

Faça como eu fiz

Salvando produtos no Firestore

Ajuste o fragment do formulário para criar um produto ao clicar em salvar. Então, envie para o Firestore a partir dos seguintes passos:

- Crie um ViewModel para o formulário que receba um repositório de produto via construtor
- Crie o método para salvar no ViewModel que envia o produto para o repositório
- Notifique o LiveData ao receber a resposta de sucesso do Firestore com true e false caso falhar
- No Fragment, volte para a lista de produtos caso o LiveData retorne true e apresente uma mensagem de falha caso apresente false

Para testar e conferir a mudança na lista de produtos, migre a busca para onResume .

Buscando em tempo real com listener

Ajuste a busca de produtos para atualizar automaticamente em tempo real. Para isso, utilize o addOnSnapshotListener , em vez do get() , ao buscar os documentos da coleção.

Então, chame novamente a busca de produtos no onCreate do Fragment da lista de produtos e confira se a lista é atualizada automaticamente.

Refatorando o código

Por fim, refatore o código considerando as seguintes técnicas:

- Extração de métodos ou constantes
- Remoção de anotações

Opcionalmente, você pode utilizar scope function ou map para simplificar a implementação.

Você pode verificar sobre [scope functions \(https://cursos.alura.com.br/course/kotlin-recursos-do-paradigma-funcional\)](https://cursos.alura.com.br/course/kotlin-recursos-do-paradigma-funcional) ou [map \(https://cursos.alura.com.br/course/kotlin-collections-set-map\)](https://cursos.alura.com.br/course/kotlin-collections-set-map) nos cursos da Alura.

Por fim, teste o app e confira se apresenta os comportamentos esperados.