Android TechTalk
Cássio Bruzasco

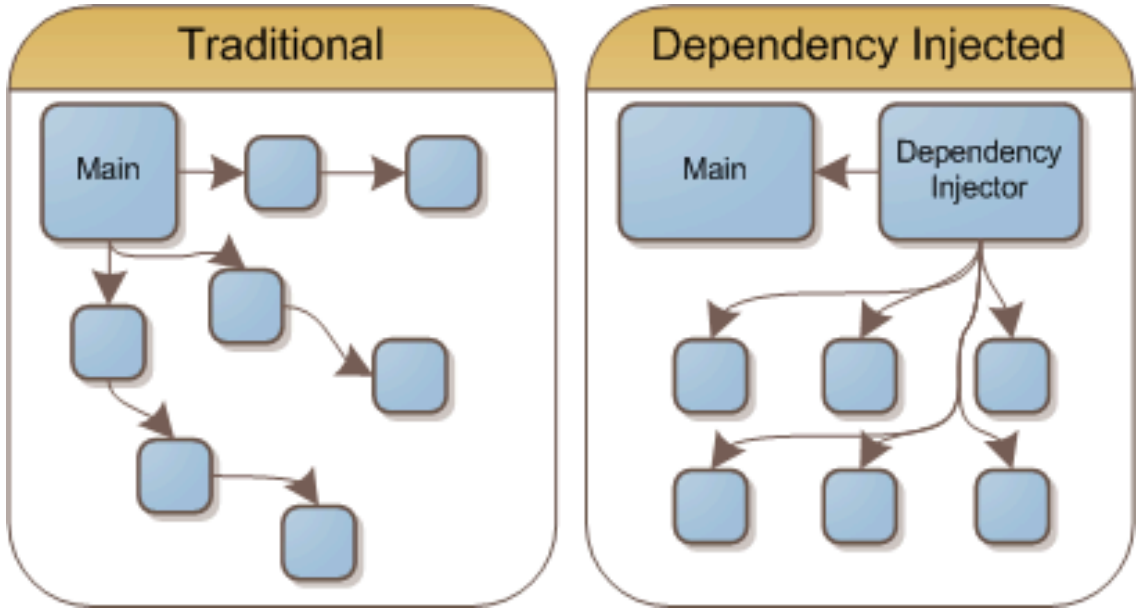
Observer pattern / Kotlin Flows e injeção de dependência.

Credenciais

- 9 anos trabalhando com desenvolvimento Android
- Product Owner, Tech lead, Digital Architect e Software Engineer
- 5 anos de Venturus entre idas e vindas
- 2 apps pessoais publicados e 1 app em desenvolvimento

Injeção de dependência

Injeção de dependência



- Koin não é um injetor de dependência. Koin utiliza da estratégia de *Service Locator Pattern*
- Grande ganho em testes unitários
- Uso mais inteligente de memória
- Facilidade de comunicação entre submódulos

	build time	runtime performance
Koin	no impact	impact
Dagger	impact	no impact

Observer Pattern

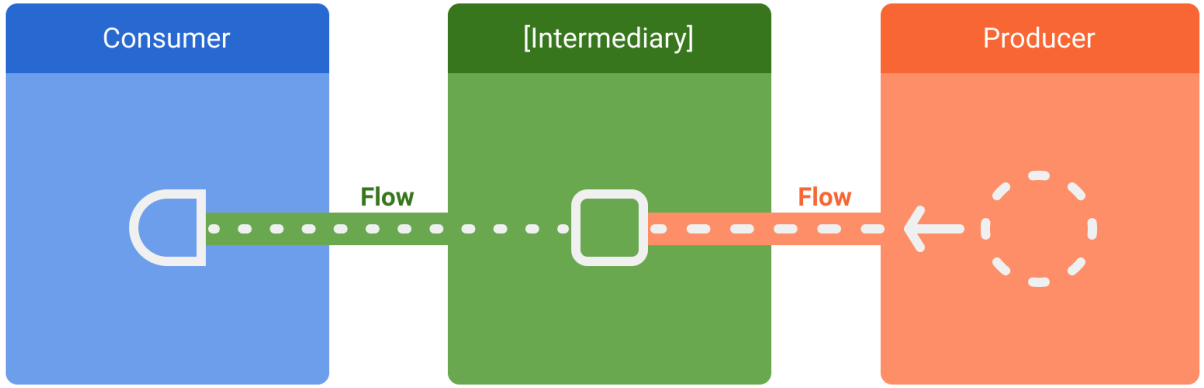
Extraindo o máximo de Flows

- Flow é parecido com um Iterator, emite resultados em sequência porém de forma assincrona.

- Emit -> Adiciona um dado a stream de dados

  Operadores -> Modificam um valor emitido para ser coletado posteriormente (OnEach, map, ...)

- Collect -> Operador final que coleta todos os dados da stream.



Producer: Produz a stream de dados de forma assincrona

Intermediary: Modifica essa stream conforme desejado (Opcional)

Consumer: Consome o dado da stream de dados

Operadores úteis

filter() -> filtra resultado de um flow pelo predicado desejado

merge() -> junta o resultado de um flow em outro flow sem preservar a ordem

zip() -> faz dois ou mais flows acontecerem em paralelo

OnEach() -> executa após um valor ser emitido no upstream

retry() -> tenta executar o flow novamente x vezes se a exception for a desejada

catch() -> processa uma exception que ocorreu na execução do flow

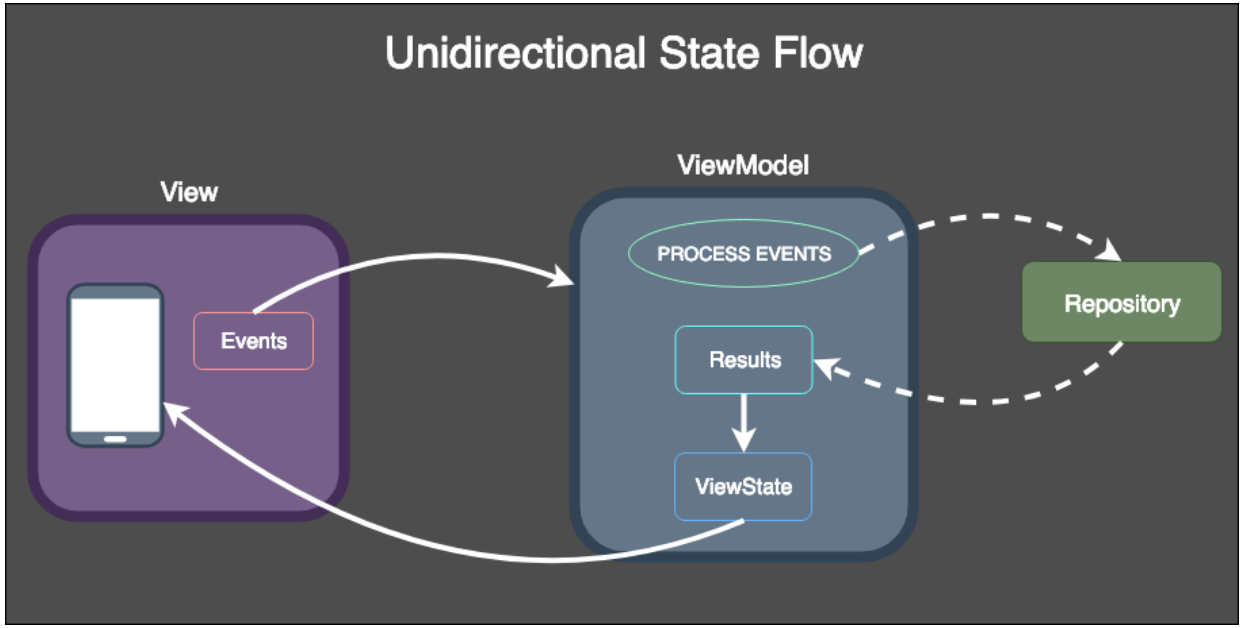
collect() -> Operador terminal que coleta o valor emitido ignorando todo o resto

collectLatest() -> Grande diferença do collect() é que se for emitido um valor novo enquanto esse operador foi chamado, ele bloqueia o antigo emit e só coleta o novo valor.

e muito mais...

Programação reativa

- Uso de LiveData não é aconselhado em projetos Compose. Funciona, mas não é o ideal.
- Coroutines e Flows são altamente recomendados pelo Google para se comunicar entre as camadas da aplicação.
- *StateFlow é state aware* e retorna valores em sequência.
- *Flow no geral tem muitos operadores que podemos aproveitar e facilitar nossa vida. Como visto anteriormente.*



Demonstração

github.com/cassiobruzasco/VntTechTalkCompose