



Injeção de Dependência com Koin

O que é injeção de dependência?

Injeção de dependência significa dar a um objeto suas instâncias de variáveis. É isso.

Falamos sobre isso na PARTE 2 da nossa aula sobre os **6 design patterns que todo desenvolvedor Android deveria saber!**

Caso não tenha visto essa aula, assista depois.

Mão na massa!

Vamos criar um exemplo MVVM com injeção de dependência manual!

1. Adicionar o botão no layout inicial e ativar o ViewBinding no projeto.
2. Adicionar as dependências iniciais do projeto.

```
// Retrofit
implementation "com.squareup.retrofit2:retrofit:2.9.0"
implementation "com.squareup.retrofit2:converter-gson:2.9.0"

//viewModelScope
implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.5.1"
```

3. Criar a implementação da API usando Retrofit.

- a. Se você não conhece como usar o Retrofit, assista as aulas [#013](#) e [#014](#)
- b. Vamos criar a model da resposta
- c. Vamos criar a interface da API
- d. Vamos criar o repositório
- e. Vamos criar a viewModel
- f. Vamos criar a factory da viewModel
- g. Vamos implementar a logica na viewModel
- h. Vamos utilizar a viewModel na Activity

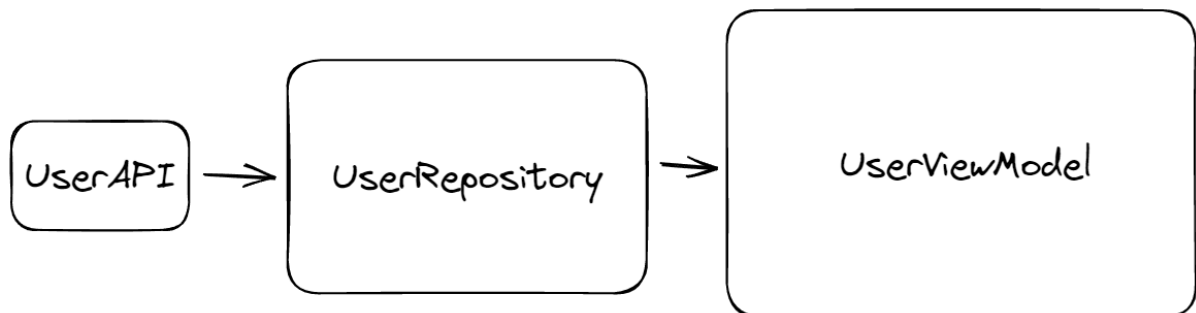
Pronto! Temos um app MVVM com Coroutines e Retrofit usando injeção de dependência manual!

Agora vamos ver como refatorar esse código para utilizar o framework do Koin!

1. Adicionar as dependências do Koin no projeto

```
// Koin  
implementation "io.insert-koin:koin-android:3.3.3"
```

2. Criação dos módulos e utilização do Koin



Singleton: Uma instância sempre!

Factory: Uma nova instância sempre que chamado!

3. Ativando Logger para debug das operações do Koin
4. Crie um arquivo AppModule para centralizar os módulos

5. Qualificadores e Escopos