

Em coroutines, um *flow* é um tipo que pode emitir vários valores sequencialmente, ao contrário das *funções de suspensão*, que retornam somente um valor. Por exemplo, você pode usar um flow para receber atualizações em tempo real de um banco de dados.

Os flows são criados com base nas coroutines e podem fornecer vários valores. Conceitualmente, um fluxo é um *stream de dados* que pode ser computado de forma assíncrona. Os valores emitidos precisam ser do mesmo tipo. Por exemplo, um `Flow<Int>` é um flow que emite valores inteiros.

Um flow é muito semelhante a um `Iterator` que produz uma sequência de valores, mas usa funções de suspensão para produzir e consumir valores de maneira assíncrona. Isso significa, por exemplo, que o fluxo pode fazer uma solicitação de rede com segurança para produzir o próximo valor sem bloquear a linha de execução principal.

Há três entidades envolvidas em streams de dados:

- Um **produtor** produz dados que são adicionados ao stream. Graças às coroutines, os flows também podem produzir dados de maneira assíncrona.
- **Intermediários (opcionais)** podem modificar cada valor emitido para o stream ou o próprio stream.
- Um **consumidor** consome os valores do stream.

No Android, uma *fonte de dados* ou um *repositório* normalmente são produtores de dados de UI que têm o `view` como o consumidor que exibe os dados. Outras vezes, a camada `view` é um produtor de eventos de entrada do usuário e outras camadas da hierarquia os consomem. As camadas entre o produtor e o consumidor geralmente agem como intermediárias que modificam o stream de dados para ajustá-lo aos requisitos da camada seguinte.

Por hoje é só pessoal, sucesso nos estudos!