Desenvolvimento Mobile

Plataforma Android

Prof. Dr. Marcelo Otone Aguiar

Universidade Federal do Espírito Santo - UFES

5 de Novembro de 2024

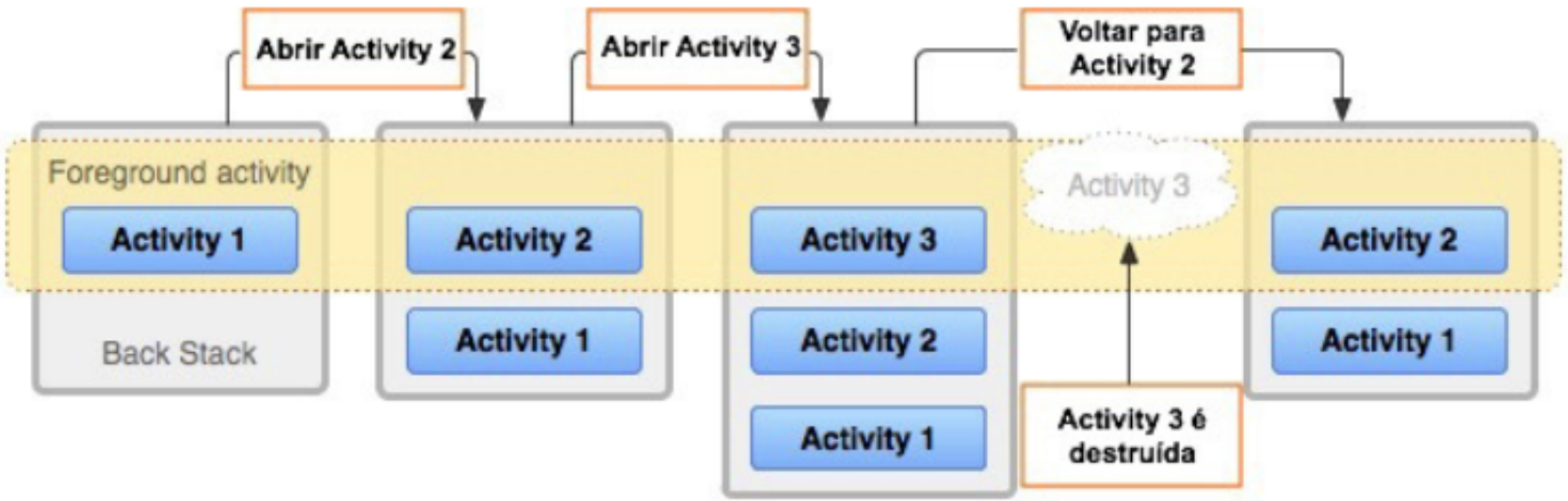
Conteúdo

Activity

- A **Activity** é responsável pela representação gráfica da tela do **app**
- É nela que adicionamos os componentes visuais de interação com usuário, como botões, caixas de texto, de seleção, entre outros.
- É comum um **app** possuir várias **Activities**, mas nada impede de possuir apenas uma, vai depender do problema abordado
- Uma **Activity** pode iniciar outras **Activities**

Activity

- Quando uma **Activity** inicia outra, a **Activity** iniciada é empilhada sobre a principal
- A medida que o botão voltar é pressionado, as **Activities** vão sendo desempilhadas
- A imagem a seguir ilustra esse fluxo de funcionamento:



Fonte: Resende (2023)

Activity

- Um erro muito comum entre desenvolvedores que não estão habituados com esse conceito, é, ao invés de desempilhar a **Activity** atual para voltar a anterior, abrir novamente a **Activity anterior**
- E, embora, na prática isso possa até funcionar, é algo indesejado, pois, criará uma situação em que a memória RAM do dispositivo poderá sobrecarregar, uma vez que, as **Activities** não são removidas da pilha de navegação

Ativando o Recurso *View Binding*

- Um recurso disponível desde a versão **Android Studio 3.6**, denominado como *View Binding* facilita o acesso a elementos da interface gráfica (**views**) no código, evitando a necessidade de usar o método *findViewById()*
- Quando o *View Binding* está ativado, o **Android Studio** gera automaticamente uma classe de *binding* para cada arquivo de layout XML, associando-o aos elementos que ele contém
- Assim, em vez de localizar manualmente cada **view**, você pode acessar os elementos da interface com segurança por meio das propriedades da classe de *binding*
- Isso reduz a probabilidade de erros, como tentativas de acessar **views** nulas, e torna o código mais claro e seguro

Ativando o Recurso *View Binding*

Exemplo usando *findViewById()*

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

    val textView: TextView = findViewById(R.id.textView)
    val button: Button = findViewById(R.id.button)

    textView.text = "Hello, World!"
```

Exemplo usando *View Binding*

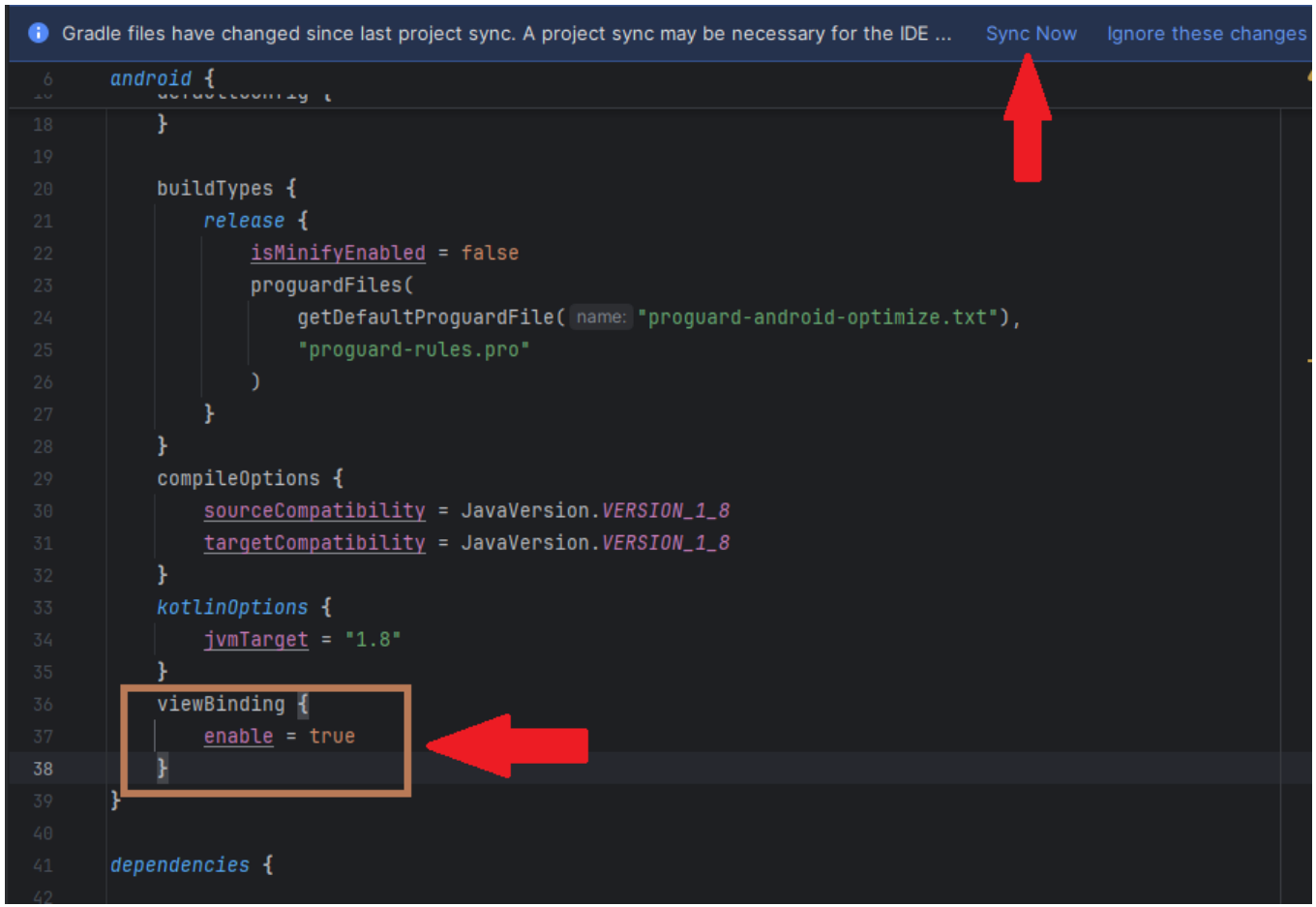
```
private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    binding = ActivityMainBinding.inflate(layoutInflater)
    setContentView(binding.root)

    binding.textView.text = "Hello, World!"
```

Ativando o Recurso *View Binding*

- Para ativar, abra o arquivo: build.gradle.kts (Module:app)
- Adicione a configuração conforme a imagem a seguir e clique em ***Sync Now***



Referências

- RESENDE, K. **Kotlin com Android:** Crie aplicativos de maneira fácil e divertida. São Paulo: Casa do Código, 2023.