

Skooler Chat - for you

Plataforma de mensagens e assistente escolar do IPP

Tiago Vieira (21122), Bruno Monteiro (3990), Miguel Rodrigues (27089)

Publication Date: 2025-11-19; Submission Date: 2025-11-19



1. Introdução

Este projeto consiste no desenvolvimento de uma aplicação móvel de chat em tempo real, utilizando o Flutter como framework de desenvolvimento multiplataforma e o Firebase como backend.

A aplicação permite a autenticação de utilizadores e a troca de mensagens instantâneas entre múltiplos utilizadores autenticados.

O principal objetivo foi criar uma solução funcional, simples e escalável, explorando serviços cloud modernos e boas práticas de desenvolvimento mobile.

2. Requisitos

A seguinte lista define os principais requisitos para o projeto:

- Implementar um sistema de autenticação de utilizadores
- Permitir login por Email/Password
- Integrar login com Google
- Desenvolver um chat em tempo real
- Garantir que mensagens são partilhadas entre diferentes utilizadores
- Utilizar uma arquitetura simples e clara
- Assegurar uma boa experiência de utilização (UI estável e responsiva)

3. Modulação conceptual

3.1. Dependências do sistema

3.1.1. Flutter

Framework da Google utilizado para o desenvolvimento da interface da aplicação. Permite criar aplicações nativas a partir de uma única base de código em Dart.[1]

3.1.2. Firebase

Plataforma Backend-as-a-Service utilizada para suportar autenticação e armazenamento de dados.[2]

Serviços Firebase utilizados:

- Autenticação Firebase[3]
 - Email e Password
 - Google Sign-In[4]
- Base de dados em tempo real Firebase[5]
- Armazenamento e sincronização das mensagens em tempo real
- Firebase Core
- Inicialização e ligação da aplicação ao projeto Firebase

3.2. Sistema de Autenticação

3.2.1. Email e Password

Os utilizadores podem criar conta e autenticar-se utilizando email e palavra-passe, através do Firebase Authentication.[3]

3.2.2. Google Sign-In

Foi integrado o login com conta Google, permitindo autenticação rápida e segura, utilizando o mesmo sistema de sessões do Firebase.[4]

3.2.3. Persistência de Sessão

A aplicação utiliza `authStateChanges()` para detetar automaticamente se o utilizador está autenticado, redirecionando-o para o chat sem necessidade de novo login.[6]

3.3. Chat em Tempo Real

O sistema de chat foi implementado com Firebase Realtime Database, garantindo:[5]

- Atualização instantânea das mensagens
- Sincronização automática entre utilizadores
- Comunicação bidirecional em tempo real

4. Implementação

4.1. Arquitetura da Aplicação

A aplicação segue uma arquitetura simples e modular, separando responsabilidades por ficheiros e pastas.

4.1.1. Estrutura do Projeto

```
lib/
├── main.dart
├── firebase_options.dart
├── message.dart
├── services/
│   ├── auth_service.dart
│   └── pages/
│       ├── login_page.dart
│       └── chat_page.dart
```

main.dart

Responsável pela inicialização do Firebase e controlo do estado de autenticação.

firebase_options.dart

Contém as configurações geradas pelo FlutterFire CLI para ligação ao Firebase.

auth_service.dart

Implementa toda a lógica de autenticação (login, registo, Google e logout).

login_page.dart

Interface de autenticação do utilizador.

chat_page.dart

Interface do chat em tempo real.

message.dart

Modelo de dados para mensagens.

4.2. Estrutura dos Dados na Base de dados em tempo real

```

messages
├── messageId
│   ├── text
│   ├── userId
│   └── email

```

Cada mensagem contém:

- O texto enviado
- O identificador do utilizador
- O email do remetente

4.3. Regras de Segurança

```

{
  "rules": {
    "messages": {
      ".read": "auth != null",
      ".write": "auth != null"
    }
  }
}

```

Estas regras garantem que apenas utilizadores autenticados podem ler e escrever mensagens.

5. Resultados Obtidos

No final do desenvolvimento, a aplicação apresenta as seguintes funcionalidades operacionais:

- Autenticação funcional por Email/Password
- Autenticação funcional com Google
- Chat em tempo real entre múltiplos utilizadores
- Mensagens sincronizadas instantaneamente
- Interface estável e adaptável a diferentes tamanhos de ecrã
- Estrutura de código organizada e fácil de manter

5.1. Dificuldades Encontradas

Durante o desenvolvimento foram enfrentados alguns desafios, nomeadamente:

- Configuração correta do Firebase e do ficheiro firebase_options.dart
- Conflitos entre diferentes serviços Firebase
- Problemas de cache e hot reload do Flutter
- Ajustes de layout para evitar overflow em ecrãs pequenos
- Integração correta entre autenticação e base de dados

Todos estes problemas foram resolvidos através de análise, testes e reconfiguração adequada do ambiente.

6. Possíveis melhorias futuras

O projeto pode ser expandido com várias funcionalidades adicionais. Em primeira fase será terminar a componente de chat, comunicação direta entre alunos e professores, turmas ou de trabalho. Numa segunda fase pretende-se integrar plataformas escolares com o Moodle ou o PAE e estender funcionalidade de assistência, realidade estendida e sensível ao contexto escolar, monitorização e bem estar.

- Componente de chat:
 - Criação de salas de chat
 - Mensagens privadas
 - Timestamps nas mensagens
 - Perfis de utilizador com avatar
 - Estado online/offline
 - Notificações push
 - Interface gráfica mais avançada (estilo WhatsApp ou Discord)
 - Migração para Firestore para chats mais complexos
- Integração em plataformas escolares como Moodle e PAE.
- Treinador pessoal personalizado
- Integração com sistemas ubíquos e sensível ao contexto:
 - Agenda escolar, com sincronização automática no contexto do aluno (avaliações, lembretes de aula...),
 - Marcação de presença nas aulas por biometria,
 - Pós marcação de presença, apresenta o sumário e conteúdo para a aula,
 - Assistente por IA (Gemini) contextualizado com o conteúdo e sumário da aula,
 - Monitorização de saúde e bem estar (sincroniza com pulseiras inteligentes),
 - Botão de pânico/emergência, que liga e envia mensagens periodicamente com localização para um número definido pelo utilizador.

7. Conclusão

Este projeto permitiu consolidar conhecimentos em desenvolvimento mobile com Flutter e integração de serviços cloud através do Firebase. A aplicação cumpre todos os objetivos propostos, funcionando de forma estável e eficiente, e serve como uma base sólida para futuras evoluções. O uso do Firebase Realtime Database mostrou-se adequado para um sistema de chat simples e eficaz, enquanto o Firebase Authentication garantiu segurança e facilidade de gestão de utilizadores.

Referências Bibliográficas

1. Google. Flutter Documentation. <https://docs.flutter.dev>, 2024. Accessed: 2026-01-19.
2. Google. Firebase Documentation. <https://firebase.google.com/docs>, 2024. Accessed: 2026-01-19.
3. Google. Firebase Authentication. <https://firebase.google.com/docs/auth>, 2024. Accessed: 2026-01-19.
4. Google. Google Sign-In Documentation. <https://developers.google.com/identity>, 2024. Accessed: 2026-01-19.
5. Google. Firebase Realtime Database. <https://firebase.google.com/docs/database>, 2024. Accessed: 2026-01-19.
6. Google. Get Started with Firebase Authentication on Flutter. <https://firebase.google.com/docs/auth/flutter/start#authstatechanges>, 2024. Accessed: 2026-01-19.