UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE COMPUTAÇÃO

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Arthur Martins Aguiar

Eduardo Lordão Oliveira

Documentação do Trabalho 3 - PDM

Uberlândia – MG – 2025

Documentação do Trabalho 3 - PDM

Trabalho avaliativo do curso de bacharelado em Sistemas de Informação, como requisito parcial para aprovação na disciplina de Programação para Dispositivos Móveis

Orientador: Alexsandro Santos Soares

UBERLÂNDIA – MG

2025

1. Estrutura Arquitetural e Integração Cloud

O projeto foi migrado de uma persistência local para uma arquitetura **Cloud-First**, fundamentada no padrão **MVVM (Model-View-ViewModel)**. A fundação do app agora repousa sobre o ecossistema **Firebase**, utilizando uma estratégia de injeção de dependências com **Dagger Hilt** para garantir um código modular, escalável e de fácil manutenção.

1.1. Gestão de Identidade (Firebase Auth)

Foi estabelecido um sistema de autenticação robusto para e-mail e senha. O fluxo controla o ciclo de vida da sessão (logado/deslogado), garantindo que o UID do usuário seja o token de acesso para suas informações privadas. O estado da autenticação é exposto de forma reativa para a UI, permitindo transições fluidas entre as telas de acesso e a área logada.

1.2. Persistência de Dados Remota (Firestore)

A camada de dados utiliza o **Cloud Firestore** como banco de dados NoSQL. Através do **Repository Pattern**, a lógica de acesso aos dados fica isolada das ViewModels, permitindo que as operações de CRUD (Criar, Ler, Atualizar e Deletar) ocorram de forma assíncrona e transparente.

1.3. Fluxo de Dados e Reatividade

A comunicação entre o Back-end e a Interface ocorre através de fluxos de dados contínuos. Ao invés de requisições únicas, o app utiliza *listeners* em tempo real. Isso significa que a interface do usuário é um reflexo direto do estado do banco de dados: qualquer modificação na nuvem é propagada instantaneamente para o dispositivo sem necessidade de *refresh* manual.

2. Modelagem de Dados

A estrutura de documentos foi planejada para operar em um ambiente multiusuário, onde a segregação de dados é prioridade.

2.1 Estrutura da Coleção tasks:

- **id (String):** Identificador único do documento, gerado de forma aleatória pelo SDK do Firebase para garantir unicidade em larga escala.
- **ownerId (String):** Chave estrangeira que armazena o UID do proprietário da tarefa, utilizada como filtro primário em todas as queries.
- **content / info:** Campos destinados ao título e detalhamento da atividade.
- **status (Boolean):** Flag booleana que indica se a tarefa foi finalizada

3. Decisões Técnicas e Experiência do Usuário (UX)

3.1. Roteamento Dinâmico

A navegação foi centralizada em um NavHost que monitora o estado do usuário. Caso a sessão expire ou o usuário faça *logout*, o sistema realiza o redirecionamento automático para a tela de autenticação, protegendo as rotas internas da aplicação.

3.2. Interface Declarativa e UDF

Utilizando as premissas do **Jetpack Compose**, implementamos o **Unidirectional Data Flow (UDF)**. Os eventos de clique disparam ações nas ViewModels, que por sua vez atualizam um estado único observado pela UI. Isso elimina bugs de interface "travada" e facilita o tratamento de erros de rede.

3.3. Escalabilidade com Firestore

A escolha pelo Firestore justifica-se pela sua flexibilidade em lidar com coleções hierárquicas e pela eficiência do seu motor de busca, permitindo consultas filtradas

por usuário de maneira muito mais performática que bancos de dados relacionais tradicionais ou o Realtime Database.

4. Planejamento de Evolução

4.1. Sincronização em Background

Implementar o suporte a operações offline estendidas, permitindo que o usuário trabalhe sem internet e o Firebase gerencie a fila de sincronização assim que o dispositivo detectar sinal.

4.2. Validação e Segurança

Adicionar camadas de segurança via **Firebase Security Rules**, impedindo que um usuário acesse documentos onde o ownerId seja diferente do seu próprio auth.uid, mesmo que tente via script externo.

4.3. Interface de Recuperação

Desenvolver telas de suporte para recuperação de credenciais e confirmação de e-mail, elevando o nível de maturidade de fluxo de segurança do aplicativo.