



Desenvolvimento de Interfaces Android

Disciplina Regular 2





Material Design

Android Layouts, Estilos e Temas





Navegação em Activities



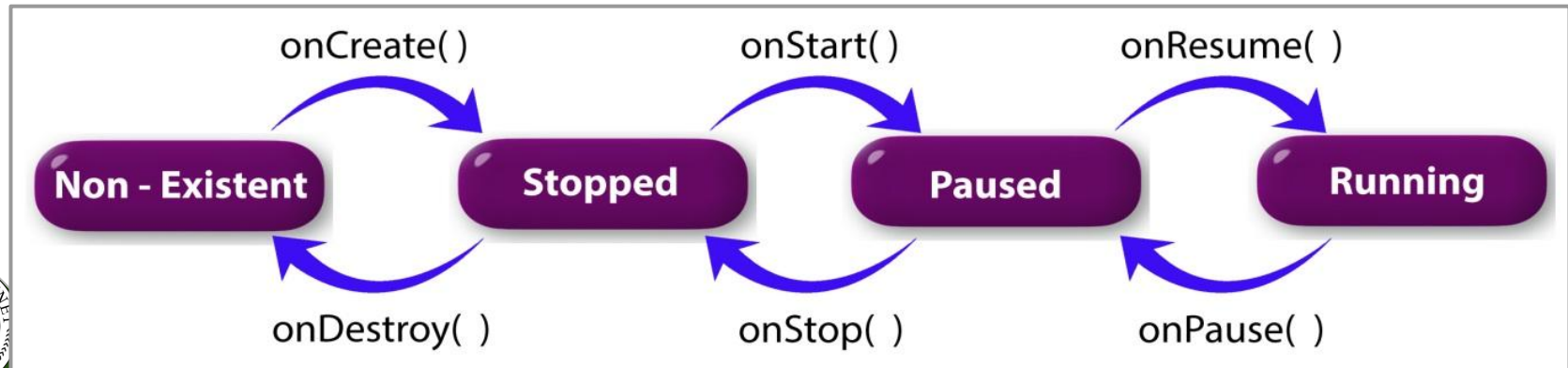


Ciclo de Vida da Activity

Activities no sistema são gerenciadas como pilhas.

Quando uma nova Activity começa, ela é colocada no topo da pilha e se torna a **atividade em execução** - a atividade anterior sempre permanece abaixo da nova na pilha e não retornará ao primeiro plano até que a nova atividade saia.

Uma atividade tem essencialmente quatro estados:



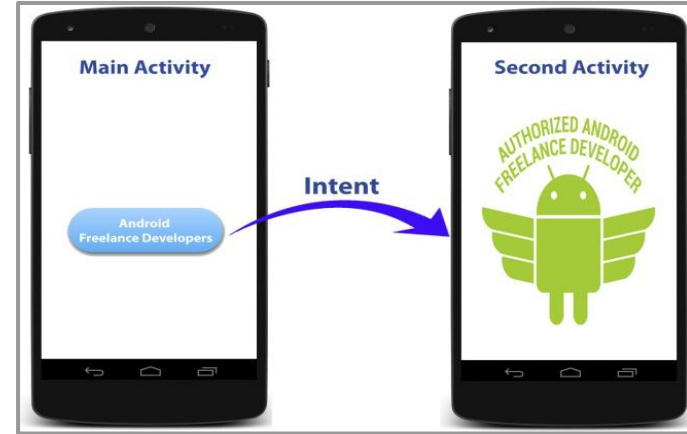


Android Intent

Os componentes do aplicativo Android podem se conectar a outros aplicativos Android.

Esta conexão é baseada em uma descrição de tarefa representada por um objeto **Intent**.

Android Intent é uma descrição de classe abstrata de uma operação a ser executada.



Intents são mensagens assíncronas que permitem que componentes do aplicativo solicitem funcionalidade de outros componentes do Android.

Os intents permitem que você interaja com componentes dos mesmos aplicativos, bem como com componentes fornecidos por outros aplicativos.



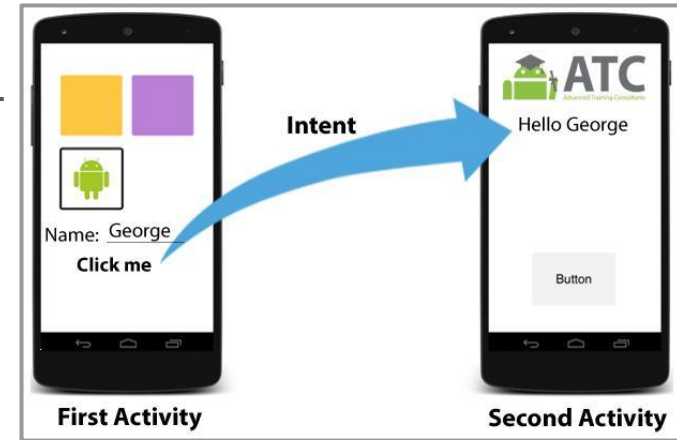


Android Intent

É muito provável que você precise transferir alguns dados para a atividade que deseja iniciar por meio do Intent.

O Android SDK fornece essa opção usando métodos extras.

Você pode anexar dados ao seu intent usando o método **putExtra()** do Intent para incluir dados extras na atividade de chamada ou o método **getStringExtra()** para recuperar dados da atividade chamada; por exemplo, o envio de um dado da primeira atividade para a segunda atividade quando ela começa.



Um objeto Intent é um pacote de informações usado pelo componente que recebe o intent e também as informações usadas pelo sistema Android.



MainActivity.kt x

```
9  class MainActivity : AppCompatActivity() {
10      override fun onCreate(savedInstanceState: Bundle?) {
11          super.onCreate(savedInstanceState)
12          setContentView(R.layout.activity_main)
13
14          val btnTela2 :Button! = this.findViewById<Button>(R.id.btnTela2)
15          btnTela2.setOnClickListener { it: View!
16
17              1 val intent = Intent( packageContext: this, MainActivity2::class.java)
18              val txtFrase :EditText! = this.findViewById<EditText>(R.id.txtFrase)
19              intent.putExtra( name: "frase", txtFrase.text.toString())
20              startActivity(intent)
21          }
22      }
23  }
```



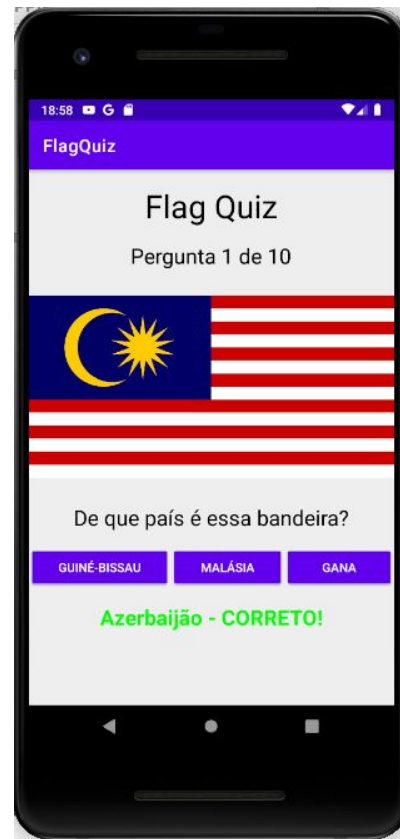
```
MainActivity.kt x MainActivity2.kt x
7  <> class MainActivity2 : AppCompatActivity() {
8  ●↑  override fun onCreate(savedInstanceState: Bundle?) {
9      super.onCreate(savedInstanceState)
10     setContentView(R.layout.activity_main2)
11
12     2 val frase :String? = intent.getStringExtra( name: "frase")
13     val lblFrase :TextView! = this.findViewById<TextView>(R.id.lblFrase)
14     lblFrase.text = frase
15     }
16 }
```




Laboratório



Flag Quiz





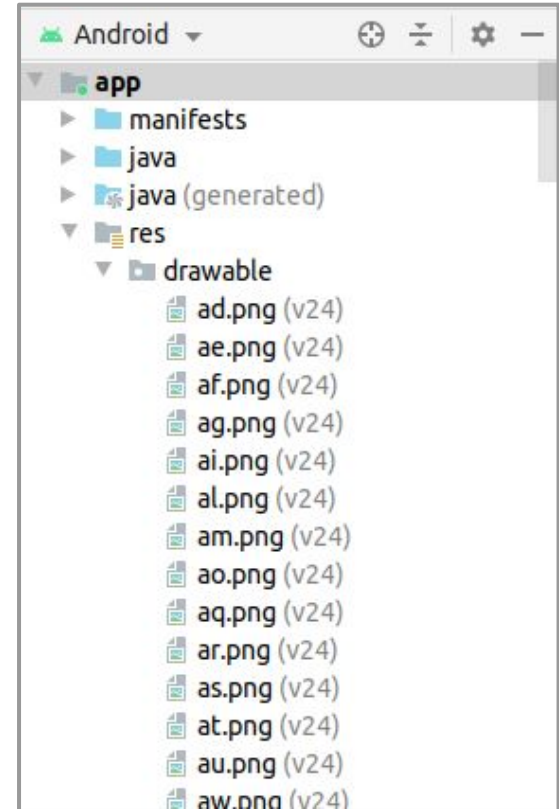
Flag Quiz

Na pasta **res** vamos carregar os arquivos de imagem referentes a cada um dos países, identificados pelo padrão **ISO 639** de duas letras.

No Android temos a classe **Locale** que representa esses países e tem a identificação de duas letras.

O objetivo é poder gerar as perguntas de forma aleatória e carregar uma `ImageView` com essas imagens, dinamicamente.

Além disso, precisamos dos nomes dos países para usar nos botões com as possíveis respostas.





MainActivity.kt x

```
14 class MainActivity : AppCompatActivity(), View.OnClickListener {  
15  
16     private lateinit var países: ArrayList<Locale>  
17     private lateinit var botoes: Array<Button>  
18     private var botaoRespostaCerta = 0
```





```
MainActivity.kt x
20  override fun onCreate(savedInstanceState: Bundle?) {
21      super.onCreate(savedInstanceState)
22      setContentView(R.layout.activity_main)
23      //Carrega os países disponíveis
24      for (locale in Locale.getAvailableLocales()) {
25
26          1      if (locale.country.length == 2) {
27
28              países.add(locale)
29          }
30      }
31      //Monta um array com os botões para facilitar trabalhar com o grupo
32      botões = arrayOf(this.findViewById<Button>(R.id.btn0), this.findViewById<Button>(R.id.btn1),
33                      this.findViewById<Button>(R.id.btn2))
34      for (i in 0..2) {
35
36          2      botões[i].setText("")
37              botões[i].setOnClickListener(this)
38      }
39      this.sorteiaPais()
40  }
```

```
42 private fun sorteiaPais() {
43
44     1 var rodada = IntArray( size: 3)
45     //Sorteio de 3 países diferentes
46     for (i in 0..2) {
47         var numero = 0
48         var ok = false
49         while (!ok) {
50             numero = (Math.random() * paises.size - 1).roundToInt() 2
51             ok = true
52             for (j in 0..2) {
53                 if (rodada[j] == numero) {
54
55                     ok = false 3
56                     break;
57                 }
58             }
59             if (ok) {
60                 val uri = "@drawable/" + paises.get(numero).country.toLowerCase()
61                 4 val resourceId = this.resources.getIdentifier(uri, defType: "drawable", packageName)
62                 if (resourceId == 0) {
63                     ok = false
64                 }
65             }
66         }
67         rodada[i] = numero
68     }
```



MainActivity.kt x

```
69 //Sorteia que botão será o certo desta rodada
70 1 botaoRespostaCerta = (Math.random() * 2).roundToInt()
71 //Carrega a imagem da bandeira
72 val uri = "@drawable/" + pais.es.get(rodada[botaoRespostaCerta]).country.toLowerCase()
73 2 val imageResource = this.resources.getIdentifier(uri, defType: null, packageName)
74 var imgBandeira = this.findViewById<ImageView>(R.id.imgBandeira)
75 imgBandeira.setImageDrawable(resources.getDrawable(imageResource))
76 //Carrega os rótulos dos botões com os nomes dos países
77 3 for (i in 0..2) {
78     botoes[i].setText(pais.es[rodada[i]].displayCountry)
79 }
80
81 }
```



```
MainActivity.kt x
83  override fun onClick(view: View?) {
84
85      val button : Button = view as Button
86      val lblResposta = this.findViewById<TextView>(R.id.lblResposta)
87      if (botoes[botaoRespostaCerta] == button) {
88
89          lblResposta.setTextColor(Color.GREEN)
90          lblResposta.setText(button.text.toString() + " - CORRETO!")
91      } else {
92
93          lblResposta.setTextColor(Color.RED)
94          lblResposta.setText(button.text.toString() + " - ERRADO!")
95      }
96      this.sorteiaPais()
97  }
```




Exercício

Implementar a aplicação Flag Quiz apresentada nesta aula.

Solucionar os problemas de interface, como por exemplo o contraste de tela.