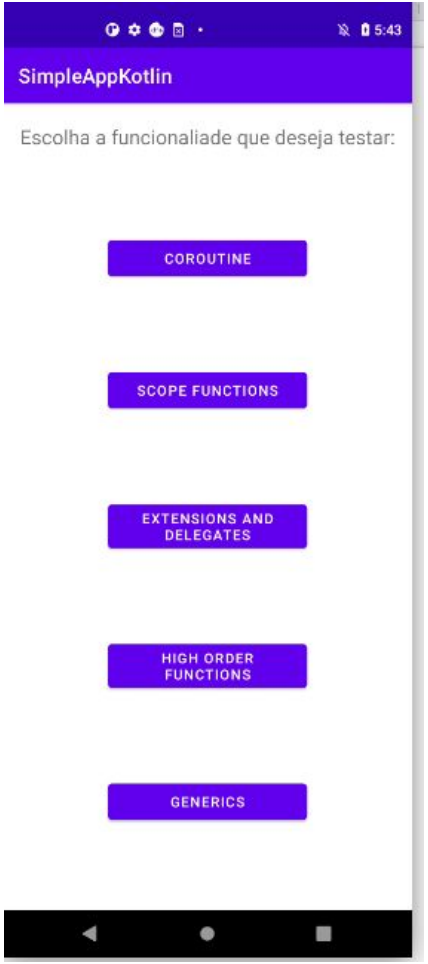
Desafío 2 - Prática Kotlin

Caio Viel
caioviel@gmail.com

Activity Inicial - Escolha uma das opções

- São 5 aplicações demos diferentes

- Cada uma ilustra a utilização de uma ou mais das features solicitadas no desafio



Coroutine

- Uma **Animation** de rotação é exibida na tela.

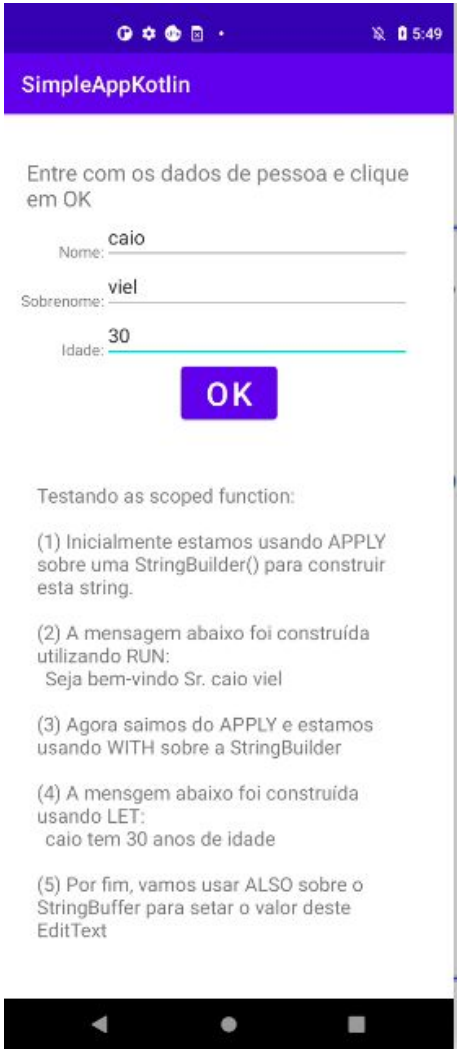
- Ao clicar no **Button** “Rodar em uma Corotina”, uma **List** com 100.000 elementos é criada e não há impacto na animação.

- Ao clicar no **Button** “Rodar na Thread Gráfica”, a mesma **List** é criada, porém há um pequeno travamento na animação

Scoped Functions

- Após apertar o botão OK, uma instância da classe **Person** é criada com os dados entrados nos **EditText**.

- Abaixo do botão, um **TextView** tem seu conteúdo modificado pelo usuário das scoped function **apply**, **run**, **with**, **let** e also sobre a instância de **Person** e um **StringBuilder**.



Extentions e Delegates

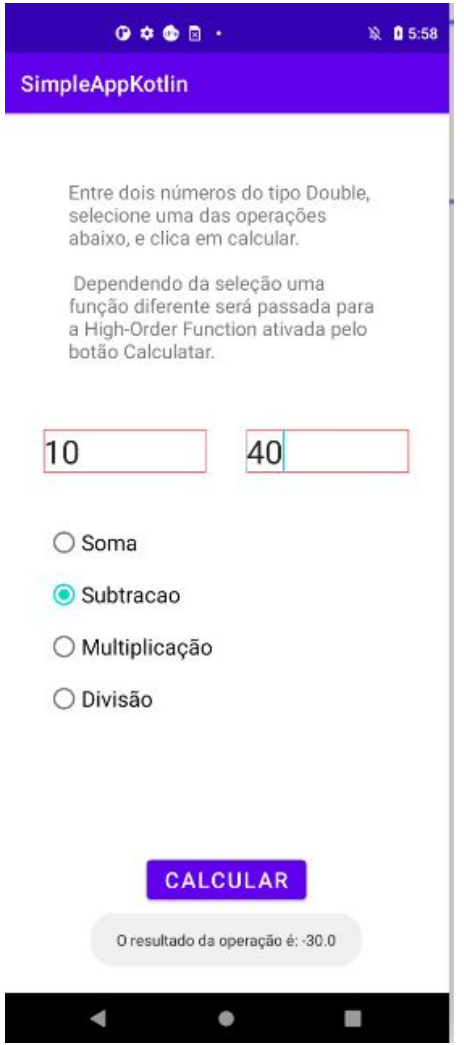
- Um novo atributo do tipo booleano (**isUnderage**) é adicionado via extention na classe **Person**. O comportamento deste atributo é delegado a classe **UnderageCheck**

- Se Age < 18, **isUnderage** é **true**; caso contrário **falso**.

- Preenchendo o formulário e clicando no **Button** OK, uma instância de **Person** é criada e através do atributo **isUnderage**, um **Toast** informativo diz se a pessoa é menor de idade ou não.

High-Order Functions

- Após entrar dois valores número e clicar no **Button** Calcular, a high-order function **performOperation** é chamada, realizando a operação e exibindo um **Toast** na tela com o resultado.
- **performOperation** recebe como argumento dois valores e a função que ela chama dentro dela com os valores recebidos.
- A função passada como parâmetro é do tipo *(Double, Double) -> Double*, e é selecionado dependendo **RadioButto**n selecionado.
- Foi adicionado um novo método (extensions) no **RadioGroup** para ajudar a encontrar a opção selecionada.



Generics

- A aplicação simula um console
- Foi criado um método genérico screenlog para printar informações neste console

```
fun <T> screenlog(obj : T) {
    val timeStamp: String = SimpleDateFormat( pattern: "yyyyMMdd_HHmmss").format(Date())
    var sb = StringBuilder().apply {   this: StringBuilder
        append(console.text.toString())
        append("\n")
        append("$timeStamp - ${obj.toString()}")
        append("\n")
    }
    console.text = sb.toString()
}
```

- Diversos tipos de objetos são então passados como paramêtro para a função genérica