

The background features a complex network of thin grey lines connecting various points, some of which are solid black dots. Scattered throughout are numerous triangles of different sizes and orientations, some outlined in grey and others in white. The overall aesthetic is modern and technical.

PROGRAMA DEV VENTURE

Desenvolvimento Android

**ADD
COMPORTAMENTO**

Manipulação de
interfaces

01

**SHARED
PREFERENCES**

Armazenamento de
dados

02

DESAFIO

Intents de acionamento
do sistema

03

Aula 06 AGENDA



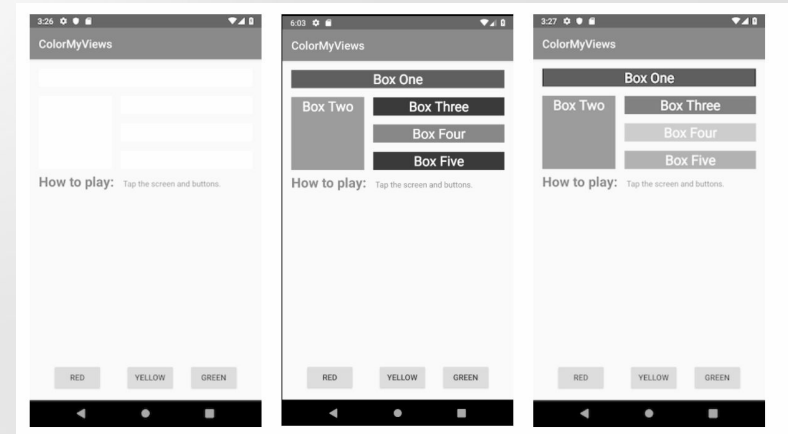
01

ADD COMPORTAMENTO

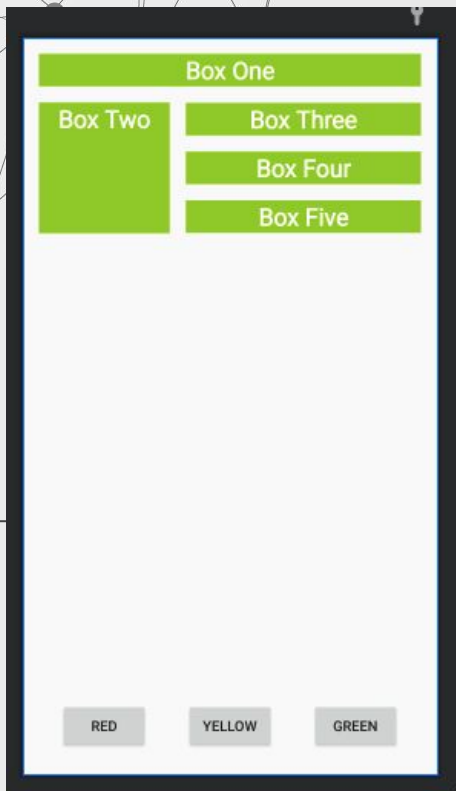


COLOR MY VIEWS

O aplicativo ColorMyViews é inspirado no artista holandês Piet Mondrian. Ele inventou um estilo de pintura chamado neoplasticismo, que usa apenas linhas verticais e horizontais e formas retangulares em preto, branco, cinza e cores primárias.



dojo / BOTÕES



- Adicione ao layout 3 botões
 - `redButton`
 - `yellowButton`
 - `greenButton`
 - Configure os botões para ficarem alinhados entre si verticalmente
 - Configure os botões para que a distância entre eles seja sempre distribuída igualmente
-
- Por fim, conecte os botões com a parte inferior da tela, adicionando uma margem de 24dp

dojo / BOTÕES

- Crie as seguintes cores no arquivo colors.xml
 - `<color name="grey">#BABABA</color>`
 - `<color name="green">#12C700</color>`
 - `<color name="red">#E54304</color>`
 - `<color name="yellow">#FFC107</color>`
- Cora os botões de acordo com seu respectivo nome





MANIPULAÇÃO DE RESOURCES

O comportamento que vamos adicionar será: quando clicarmos no botão "yellow", por exemplo, e em seguida clicarmos na *box one*, então a *box one* ficará amarela.

Para isso, precisaremos na classe MainActivity.kt manipular elementos da view e alguns resources.

Todos itens pertencentes a pasta *res/* são considerados resources

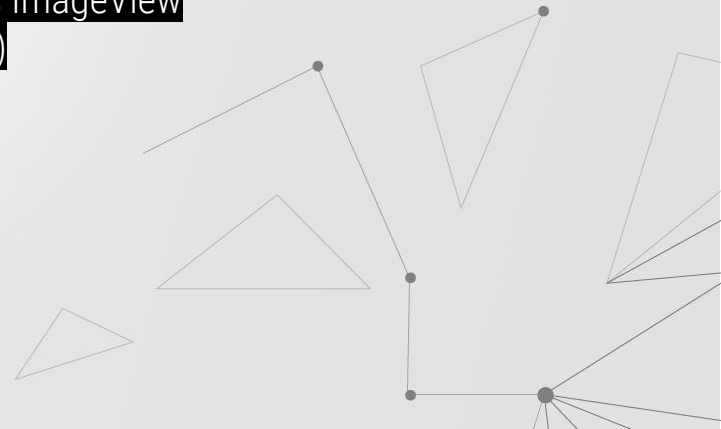




MANIPULAÇÃO DE RESOURCES

É possível usar um *resource* no código passando o código do *resource* como parâmetro do método. Por exemplo, é possível definir uma *ImageView* para usar o recurso ***res/drawable/myimage.png*** usando `setImageResource()`:

```
val imageView = findViewById(R.id.myimageview) as ImageView  
imageView.setImageResource(R.drawable.myimage)
```



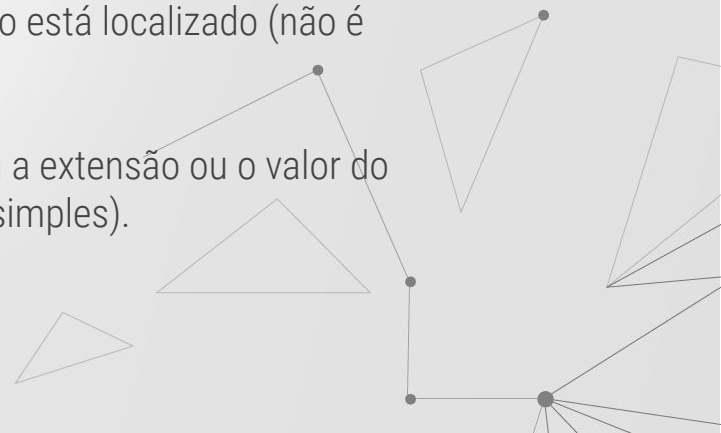


MANIPULAÇÃO DE RESOURCES

Esta é a sintaxe para referenciar um recurso no código:

[<package_name>.]R.<resource_type>.<resource_name>

- **<package_name>** é o nome do pacote em que o recurso está localizado (não é obrigatório ao referenciar recursos de seu pacote).
 - **<resource_type>** é a subclasse R do tipo de recurso.
 - **<resource_name>** é o nome do arquivo do recurso sem a extensão ou o valor do atributo android:name no elemento XML (para valores simples).
-



dojo / CURRENT COLOR

- Crie um mecanismo para armazenar o resource da cor correspondente ao botão que foi clicado.
 - Clique no botão vermelho
 - Cor armazenada: R.color.red
 - Clique no botão amarelo
 - Cor armazenada: R.color.yellow





dojo / COLOR MY VIEW

Adicione a *box_one_text* o seguinte comportamento:

- Quando clicamos na caixa, a cor armazenada deve ser aplicada como cor de fundo.

OBS: uma forma de colorir o fundo de uma view é usando o método `setBackgroundResource()`





dojo / COLOR MY VIEW

Replique este comportamento para todas as caixas.

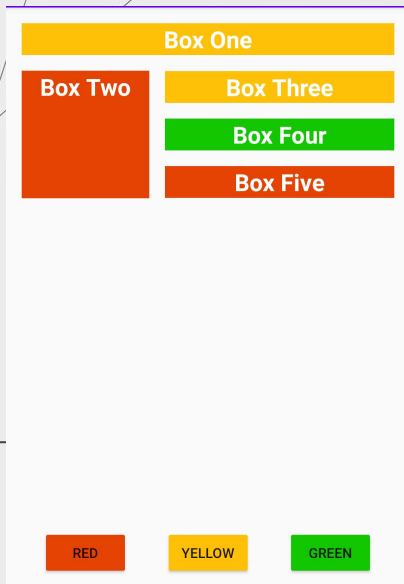


02

SHARED PREFERENCES



CICLO DE VIDA



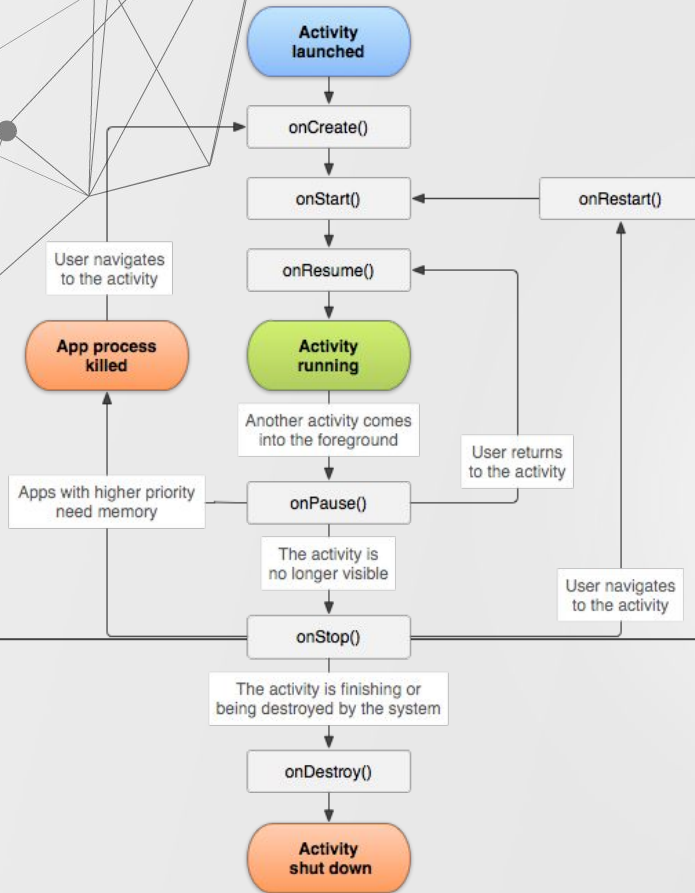
O que acontece se a app for colocada em background depois de colorirmos as caixas?

Por que isso acontece?

CICLO DE VIDA

Principais callbacks do ciclo de vida de uma activity

onCreate()
onStart()
onResume()
onPause
onStop()
onDestroy()

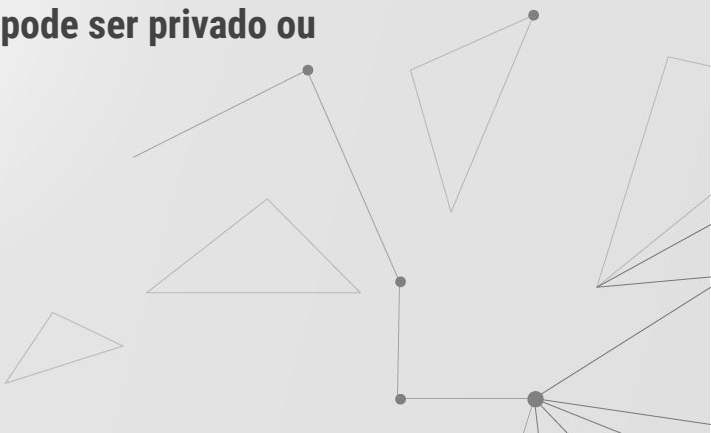




SHARED PREFERENCES

Caso você tenha um conjunto relativamente pequeno de chaves-valor para salvar, as SharedPreferences podem ser um bom jeito de gerenciar o armazenamento.

Um objeto SharedPreferences indica **um arquivo** que contém pares de chave-valor e fornece métodos simples para leitura e gravação. **Cada arquivo SharedPreferences é gerenciado pelo framework e pode ser privado ou compartilhado.**

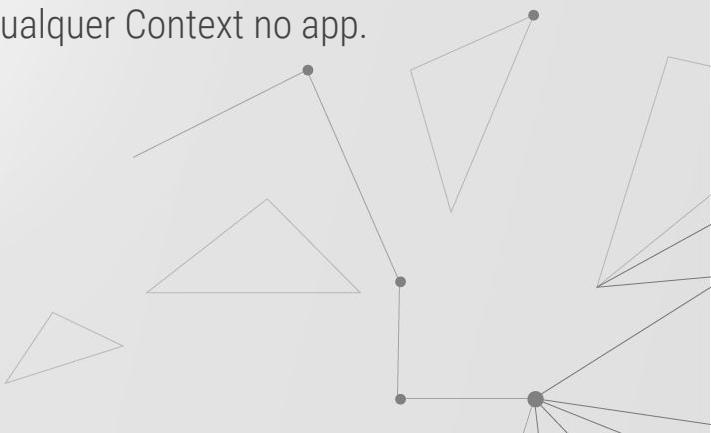




SHARED PREFERENCES

É possível criar um novo arquivo de preferências compartilhadas ou acessar um já existente chamando um dos seguintes métodos:


getSharedPreferences(): use esse método se precisar de vários arquivos de preferências compartilhadas identificados por nome, que são especificados com o primeiro parâmetro. É possível chamá-lo em qualquer Context no app.





SHARED PREFERENCES

getPreferences(): use esse método por uma Activity se precisar usar apenas um arquivo de preferências compartilhadas para a activity. Como ele recupera um arquivo padrão de preferências compartilhadas que pertence à activity, não é necessário fornecer um nome.





SHARED PREFERENCES

Acessar instância do shared preferences

```
val sharedPreferences = activity.getSharedPreferences("colors", Context.MODE_PRIVATE)
```

```
val sharedPreferences = activity.getPreferences(Context.MODE_PRIVATE)
```

O parâmetro colors é o nome do arquivo que estamos criando para armazenamento.





SHARED PREFERENCES

Para salvar dados no shared preferences

```
val sharedPreferences = activity.getSharedPreferences(Context.MODE_PRIVATE)
val editor = sharedPreferences.edit()
editor.putString("chave", valor)
editor.putInt("outra_chave", outroValor)
editor.commit()
```





dojo / SALVAR ESTADO

Como salvar o estado das caixas para que ele não seja perdido quando app for fechada ou colocada em plano de fundo?





dojo / SALVAR ESTADO

- Sobreescreva os métodos
 - onCreate()
 - onStart()
 - onResume()
 - onPause
 - onStop()
 - onDestroy()

Com uma chamada de log (Log.i("lifecycle:", "onCreate"))



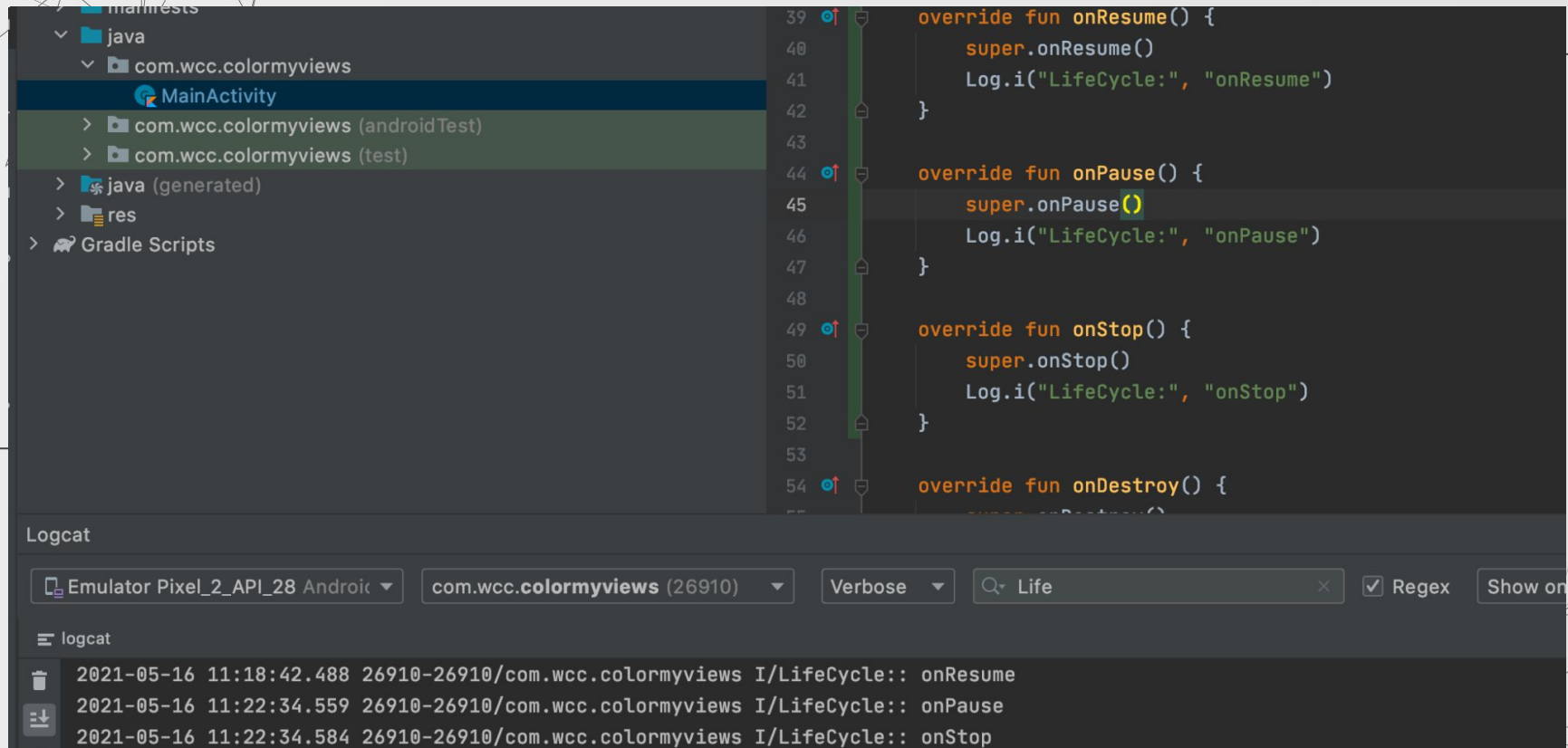


dojo / **SALVAR ESTADO**

- Execute a app e fique de olho na aba Logcat do AndroidStudio
 - Abra e feche a app e verifique os logs
-



dojo / SALVAR ESTADO



The screenshot displays the Android Studio interface. On the left, the Project Explorer shows the package structure: `com.wcc.colormyviews` containing `MainActivity`, `com.wcc.colormyviews (androidTest)`, `com.wcc.colormyviews (test)`, `java (generated)`, `res`, and `Gradle Scripts`. The `MainActivity` file is selected.

The main editor shows the `MainActivity` code with the following lifecycle methods:

```
39 override fun onResume() {  
40     super.onResume()  
41     Log.i("LifeCycle:", "onResume")  
42 }  
43  
44 override fun onPause() {  
45     super.onPause()  
46     Log.i("LifeCycle:", "onPause")  
47 }  
48  
49 override fun onStop() {  
50     super.onStop()  
51     Log.i("LifeCycle:", "onStop")  
52 }  
53  
54 override fun onDestroy() {  
55     super.onDestroy()  
56 }
```

The Logcat window at the bottom shows the following log entries:

```
2021-05-16 11:18:42.488 26910-26910/com.wcc.colormyviews I/LifeCycle:: onResume  
2021-05-16 11:22:34.559 26910-26910/com.wcc.colormyviews I/LifeCycle:: onPause  
2021-05-16 11:22:34.584 26910-26910/com.wcc.colormyviews I/LifeCycle:: onStop
```




dojo / SALVAR ESTADO

Como salvar o estado das caixas para que ele não seja perdido quando app for fechada ou colocada em plano de fundo?





dojo / SALVAR ESTADO

No método `onStop()`:
Salve as cores atuais de cada caixa





dojo / SALVAR ESTADO

No método onCreate():
Recupere as cores de cada caixa



03

DESAFIO





SHARE

Em duplas, incremente a app com uma nova funcionalidade: compartilhar a obra de arte criada!

Dicas:

Talvez você precise:

- Criar um bitmap a partir da view que você queira compartilhar
 - Salvar o bitmap criado
 - Usar intents para enviar o bitmap salvo
 - Extra: algumas extras permissões possam ser necessárias
-



Palavras chave da aula de hoje:

- CICLO DE VIDA
- LOGCAT
- SHARED PREFERENCES
- INTENTS
- VAR
- GET/SET

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.