

Kotlin

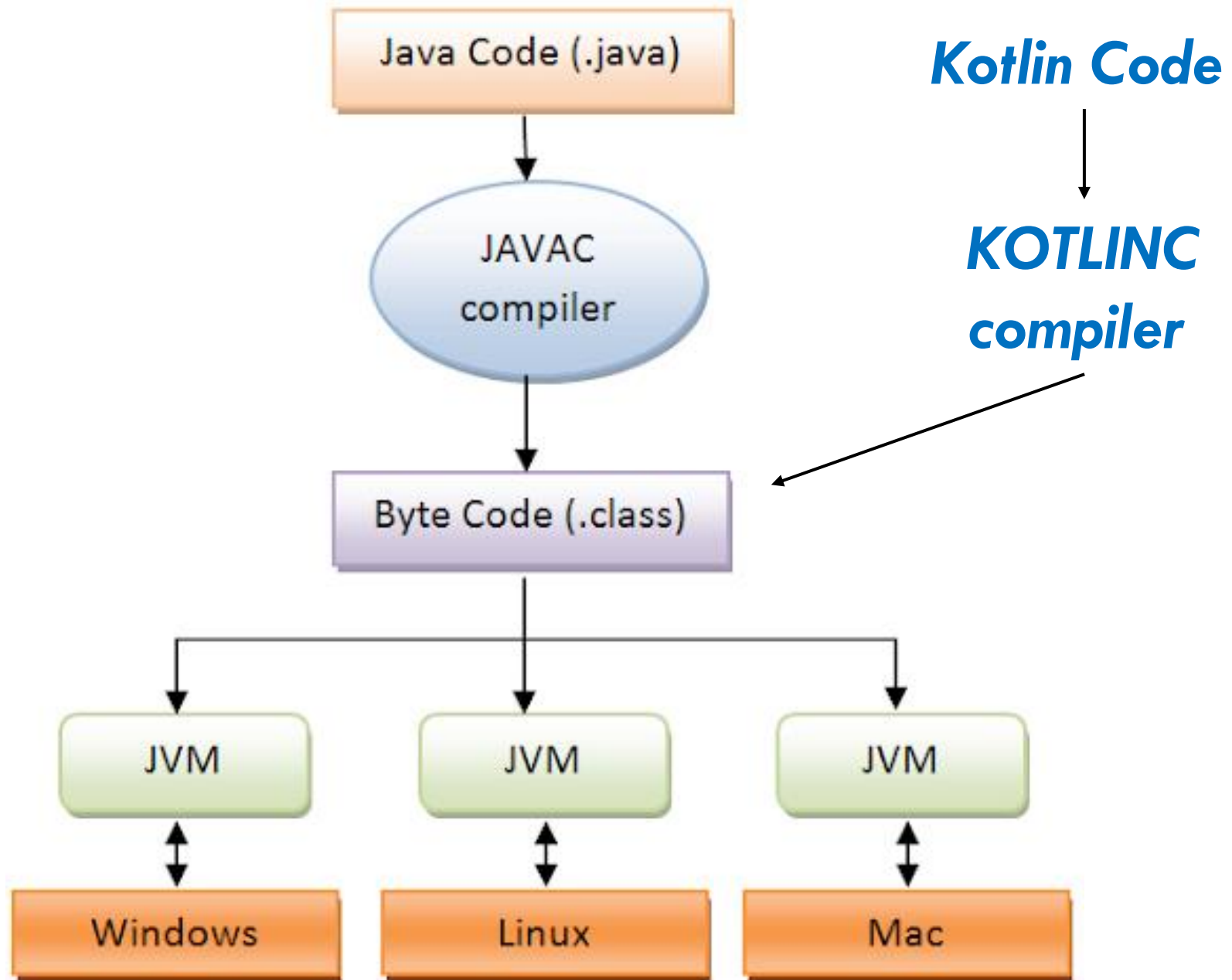
for Android App Development

DESENVOLVIMENTO MOBILE

AGENDA

1. Recapitulação
2. Elementos gráficos Android & Constraints
3. Navegação de Telas & Fragmentos

KOTLIN



KOTLIN - VARIÁVEIS

```
fun main() {  
    val imutável = 5  
    ! imutável = imutável + 3  
  
    println(imutável)  
}
```

! Val cannot be reassigned

```
fun main() {  
    var mutável = 5  
    mutável = mutável + 3  
  
    println(mutável)  
}
```

8

KOTLIN – NULO

```
var PODE_ser_nulo_me_Salve: String? = null

if (PODE_ser_nulo_me_Salve != null)
    println(PODE_ser_nulo_me_Salve.length)
```

KOTLIN – NULO

```
var PODE_ser_nulo_me_Salve: String? = null
```

```
//?: não faz nada se NÃO for nulo
```

```
//mas se for nulo troca a expressão inteira por um valor
```

```
println(PODE_ser_nulo_me_Salve?.length ?: 0)
```

0

KOTLIN - COLEÇÕES

```
fun main() {  
    //Lista (Vetor dinâmico)  
    val chamada = listOf("Amanda", "João", "Alex", "Maria")  
    val ordenada = chamada.sorted()  
    println(chamada.joinToString(","))  
}
```

Amanda,João,Alex,Maria

KOTLIN - COLEÇÕES

```
fun main() {  
    //Conjuntos  
    val frutas = setOf("Melancia", "Morango", "Açaí")  
  
    println(if (frutas.contains("Morango")) "tem morango!" else "não tem...")  
}
```

```
val resposta =  
    if ("Morango" in frutas)  
        "tem morango!"  
    else  
        "não tem..."
```


KOTLIN - COLEÇÕES

```
// Mapa (Vetor Associativo)  
val mapa_associação = mapOf(  
    "Chave1" to "Valor1",  
    "Chave2" to "Valor2",  
    "Chave3" to "Valor3"  
)  
for((k, v) in mapa_associação) {  
    println("$k: $v")  
}
```

```
Chave1: Valor1  
Chave2: Valor2  
Chave3: Valor3
```

KOTLIN – CONDICIONAIS

```
val nota = 7.2
val avaliação =
    when {
        nota > 8 -> "Excelente"
        nota > 5 -> "Assistível"
        else -> "Lixo"
    }

println(avaliação)
```

KOTLIN – LAÇOS

```
for (i in 1..10)
    println(i)
for (i in 6 downTo 0 step 2)
    println(i)
for (i in listOf(7, 15, 23))
    println(i)
for ((idx, i) in listOf(7, 15, 23).withIndex())
    println("$idx $i")
```

KOTLIN – CLASSES

```
//parâmetros var e val formam parte do "construtor primário"  
class Classe(var texto: String = "", qtd: Int = 1) {  
    //var e val 'top-level' também fazem parte da construção primária  
    val vezes = if (qtd < 1) 1 else qtd  
  
    //código de construção primária deve ser colocado em bloco init  
    init {  
        texto = "|" + texto + "|"  
    }  
}
```

KOTLIN – CLASSES

```
open class Pessoa(var nome: String) {  
    open fun apresentar() {  
        println("Pode me chamar $nome")  
    }  
}  
  
class Professor(nome: String): Pessoa(nome) {  
    override fun apresentar() {  
        println("Pode me chamar de Prof. $nome")  
    }  
}
```

KOTLIN – FUNÇÕES

```
val numeros = listOf(1,2,3,4,5,6)
```

```
var pares: List<Int> = listOf()
```

```
pares = numeros.filter( fun (it: Int) = it % 2 == 0)
```

```
pares = numeros.filter( {it: Int -> it % 2 == 0})
```

```
pares = numeros.filter( {it % 2 == 0} )
```

```
pares = numeros.filter {  
    it % 2 == 0  
}
```

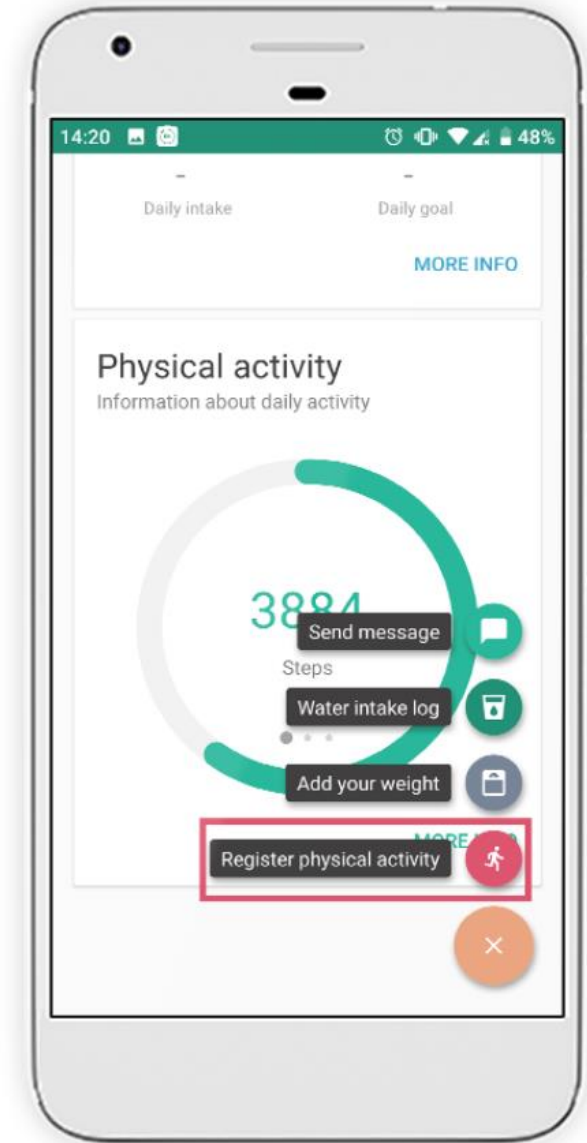
ANDROID – COMPONENTES BÁSICOS

Estrutura de Arquivos:

- *manifests/AndroidManifest.xml* : configurações gerais, define coisas como Nome, Permissões, Ícone...
- *java* : códigos Java/Kotlin, implementação
- *res* : recursos como imagens, definições de interface, strings...

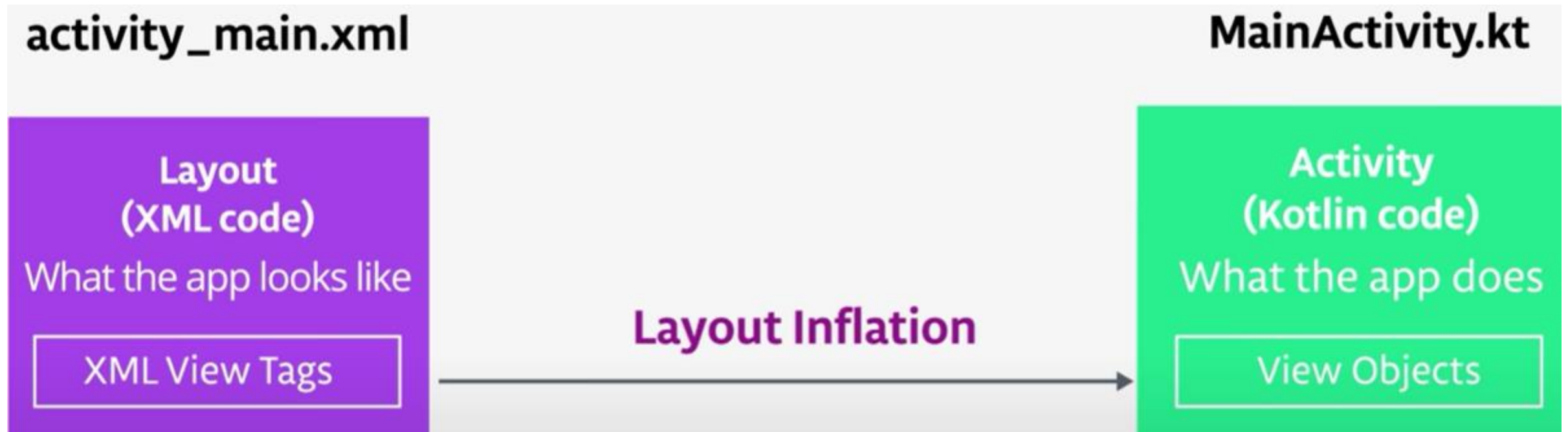
ANDROID – COMPONENTES

Activity : Uma Janela/Tela interativa.



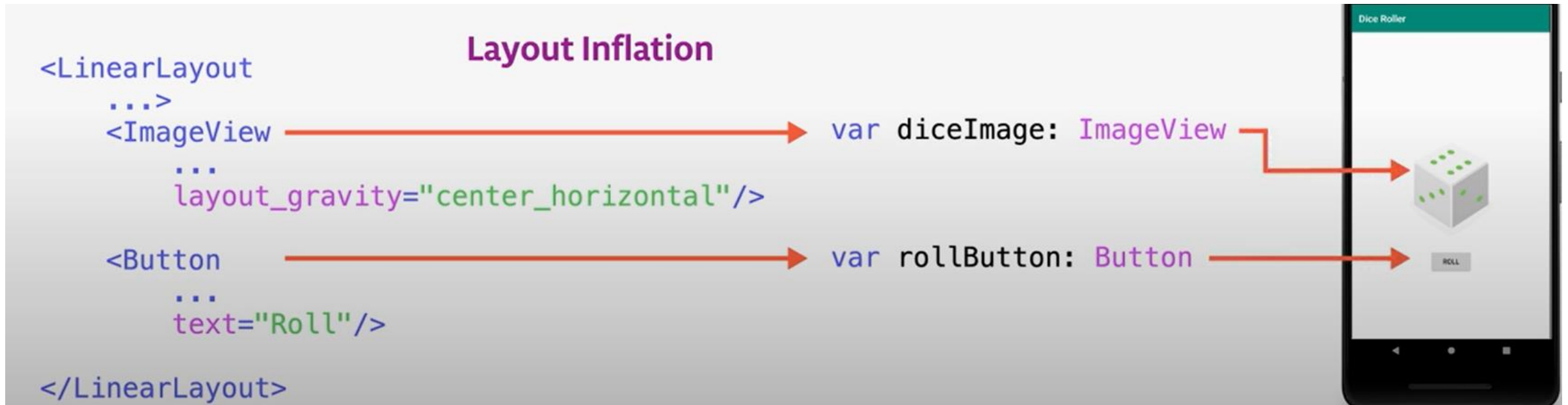
ANDROID – COMPONENTES BÁSICOS

Inflation : Seguir uma “receita” construindo objetos gráficos

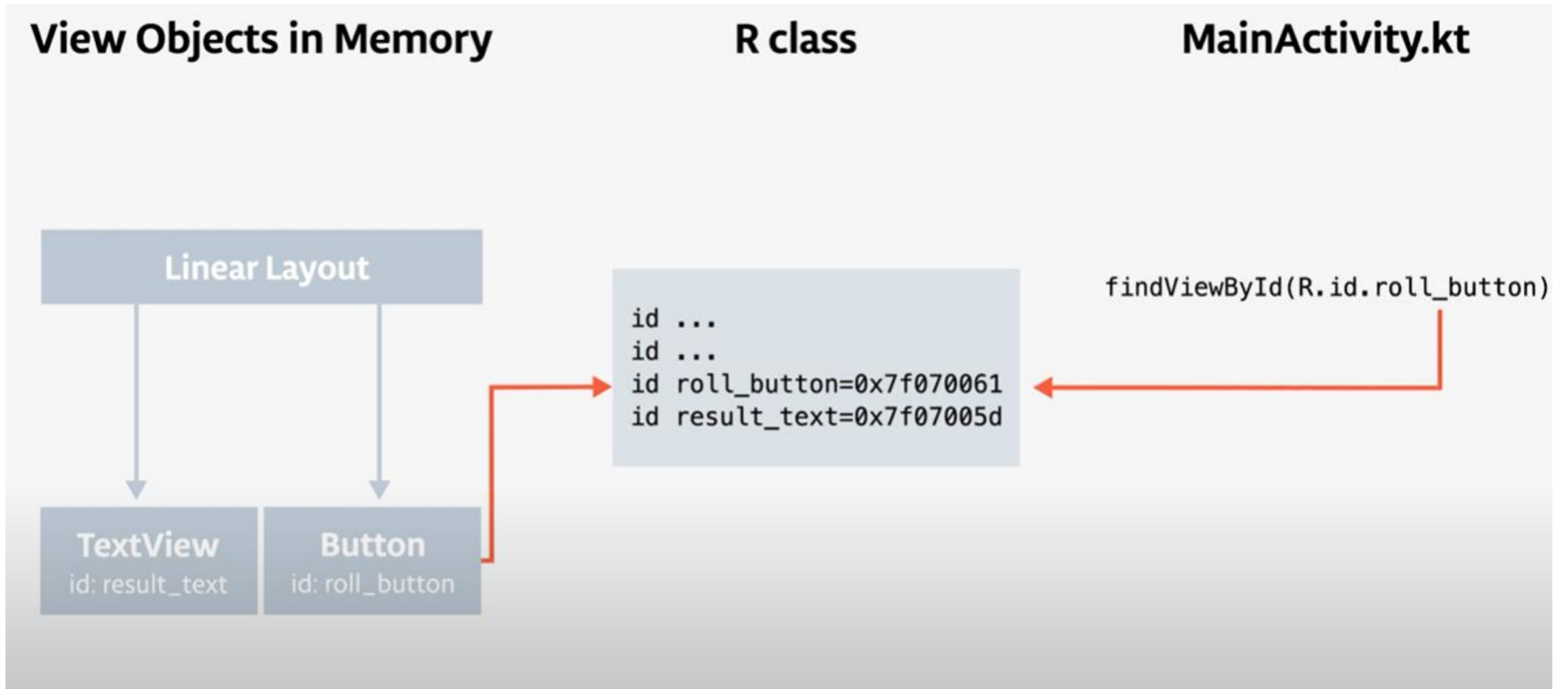


ANDROID – COMPONENTES BÁSICOS

Inflation : Seguir uma “receita” construindo objetos gráficos



ANDROID – IDENTIFICADORES



ANDROID – PROGRAMATICAMENTE

```
val botão: Button = findViewById(R.id.roll_button)  
  
botão.text = "Tá na hora do Show"
```

ANDROID – RESOURCES

```
val dado_imagem: ImageView = findViewById(R.id.dado_imagem)
val de1a6 = Random().nextInt(bound: 6) + 1
val faces = listOf(
    R.drawable.empty_dice,
    R.drawable.dice_1,
    R.drawable.dice_2,
    R.drawable.dice_3,
    R.drawable.dice_4,
    R.drawable.dice_5,
    R.drawable.dice_6,
)
dado_imagem.setImageResource(faces[de1a6])
```

ANDROID – OTIMIZANDO FINDVIEW

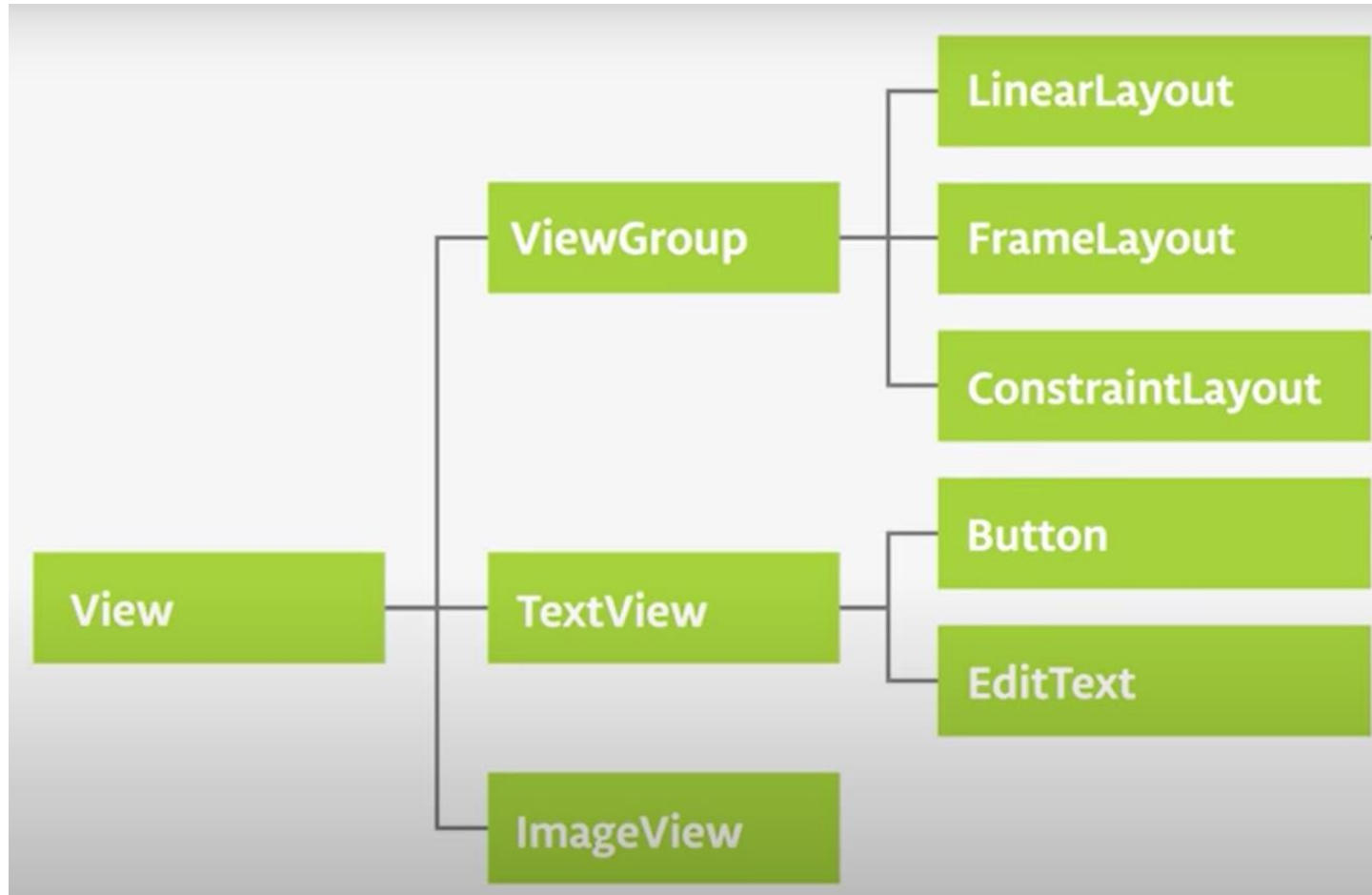
```
//lateinit = "Confia que vou inicializar antes de usar"  
lateinit var dado_imagem: ImageView  
  
override fun onCreate(savedInstanceState: Bundle?) {  
    super.onCreate(savedInstanceState)  
    setContentView(R.layout.activity_main)  
  
    var botão: Button = findViewById(R.id.roll_button)  
  
    dado_imagem = findViewById(R.id.dado_imagem)
```

ANDROID – COMPONENTES BÁSICOS

View : Um objeto gráfico visível, exemplos como TextView, Button, ImageView, CheckBox, RadioButton, EditText, ProgressBar...

ANDROID – VIEWGROUP

Viewgroup : Uma View que contém Views.



ANDROID – SCALE-INDEPENDENT PIXEL

sp : Pixel proporcional ao tamanho da tela. (igual a dp mas respeita configurações de acessabilidade)

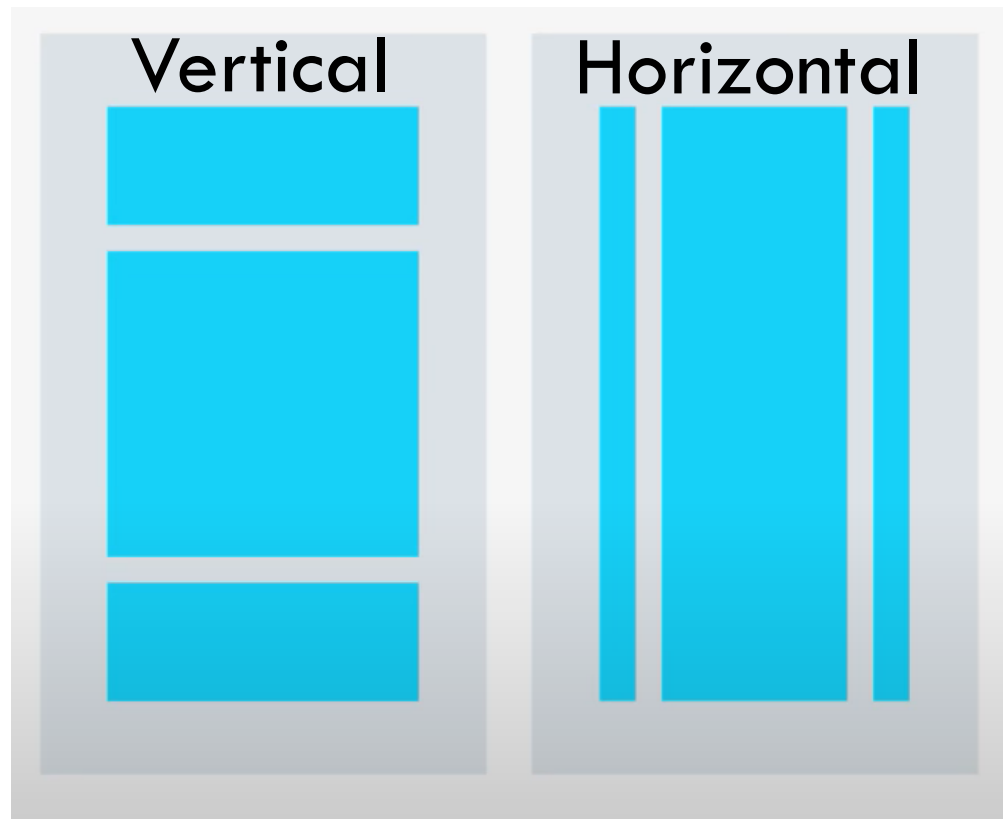
1 sp =

160dpi = 1 pixel

480dpi = 3 pixel

ANDROID – LINEAR LAYOUT

Linear Layout : Viewgroup que sequencia elementos



APLICATIVO – “SOBRE MIM”

Vamos criar um novo Projeto

- Crie como “Empty Activity”
- Converta o Layout para Linear



APLICATIVO – “SOBRE MIM”

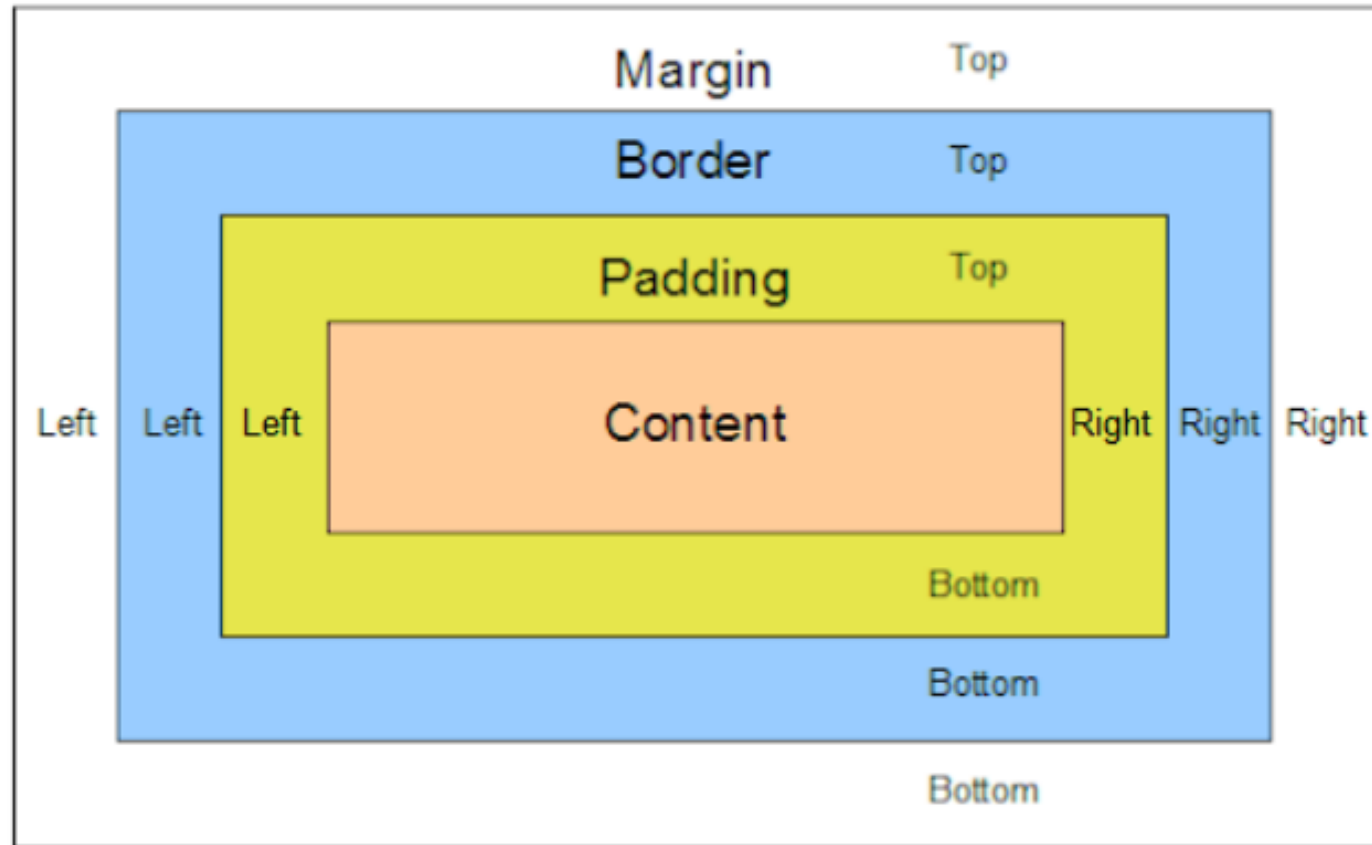
Alterações

- Altere width Layout p/ match
- Escolha black para o texto
- Escolha center para alinhamento
- 18sp, extraía p/ dimens
- Fonte “Roboto”



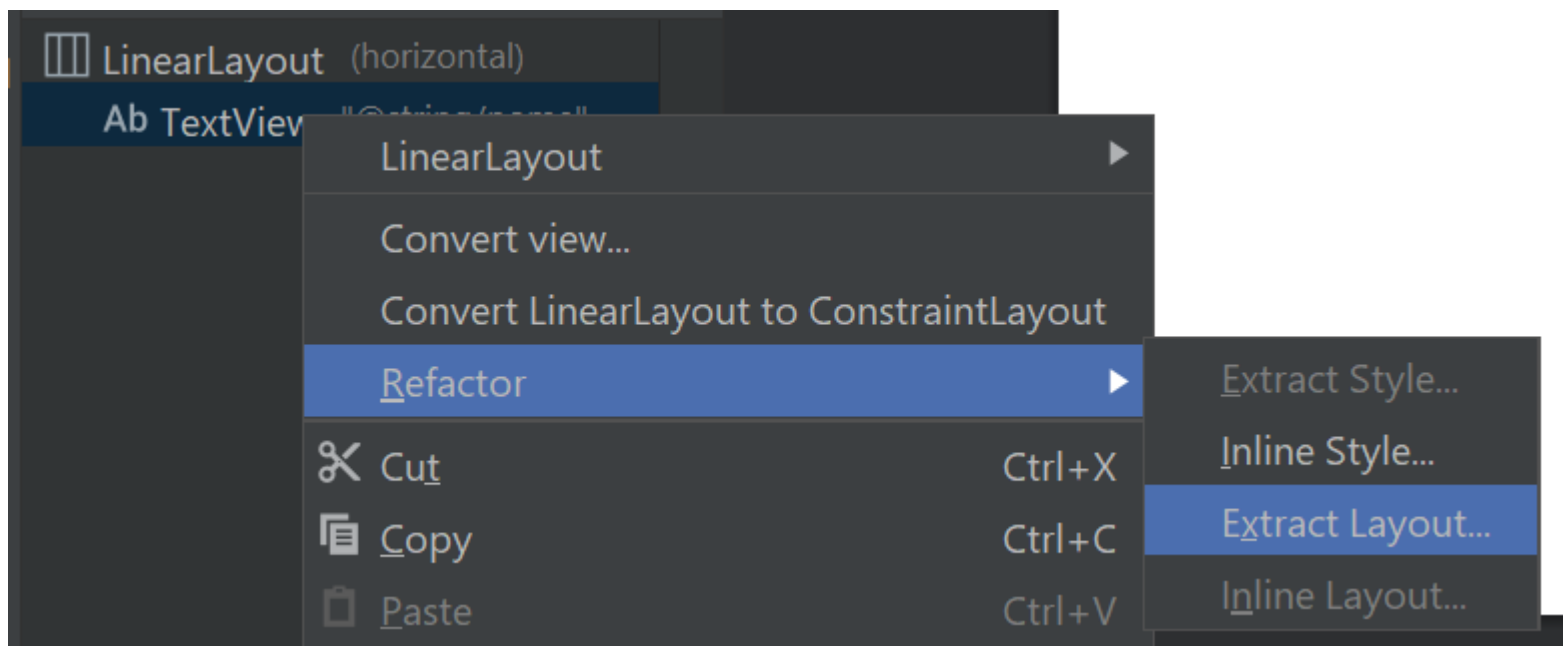
APLICATIVO – “SOBRE MIM”

Margin Top 16, Padding Top 8



APLICATIVO – “SOBRE MIM”

Refatore para um Style



APLICATIVO – “SOBRE MIM”

Alterações

- Adicione um ImageView “Yellow Star”
- Width match para centralizar
- Content description
- Padding Top 16

ANDROID – SCROLLVIEW

Scroll View: Viewgroup que trabalha com rolagem



ANDROID – SCROLLVIEW

The screenshot displays the Android Studio interface with three main panels:

- Component Tree:** Shows a vertical layout containing a `TextView` with the text `"@string/nome"`, an `ImageView` with the resource `estrela_amarela`, and a `ScrollView` containing a `TextView` with the text `"@string/biografia_t..."`.
- Text Preview:** Displays a paragraph of text from the `bio` string resource: "a Terra logo após a concepção de Peter, a fim de continuar sua guerra intergaláctica, deixando Meredith na Terra para protegê-la. Nove meses mais tarde, durante um fenômeno astronômico raro, quando muitos dos planetas estavam alinhados, Peter nasceu e pouco tempo depois sua mãe Meredith, por razões que ela não conseguia explicar, teve o desejo de levá-lo para fora e apresentá-lo para as estrelas. Durante dez anos, Quill foi criado por Meredith, a qual permaneceu uma mãe solteira, mas teve a ajuda de sua melhor amiga Lisa Chang e a filha dela Monica. Peter logo começou a desenvolver um grande interesse em programas de ficção científica, e após uma visita a um parque de diversões onde ele ficou impressionado com as maravilhas do espaço, ele decidiu que queria ser um astronauta quando crescesse.[1] No entanto, ele mais tarde acabou ficando um pouco desapontado após descobrir que os alienígenas no passeio não eram reais. Um dia, Peter retornou para casa, ferido por tentar impedir
- Declared Attributes:** Lists the attributes for the selected `TextView` in the `ScrollView`.

Declared Attributes		
<code>layout_width</code>	<code>match_parent</code>	
<code>layout_height</code>	<code>wrap_content</code>	
<code>layout_marginLeft</code>	<code>8sp</code>	
<code>layout_marginRi...</code>	<code>8sp</code>	
<code>id</code>	<code>bio</code>	
<code>lineSpacingMulti...</code>	<code>1.2</code>	
<code>paddingBottom</code>	<code>32sp</code>	
<code>text</code>	<code>@string/biografia_textc</code>	
<code>textAlignment</code>	<code>textStart</code>	
<code>textSize</code>	<code>18sp</code>	
Layout		
<code>layout_width</code>	<code>match_parent</code>	
<code>layout_height</code>	<code>wrap_content</code>	

ANDROID – SCROLLVIEW

The screenshot displays the Android Studio interface with a TextView widget selected in a ScrollView. The widget is titled "Escolha um apelido" and contains a paragraph of text. The Component Tree on the left shows the hierarchy: layouti (vertical) > TextView "@string/nome" > estrela_amarela > apelido (Plain Text) > scroll > bio "@string/biografia_texto". The Properties panel on the right shows the attributes for the "apelido" widget, including layout_width, layout_height, layout_marginBottom, ems, hint, id, inputType, textAlignment, textStyle, and layout.

Component Tree:

- layouti (vertical)
 - Ab TextView "@string/nome"
 - estrela_amarela
 - Ab apelido (Plain Text) ⚠
 - ▼ scroll
 - Ab bio "@string/biografia_texto"

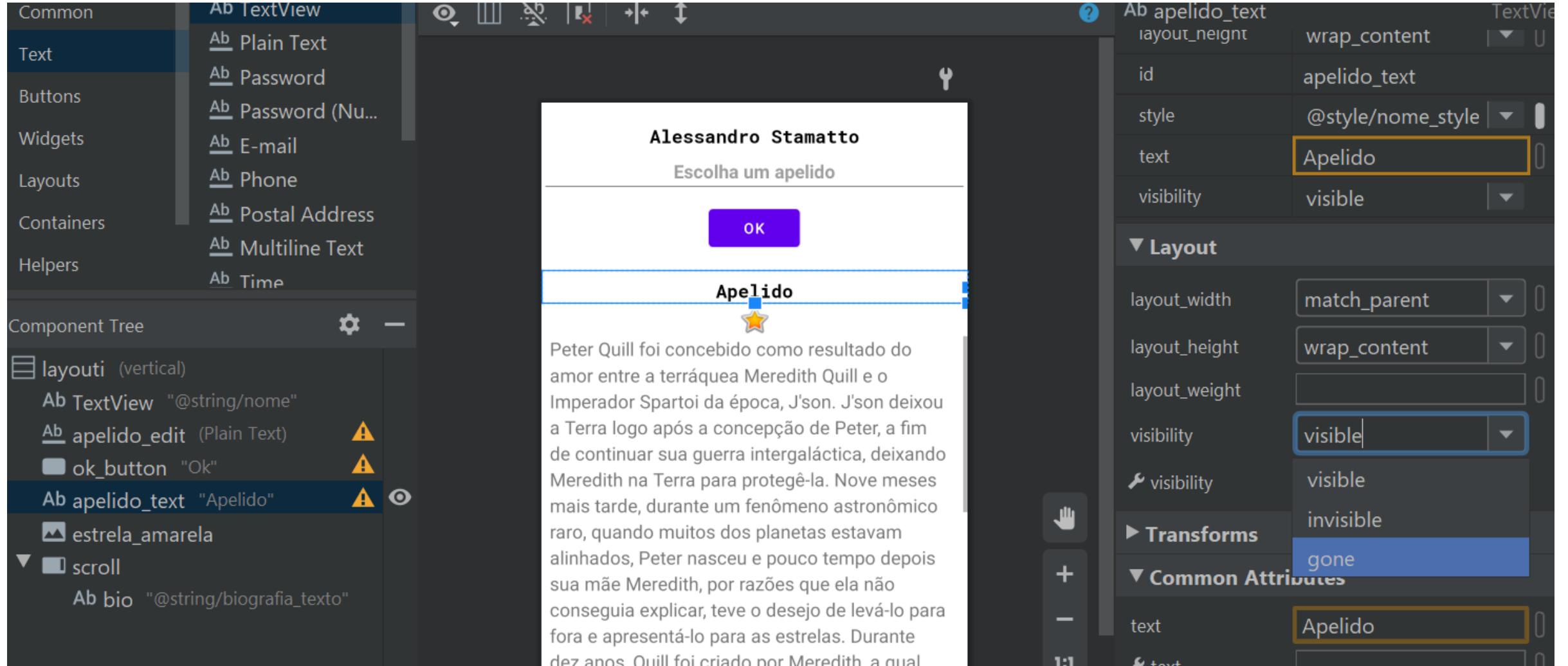
Properties Panel (Ab apelido):

Declared Attributes	
layout_width	match_parent
layout_height	wrap_content
layout_marginBo...	8sp
ems	10
hint	Escolha um apelido
id	apelido
inputType	textPersonName
textAlignment	center
textStyle	bold

Layout:

Layout	
layout_width	match_parent
layout_height	wrap_content

ANDROID – SCROLLVIEW

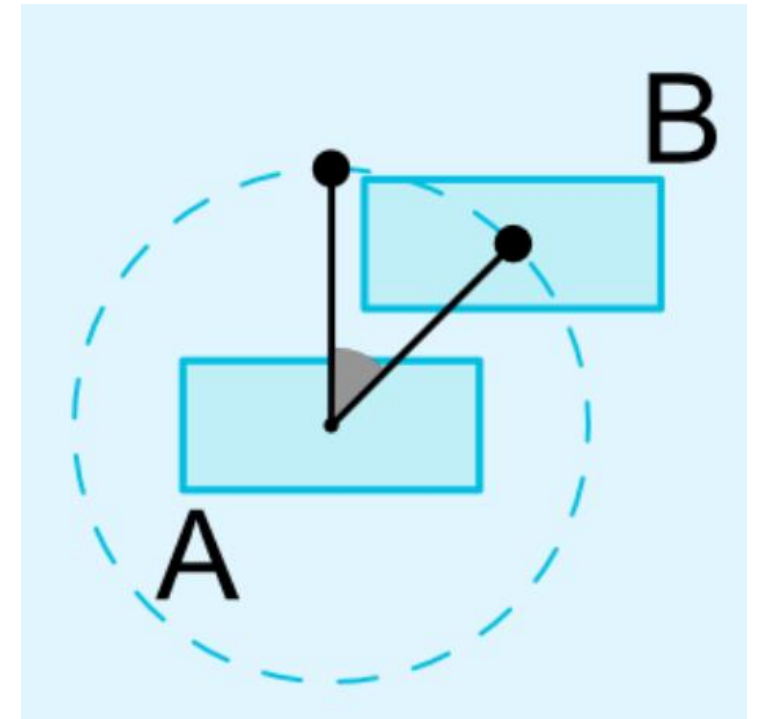


```
fun inicializarViewVars() {  
    edit = findViewById(R.id.apelido_edit)  
    apelido = findViewById(R.id.apelido_text)  
    ok_button = findViewById(R.id.ok_button)  
    ok_button.setOnClickListener {atualizarApelido(it)}  
}  
  
fun atualizarApelido(view: View) {  
    if (edit.text.toString().trim().isEmpty()) return;  
  
    apelido.text = edit.text  
  
    edit.visibility = View.GONE  
    ok_button.visibility = View.GONE  
  
    apelido.visibility = View.VISIBLE  
  
    (getSystemService(Context.INPUT_METHOD_SERVICE)  
        as InputMethodManager)  
        .hideSoftInputFromWindow(view.windowToken, flags: 0)  
}
```

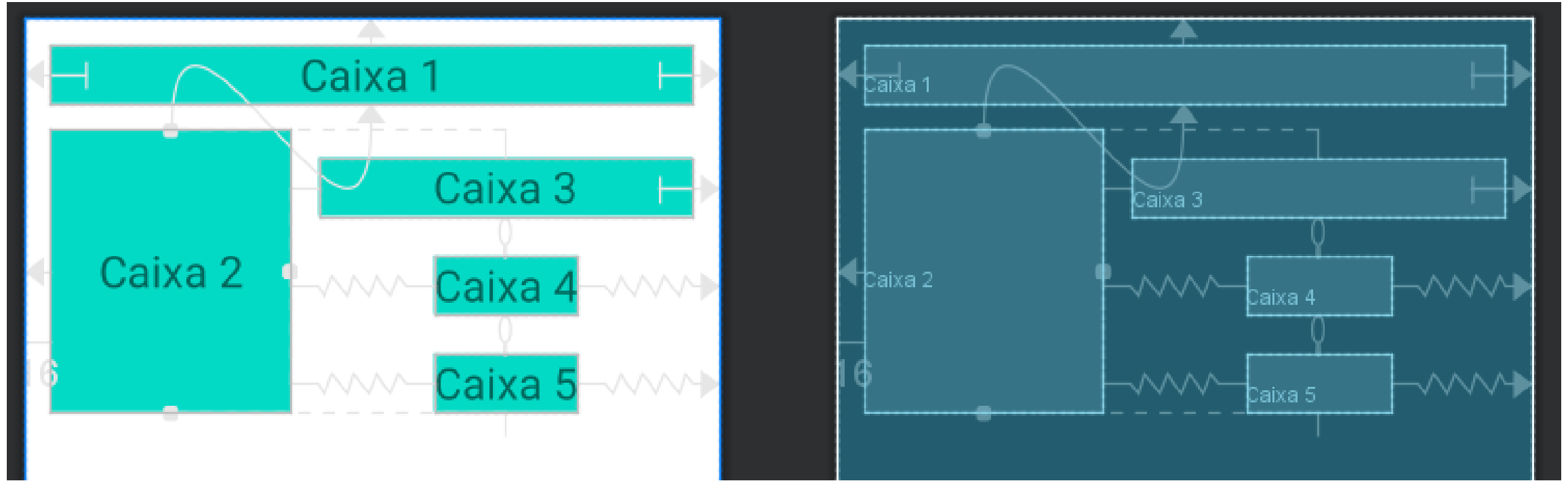
ANDROID – CONSTRAINT LAYOUT

Constraint Layout: Viewgroup que trabalha com regras (**constraints**) de posicionamento **relativo**.

- Coloque A perto da borda esq.
- Coloque B a 45 graus de A



ANDROID – CONSTRAINT LAYOUT



ANDROID – EXERCÍCIO

Faça as caixas começarem em branco, e fiquem colorida quando se clica nela. Dica:

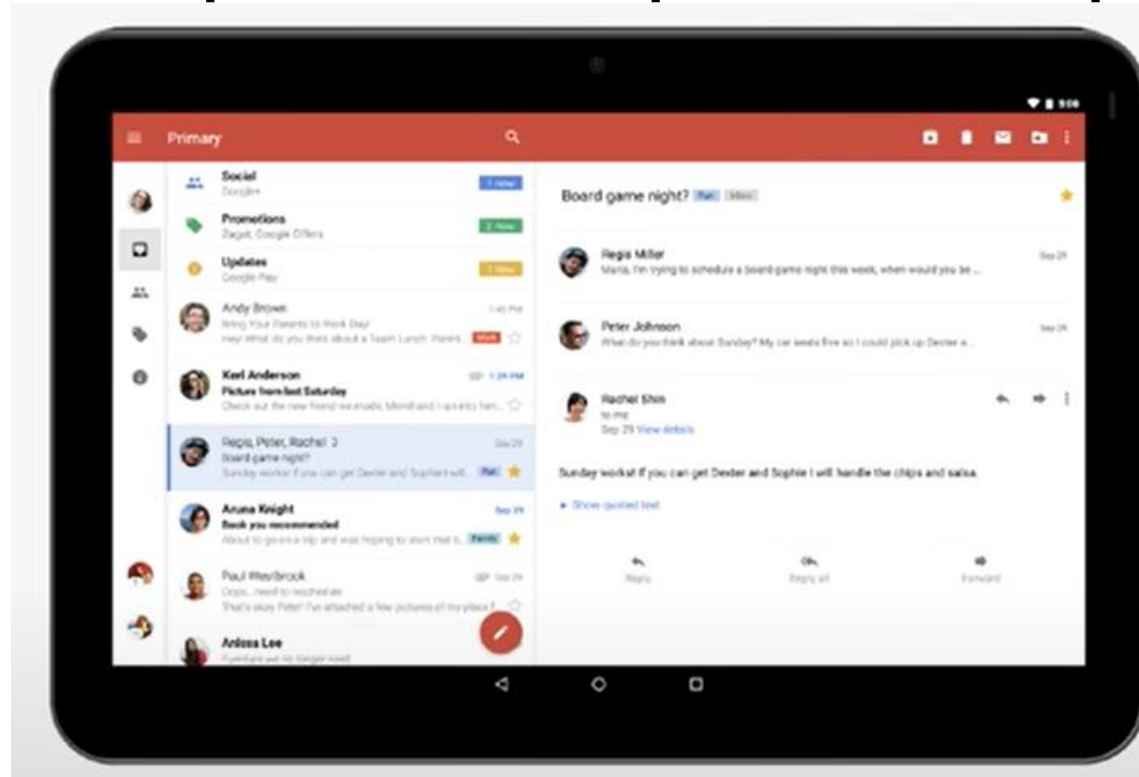
```
// Boxes using Color class colors for background
R.id.box_one_text -> view.setBackgroundColor(Color.DKGRAY)
R.id.box_two_text -> view.setBackgroundColor(Color.GRAY)

// Boxes using Android color resources for background
R.id.box_three_text -> view.setBackgroundResource(android.R.color.holo_green_light)
R.id.box_four_text -> view.setBackgroundResource(android.R.color.holo_green_dark)
R.id.box_five_text -> view.setBackgroundResource(android.R.color.holo_green_light)

else -> view.setBackgroundColor(Color.LTGRAY)
```

ANDROID – FRAGMENTS

Fragments: Uma Activity, só que ao invés de representar a tela toda pode representar apenas uma parte da tela.

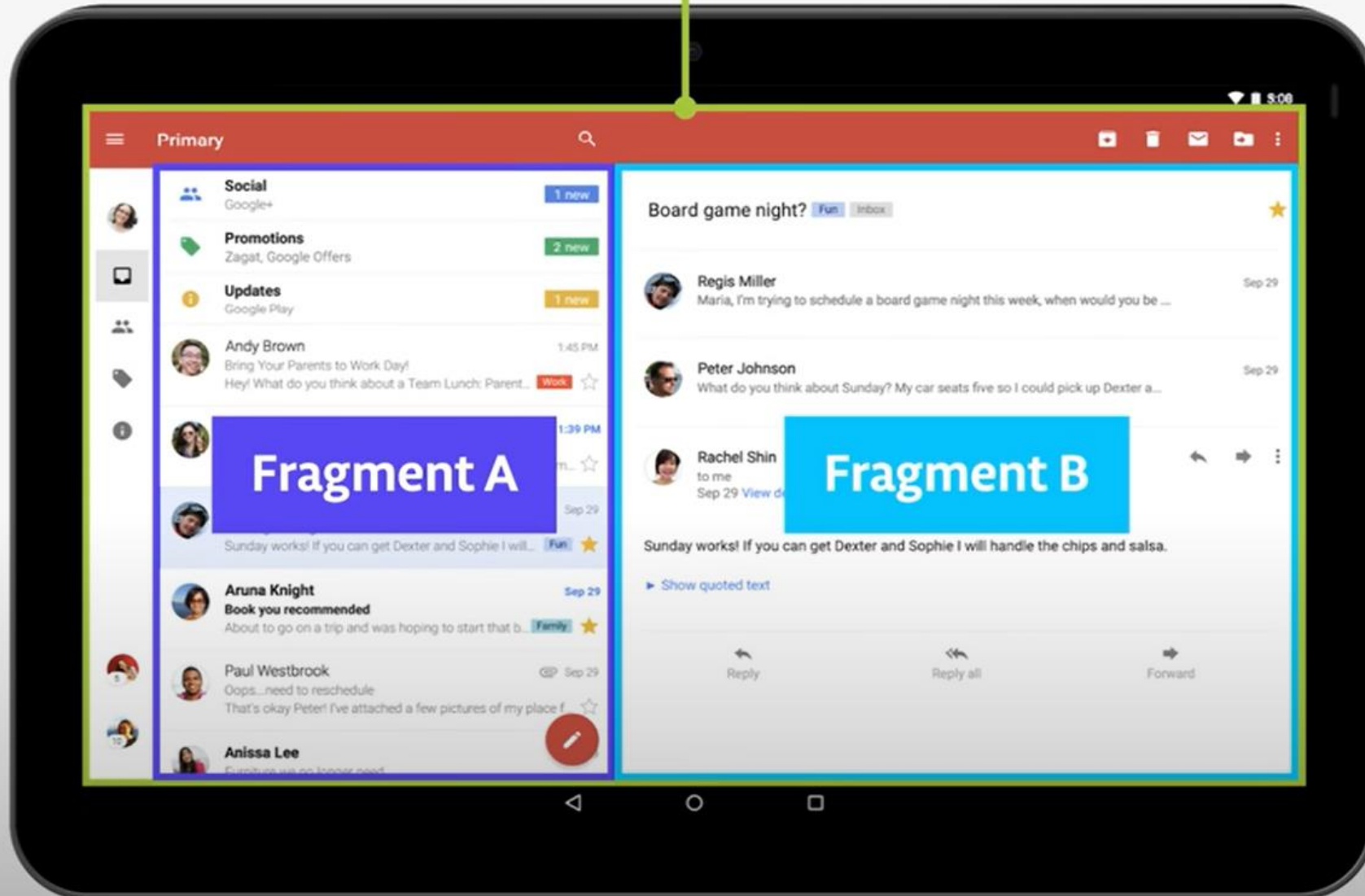


ANDR

Fragment

tela to

Activity



enter a
la.

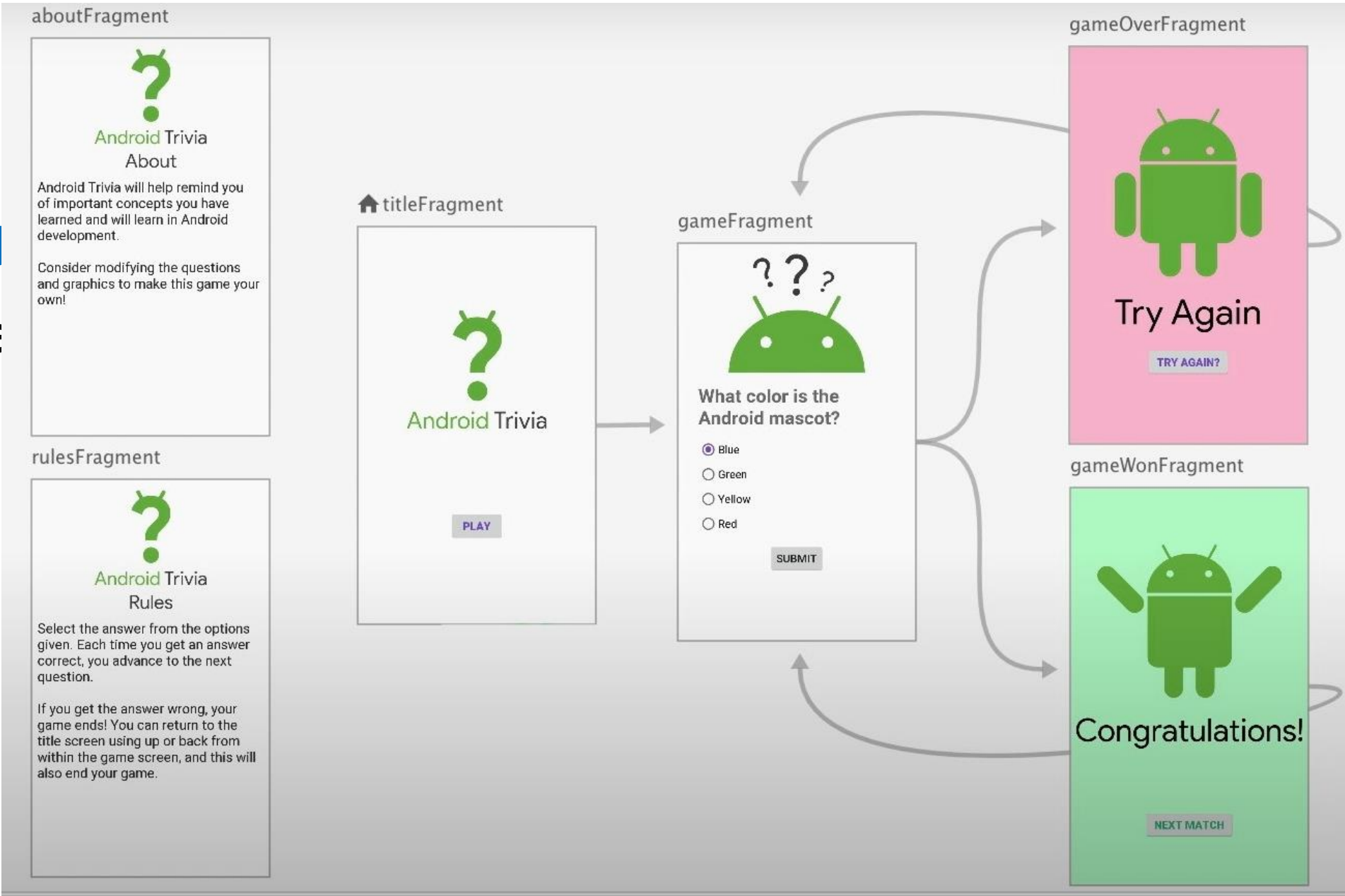
ANDROID – FRAGMENTS

Fragments: Também oferece recursos de troca de tela e navegação entre telas.

A N D

Fragment

nave



1 e

ANDROID – FRAGMENTS

Fragments: Usando Fragments a Activity representa seu aplicativo (o ponto de partida) e Fragmentos as suas telas.