

FIAP

# ConstraintLayout e Intents

## Hybrid Mobile App Development



Prof. Vinny Albuquerque [[profvinny.albuquerque@fiap.com.br](mailto:profvinny.albuquerque@fiap.com.br)]

## Layouts

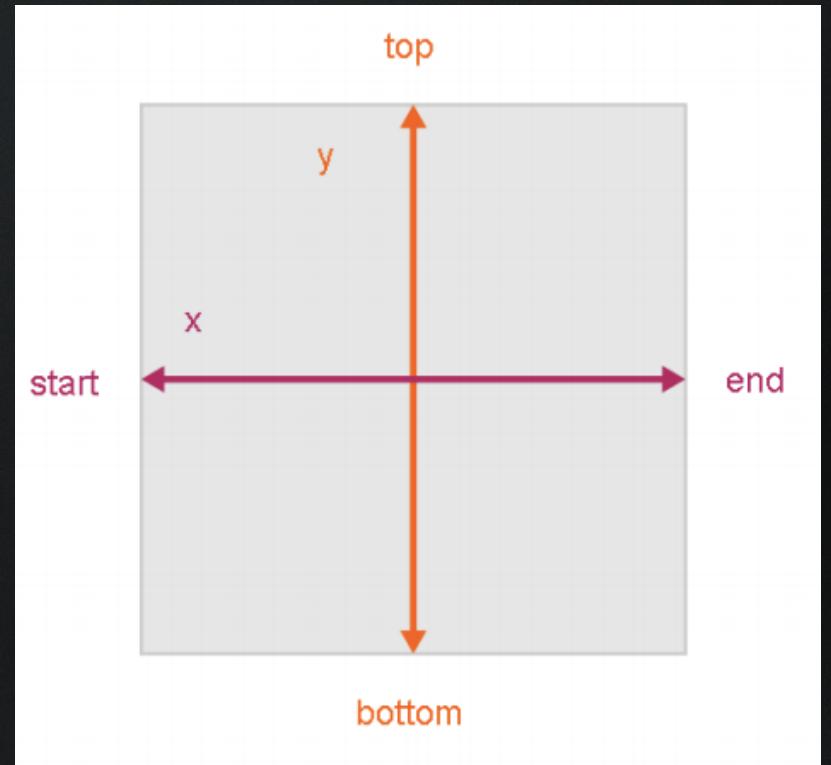
- Os layouts em Android consistem em práticas importantes para o desenvolvimento de telas responsivas e adequadas para os dispositivos dos usuários.
- Existem diferentes abordagens de desenvolvimento de layouts as quais trazem vantagens e desvantagens para o desenvolvedor.
- Não existe um layout melhor ou pior, mas aquele mais adequado às habilidades do desenvolvedor para implementar tais funcionalidades.

- O ConstraintLayout é o mais recente dos layouts disponíveis para uso no SDK do Android e consiste em configurar as Views e Widgets de acordo com restrições de posicionamento em relação aos elementos adjacentes.
- A recomendação é utilizar o ConstraintLayout em modo Design, ou seja, utilizando a interface gráfica para modelar as telas do aplicativo.

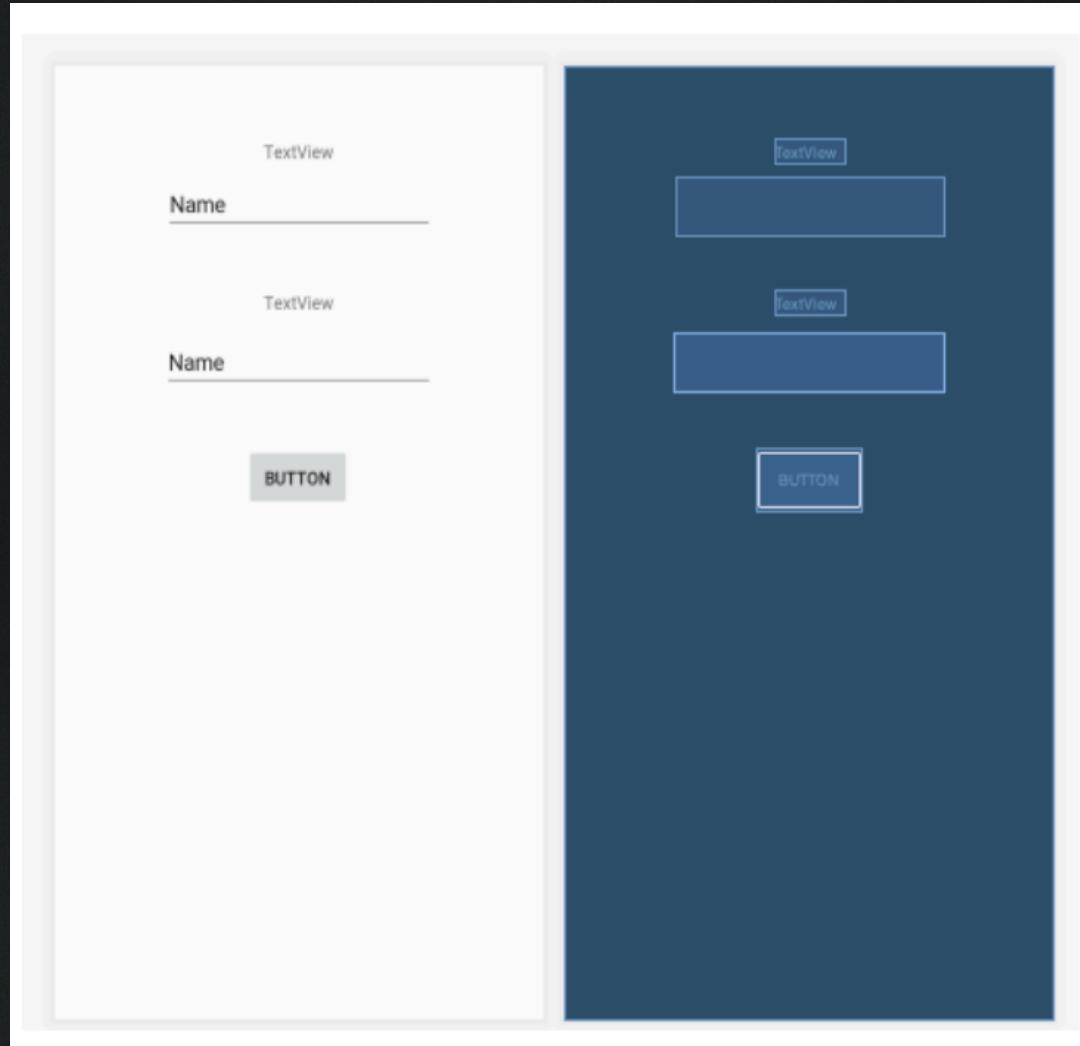
## ConstraintLayout

É importante entender, para qualquer layout, os eixos para dimensionamento dos objetos.

A vantagem no uso do ConstraintLayout é a definição prévia do posicionamento dos objetos de tela em modo gráfico (arquivo activity\_main.xml, aba Design) e, depois, a definição de restrições para o posicionamento de acordo com as necessidades do usuário.



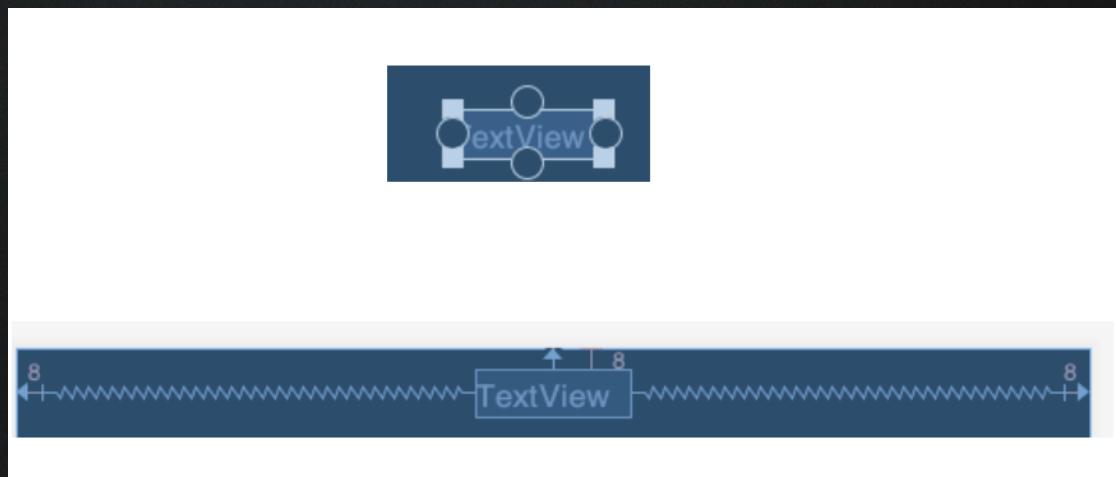
- **Depois de distribuídos os objetos, deve ser definido as constraints para cada objeto.**
- **O que deve ser avaliado na aplicação das restrições é quais dimensões precisam ser fixas e quais outras necessitam ser relativas ao tamanho de tela.**



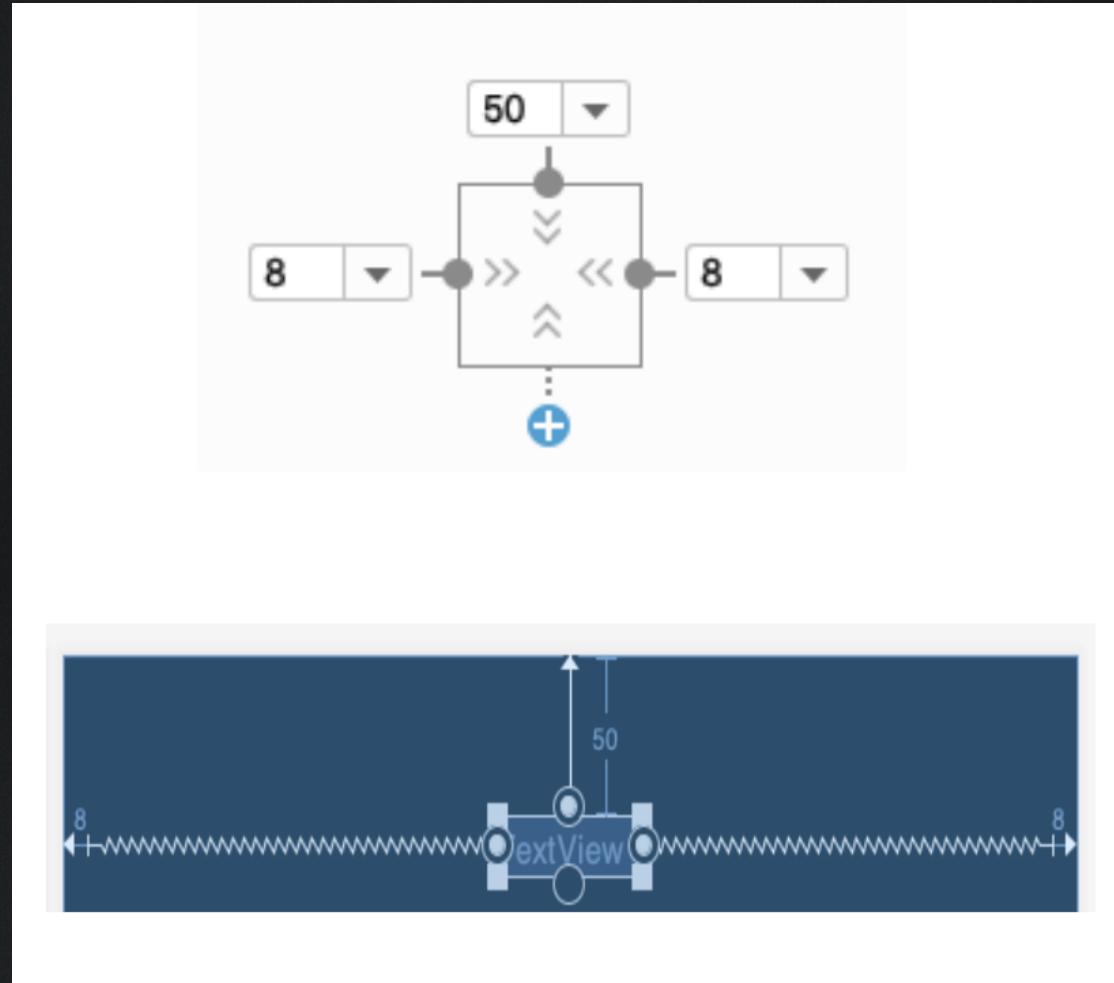
## ConstraintLayout

A dica para configurar cada objeto é:

- **Ajustar as “forças” para as dimensões vertical e horizontal.**
- **Depois, caso necessário definir dimensões absolutas entre os objetos (ou entre o objeto e a borda).**



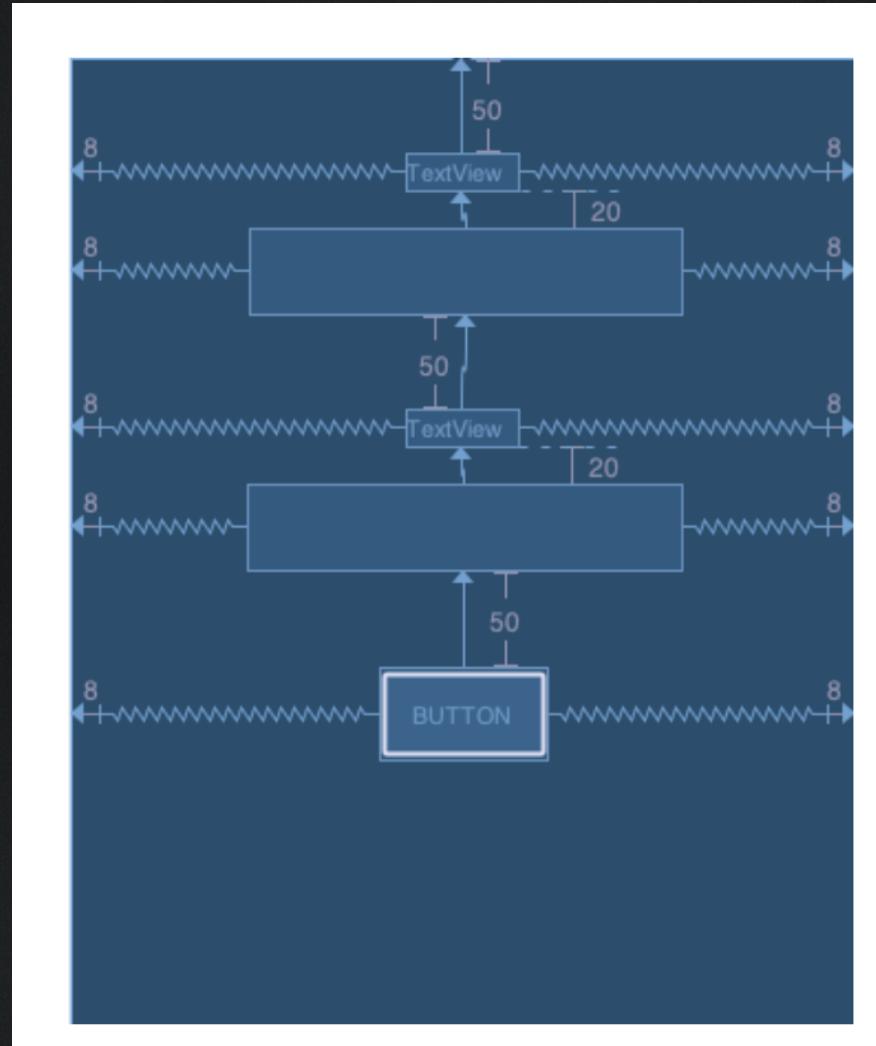
- Quando um objeto é selecionado são apresentadas as propriedades do Layout na área Attributes, no qual as margens para cada elemento pode ser ajustada.



## ConstraintLayout

FIAP

- Layout configurado após ajuste de forças e margens de cada elemento de tela.



## Qual combustível é mais vantajoso?

FIAP

Vamos desenvolver um aplicativo para calcular qual combustível é vantajoso abastecer:

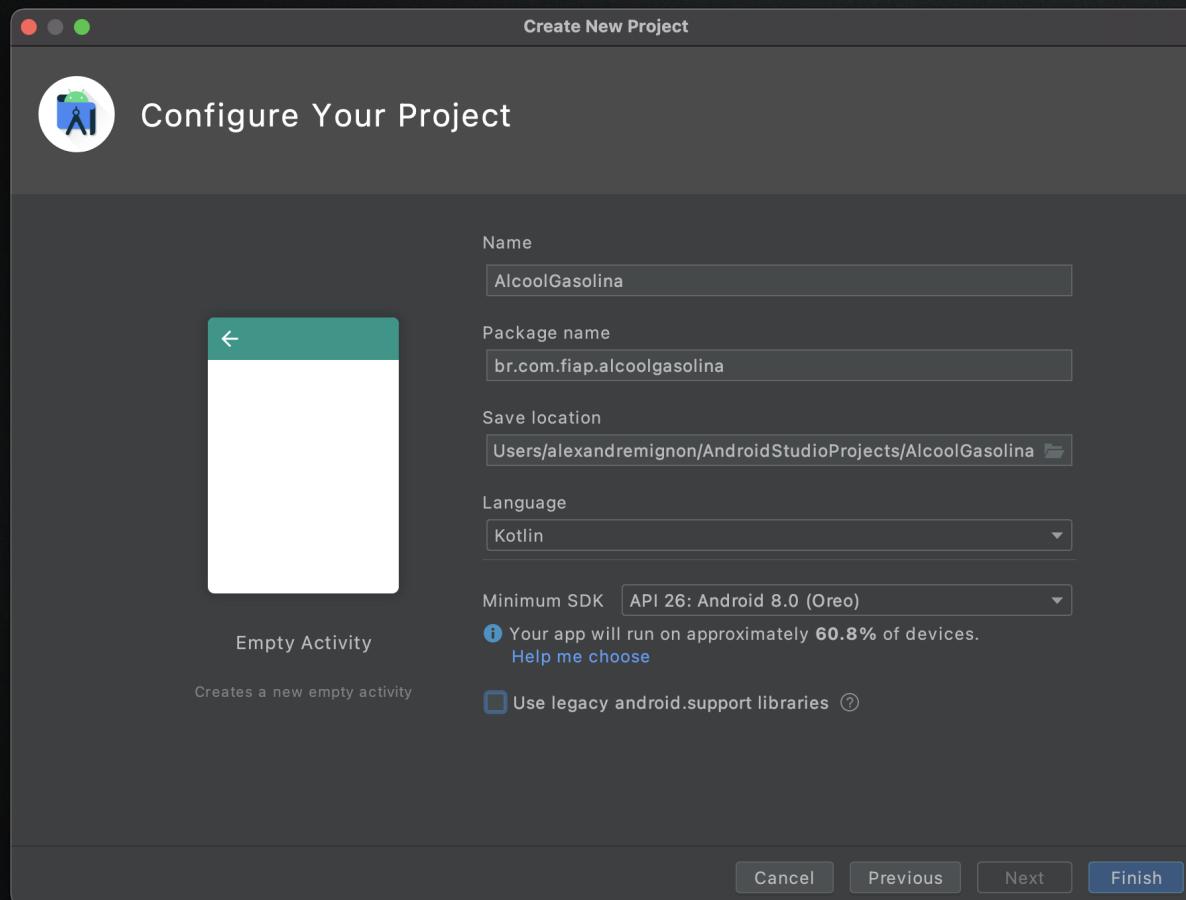
Álcool ou Gasolina.

A relação Preço Álcool / Preço Gasolina informa que:

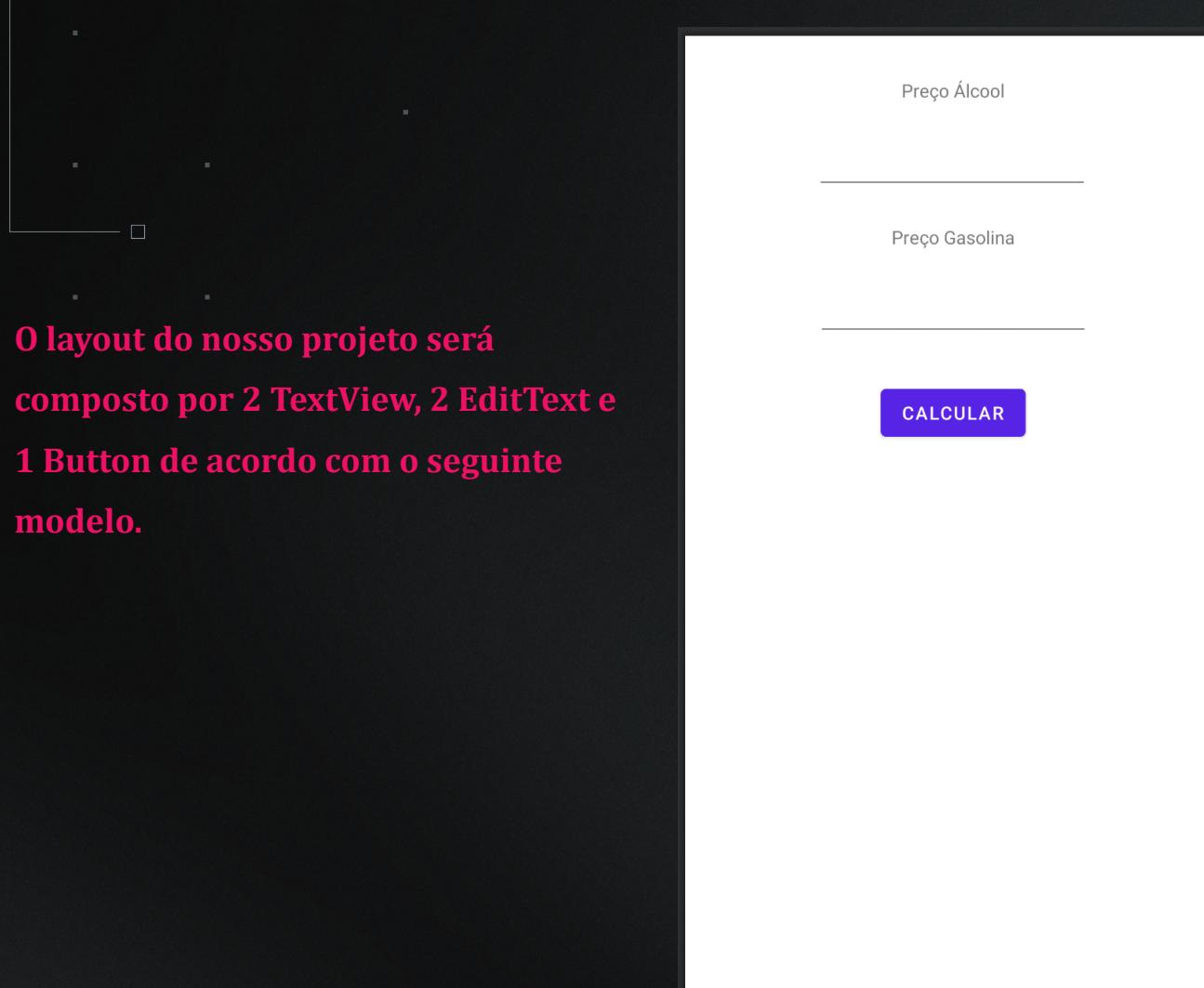
- **Valor acima de 0.7 ⇒ Compensa a Gasolina**
- **Valor abaixo de 0.7 ⇒ Compensa o Álcool**
- **Valor igual a 0.7      ⇒ Tanto faz**

## Criação do Projeto

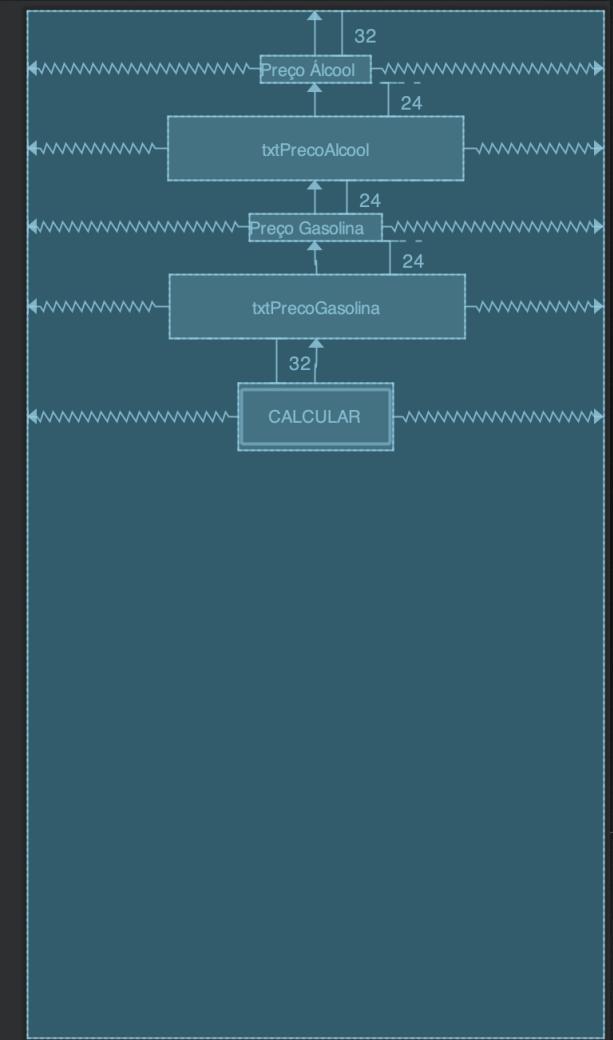
FIAP



## Layout



O layout do nosso projeto será  
composto por 2 TextView, 2 EditText e  
1 Button de acordo com o seguinte  
modelo.



## Code Behind

- O code behind de layouts em Android permite programar as ações sobre os diferentes eventos da tela.
- Toda tela é representada por uma Activity na qual possui diferentes estados.
- Iremos programar eventos no método onCreate.

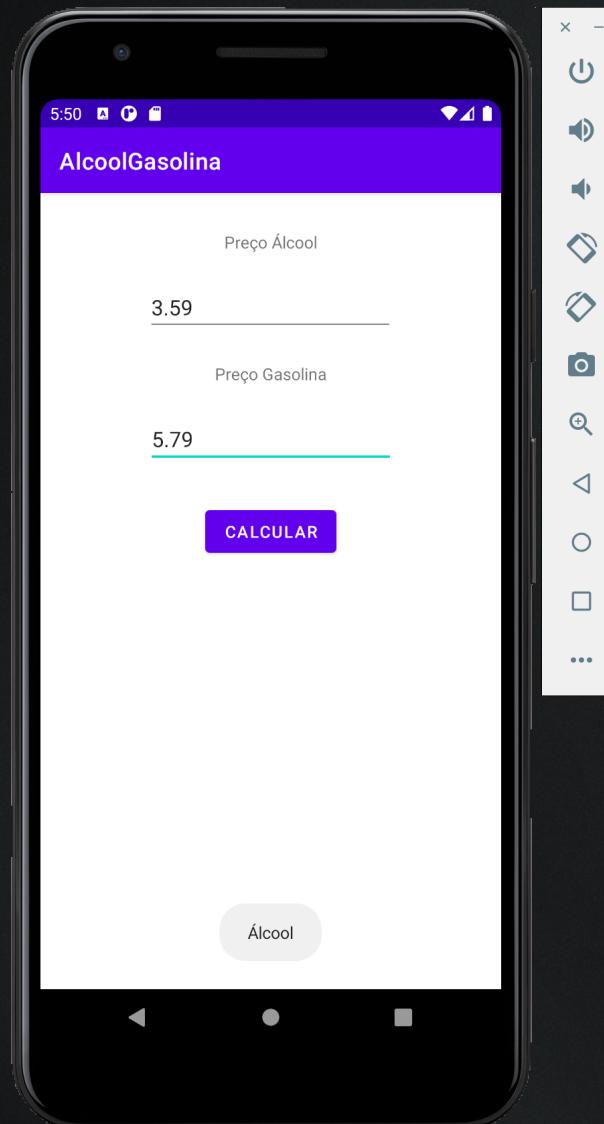
## Code Behind

FIAP

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var btnCalcular = findViewById<Button>(R.id.btnCalcular)  
        // Função que implementa o evento de clique em um botão  
        btnCalcular.setOnClickListener {view : View? ->  
            var txtAlcool = findViewById<EditText>(R.id.txtPrecoAlcool)  
            // declaração implícita do tipo da variável  
            var precoAlcool = txtAlcool.text.toString().toDouble()  
            var txtGasolina = findViewById<EditText>(R.id.txtPrecoGasolina)  
            var precoGasolina = txtGasolina.text.toString().toDouble()  
            // declaração explícita do tipo da variável  
            var resultado : Double = precoAlcool / precoGasolina  
            var mensagem = ""  
            if (resultado > 0.7) {  
                mensagem = "Gasolina"  
            }  
            else if (resultado < 0.7) {  
                mensagem = "Álcool"  
            }  
            else {  
                mensagem = "Tanto faz"  
            }  
            Toast.makeText(context: this, mensagem, Toast.LENGTH_LONG).show()  
        }  
    }  
}
```

## Resultado

FIAP



## Code Behind - Kotlin Extensions



Modificar o arquivo build.gradle e adicionar o seguinte import na classe MainActivity.

### build.gradle

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'kotlin-android-extensions'
}
```

### MainActivity

```
import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*
```

## Code Behind

FIAP

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

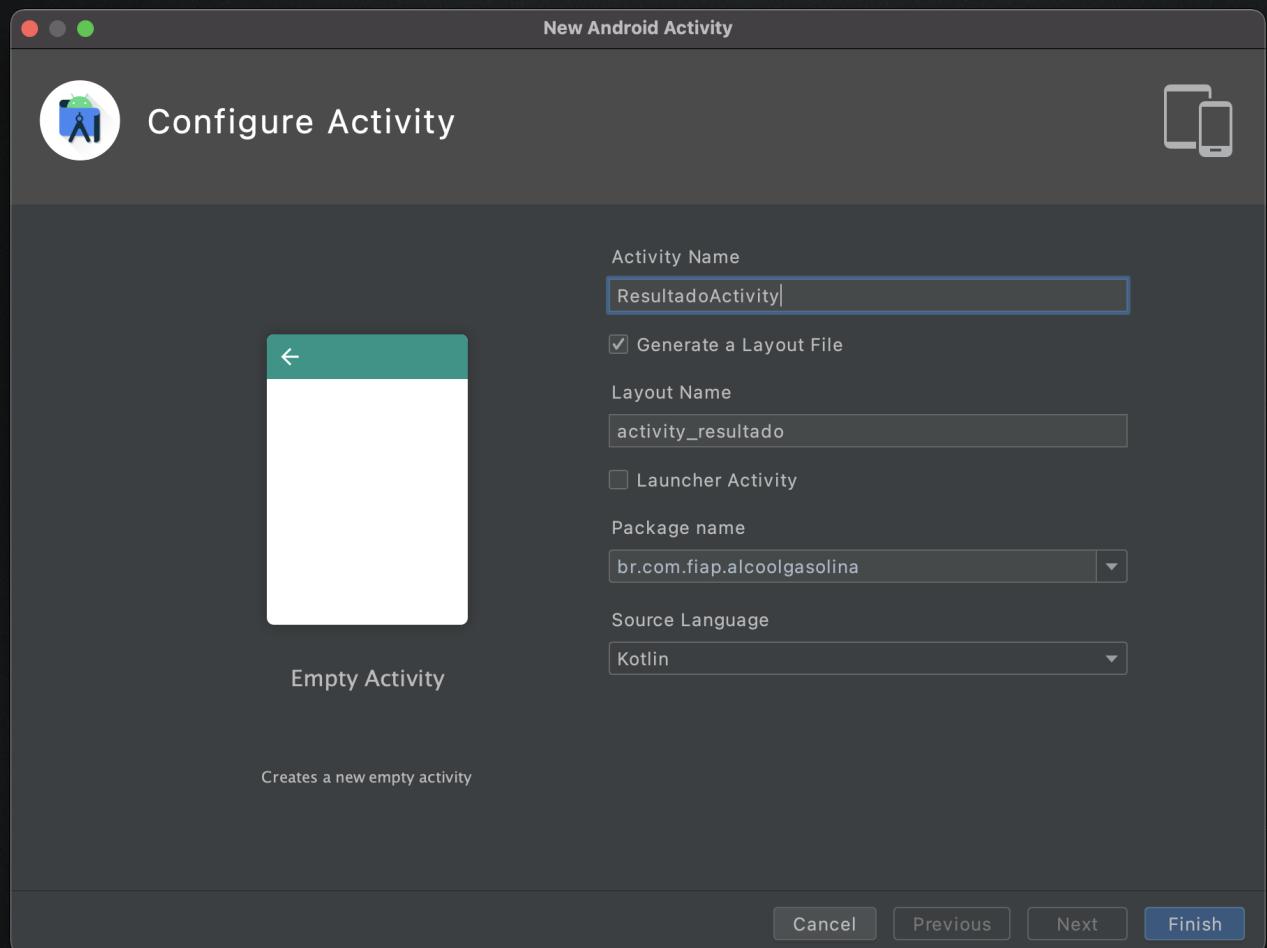
        // Função que implementa o evento de clique em um botão
        btnCalcular.setOnClickListener {view : View? ->
            // declaração implícita do tipo da variável
            var precoAlcool = txtPrecoAlcool.text.toString().toDouble()
            var precoGasolina = txtPrecoGasolina.text.toString().toDouble()
            // declaração explícita do tipo da variável
            var resultado : Double = precoAlcool / precoGasolina
            var mensagem = ""
            if (resultado > 0.7) {
                mensagem = "Gasolina"
            }
            else if (resultado < 0.7) {
                mensagem = "Álcool"
            }
            else {
                mensagem = "Tanto faz"
            }
            Toast.makeText( context: this, mensagem, Toast.LENGTH_LONG).show()
        }
    }
}
```

- **Outras telas podem ser implementadas no Android Studio e prover navegabilidade entre as mesmas.**
- **O uso do recurso de Intent permite a carga da nova tela a ser exibida além de transitar os dados da tela anterior para a posterior.**

## Transição entre Telas

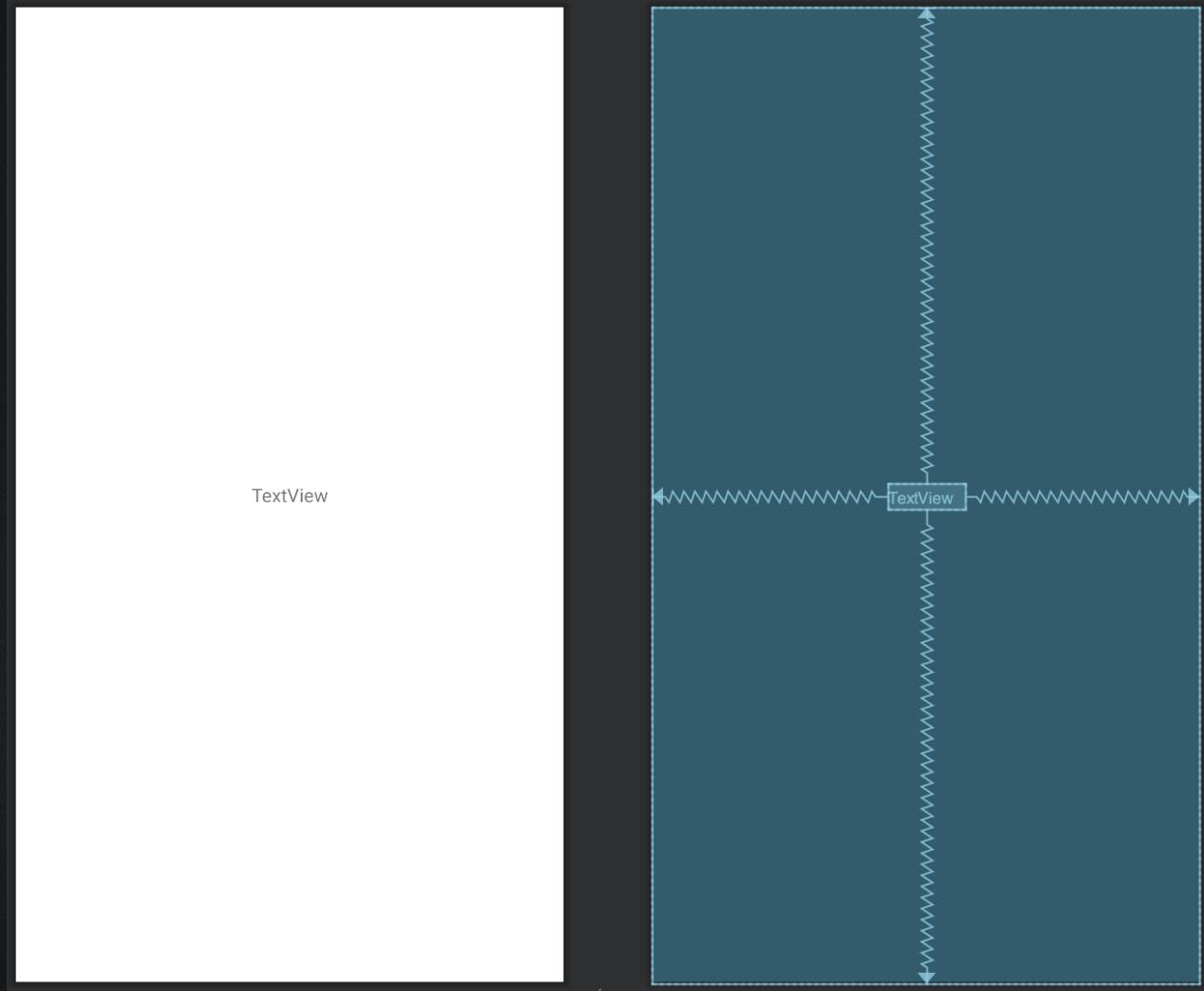
FIAP

- Criar uma nova Activity.



## Layout

O layout da nova tela será composto por um TextView (id: lbResultado).



- Carga da nova tela através do recurso de Intent e envio da mensagem de combustível vantajoso para a tela posterior.

```
// instancia do objeto Intent, apontando a Activity alvo.  
var intentResultado = Intent(packageContext: this, ResultadoActivity::class.java)  
// armazenamento por conjuntos de chave-valor  
intentResultado.putExtra(name: "msg", mensagem)  
// disparo da Intent para carregar a nova tela  
startActivity(intentResultado)
```

## Transição entre Telas



Recuperação da resposta da tela anterior na nova tela carregada.

Adicionar o código abaixo no onCreate da ResultadoActivity.

```
// o objeto intent permite recuperar todos os valores definidos na tela anterior  
lbResultado.text = intent.getStringExtra( name: "msg")
```

## Transição entre Telas

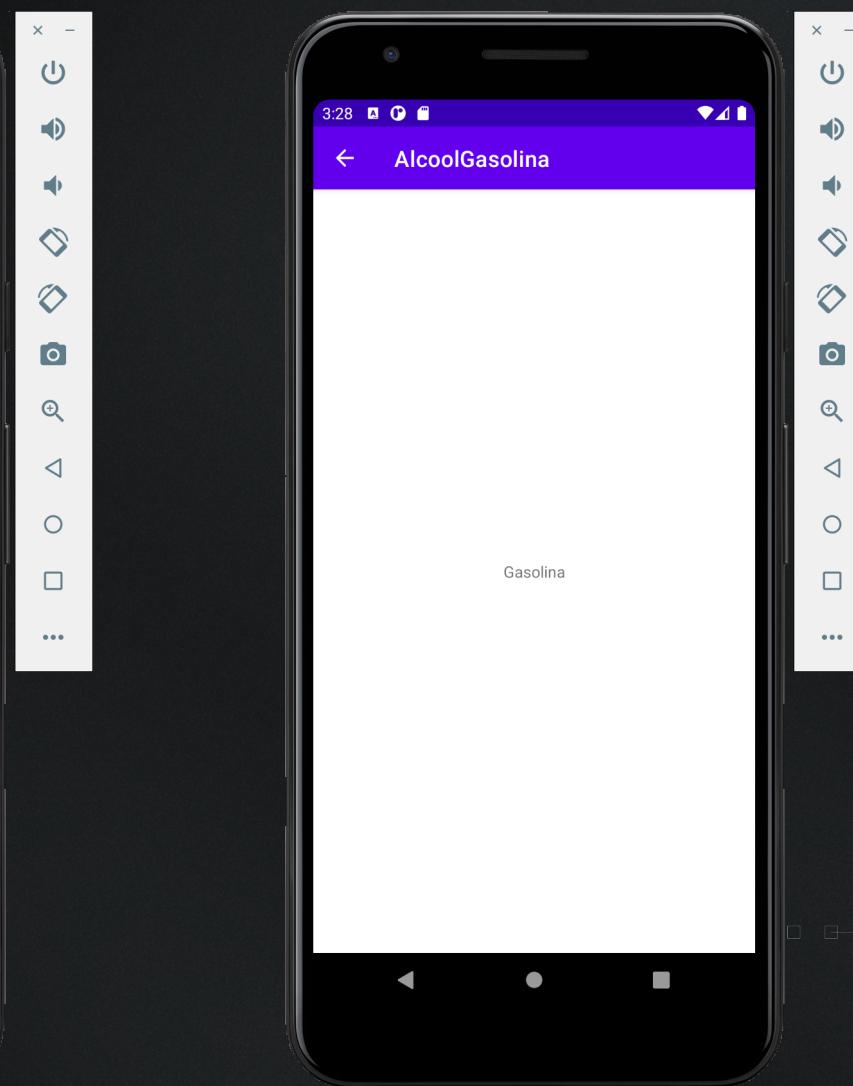
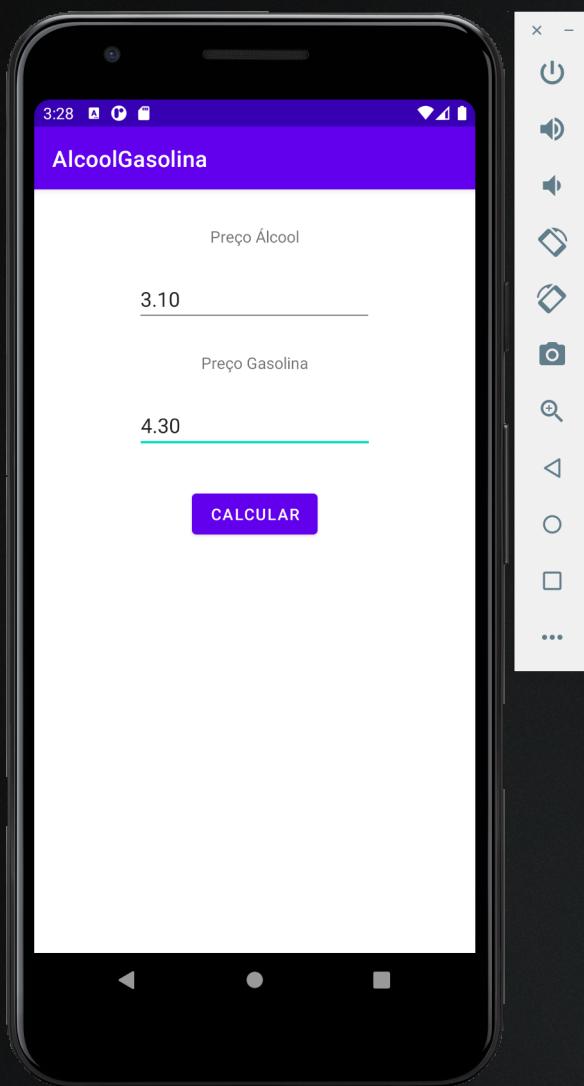


Configurar botão voltar nativo do Android na tela de resultado no arquivo `AndroidManifest.xml`.

```
<activity android:name=".ResultadoActivity"  
        android:parentActivityName=".MainActivity"></activity>
```

## Resultado

FIAP



Dúvidas?



# Tuíte

- By Steve Jobs

"Se você não é apaixonado pelo que faz, deveria tentar fazer outra coisa."