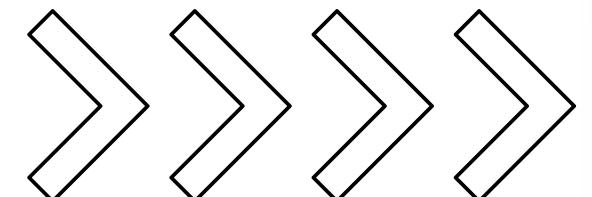


TO DO LIST COM FIREBASE

SISTEMAS DE INFORMAÇÃO

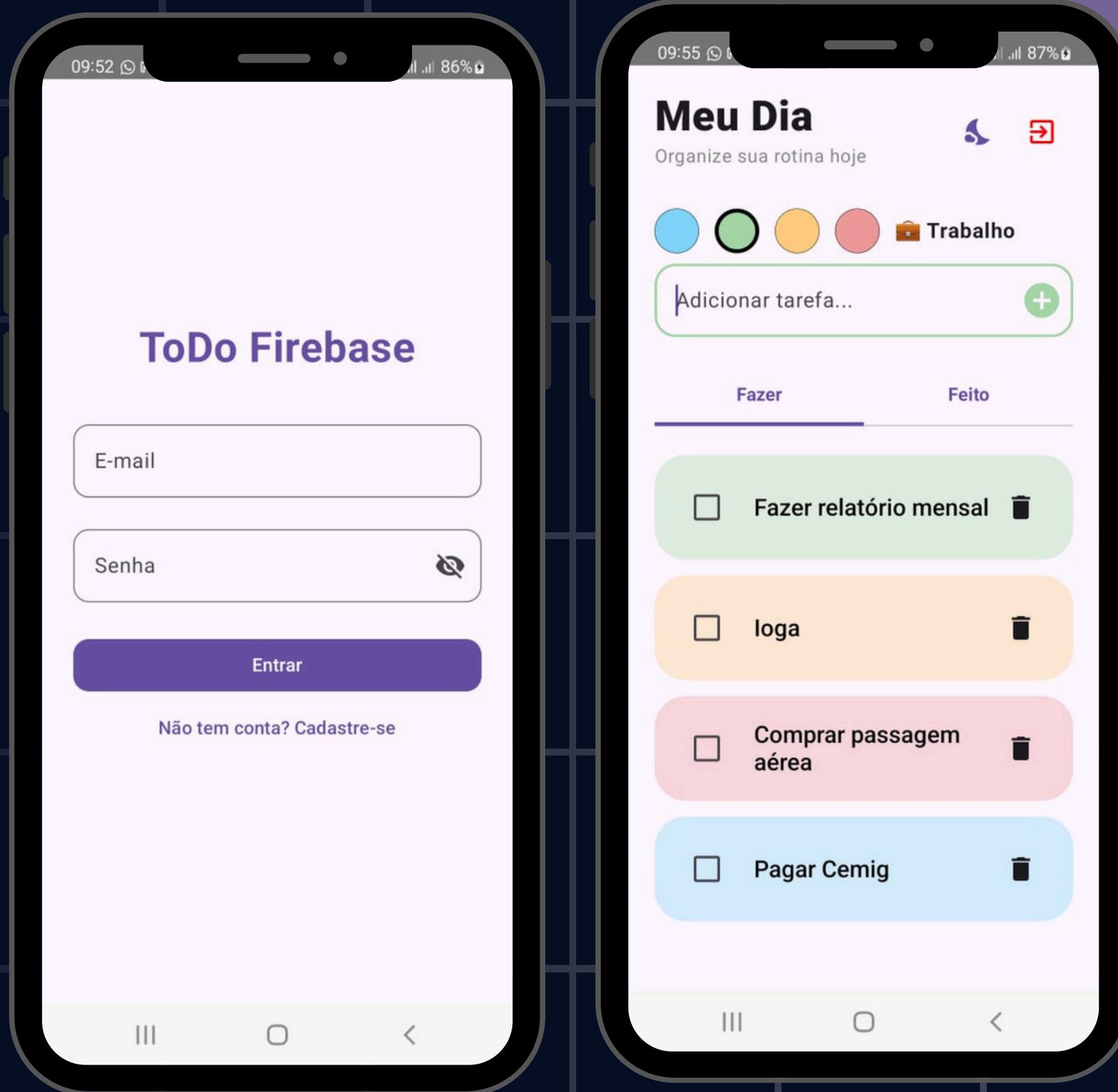


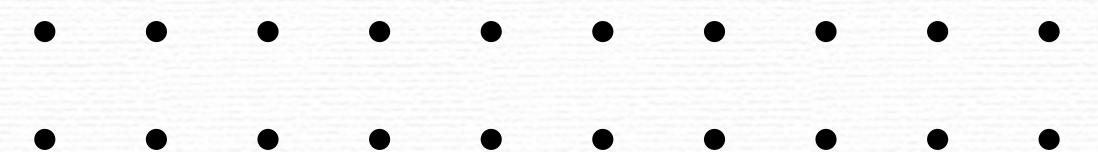
Discentes:
Andreza Batista Alves
Keila Almeida Santana.

Funcionalidades, arquitetura e persistência

- Aplicativo Android desenvolvido em Kotlin e Jetpack Compose.
- Autenticação com Firebase Authentication.
- Persistência em nuvem com Firebase Firestore.
- Gerenciamento de tarefas por usuário.

- Cadastro de usuários com e-mail e senha
- Login e logout com Firebase Authentication
- Criação e exclusão de tarefas
- Separação entre tarefas a Fazer e Feitas
- Persistência das tarefas em nuvem (Firebase Firestore)
- Recuperação automática das tarefas do usuário autenticado.
- Modo noturno.





Uso de LLM's

Gemini 3 Flash e Chat GPT - 5.2

- Apoio na integração do Firebase Authentication.
- Auxílio na implementação do login.
- Resolução de problemas com bibliotecas.
- Sugestões de cores, ícones e modo escuro.
- Criação das telas “a Fazer” e “Feitas”.

Opinião do grupo

As LLMs foram utilizadas como ferramenta de apoio ao aprendizado, auxiliando na compreensão de conceitos, resolução de erros e organização do código. As soluções sugeridas foram sempre analisadas e adaptadas manualmente ao projeto.

Exemplos de Prompt's:

"Não estou conseguindo implementar o login com Firebase Authentication. Você pode me explicar passo a passo como configurar o Firebase no projeto e como fazer o login usando e-mail e senha?"

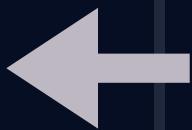
"Como posso separar as tarefas em duas telas no Jetpack Compose, uma para atividades a fazer e outra para atividades já concluídas, mantendo o estado atualizado?"



Dificuldades encontradas:

- Primeiro contato com o Firebase e sua configuração no Android
- Erros iniciais na integração do Firebase Authentication e Firestore
- Pouco domínio prévio do desenvolvimento Android e da linguagem Kotlin
- Dificuldades no gerenciamento do estado e navegação entre telas
- Desafios na definição do design da interface (cores, fontes e organização visual)

```
fun loadTasks(userId: String) {  
    db.collection("users")  
        .document(userId)  
        .collection("tasks")  
        .get()  
        .addOnSuccessListener { result ->  
            val tasks = result.toObjects(Task::class.java)  
            _taskList.value = tasks  
        }  
}
```



Utiliza o userId do usuário autenticado para recuperar do Firestore apenas as tarefas do usuário logado.

Melhorias do Projeto:

Possibilidade
de edição das
tarefas

Filtros e
ordenação das
tarefas

Implementação
de testes
automatizados

Inclusão de
notificações
para lembretes