Meducation

A white letter on a purple background

Description automatically generated

Unidade Curricular de Portfólio

Manual de Utilizador

**Índice**

* Introdução
* Arquitetura do Sistema
* Entidades e sua implementação
* Algoritmos e sua implementação
* Limitações técnicas e ideias para desenvolvimento futuro

**Introdução**

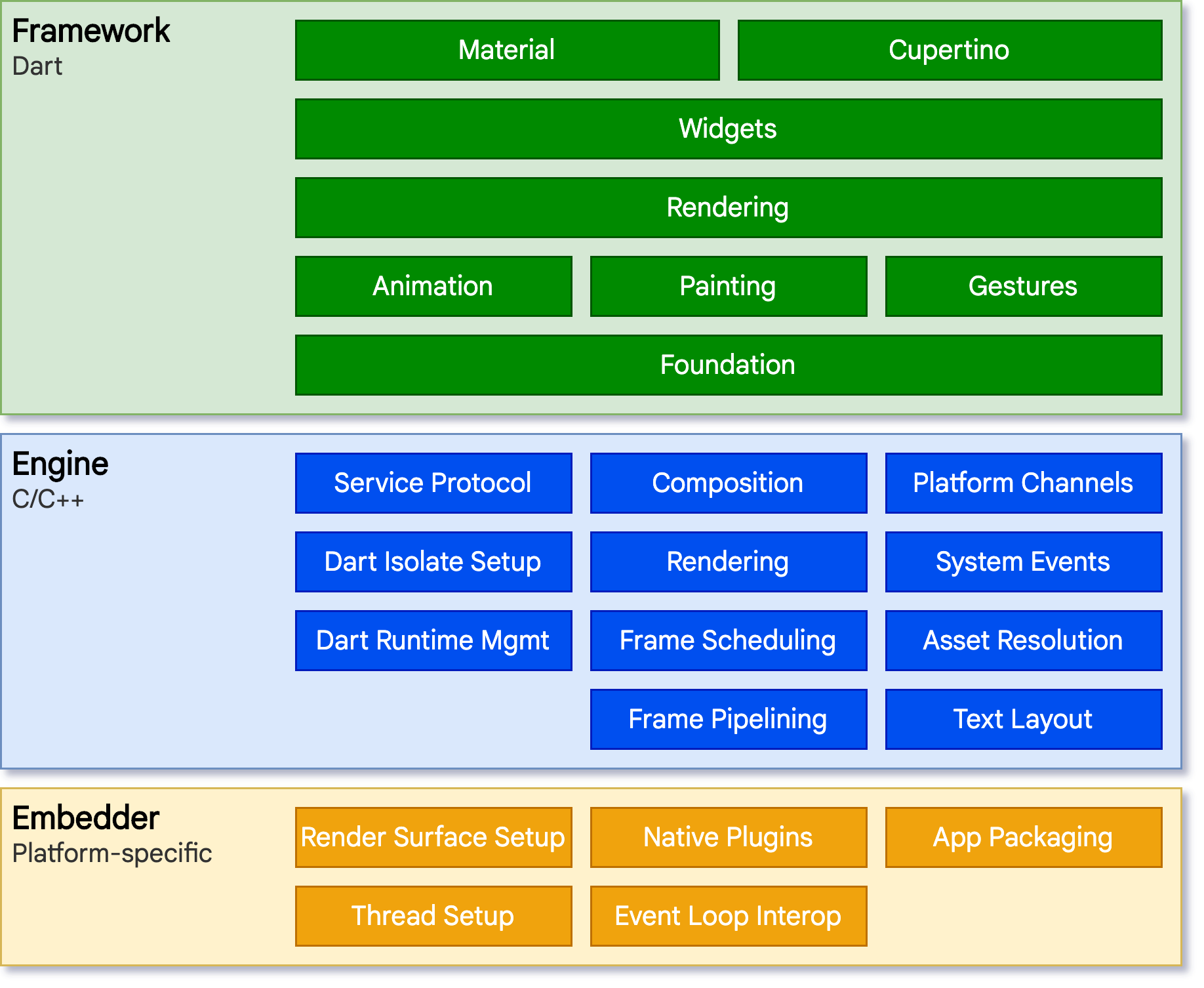
Este documento foi elaborado com o objetivo de documentar a aplicação e ajudar a entender o funcionamento da aplicação, desta forma apresentamos o manual técnico da aplicação Meducation.

A aplicação Meducation foi proposta e criada com o intuito de desenvolver competências na área de engenharia de software com base nos conhecimentos adquiridos no decorrer da licenciatura em engenharia informática, desta forma e em conjunto com a disciplina de portefólio e com a ajuda da professora Rossana Santos e Helena Caria, juntámos a informática à saúde e surgiu a Meducation.

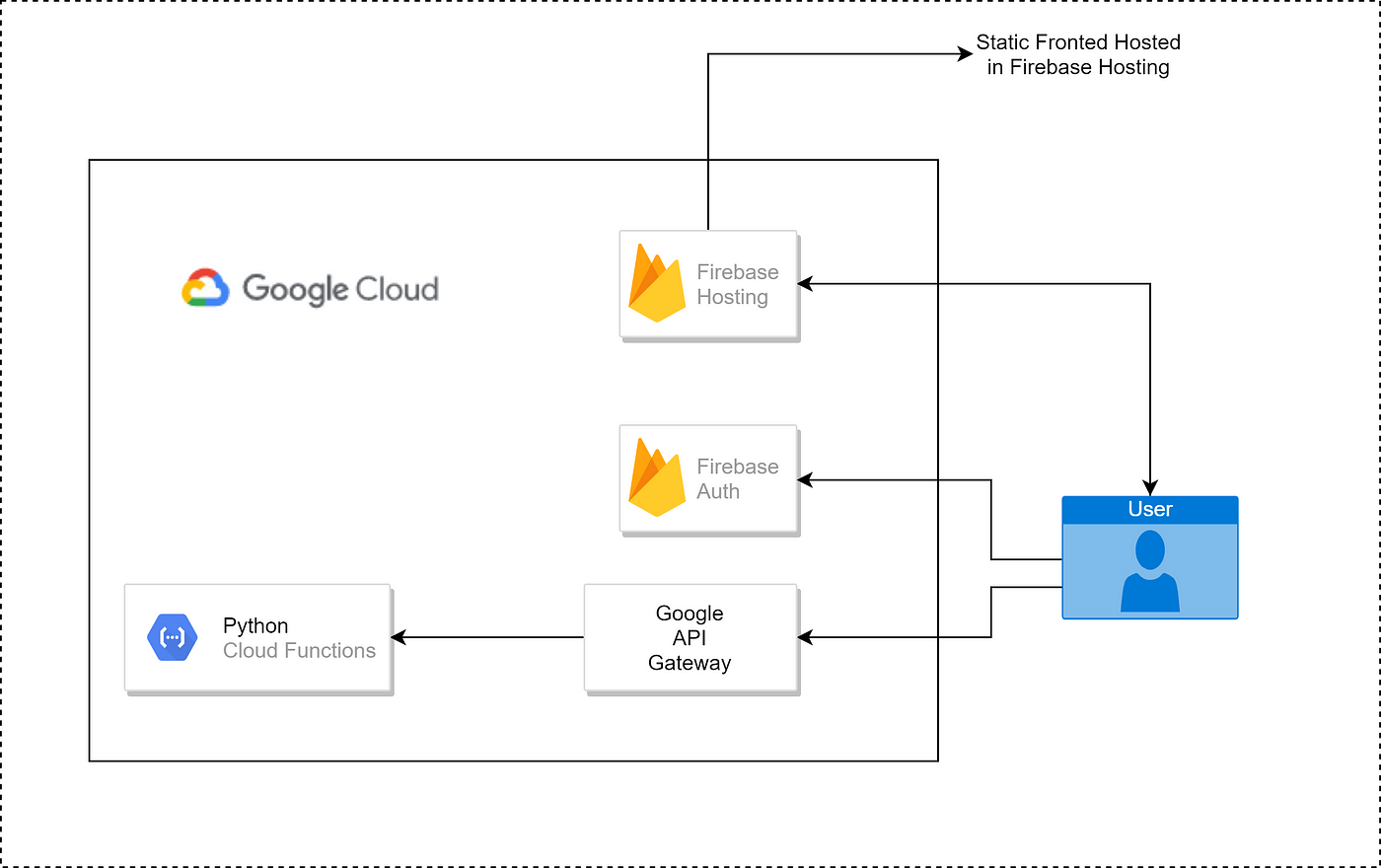
**Arquitetura do sistema**

O sistema desenvolvido foi implementado em linguagem dart e é uma aplicação flutter para android. Para a criação do sistema foram usados recursos externos da google para conseguir implementar algumas das funcionalidades, também foram criadas algumas subpastas organizacionais, sendo estas:

* Dataset
* Imports
* Models
* Screens
* Widgets



Flutter Architecture



*Firebase Auth Architecture*

**Entidades e sua 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 usuário. Inclui o ID do usuário, ID do quiz, pontuação do usuário, 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,

};

}

}

**Algoritmos e sua implementação**

A aplicação Meducation utiliza uma série de algoritmos para autenticação de usuários, permitindo que eles se registrem, façam login, e recuperem suas senhas. Abaixo, apresentamos os principais fluxos e algoritmos utilizados para essas funcionalidades.

**Registro de Usuário**

O registro de usuário envolve várias etapas, incluindo a coleta de informações do usuário, 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 usuário insere seu nome, e-mail e senha.
   * Opcionalmente, o usuário pode fazer upload de uma imagem de perfil.
2. **Upload da Imagem de Perfil**:
   * Se o usuário 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 usuário.
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 usuário 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 e ideias para desenvolvimento futuro**

Embora a aplicação Meducation tenha sido projetada para ser funcional e útil, existem algumas limitações técnicas que precisam ser consideradas:

**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 usuários simultâneos.

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

**Ideias para Desenvolvimento Futuro**

**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 usuários.