



Curso: Tecnologia em Desenvolvimento de Sistemas - 2TDS	
Professor: Fernando Pinéo	Componente Curricular: Mobile Application Development
Aluno:	
Aluno:	
Aluno:	
Orientações: <ul style="list-style-type: none">• Um integrante do grupo deverá entregar o link do repositório GitHub na tarefa, incluir no README os nomes dos integrantes juntamente com o RM.• Permitidas consultas na WEB.• Repositório com commits realizado posterior ao prazo estipulado não serão aceitos.• Entrega do repositório do app: (durante horário da aula)• Peso 8,5 pontos(completando a segunda parte do CP4)	

CheckPoint 4

Projeto: Lista Tarefas Plus

Funcionalidades obrigatórias:

1. Autenticação com Google e com o provedor E-mail/ (via Firebase)
2. Login persistente (auto-login)
3. Armazenamento de tarefas no Firestore (por usuário)
4. Lista de tarefas com sincronização em tempo real
5. Tema claro/escuro com persistência
6. Internacionalização (PT/EN) com troca dinâmica de idioma
7. Notificações locais com agendamento
8. Uso de TanStack Query para alguma API externa (ex: motivacional, clima etc.)

Obs: Cada usuário tem sua coleção de tarefas.

Tarefa deverá ser criado com os seguintes campos:

```
{
  title: 'Estudar React Native',
  description: 'Focar no TanStack Query e Notificações',
  completed: false,
  dueDate: '2025-09-10T14:00:00Z',
  createdAt: Timestamp,
  updatedAt: Timestamp
}
```

Ferramentas e libs recomendadas:

- @react-native-firebase/app
- @react-native-firebase/auth
- @react-native-firebase/firestore
- @react-navigation/native
- react-native-paper ou native-base (UI)
- i18n-js ou react-i18next
- expo-notifications (ou notifee para apps mais avançados)
- @tanstack/react-query

Entrega

- ☒ Código hospedado no GitHub
- ☒ APK gerado (usando Expo EAS Build)
- ☒ Vídeo demonstrando a aplicação (máx. 5 min)
- ☒ Documentação curta (README com instruções e arquitetura)

Tabela de pontuação:

Critério	Descrição	Peso
1. Funcionalidade	O app cumpre todos os requisitos mínimos: login com Google, Firestore funcional, tarefas visíveis, adicionáveis e editáveis.	2.0
2. Firestore	Uso adequado do Firestore: estrutura bem pensada, uso de onSnapshot, criação, leitura, edição e deleção (CRUD).	1.5
3. Tema Claro/Escuro	Implementação do tema com persistência (ex: AsyncStorage). A troca funciona corretamente.	1.0
4. Internacionalização (i18n)	Suporte ao menos para português e inglês. Permite troca dinâmica e textos estão traduzidos.	1.5
5. Notificações Locais	Notificações funcionam corretamente com agendamento por data/hora associada à tarefa.	1.0
6. TanStack Query	Uso apropriado para consumir e exibir dados externos (ex: API de clima, frases, etc.). Integração limpa e eficiente.	1.0
7. Design e UX	Navegação fluida, boa escolha de componentes, responsividade, experiência do usuário clara e intuitiva.	0.5
8. Código e Arquitetura	Código limpo, organizado, com separação de responsabilidades. Uso de hooks, pastas, tipagem (se usarem TS).	0.5
9. GitHub + README	Projeto disponível no GitHub, com README explicativo (como rodar, funcionalidades, libs usadas, etc.).	0.5
10. Apresentação/Vídeo	Vídeo curto (até 5 min) demonstrando o app, com explicação clara do que foi implementado.	0.5