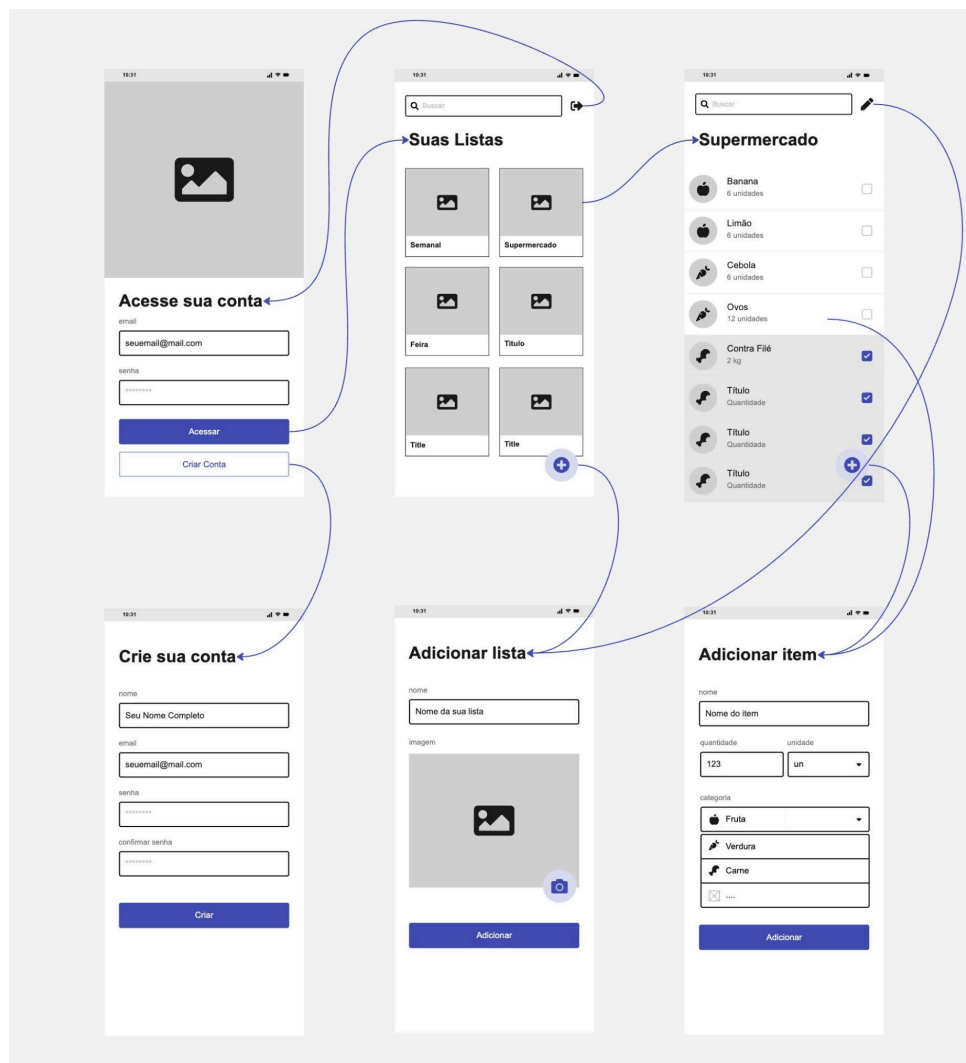


Projeto Prático (Final)

Objetivo

Desenvolver um **Aplicativo de Lista de Compras** para Android Nativo com Kotlin, aplicando arquitetura MVVM, Firebase (Authentication, Firestore e Storage) e o padrão Repository.

Protótipos



Contextualização

O Aplicativo de Lista de Compras é projetado para ajudar os usuários a criar e gerenciar listas de compras de forma eficiente. Ele permite que os usuários adicionem itens, marquem como comprados e removam itens conforme necessário.

Nesta etapa final, o aplicativo deverá ser totalmente funcional, com persistência de dados em nuvem e autenticação real de usuários, utilizando os serviços do Firebase.

Requisitos Funcionais

RF001	Login / Logout / Recuperação de Senha
	<p>O aplicativo deve ter uma tela de login com campos de email, senha e o logotipo da aplicação.</p> <p>Deve ser utilizada a Firebase Authentication para realizar login, logout, cadastro e recuperação de senha.</p> <p>O login deve validar e-mail e senha, apresentando mensagens adequadas em caso de erro.</p> <p>Ao realizar logout, o usuário deve ser redirecionado para a tela de login.</p> <p>Deve haver uma opção de recuperação de senha que envia email ao usuário com link para recuperação de seu acesso.</p>
RF002	Cadastro de Usuário
	<p>O cadastro de novos usuários deve ser realizado através da Firebase Authentication.</p> <p>O formulário deve conter os campos: nome, e-mail, senha e confirmação de senha.</p> <p>O nome do usuário deve ser salvo no Firebase Firestore em um documento vinculado ao UID retornado pelo Authentication.</p>

	Deve haver validação de todos os campos e tratamento adequado de erros do Firebase.
--	---

RF003	Gestão de Listas de Compras
	<p>O aplicativo deve permitir criar, editar, listar e excluir listas de compras, armazenando-as no Firebase Firestore.</p> <p>Cada lista deve conter título, imagem (opcional) e referência ao usuário logado.</p> <p>As imagens devem ser salvas no Firebase Storage, e a URL correspondente deve ser armazenada no Firestore.</p> <p>A listagem deve ser ordenada alfabeticamente e exibida em uma RecyclerView com título e imagem (ou placeholder).</p> <p>Ao excluir uma lista, todos os itens associados também devem ser removidos do Firestore e, se houver imagem, do Storage.</p>

RF004	Gestão de Itens da Lista
	<p>O aplicativo deve permitir criar, editar, listar e excluir itens dentro de cada lista de compras.</p> <p>Cada item deve conter nome, quantidade, unidade e categoria.</p> <p>Os itens devem ser armazenados no Firebase Firestore, dentro da lista correspondente.</p> <p>Os itens devem ser apresentados em uma RecyclerView, ordenados alfabeticamente e agrupados por categoria.</p> <p>Deve ser possível marcar itens como comprados, exibindo-os de forma diferenciada (por exemplo, tachado ou em outra cor).</p>

RF005	Buscas
--------------	---------------

	<p>Deve ser possível realizar buscas por listas e por itens, com filtragem eficiente a partir do nome.</p> <p>A busca deve ser aplicada sobre os dados do Firebase Firestore, utilizando consultas (queries).</p>
--	---

Requisitos Não-Funcionais

- O aplicativo deve ser desenvolvido com Android Nativo em Kotlin.
- A arquitetura deve seguir o padrão MVVM (Model-View-ViewModel).
- A camada de dados deve seguir o padrão Repository, separando regras de acesso ao Firebase das camadas de interface e lógica.
- Deve ser utilizado o Firebase Authentication, Firestore e ~~Storage~~ conforme especificado.
- Deve-se utilizar ViewBinding.
- O design deve seguir as diretrizes do Material Design, com navegação fluida e interface intuitiva.
- O aplicativo poderá ser desenvolvido em grupos de até 3 pessoas.

O que deverá ser entregue?

- Arquivo texto com nome e código dos alunos envolvidos;
- Link para repositório público no GitHub contendo o código-fonte do aplicativo;
- Prints no formato JPG das telas do Firebase (Auth, Firestore e ~~Storage~~) mostrando os dados da aplicação.
- Um vídeo no formato MP4 ou YouTube de até 5 minutos apresentando:
 - O aplicativo em funcionamento demonstrando as principais funcionalidades e evidenciando qualidade da entrega;
 - As principais partes do código, com foco na arquitetura MVVM, Repository e integração com Firebase.
 - Painel do Firebase com dados cadastrados pelo aplicativo.

Critérios de Avaliação

[15%] RF001 Login / Logout / Recuperação de senha	Pontuação
<input type="checkbox"/> Implementação correta do fluxo de autenticação via Firebase Authentication (login, logout, cadastro, recuperação de senha). <input type="checkbox"/> Validação e feedbacks adequados na interface.	
[10%] RF002 Cadastro de Usuário	Pontuação
<input type="checkbox"/> Cadastro completo e integração entre Authentication e Firestore. <input type="checkbox"/> Armazenamento do nome do usuário e tratamento de erros.	
[25%] RF003 Gestão de Listas de Compras	Pontuação
<input type="checkbox"/> Criação, edição, listagem e exclusão de listas integradas ao Firestore. <input type="checkbox"/> Armazenamento correto das imagens no Storage e link no Firestore. <input type="checkbox"/> Sincronização com o usuário autenticado.	
[25%] RF004 Gestão de Itens da Lista	Pontuação
<input type="checkbox"/> Operações CRUD completas para itens no Firestore. <input type="checkbox"/> Exibição organizada por categoria, marcação de itens comprados e atualização visual em tempo real.	
[10%] RF005 Buscas	Pontuação
<input type="checkbox"/> Funcionalidade de busca por listas e itens com consultas no Firestore.	
[15%] Arquitetura e Padrões	Pontuação
<input type="checkbox"/> Implementação da arquitetura MVVM, uso adequado do padrão Repository e separação clara entre camadas (View, ViewModel, Repository).	

Considerações Finais

- A identificação de cópias indiscriminadas de código fonte da Internet, IA ou de colegas da turma, ocasionará a perda da pontuação. Caso haja suspeita de fraude, o grupo poderá passar por uma avaliação oral acerca do projeto desenvolvido.
- A entrega fora do prazo estipulado acarretará em nota zero. O repositório no Github será avaliado levando em consideração somente os commits realizados até a data de entrega.

Dicas

- Tente baixar seu código em outra máquina e executar a aplicação para ter a certeza de que não está faltando nenhum arquivo de configuração que possa vir a prejudicar sua nota.
- Não deixe para apresentar seu projeto apenas na entrega final, aproveite as aulas para tirar dúvidas e obter feedbacks antecipados. Isso fará com que a entrega do grupo se aproxime cada vez mais do que é esperado.