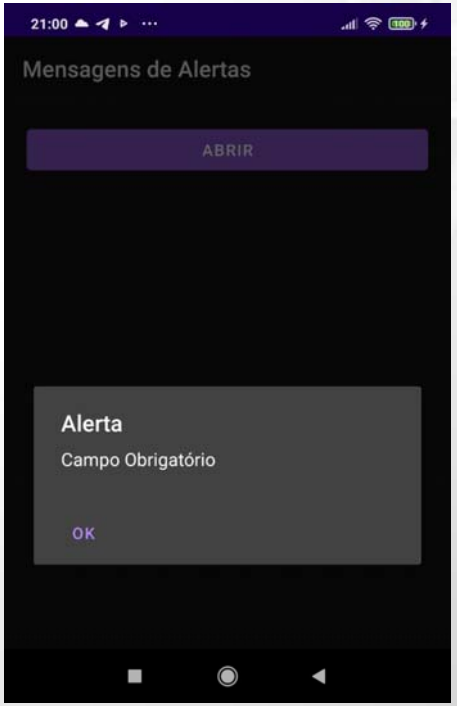
```
public void abrirToast (View v){

    Toast.makeText(this, "Campo Obrigatório", Toast.LENGTH_SHORT).show();

}
```



AlertDialog

Outra forma de apresentar uma mensagem ao usuário é por meio do **AlertDialog**

```
public void abrirAlerta (View v){

    AlertDialog.Builder dialogo = new AlertDialog.Builder(this);
    dialogo.setTitle("Alerta");
    dialogo.setMessage("Campo Obrigatório");
    dialogo.setNeutralButton("ok", null);
    dialogo.show();

}
```



Atividade

Criar um App de combustível que calcule a melhor opção de abastecimento para o usuário, se é melhor usar álcool ou gasolina.

Se (valor_do_álcool / valor_da_gasolina < 0.7) é melhor usar gasolina

O campo valores é obrigatório