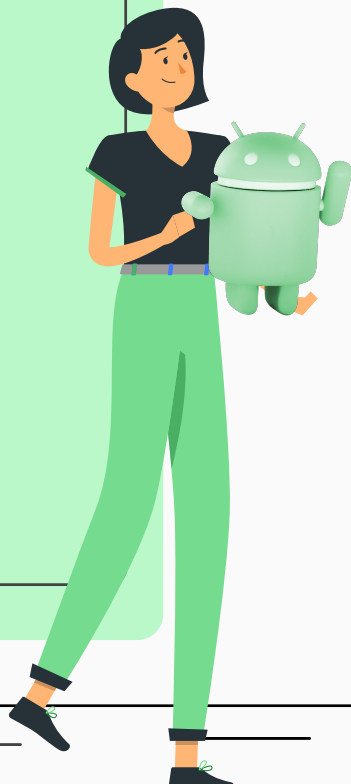


AULA IO

Interfaces interativas & SOLID





01 COLOR MY VIEWS

Evolução do app

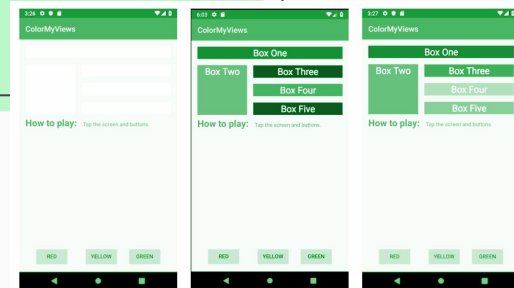
02 DESAFIO

Persistindo dados com
Shared Preferences

02 VINCULANDO AÇÕES

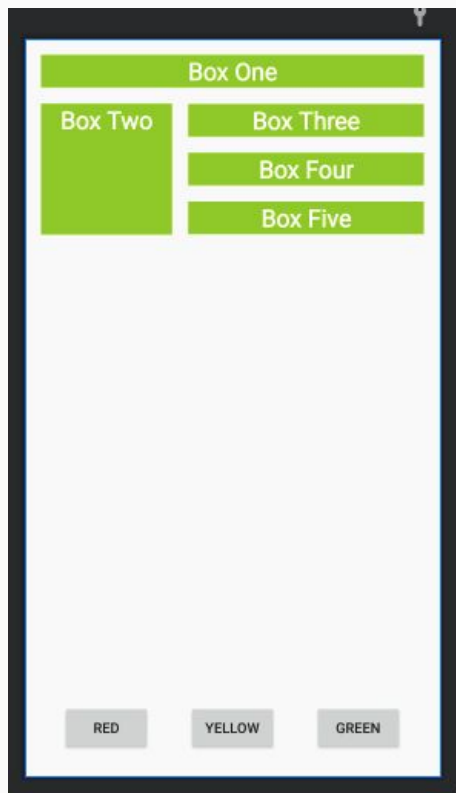
COLORMYVIEWS

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.



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

BOTÕES COLORIR VIEWS



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>
```

Adicione nos botões o seguinte comportamento:

1. Crie uma variável `currentColor`, e a inicialize com a cor cinza
2. Ao clicar no botão vermelho, a variável `currentColor` recebe o valor da cor red que definimos nos resources
3. Ao clicar no botão amarelo, a variável `currentColor` recebe o valor da cor yellow que definimos nos resources
4. Ao clicar no botão verde, a variável `currentColor` recebe o valor da cor green que definimos nos resources

OBS: lembre-se que a classe R nos permite acesso aos resources do app (R.color.red é uma forma de acessar o valor da cor vermelha que foi declarada no arquivo color.xml)

Adicione a `box_one_text` o seguinte comportamento:

Quando clicamos na caixa, a cor armazenada em *currentColor* deve ser aplicada como fundo da dela.

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

DESAFIO

Ajuste os botões de cor para que eles tenham a mesma cor de fundo que eles representam.



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.

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.

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.

Para salvar itens no shared preferences:

1) Acessar instância do shared preferences

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

Ou

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

Para salvar itens no shared preferences:

2) Acessar editor do shared preferences

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

Para salvar itens no shared preferences:

3) Adicionar itens

```
editor.putString("chave", valor)
```

```
editor.putInt("outra_chave", outroValor)
```

Para salvar itens no shared preferences:

4) Committed mudanças
editor.commit()

Para acessar itens do shared preferences:

1) Acessar instância do shared preferences

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

2) Recuperar valor

```
val value = sharedPreferences.getString("chave", valorDefaultCasoNaoAcheChave)
```

Salve a cor definida na `box_one_text`

Dica: você pode executar a operação de salvar no shared preferences, apenas quando o aplicativo for fechado.