



Insert Koin @ Cookpad

Eduardo Carrara

Desenvolvedor Android



@DuCarrara





"Make everyday cooking fun!"

10,000,000+ Play Store Downloads

2,5+ milhões de receitas caseiras publicadas

20+ regiões/países

Venha trabalhar conosco! (cookpad.workable.com)



@DuCarrara



O Início: Android Camp 2018*



*Na verdade começou antes em discussões em issues do Github issues, mas finja comigo pela história 😊



@DuCarrara



"Devemos usar um framework de DI?"

-  *Tópico quente no Android Camp
2018*



@DuCarrara



Se não há framework de DI...

... como fazemos?



@DuCarrara



Plain Old **Interfaces** +
Objetos Singleton +
Funções Estáticas



@DuCarrara



🤔 O que queríamos resolver afinal?



@DuCarrara



Reduzir **Ambiguidade**

Entender nossa árvore de dependências

Facilmente conectar as coisas



@DuCarrara



~~"Devemos utilizar um framework de DI?"~~

"Um framework de DI nos ajudaria?"

👍 Vale a pena tentar! **Mas qual usar???**



@DuCarrara



Selecionando o Framework



Um *Especialista*



Avaliação Técnica



Votação



@DuCarrara



	DAGGER2	KOIN	KODEIN
	JAVA	KOTLIN	KOTLIN
Pros	<p>Conhecimento no Time Robusto</p> <p>Muitos usuários</p> <p>Mantido pelo Google</p> <p>Muito Rápido*</p> <p>APIs para Android</p>	<p>Feito com Kotlin</p> <p>Ótimas Docs</p> <p>Comunidade Crescendo</p> <p>Muito Simples</p> <p>APIs para Android</p>	<p>Feito com Kotlin</p> <p>Docs Muito Boas</p> <p>Alguma Comunidade</p> <p>Simples</p> <p>APIs para Android</p>
Cons	<p>Setup pode ser assustador</p> <p>Difícil de Debugar</p> <p>Aumento no tempo de build</p>	<p>Em evolução</p> <p>Sem verificação em tempo de build</p> <p>Mais Lento*</p>	<p>Em evolução</p> <p>Sem verificação em tempo de build</p> <p>Muito mais lento*</p>

*Performance Reference: <https://github.com/Sloy/android-dependency-injection-performance>



@DuCarrara



Finalmente, **por que** Koin? 



@DuCarrara



Feito com **Kotlin** ❤️

Tentar algo **Novo e Simples**

Em caso de 💣 mudamos de novo 🙄



@DuCarrara



Criamos um **Plano de 3 Passos**



@DuCarrara



1. Limpar/Organizar Dependências



@DuCarrara





2. **Aplicar Koin em um subconjunto** da base & avaliar



@DuCarrara




3. Adotar Koin no codebase todo



@DuCarrara



Quão simples é o Koin? 

Vamos ver o **Básico do Koin...**



@DuCarrara



Configurando Dependências

```
1 interface BaseRepository
2 class Repository : BaseRepository
3 class UseCase(val repository: BaseRepository)
4 class OneViewModel(val useCase: UseCase) : ViewModel()
5
6 val featureModule = module {
7
8     single<BaseRepository>(named("MyRepository")) { Repository() }
9
10    factory { UseCase(repository = get(named("MyRepository"))) }
11
12    viewModel { OneViewModel(useCase = get()) }
13
14 }
```



@DuCarrara



Inicializando no Android

```
1 class YourApplication : Application() {  
2     override fun onCreate() {  
3         super.onCreate()  
4  
5         startKoin {  
6             androidContext(this@YourApplication)  
7             modules(networkingModule, dataModule, viewModule)  
8         }  
9     }  
10 }
```



@DuCarrara



Provendo Dependências

```
1 class OneActivity : AppCompatActivity() {  
2  
3     val viewModel: OneViewModel by viewModel()  
4  
5     val useCase: UseCase by inject()  
6  
7     val repository: BaseRepository = get(named("MyRepository"))  
8  
9 }
```



@DuCarrara



KoinComponent

```
1 class ConsumerClass : KoinComponent {  
2  
3     val dataRepository: DataRepository = get()  
4  
5     val presenter: Presenter by inject()  
6  
7 }
```



@DuCarrara



Koin @  **cookpad**



@DuCarrara



Nosso Time & Codebase

Agosto 2018	Junho 2019
13 Android Devs	17 Android Devs
20 modules	21 modules
+68K LoC	+92K LoC
Internal Dep. Resolution Solution	Koin 2.0 RC-3
MVP	MVP -> MVVM
100% Kotlin	

Codebase com ~4.5 anos de idade



@DuCarrara



Quão grande ficou?

26 Koin Modules

~181 *single*, **~76** *factory* and **~29** *viewModel*

39 KoinComponents Implementations

132 *by inject* dependency requests



@DuCarrara



Impactos em **Performance***?

Tempo de Setup do Koin: **~443.28ms**

Inicialização do App**: De **1.88s** Para **2.75s**

Memória depois da Inic: De **~48MB** Para **~47MB**

*Measured in Android Emulator API 28, 1.5GB RAM, App Release Version using Debug class facilities and AS Profiler

**From Application.onCreate to LoginActivity.onResume



@DuCarrara



O que aprendemos?



@DuCarrara



Modularização ainda pode ser complicada



@DuCarrara



Entenda o **KoinComponent**



@DuCarrara



Acoplamento com Koin



@DuCarrara



Cuidado com os **Crashes em Runtime**



@DuCarrara



Valeu a pena?



@DuCarrara



Feedback do Time

75% está **satisfeito**

100% concorda que Koin is **fácil de usar**

75% acha que ainda **podemos melhorar**



@DuCarrara



Próximos Passos

Explorar **escopos**

Melhorar o uso de **modules**

Avaliação de **performance** profunda

Explorar mais **Testes**



@DuCarrara



Koin valeu a pena!



@DuCarrara



Obrigado!
Thanks!



Eduardo Carrara

@DuCarrara

github.com/ecarrara-araujo



@DuCarrara



Referências

1. [Koin Getting Started Documentation](#)
2. [Ready for Koin 2.0](#)
3. [Koin Reference Documentation](#)
4. [Simplify your Android Development with Koin \(YouTube\)](#)



@DuCarrara

