****

**Conteúdo**

[1. Resumo 3](#_Toc171155132)

[2. Introdução ao projeto 4](#_Toc171155133)

[3. Arquitetura do sistema 5](#_Toc171155134)

[3.1 Tecnologias utilizadas 5](#_Toc171155135)

[3.2 Estrutura e organização 6](#_Toc171155136)

[3.3 Entidades informacionais 7](#_Toc171155137)

[4. Implementação 8](#_Toc171155138)

[**Pergunta (Question)** 8](#_Toc171155139)

[**Caso (Case)** 8](#_Toc171155140)

[**Resultado do Quiz (QuizResult)** 8](#_Toc171155141)

[**Registro de Utilizador** 9](#_Toc171155142)

[**Login do Utilizador** 10](#_Toc171155143)

[5. Limitações técnicas 12](#_Toc171155144)

[6. Sugestões de melhoria 13](#_Toc171155145)

# Resumo

Este documento tem como objetivo fornecer uma documentação técnica abrangente da aplicação, visando auxiliar na compreensão de seu funcionamento e arquitetura. É direcionado à aplicação Meducation e destina-se a um público com conhecimento técnico em desenvolvimento de software.

Este projeto foi proposto pela professora Rossana Santos e Helena Caria com o intuito de desenvolver uma aplicação que serve de um meio à adquirição de conhecimento na área da saúde, sendo direcionado para cursos da Escola Superior de Saúde do Instituto Politécnico de Setúbal. Paralelemente o projeto tem também o objetivo de desenvolver competências aplicadas a um contexto real na área de engenharia de software com base nos conhecimentos adquiridos no decorrer da Licenciatura em Engenharia Informática. Desta forma, e em associação com a professora Rossana Santos e Helena Caria, juntámos a informática à saúde e surgiu a aplicação Meducation.

# Introdução ao projeto

A aplicação Meducation é uma aplicação móvel desenvolvida para auxiliar no ensino e na aprendizagem de alunos de cursos da área da saúde no Instituto Politécnico de Setúbal. Surge como uma ferramenta inovadora para o ensino e a aprendizagem na Escola Superior de Saúde de Setúbal. Através de um aplicativo mobile nativo, os alunos podem ter uma experiência de aprendizagem personalizada e dinâmica, otimizando o seu desenvolvimento profissional e preparando-os para desafios de teor tanto prático como teórico.

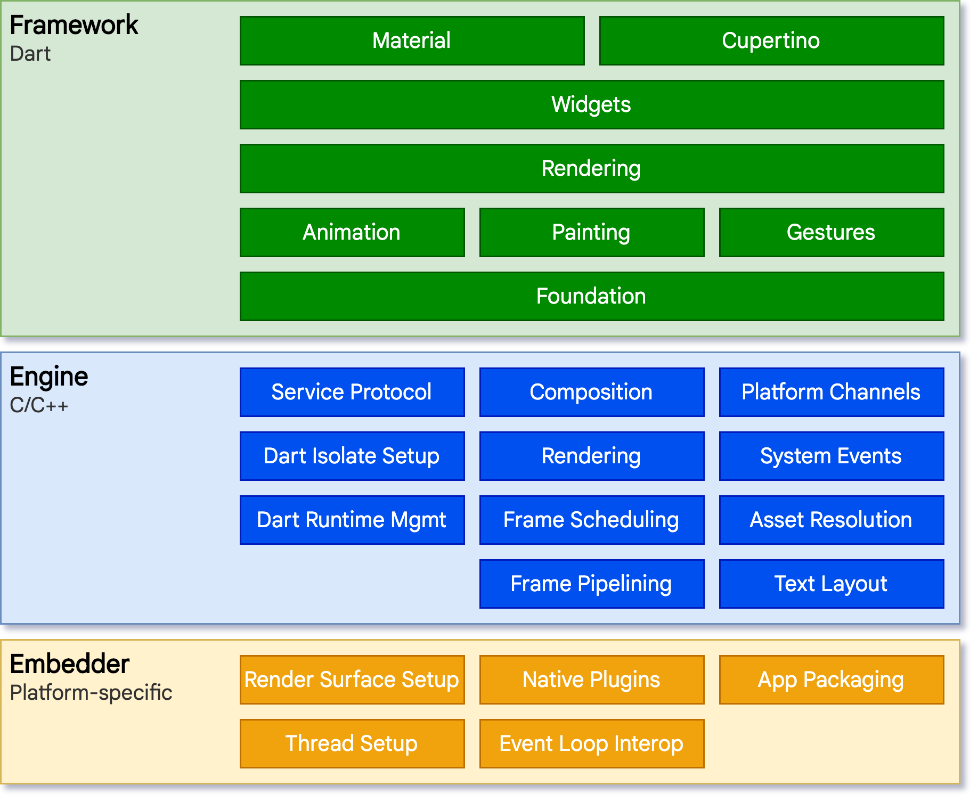
Este sistema oferece aos utilizadores funcionalidades como a personalização da mascote da aplicação, a interação com um jogo onde o objetivo é responder a um caso clínico (com teor teórico e prático), análise de dados e estatísticas tanto pessoais como globais, entre outras.

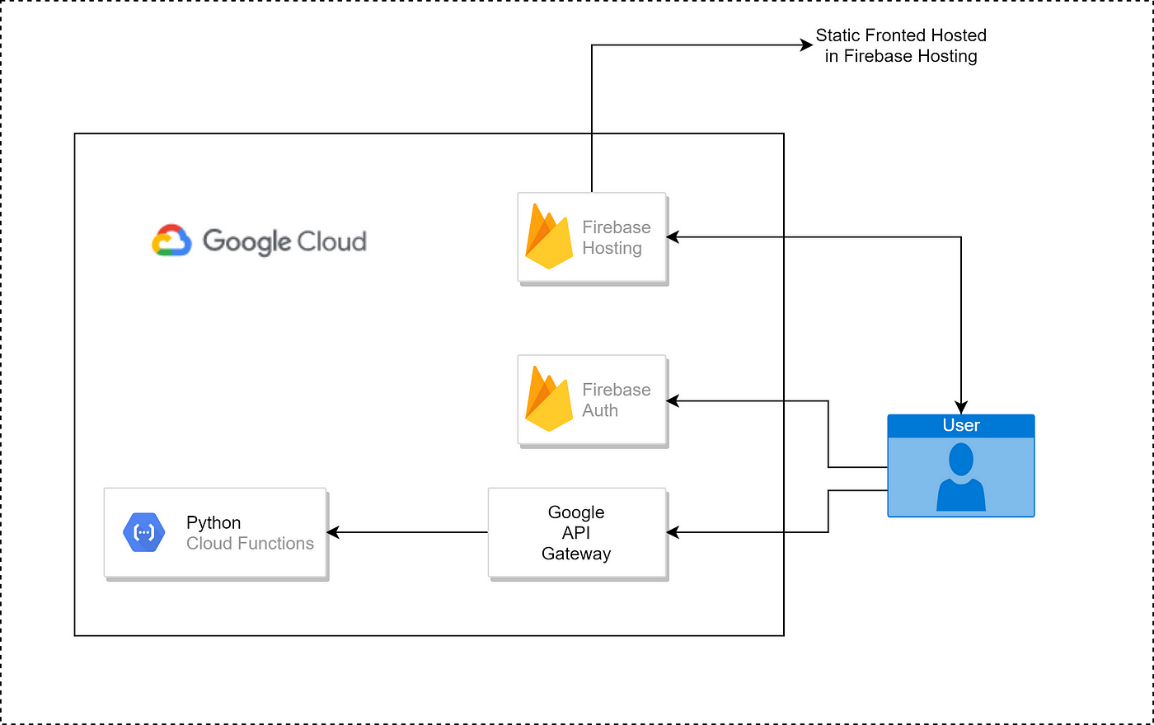
# Arquitetura do sistema

A arquitetura da aplicação Meducation foi planeada para proporcionar uma experiência de utilizador simplista e eficiente, enquanto utiliza tecnologias modernas e robustas para garantir a segurança e o desempenho do sistema. A arquitetura está dividida em camadas, cada uma responsável por um aspeto específico da aplicação.

# Tecnologias utilizadas

O sistema desenvolvido foi implementado na linguagem Dart com base na framework Flutter. Externamente foi utilizado o serviço cloud Firebase para suportar os dados da aplicação, assim como recursos externos da Google para a autenticação.





# Estrutura e organização

Para a criação do sistema foram criadas algumas subpastas organizacionais, de modo a facilitar a integração do sistema e fazer melhorias ao nível de performance:

* Dataset
* Imports
* Models
* Screens
* Auth
* Profile
* Quiz
* Splash
* Widgets

# Entidades informacionais

As entidades informacionais da aplicação Meducation foram modeladas para representar os elementos essenciais da experiência dos utilizadores. Abaixo está um diagrama UML representando as principais entidades.

+------------------+

| Case |

+------------------+

| - id: String |

| - title: String |

| - questions: List|

+------------------+

|

v

+------------------+

| Question |

+------------------+

| - title: String |

| - answers: List |

+------------------+

|

v

+------------------+

| Answer |

+------------------+

| - text: String |

| - isCorrect: bool|

+------------------+

# Implementação

**Pergunta (Question)**

A entidade Pergunta representa uma pergunta dentro de um caso clínico. Inclui o título da pergunta e uma lista de respostas possíveis.

dart

Copy code

import 'package:portefolio/src/models/answer\_model.dart';

class Question {

const Question({

required this.title,

required this.answers,

});

final String title;

final List<Answer> answers;

}

**Caso (Case)**

A entidade Caso representa um caso clínico que contém várias perguntas. Inclui um ID, título e uma lista de perguntas.

dart

Copy code

import 'package:portefolio/src/models/question\_model.dart';

class Case {

const Case({

required this.id,

required this.title,

required this.questions,

});

final String id;

final String title;

final List<Question> questions;

}

**Resultado do Quiz (QuizResult)**

A entidade Resultado do Quiz armazena os resultados de um quiz realizado por um utilizador. Inclui o ID do utilizador, ID do quiz, pontuação do utilizador, pontuação total do quiz e o timestamp.

dart

Copy code

import 'package:portefolio/src/imports/imports.dart';

class QuizResult {

const QuizResult({

required this.userId,

required this.quizId,

required this.userScore,

required this.quizScore,

required this.timestamp,

});

final String? userId;

final String quizId;

final int userScore;

final int quizScore;

final Timestamp timestamp;

toJson()

{

return {

"user\_id": userId,

"quiz\_id": quizId,

"user\_score": userScore,

"quiz\_score": quizScore,

"timestamp": timestamp,

};

}

}

**Registro de Utilizador**

O registro de utilizador envolve várias etapas, incluindo a coleta de informações do utilizador, upload de uma imagem de perfil e envio de um e-mail de verificação. Aqui está uma descrição passo a passo do processo:

1. **Recolha de Dados do Utilizador**:
   * O utilizador insere seu nome, e-mail e senha.
   * Opcionalmente, o utilizador pode fazer upload de uma imagem de perfil.
2. **Upload da Imagem de Perfil**:
   * Se o utilizador escolhe uma imagem, ela é carregada a partir do dispositivo.
   * A imagem é enviada para o Firebase Storage, e o URL da imagem é obtida.
3. **Criação da Conta no Firebase Auth**:
   * Utilizando o e-mail e a senha fornecidos, uma nova conta é criada no Firebase Authentication.
   * Um e-mail de verificação é enviado para o endereço de e-mail do utilizador.
4. **Armazenamento de Dados na Firestore**:
   * Os dados do utilizador, incluindo o nome, e-mail e URL da imagem de perfil (ou uma URL padrão se nenhuma imagem for fornecida), são armazenados na Firebase Firestore.
5. **Navegação para a Página de Verificação**:
   * Após o registro bem-sucedido, o utilizador é redirecionado para uma página que instrui a verificar seu e-mail.

**Fluxograma:**

+----------------------+

| Entrada de Dados |

| (nome, e-mail, senha)|

+----------+-----------+

|

v

+-----------------------+

| Upload da Imagem (op.)|

+----------+------------+

|

v

+----------------------+

| Criação da Conta |

| no Firebase Auth |

+----------+-----------+

|

v

+----------------------+

| Envio de E-mail de |

| Verificação |

+----------+-----------+

|

v

+----------------------+

| Armazenamento de Dados|

| no Firestore |

+----------+-----------+

|

v

+----------------------+

| Redirecionamento para|

| Página de Verificação|

+----------------------+

### **Login do Utilizador**

O login do utilizador pode ser feito utilizando o e-mail e senha, ou através do Google Sign-In. O fluxo de login inclui a verificação de e-mail e a navegação para a página principal da aplicação.

1. **Recolha de Credenciais**:
   * O utilizador insere seu e-mail e senha.
2. **Autenticação com Firebase Auth**:
   * As credenciais são verificadas no Firebase Authentication.
   * Se o e-mail do utilizador não foi verificado, um aviso é exibido, solicitando a verificação.
3. **Login com Google**:
   * O utilizador pode optar por fazer login com sua conta Google.
   * O Google Sign-In é utilizado para autenticar o utilizador e obter suas informações.
4. **Armazenamento de Dados no Firestore**:
   * Se o login for bem-sucedido, os dados do utilizador são armazenados no Firestore.
   * Se o utilizador já existe no Firestore, seus dados são atualizados.
5. **Navegação para a Página Principal**:
   * Após o login bem-sucedido, o utilizador é redirecionado para a página principal da aplicação.

#### **Fluxograma**:

+----------------------+

| Entrada de Credenciais|

| (e-mail, senha) |

+----------+-----------+

|

v

+----------------------+

| Autenticação no |

| Firebase Auth |

+----------+-----------+

|

+------------------------+

| |

v v

+---------------------+ +----------------------+

| E-mail Verificado? | | E-mail Não Verificado|

| (Sim) | | (Não) |

+----------+----------+ +----------+-----------+

| |

v |

+----------------------+ |

| Armazenamento de Dados| |

| no Firestore | |

+----------+------------+ |

| |

v v

+----------------------+ +----------------------+

| Redirecionamento para| | Aviso de Verificação |

| Página Principal | +----------------------+

+----------------------+

# Limitações técnicas

Embora a aplicação Meducation tenha sido projetada para ser completamente funcional, útil e servir ao propósito final, que é fornecer um sistema que permita a aprendizagem contínua dos alunos da área da saúde, existem algumas limitações técnicas que precisam de ser consideradas após a entrega final do projeto.

**Desempenho**: A aplicação pode apresentar lentidão em dispositivos Android mais antigos devido ao uso intensivo de recursos

**Escalabilidade**: A aplicação atual não foi projetada para suportar um grande número de utilizadores simultâneos

**Segurança**: Melhorias na segurança dos dados dos utilizadores podem ser necessárias, incluindo criptografia de dados sensíveis

# Sugestões de melhoria

Embora a aplicação Meducation tenha sido projetada para ser completamente funcional, útil e servir ao propósito final, que é fornecer um sistema que permita a aprendizagem contínua dos alunos da área da saúde, existem algumas limitações técnicas que precisam de ser consideradas após a entrega final do projeto.

**Versão iOS**: Expandir a aplicação para suportar dispositivos iOS

**Integração com APIs de Saúde**: Integrar com APIs de saúde para fornecer dados mais precisos e atualizados

**Funcionalidades Sociais**: Adicionar funcionalidades sociais, como fóruns e grupos de estudo

**Personalização de Conteúdo**: Melhorar os algoritmos de recomendação para fornecer conteúdos mais personalizados aos utilizadores