

The background features several overlapping, organic, grey shapes that resemble stones or abstract forms. These shapes are positioned in the lower half of the frame, with the central text overlaid on them.

Koin — Hilt - Dagger



Injeção de dependência

Injeção de dependência é um padrão que ajuda a gerenciar o acoplamento de componentes em um aplicativo. Ele torna o código mais modular e fácil de testar, permitindo injetar dependências em um componente a partir de uma fonte externa. Isso reduz a duplicação de código e aumenta a coesão entre os componentes, resultando em um código mais limpo e organizado. A Injeção de dependência é amplamente usada no desenvolvimento de aplicativos Android e é uma prática recomendada para projetos de software em geral.



Koin

Koin é uma biblioteca de injeção de dependência mais simples e fácil de usar que não requer processamento em tempo de compilação. Ele usa Kotlin DSL para definir módulos de injeção de dependência, tornando a configuração mais rápida e fácil. É uma boa opção para projetos menores ou médios.

Apps: Ifood, Mercado Livre, Nubank

*Kotlin DSL significa "Domain-Specific Language" (Linguagem Específica do Domínio) em Kotlin.

É uma forma de criar uma linguagem de programação customizada para um domínio específico dentro de um projeto

Koin – Exemplo de Implementação

Crie um módulo Koin para fornecer as dependências necessárias. Por exemplo, vamos supor que precisamos injetar uma instância da classe **MyRepository** em uma ViewModel:

```
val myModule = module {  
    single { MyRepository() }  
    viewModel { MyViewModel(get()) }  
}
```

Neste exemplo, **MyRepository** é uma classe que representa a camada de acesso a dados da aplicação, e **MyViewModel** é uma ViewModel que depende da instância de **MyRepository**

Koin – Exemplo de Implementação

Inicie o Koin no início do aplicativo, geralmente no método **onCreate** da classe **Application**:

```
class MyApplication : Application() {  
  
    override fun onCreate() {  
        super.onCreate()  
        startKoin {  
            androidLogger()  
            androidContext(this@MyApplication)  
            modules(myModule)  
        }  
    }  
}
```

Koin – Exemplo de Implementação

Agora você pode injetar as dependências em qualquer lugar do aplicativo usando a função **get()** do Koin. Por exemplo, para obter uma instância de **MyViewModel** em uma Activity:

```
class MainActivity : AppCompatActivity() {  
  
    private val viewModel: MyViewModel by viewModel()  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```



Dagger

Dagger é uma biblioteca de injeção de dependência poderosa e flexível que usa anotações de processamento em tempo de compilação para gerar código de injeção de dependência altamente otimizado. Ele é adequado para projetos grandes e complexos, mas pode ter uma curva de aprendizado íngreme.

Apps: Google Maps, Google Pay, Trello

Hilt é uma biblioteca de injeção de dependência oficial do Google para o desenvolvimento de aplicativos Android que usa o Dagger como mecanismo subjacente. É projetado para simplificar a configuração do Dagger e reduzir a quantidade de código necessário para usar a injeção de dependência. Com o Hilt, podemos definir módulos e componentes de injeção de dependência em nosso aplicativo usando anotações e sem a necessidade de escrever código repetitivo.

Apps: Banco Inter, C6 Bank, PicPay



Conclusão

Koin é uma escolha sólida para aplicativos menores e desenvolvedores que procuram uma biblioteca fácil de usar. Dagger é a melhor escolha para aplicativos maiores que exigem alta performance e escalabilidade. Hilt é uma opção intermediária que oferece um equilíbrio entre facilidade de uso e desempenho, enquanto ainda sendo otimizado para o uso com Kotlin.