

## Faça como eu fiz

### Listando produtos no app

Migre o código de teste com o Firestore para o repositório de produto e, então, faça as seguintes implementações:

- Integre a busca com um `LiveData<List<Produto>>`
- Ao identificar o sucesso na busca, converta a lista de documentos para uma lista de produtos
- Atualize o `LiveData` após criar a lista de produtos
- Modifique o `ViewModel` de produto para usar o método de busca do Firestore

Confira se, ao executar o projeto, é apresentada a lista de produtos do Firestore.

### Melhorando a conversão de documentos para objeto

Crie uma classe para representar o produto como um documento e, então, utilize o `toObject` do `DocumentSnapshot` para converter a classe que representa o documento do produto para `Produto`.

Teste o app e confira se apresenta a lista de produtos sem problemas.

### Adicionando o fluxo de tela para o cadastro

Adicione um `FAB` na lista de produtos, crie o `Fragment` para o formulário de produto e, então, [utilize este layout no Fragment \(https://github.com/alura-cursos/firebase-cloud-firestore-android/commit/f73aa8fb926a5db6cfd9ecb6ff53a5f8284e53f7\)](https://github.com/alura-cursos/firebase-cloud-firestore-android/commit/f73aa8fb926a5db6cfd9ecb6ff53a5f8284e53f7). Com tudo pronto, integre ambas telas com os seguintes passos:

- Configure o novo destino do `Fragment` adicionando também a ação da lista de produtos para o formulário;
- Configure o `FAB` para acessar o formulário ao clicá-lo.

Teste o app e confira se é possível acessar a tela formulário ao clicar no `FAB` na lista de produtos.