

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE RONDÔNIA CAMPUS ARIQUEMES**

**ESPECIALIDADESON: SISTEMA MULTIPLATAFORMA PARA GESTÃO
DE PROVAS E CERTIFICADOS DIGITAIS PARA CLUBE DE
DESBRAVADORES**

ENEAS SILVA DE QUEIROZ

Ariquemes - RO

2025

ENEAS SILVA DE QUEIROZ

**ESPECIALIDADESON: SISTEMA MULTIPLATAFORMA
PARA GESTÃO DE PROVAS E CERTIFICADOS DIGITAIS
PARA CLUBE DE DESBRAVADORES**

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia de Rondônia, Campus Ariquemes, como requisito parcial para obtenção do título de **Tecnólogo em Análise e Desenvolvimento de Sistemas**.

Orientador: Prof. Me Andrey A. Quadros

Ariquemes/RO

2025.

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

Queiroz, Eneas Silva de.

Especialidadeson: sistema multiplataforma para gestão de provas e certificados digitais para clube de desbravadores / Eneas Silva de Queiroz. - Ariquemes, 2025.

49 f. : il.

Orientador(a): Prof. Dr. Andrey Alencar Quadros.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Ariquemes, 2025.

1. Avaliação digital. 2. Certificação. 3. Multiplataforma. 4. Desenvolvimento de software. 5. Gestão educacional. I. Quadros, Andrey Alencar (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

Bibliotecário(a) Responsável: Renilce Silva Moraes, CRB-11/906

ENEAS SILVA DE QUEIROZ

ESPECIALIDADESON

Sistema multiplataforma para gestão de provas e certificados digitais para Clube de Desbravadores

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Analista e Desenvolvedor de Sistemas” e aprovado em sua forma final pelo Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

Ariquemes/RO, 16 de dezembro de 2025.

Banca Examinadora:

Prof. Me Andrey A. Quadros

(orientador)

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO – Campus
Ariquemes)

Prof. Me Luciano Topolniak

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO – Campus
Ariquemes)

Prof. Esp. Marcos Alves Faino

Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO – Campus
Ariquemes)

Ariquemes/RO

2025.

AGRADECIMENTOS

Agradeço, em primeiro lugar, a Deus, por me conceder força, sabedoria e perseverança em cada etapa desta jornada acadêmica. Sua luz me sustentou mesmo nos momentos mais desafiadores, e sem Ele eu não teria chegado até aqui.

À minha mãe, Quesia Olegário da Silva, cuja presença constante, apoio incondicional e palavras de incentivo sempre foram fundamentais. Sua confiança nas minhas escolhas e o amor que nunca faltou foram a base de todas as minhas conquistas.

À minha esposa, Ana Selma Martinovsk Queiroz, companheira fiel em todos os momentos. Sua compreensão, paciência e apoio diário foram essenciais para que eu enfrentasse cada desafio com coragem e esperança. Sua presença foi força e motivação nesta caminhada.

Aos meus professores, em especial ao meu orientador Andrey Alencar Quadros, pela dedicação, orientação, paciência e confiança depositada em mim desde o início do curso. Agradeço também aos demais docentes, que contribuíram significativamente para minha formação acadêmica e profissional. Estendo minha gratidão ao IFRO – Instituto Federal de Rondônia, instituição que proporcionou conhecimento, estrutura e oportunidades valiosas para meu crescimento.

Aos meus amigos e irmãos, que sempre torceram pelo meu sucesso e estiveram ao meu lado com palavras de apoio e encorajamento.

E ao Clube de Desbravadores, do qual faço parte com orgulho, pela inspiração, pelos valores que fortaleço diariamente e pelo incentivo ao desenvolvimento pessoal, espiritual e comunitário.

A todos que contribuíram, direta ou indiretamente, deixo meu mais profundo agradecimento. Este trabalho também é fruto da colaboração, do carinho e da confiança de cada um de vocês.

RESUMO

O Clube de Desbravadores oferece um programa educativo que inclui um sistema de especialidades no qual os participantes desenvolvem conhecimentos e habilidades em diversas áreas. O processo tradicional de avaliação ainda ocorre de forma manual, exigindo tempo e esforço de instrutores para criar provas e emitir certificados. Este trabalho apresenta o sistema EspecialidadesOn, que tem como objetivo modernizar e digitalizar esse processo, permitindo a realização de avaliações on-line e a emissão automatizada de certificados digitais. O sistema foi desenvolvido para fornecer uma solução segura, acessível e eficiente, utilizando tecnologias modernas que ampliam a qualidade e agilidade das atividades educativas. A construção do sistema empregou o framework Flutter e os serviços do Firebase, tendo sido seguidas etapas de modelagem, implementação, testes e validação. Foram aplicadas técnicas de autenticação segura, armazenamento em nuvem, geração de certificados em PDF e validação por hash criptográfico. A arquitetura MVC orientou a organização do código, garantindo maior clareza e manutenibilidade. Os resultados demonstraram que a aplicação funciona de forma estável, responsiva e multiplataforma, atendendo satisfatoriamente aos requisitos estabelecidos e reduzindo significativamente o tempo dedicado pelos instrutores ao processo de avaliação. O sistema apresenta-se como uma alternativa moderna e funcional que contribui para a digitalização das atividades do Clube de Desbravadores.

Palavras-chave: avaliação digital; certificação; multiplataforma; desenvolvimento de software; gestão educacional.

ABSTRACT

The EspecialidadesOn system was developed to modernize the evaluation process of the Pathfinder Club specialties, providing greater agility, standardization, and efficiency for instructors and participants. The solution addresses the need to eliminate manual procedures such as creating tests and issuing certificates, optimizing time and reducing errors. The main objective was to build a multiplatform digital application that enables online testing and digitally validated certificate issuance. The system was developed using the Flutter framework and Firebase services, including authentication, database, storage, and digital certificate generation. The methodology involved modeling functionalities, implementing the MVC architecture, and performing functional, security, and usability tests to ensure correct operation on different devices. The results showed that the system fully meets the proposed requirements, offering good performance, intuitive usability, and secure infrastructure, establishing itself as an effective tool for managing specialty assessments.

Keywords: digital assessment; certification; multiplatform; educational management; systems development.

LISTA DE FIGURAS

Figura 1 – Fluxo de desenvolvimento do aplicativo <i>EspecialidadesON</i>	22
Figura 2 – Arquitetura MVC do sistema <i>EspecialidadesON</i>	26
Figura 3 – Estrutura de diretórios do projeto <i>EspecialidadesON</i>	27
Figura 4 – Fluxo de dados entre as camadas do sistema <i>EspecialidadesON</i>	31
Figura 5 – Tela de login do sistema	33
Figura 6 – Tela de listagem de especialidades disponíveis	34
Figura 7 – Tela de realização de prova	34
Figura 8 – Exemplo de certificado gerado pelo sistema	35
Figura 9 – Dashboard administrativo do sistema	36

LISTA DE TABELAS

Tabela 1 – Principais tecnologias e dependências.	28
Tabela 2 – Funcionalidades implementadas.	32

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
ADS	Análise e Desenvolvimento de Sistemas
IFRO	Instituto Federal de Educação, Ciência e Tecnologia de Rondônia
TCC	Trabalho de Conclusão de Curso
API	Application Programming Interface
PDF	Portable Document Format
TI	Tecnologia da Informação
BD	Banco de Dados

SUMÁRIO

1	INTRODUÇÃO	14
1.1	CONTEXTUALIZAÇÃO	14
1.2	PROBLEMA	14
1.3	OBJETIVOS	15
1.3.1	Objetivo Geral	15
1.3.2	Objetivos Específicos	15
1.4	JUSTIFICATIVA	15
1.5	ESTRUTURA DO TRABALHO	16
2	REFERENCIAL TEÓRICO	17
2.1	FLUTTER	17
2.1.1	Características Principais	17
2.1.2	Vantagens do Flutter	17
2.2	FIREBASE	18
2.2.1	Firestore Authentication	18
2.2.2	Cloud Firestore	18
2.2.3	Firestore Storage	18
2.3	ARQUITETURA MVC	18
2.3.1	Model (Modelo)	18
2.3.2	View (Visão)	18
2.3.3	Controller (Controlador)	19
2.3.4	Vantagens do MVC	19
2.4	CERTIFICADOS DIGITAIS	19
2.4.1	Validação de Certificados	19
2.4.2	Hash SHA-256	19
2.5	PDF (PORTABLE DOCUMENT FORMAT)	19
2.5.1	Vantagens do PDF para Certificados	19
3	METODOLOGIA	21
3.1	TIPO DE PESQUISA	21
3.2	METODOLOGIA DE DESENVOLVIMENTO	21
3.2.1	Etapas do Desenvolvimento	23
<i>3.2.1.1</i>	<i>Planejamento</i>	<i>23</i>
<i>3.2.1.2</i>	<i>Desenvolvimento</i>	<i>23</i>
<i>3.2.1.3</i>	<i>Testes</i>	<i>23</i>
3.3	FERRAMENTAS UTILIZADAS	23
3.3.1	Ambiente de Desenvolvimento	24
3.3.2	Gerenciamento de Dependências	24
3.3.3	Backend e Serviços	24

3.3.4	Design e Interface	24
3.4	METODOLOGIA DE VALIDAÇÃO	24
4	DESENVOLVIMENTO	26
4.1	ARQUITETURA DO SISTEMA	26
4.2	ESTRUTURA DO PROJETO	27
4.3	TECNOLOGIAS E DEPENDÊNCIAS	27
4.4	FUNCIONALIDADES IMPLEMENTADAS	28
4.4.1	Autenticação	28
4.4.2	Gestão de Provas (Admin)	28
4.4.3	Realização de Provas	29
4.4.4	Sistema de Certificados	29
4.4.5	Gerenciamento de Usuários	29
4.5	ESTRUTURA DE DADOS	30
4.5.1	users	30
4.5.2	provas	30
4.5.3	provasaprovadas	30
4.5.4	certificados	30
4.5.5	certificado Config	30
4.6	SEGURANÇA	30
4.6.1	Regras do Firestore	30
4.6.2	Regras do Storage	31
4.6.3	Validações no Cliente	31
4.7	INTERFACE DO USUÁRIO	31
5	RESULTADOS	32
5.1	FUNCIONALIDADES IMPLEMENTADAS	32
5.1.1	Comparação com os Objetivos	32
5.2	PLATAFORMAS SUPORTADAS	32
5.3	INTERFACE DO SISTEMA	33
5.4	TESTES REALIZADOS	36
5.4.1	Testes Funcionais	36
5.4.2	Testes de Segurança	37
5.4.3	Testes de Usabilidade	37
5.4.4	Testes de Performance	37
5.5	DESEMPENHO	37
5.6	LIMITAÇÕES IDENTIFICADAS	37
5.7	MÉTRICAS DO PROJETO	38
6	CONSIDERAÇÕES FINAIS	39
6.1	CONCLUSÃO	39
6.2	CONTRIBUIÇÕES	39

6.3	TRABALHOS FUTUROS	40
6.3.1	Funcionalidades Planejadas	40
6.3.2	Melhorias Técnicas	41
6.4	CONSIDERAÇÕES FINAIS	41
	Referências	42
	APÊNDICE A – Descrição	44
	APÊNDICE B – Regras de Segurança do Firestore e Storage	46
	APÊNDICE C – Estrutura das Coleções do Firestore	49

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O Clube de Desbravadores é uma organização mundial pertencente à Igreja Adventista do Sétimo Dia, atuando há décadas na formação de crianças e adolescentes, promovendo educação, disciplina, liderança e desenvolvimento espiritual (IASD, 2022). O programa inclui atividades ao ar livre, desenvolvimento de habilidades práticas e o sistema de especialidades, que permite aos participantes adquirirem conhecimentos em diversas áreas temáticas (White, 2015).

As especialidades constituem um componente essencial do programa educacional dos Desbravadores, abrangendo temas como ciências, habilidades manuais, vida ao ar livre, arte, natureza e recreação. Cada especialidade possui requisitos específicos, avaliados por instrutores capacitados. Após cumprir os requisitos, o desbravador recebe um certificado validando sua conquista.

Tradicionalmente, esse processo de avaliação ocorre de maneira totalmente manual, exigindo tempo extensivo dos instrutores para criação de provas, correção e emissão de certificados (Santos; Oliveira, 2020). Esse modelo analógico gera grande demanda operacional, reduzindo o tempo disponível para orientação e acompanhamento dos participantes.

Com a crescente digitalização e expansão de tecnologias móveis e sistemas educacionais online, torna-se necessário desenvolver soluções tecnológicas que automatizem processos pedagógicos e administrativos. A transformação digital tem impactado positivamente diversos setores, incluindo instituições de ensino e programas educacionais.

Nesse contexto, o sistema *EspecialidadesOn* foi idealizado com o objetivo de modernizar e digitalizar o processo de avaliação. Trata-se de uma solução multiplataforma, desenvolvida com Flutter e Firebase, que permite aos instrutores criar provas, avaliar desbravadores e emitir certificados automaticamente, eliminando a necessidade de tarefas manuais repetitivas (Google Flutter Team, 2024; Google Firebase, 2024).

1.2 PROBLEMA

O processo tradicional manual apresenta diversos desafios que comprometem a eficiência operacional dos instrutores e a qualidade da gestão das especialidades. Segundo estudos sobre automação de processos educacionais, atividades manuais excessivas reduzem a produtividade e dificultam a padronização (Santos; Oliveira, 2020).

Os principais problemas identificados incluem:

- a) **Tempo excessivo na criação de provas:** Instrutores demandam horas para elaborar avaliações, reduzindo o tempo disponível para orientação pedagógica;
- b) **Demora na emissão de certificados:** Processos manuais retardam a entrega de resultados e criam acúmulo de tarefas administrativas;

- c) **Falta de padronização:** Materiais criados individualmente dificultam consistência e uniformidade (Santos; Oliveira, 2020);
- d) **Dificuldade de organizar e armazenar documentos:** Arquivos físicos ou não estruturados dificultam busca e histórico;
- e) **Necessidade de presença física:** O modelo impede que desbravadores realizem provas remotamente;
- f) **Risco de falsificação de certificados:** Processos manuais não garantem autenticidade documental (Barbosa, 2019);
- g) **Limitada acessibilidade tecnológica:** Ausência de plataforma unificada impede acesso em múltiplos dispositivos.

1.3 OBJETIVOS

1.3.1 OBJETIVO GERAL

Desenvolver um sistema multiplataforma para digitalizar e automatizar o processo de avaliação de especialidades do Clube de Desbravadores.

1.3.2 OBJETIVOS ESPECÍFICOS

- a) Criar interface administrativa para cadastro e gerenciamento de provas;
- b) Desenvolver sistema automático de avaliação com cálculo de pontuação;
- c) Implementar geração de certificados digitais em PDF;
- d) Implementar controle de permissão por tipos de usuário;

1.4 JUSTIFICATIVA

A modernização do processo de avaliação de especialidades é necessária para melhorar a eficiência da gestão, garantir segurança documental e promover maior acessibilidade aos participantes. A digitalização reduz significativamente o tempo gasto em tarefas repetitivas e melhora a organização dos registros.

O uso de um sistema automatizado proporciona:

- a) **Economia de tempo:** Redução de esforço manual na criação de provas e certificados (Santos; Oliveira, 2020);
- b) **Automatização eficiente:** O sistema realiza cálculos, validações e geração documental automaticamente;
- c) **Padronização:** Conteúdos seguem estruturas uniformes, reduzindo erros e inconsistências;
- d) **Acessibilidade:** Provas podem ser realizadas de qualquer dispositivo conectado

- e) **Escalabilidade:** A solução suporta grande volume de usuários devido ao uso do Firebase (Google Firebase, 2024);
- f) **Tecnologia moderna:** Flutter permite desenvolvimento de um único código que funciona em várias plataformas (Google Flutter Team, 2024).

1.5 ESTRUTURA DO TRABALHO

O presente trabalho está organizado em seis capítulos, estruturados de forma a permitir uma compreensão progressiva do tema abordado.

O Capítulo 1 apresenta a introdução do trabalho, contextualizando o problema de pesquisa, os objetivos geral e específicos, bem como a justificativa e a relevância da proposta desenvolvida.

O Capítulo 2 contempla o referencial teórico, no qual são discutidos os principais conceitos, fundamentos e tecnologias relacionados à digitalização de processos avaliativos, certificação digital e desenvolvimento de sistemas multiplataforma, servindo de base conceitual para o desenvolvimento do sistema proposto.

O Capítulo 3 descreve a metodologia adotada para a realização do trabalho, abordando as etapas de planejamento, modelagem, desenvolvimento e testes do sistema EspecialidadesON, bem como a definição das ferramentas e técnicas utilizadas ao longo do processo.

O Capítulo 4 apresenta o desenvolvimento do sistema, detalhando a arquitetura adotada, as funcionalidades implementadas, o fluxo de funcionamento da aplicação e as principais decisões técnicas tomadas durante a implementação.

O Capítulo 5 expõe os resultados obtidos com a implementação do sistema, incluindo a análise do funcionamento da aplicação, a validação das funcionalidades propostas e a avaliação dos benefícios alcançados em relação aos objetivos estabelecidos.

Por fim, o Capítulo 6 reúne as considerações finais, destacando as conclusões do trabalho, suas contribuições, limitações identificadas e sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos que embasam o desenvolvimento do sistema *EspecialidadesOn*, abordando as principais tecnologias, conceitos e padrões arquiteturais utilizados no projeto. A compreensão dessas bases é fundamental para justificar as escolhas técnicas e a arquitetura adotada.

2.1 FLUTTER

Flutter é um framework de código aberto desenvolvido pelo Google para criação de aplicativos multiplataforma. Ele permite desenvolver aplicações nativas para Android, iOS, Web, Windows, macOS e Linux utilizando um único código base, sendo atualmente um dos frameworks mais populares no desenvolvimento multiplataforma (Google Flutter Team, 2024).

2.1.1 CARACTERÍSTICAS PRINCIPAIS

O Flutter utiliza a linguagem Dart, também criada pelo Google, que apresenta alta performance e recursos modernos para construção de interfaces reativas. O framework adota widgets como elementos fundamentais da interface, compondo desde elementos simples até telas completas (Google Flutter Team, 2024).

Além disso, o Flutter conta com um motor de renderização próprio, garantindo que a interface seja desenhada diretamente na tela, sem depender de elementos nativos. Isso resulta em alta performance e consistência visual entre plataformas.

2.1.2 VANTAGENS DO FLUTTER

Entre as principais vantagens do Flutter, destacam-se:

- a) **Desenvolvimento multiplataforma:** Um único código base gera aplicações para diversas plataformas, reduzindo custo de desenvolvimento;
- b) **Alta performance:** Compilação para código nativo garante fluidez;
- c) **Interface rica:** Widgets altamente customizáveis possibilitam interfaces modernas;
- d) **Hot Reload:** Agilidade no desenvolvimento ao aplicar mudanças instantaneamente;
- e) **Ecosistema robusto:** Grande quantidade de pacotes e documentação oficial extensa;
- f) **Comunidade crescente:** Forte comunidade global de suporte.

2.2 FIREBASE

Firebase é uma plataforma BaaS (Backend as a Service) que fornece serviços como autenticação, banco de dados, armazenamento e hospedagem. Isso permite que aplicações tenham estruturas robustas sem necessidade de servidores próprios (Google Firebase, 2024).

2.2.1 FIREBASE AUTHENTICATION

O Firebase Authentication fornece autenticação segura utilizando provedores como email/senha, Google e outros. Ele implementa segurança baseada em tokens JWT, garantindo integridade e proteção dos dados do usuário (Google Firebase, 2024).

2.2.2 CLOUD FIRESTORE

O Cloud Firestore é um banco de dados NoSQL escalável e flexível, baseado em documentos e coleções. Ele atende aplicações que exigem sincronização em tempo real e escalabilidade automatizada (Google Firebase, 2024).

2.2.3 FIREBASE STORAGE

O Firebase Storage permite armazenar arquivos com segurança, possuindo regras configuráveis que controlam permissão de acesso. Ele é amplamente utilizado para armazenar imagens, PDFs e arquivos gerados dinamicamente (Google Firebase, 2024).

2.3 ARQUITETURA MVC

O padrão arquitetural MVC (Model-View-Controller) divide a aplicação em três componentes, promovendo maior organização, reuso e testabilidade do código (Krasner; Pope, 1988).

2.3.1 MODEL (MODELO)

O Model representa estruturas de dados e regras de negócio, sendo responsável por gerenciar o estado da aplicação.

2.3.2 VIEW (VISÃO)

A View exibe informações ao usuário. No Flutter, as views são construídas por widgets que formam a interface gráfica.

2.3.3 CONTROLLER (CONTROLADOR)

O Controller recebe ações do usuário, atualiza o Model e instrui a View a se atualizar. Essa separação melhora a manutenção e evolução do software.

2.3.4 VANTAGENS DO MVC

- a) **Separação de responsabilidades:** Torna o código mais limpo e modular;
- b) **Testabilidade:** Permite testes isolados de cada camada;
- c) **Reutilização:** Facilita o reaproveitamento de componentes;
- d) **Organização:** Estrutura robusta e amplamente recomendada pela engenharia de software.

2.4 CERTIFICADOS DIGITAIS

Certificados digitais são documentos eletrônicos usados para comprovar autenticidade, conclusão de atividades ou competências adquiridas. Em sistemas modernos, são amplamente utilizados devido à facilidade de distribuição e verificação (Barbosa, 2019).

2.4.1 VALIDAÇÃO DE CERTIFICADOS

Para garantir a autenticidade de um certificado, utiliza-se um identificador único, geralmente baseado em hash criptográfico. Isso impede falsificação e facilita a verificação.

2.4.2 HASH SHA-256

O SHA-256 é uma função de hash criptográfica amplamente utilizada. Pequenas alterações nos dados geram um hash completamente diferente, garantindo integridade e segurança.

2.5 PDF (PORTABLE DOCUMENT FORMAT)

O formato PDF é amplamente utilizado por sua consistência visual em diferentes plataformas. É ideal para certificados por manter alta qualidade, padronização e facilidade de impressão.

2.5.1 VANTAGENS DO PDF PARA CERTIFICADOS

- a) Padronização universal;
- b) Segurança e possibilidade de proteção;
- c) Alta qualidade visual;
- d) Portabilidade entre dispositivos;

- e) Fácil impressão e distribuição.

3 METODOLOGIA

Este capítulo apresenta a metodologia adotada para o desenvolvimento do sistema *EspecialidadesOn*, descrevendo o tipo de pesquisa, a metodologia de desenvolvimento e as ferramentas utilizadas (Gil, 2019; Marconi Marina; Lakatos, 2021).

3.1 TIPO DE PESQUISA

Este trabalho caracteriza-se como uma pesquisa aplicada, de natureza tecnológica, com abordagem qualitativa. A pesquisa aplicada visa resolver problemas práticos e gerar conhecimentos voltados à solução de necessidades específicas (Marconi Marina; Lakatos, 2021).

A natureza tecnológica do trabalho está associada ao desenvolvimento de um produto computacional — um sistema de software — destinado a otimizar processos manuais existentes. A abordagem qualitativa permite analisar, descrever e interpretar as funcionalidades implementadas, arquitetura do sistema e resultados obtidos (Gil, 2019).

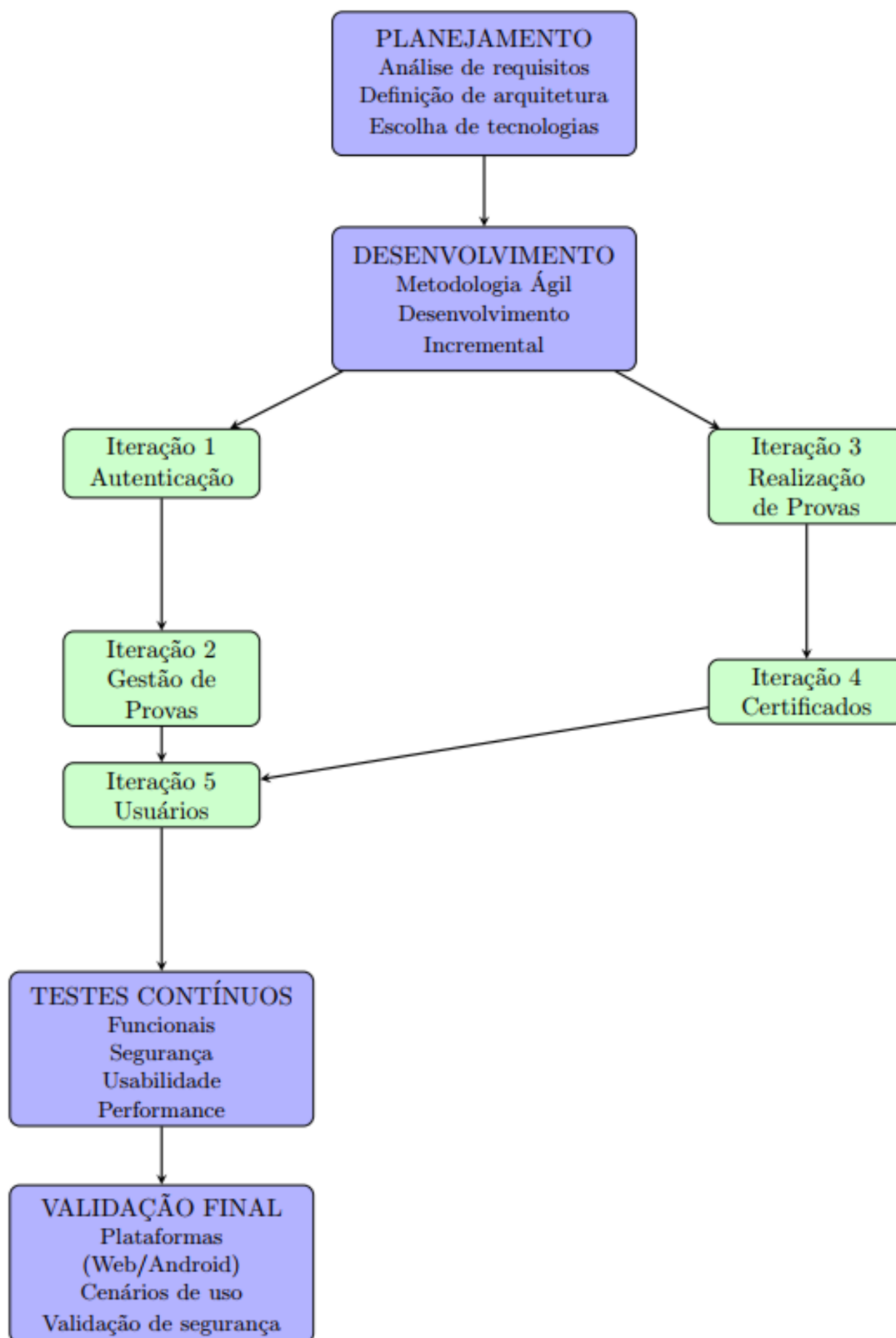
3.2 METODOLOGIA DE DESENVOLVIMENTO

O desenvolvimento do sistema seguiu uma metodologia ágil, baseada em ciclos iterativos e entregas incrementais, permitindo adaptar rapidamente requisitos e prioridades conforme a evolução do projeto (MANIFESTO... , 2001).

Além disso, boas práticas de engenharia de software foram aplicadas durante o processo de construção e organização do código.

O fluxo do processo de desenvolvimento é ilustrado na Figura 1.

Figura 1 – Fluxo de desenvolvimento do aplicativo *EspecialidadesON*.



Fonte: Autor (2025).

3.2.1 ETAPAS DO DESENVOLVIMENTO

O desenvolvimento foi dividido em etapas seguindo práticas recomendadas pela engenharia de software:

3.2.1.1 PLANEJAMENTO

Nesta etapa, foram definidos os requisitos funcionais e não funcionais do sistema, junto da arquitetura geral, tecnologias utilizadas e cronograma de execução. Também foram analisadas as necessidades dos usuários e priorizadas as funcionalidades mais relevantes.

3.2.1.2 DESENVOLVIMENTO

A implementação ocorreu de forma incremental, priorizando entregas contínuas conforme a metodologia ágil recomenda (MANIFESTO... , 2001; Schwaber; Sutherland, 2020).

As funcionalidades foram construídas na seguinte ordem:

- a) Sistema de autenticação;
- b) Interface de cadastro e gerenciamento de provas;
- c) Sistema de realização de provas;
- d) Geração de certificados digitais;
- e) Sistema de validação de certificados;
- f) Gerenciamento de usuários;
- g) Funcionalidades administrativas.

3.2.1.3 TESTES

Os testes foram executados continuamente ao longo do desenvolvimento, incluindo:

- a) Testes funcionais para verificar o correto funcionamento das features;
- b) Testes de segurança para validar regras de acesso e proteção de dados;
- c) Testes de usabilidade seguindo boas práticas acadêmicas e de UX;
- d) Testes de performance para avaliar o comportamento do sistema em uso real.

Essas práticas são recomendadas pela engenharia de software moderna .

3.3 FERRAMENTAS UTILIZADAS

As ferramentas adotadas neste projeto foram selecionadas por sua eficiência, popularidade e integração com tecnologias utilizadas em desenvolvimento mobile e web.

3.3.1 AMBIENTE DE DESENVOLVIMENTO

- a) **Visual Studio Code:** Editor amplamente utilizado para Flutter devido à sua leveza e extensões dedicadas (Google Flutter Team, 2024);
- b) **Android Studio:** IDE recomendada pelo Google para testes Android e ferramentas avançadas de emulação (Google Flutter Team, 2024);
- c) **Git:** Sistema de controle de versão distribuído usado em larga escala (Chacon; Straub, 2014);
- d) **GitHub:** Plataforma utilizada para repositório remoto e colaboração (GITHUB..., 2024).

3.3.2 GERENCIAMENTO DE DEPENDÊNCIAS

- a) **Pub:** Gerenciador de dependências oficial do Flutter e da linguagem Dart, responsável pelo controle, versionamento e resolução das bibliotecas utilizadas no projeto (Google Flutter Team, 2024);
- b) **Pub.dev:** Repositório oficial de bibliotecas e pacotes do ecossistema Flutter e Dart, mantido pela comunidade e pelo Google (PUB-DEV..., 2024).

3.3.3 BACKEND E SERVIÇOS

- a) **Firebase Console:** Ferramenta de gerenciamento da plataforma Firebase (Google Firebase, 2024);
- b) **Firebase Authentication:** Serviço de autenticação integrado ao sistema (Google Firebase, 2024);
- c) **Cloud Firestore:** Banco de dados NoSQL escalável (Google Firebase, 2024);
- d) **Firebase Storage:** Armazenamento seguro de arquivos e imagens dos usuários (Google Firebase, 2024).

3.3.4 DESIGN E INTERFACE

- a) **Material Design:** Sistema de design utilizado como base visual da aplicação (Google Material Design Team, 2024);
- b) **Flutter Widgets:** Componentes estruturais da interface gráfica do Flutter (Google Flutter Team, 2024).

3.4 METODOLOGIA DE VALIDAÇÃO

A validação do sistema foi conduzida por meio de testes práticos em múltiplas plataformas (Web e Android), garantindo comportamento uniforme em diferentes ambientes.

Além disso, foram realizadas validações de segurança, analisando regras de permissão e fluxo de dados conforme boas práticas da engenharia de software.

A usabilidade foi validada com base em testes com usuários, seguindo diretrizes de experiência do usuário recomendadas por estudos acadêmicos e frameworks de design (Google Material Design Team, 2024).

4 DESENVOLVIMENTO

Este capítulo apresenta o desenvolvimento do sistema *EspecialidadesOn*, abordando a arquitetura, estrutura do projeto, tecnologias utilizadas, funcionalidades implementadas, estrutura de dados e aspectos de segurança. As decisões técnicas adotadas baseiam-se em referências consolidadas em engenharia de software e desenvolvimento multiplataforma.

4.1 ARQUITETURA DO SISTEMA

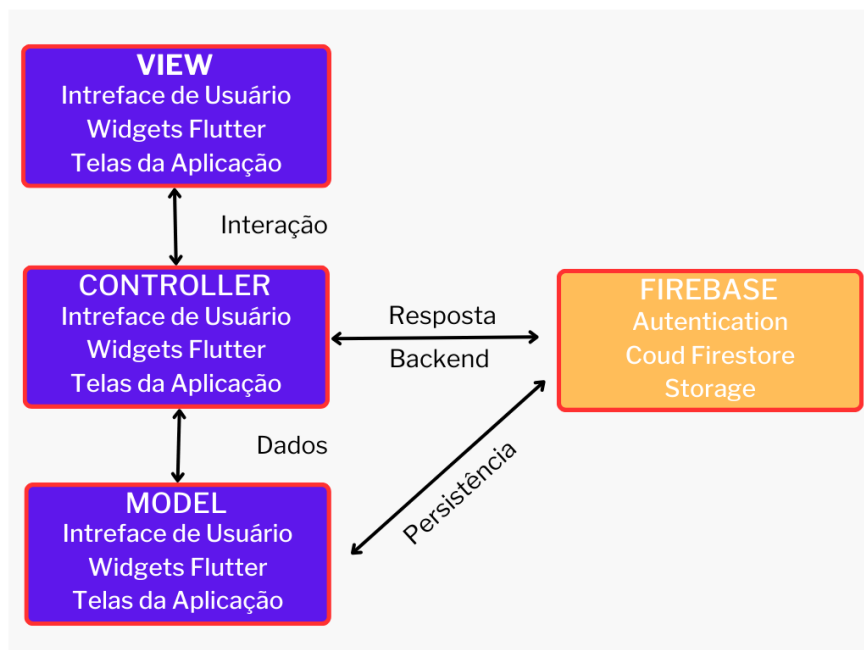
O sistema segue o padrão arquitetural MVC (Model-View-Controller), amplamente utilizado para organização de aplicações por promover separação de responsabilidades e facilitar manutenção e escalabilidade (Reenskaug, 1979).

Na arquitetura:

- View:** responsável pela interface gráfica com widgets Flutter;
- Controller:** gerencia a lógica de negócio, validações e fluxo;
- Model:** representa os dados e estruturas essenciais do sistema;
- Firestore Backend:** fornece autenticação, banco de dados e armazenamento em nuvem (Google Firebase, 2024).

A arquitetura geral é ilustrada na Figura 2.

Figura 2 – Arquitetura MVC do sistema *EspecialidadesON*.



Fonte: Autor (2025).

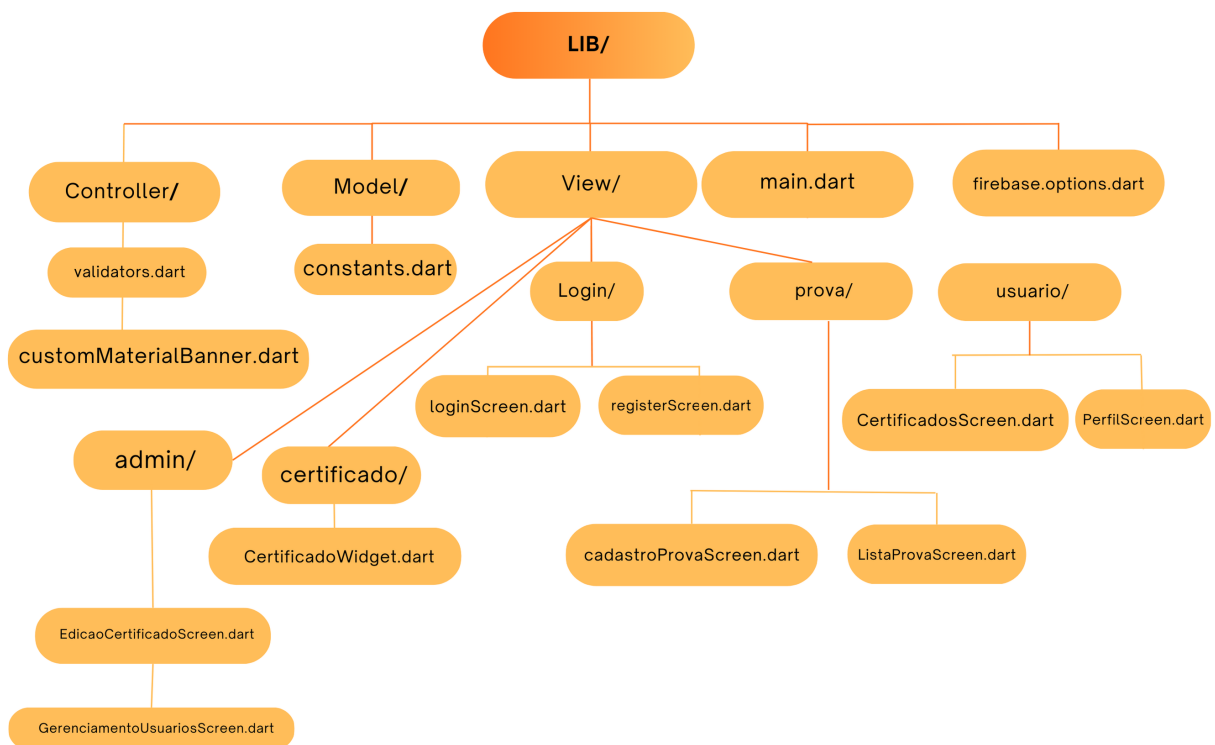
4.2 ESTRUTURA DO PROJETO

A organização do projeto segue boas práticas recomendadas para aplicações Flutter estruturadas em MVC (Google Flutter Team, 2024).

A estrutura está descrita na Figura 3 e inclui:

- Controller/**: lógica de interação e regras de negócio;
- Model/**: definição dos modelos e constantes;
- View/**: telas da aplicação divididas em módulos.

Figura 3 – Estrutura de diretórios do projeto *EspecialidadesON*.



Fonte: Autor (2025).

4.3 TECNOLOGIAS E DEPENDÊNCIAS

Flutter foi escolhido por permitir desenvolvimento multiplataforma em um único código base (**flutter2024**). Firebase foi adotado pela integração nativa com Flutter e infraestrutura escalável (Google Firebase, 2024).

A Tabela 1 apresenta as dependências principais.

Tabela 1 – Principais tecnologias e dependências.

Tecnologia	Descrição	Versão
Flutter	Framework multiplataforma para desenvolvimento de aplicações	3.3.0
Dart	Linguagem de programação utilizada pelo Flutter	3.0.0
firebase_core	Serviço central de inicialização do Firebase	4.2.0
firebase_auth	Serviço de autenticação de usuários	6.1.1
cloud_firestore	Banco de dados NoSQL em tempo real	4.0.3
firebase_storage	Serviço de armazenamento de arquivos	13.0.3
google_sign_in	Autenticação por conta Google	6.0.1
pdf	Geração de documentos em formato PDF	3.11.1
printing	Impressão e visualização de documentos	5.13.3
crypto	Geração de hash criptográfico SHA-256	3.0.3

Fonte: Autor (2025).

4.4 FUNCIONALIDADES IMPLEMENTADAS

4.4.1 AUTENTICAÇÃO

A autenticação foi implementada com Firebase Authentication, que oferece segurança e suporte a múltiplos provedores (Google Firebase, 2024). Inclui:

- a) Login com email e senha;
- b) Login com Google (OAuth 2.0) (Hardt, 2012);
- c) Recuperação de senha;
- d) Controle de perfis (usuário/admin).

As validações seguem diretrizes de segurança recomendadas (OWASP Foundation, 2023).

4.4.2 GESTÃO DE PROVAS (ADMIN)

Os instrutores possuem acesso a uma interface administrativa que permite a criação, edição e gerenciamento das provas aplicadas no sistema. Cada prova é composta por um conjunto de campos que definem suas características e funcionamento, conforme descrito a seguir:

- a) **Título da prova:** Identificação textual da prova, geralmente associada ao nome da especialidade avaliada;
- b) **Área da especialidade:** Campo utilizado para vincular a prova à especialidade correspondente dentro do Clube de Desbravadores;
- c) **Tipo da prova:** Define o formato da avaliação, sendo atualmente configurada como prova objetiva de múltipla escolha;
- d) **Imagem ilustrativa:** Elemento visual opcional utilizado para identificação da especialidade ou contextualização da prova;

- e) **Questões e respostas:** Conjunto de perguntas cadastradas pelo instrutor, cada uma contendo alternativas de resposta;
- f) **Pontuação:** Valor atribuído a cada questão, utilizado para o cálculo automático do desempenho do desbravador;
- g) **Definição da resposta correta:** Indicação da alternativa correta para cada questão, necessária para validação automática da prova.

Esse fluxo de gerenciamento segue modelos consolidados de sistemas de avaliação digital, permitindo padronização, automação do processo avaliativo e redução de erros operacionais (Juric; Kovacic, 2020).

4.4.3 REALIZAÇÃO DE PROVAS

Os desbravadores podem realizar provas digitalmente, com:

- a) cálculo automático de pontuação;
- b) regra de aprovação de 70%;
- c) registro de conclusão;
- d) acesso ao certificado após aprovação.

Provas online têm ampla aplicação em ambientes educacionais (Juric; Kovacic, 2020).

4.4.4 SISTEMA DE CERTIFICADOS

A geração automática de certificados utiliza o pacote pdf do Flutter combinada com hash SHA-256 para validação, conforme práticas de segurança criptográfica modernas (Stallings, 2017). Inclui:

- a) template dinâmico por área;
- b) código de validação formatado (XXXX-XXXX-XXXX);
- c) armazenamento de configuração visual no Firestore;
- d) geração de PDF pronta para impressão.

4.4.5 GERENCIAMENTO DE USUÁRIOS

Administradores podem:

- a) visualizar usuários;
- b) promover/demitir administradores;
- c) excluir contas;
- d) ver estatísticas (provas realizadas, certificados).

A prática segue diretrizes de RBAC (Role-Based Access Control) (Sandhu et al., 1996).

4.5 ESTRUTURA DE DADOS

O sistema utiliza Cloud Firestore (NoSQL), adequado para aplicações em tempo real e escaláveis (Google Firebase, 2024).

As coleções principais são:

4.5.1 USERS

Armazena dados do usuário e privilégios.

4.5.2 PROVAS

Estrutura de provas e questões.

4.5.3 PROVASAPROVADAS

Registro de aprovações, pontuação e data.

4.5.4 CERTIFICADOS

Certificados gerados e dados associados ao hash.

4.5.5 CERTIFICADO CONFIG

Configurações visuais personalizadas.

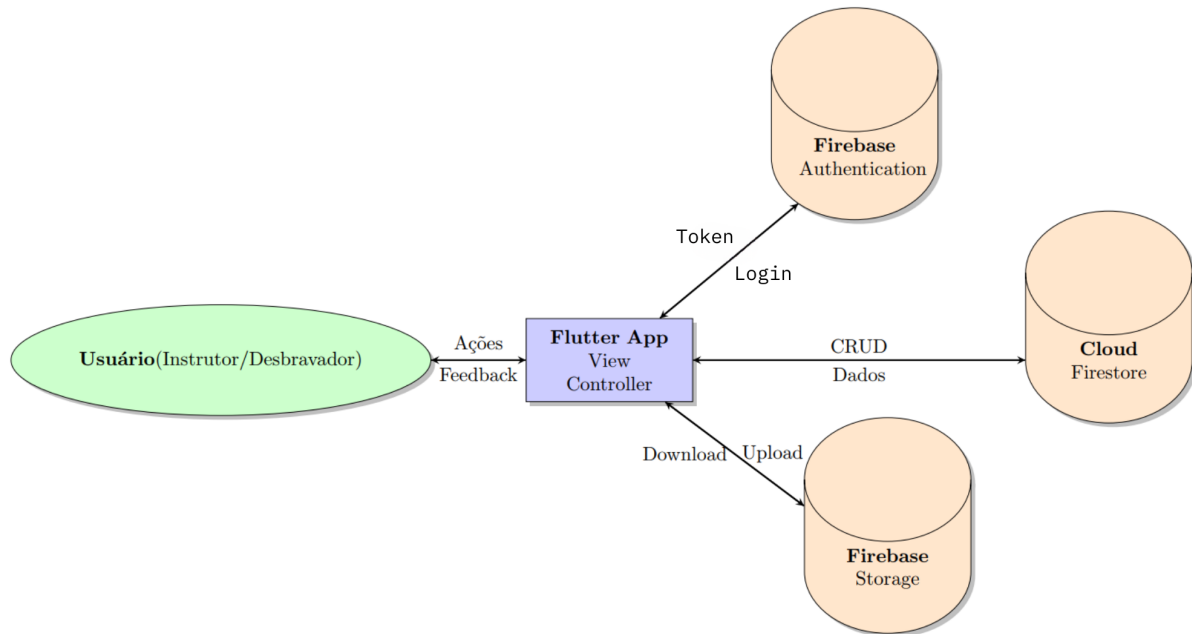
O fluxo de dados é ilustrado na Figura 4.

4.6 SEGURANÇA

Segurança foi priorizada em todas as etapas, seguindo recomendações da OWASP para aplicações móveis e web (OWASP Foundation, 2023).

4.6.1 REGRAS DO FIRESTORE

- a) acesso limitado por autenticação;
- b) privilégios diferenciados (usuário/admin);
- c) leituras e escritas restritas ao próprio usuário;
- d) controle de exclusões apenas por administradores.

Figura 4 – Fluxo de dados entre as camadas do sistema *EspecialidadesON*.

Fonte: Autor (2025).

4.6.2 REGRAS DO STORAGE

- uploads permitidos somente a usuários autenticados;
- limitação de tamanho e tipo de arquivo;
- uploads administrativos apenas para imagens de provas.

4.6.3 VALIDAÇÕES NO CLIENTE

- validação de email com regex;
- validação de senha (mín. 6 caracteres);
- sanitização de entradas;
- verificação de tamanho e tipo de arquivo.

4.7 INTERFACE DO USUÁRIO

A interface segue diretrizes do Material Design, proporcionando simplicidade, consistência e acessibilidade (Google Material Design Team, 2024).

O Flutter permite interfaces responsivas e adaptáveis a diferentes tamanhos de tela (Google Flutter Team, 2024).

5 RESULTADOS

Este capítulo apresenta os resultados obtidos com o desenvolvimento do sistema *EspecialidadesON*, contemplando as funcionalidades implementadas, plataformas suportadas, interfaces desenvolvidas, testes realizados, desempenho do sistema e limitações identificadas.

5.1 FUNCIONALIDADES IMPLEMENTADAS

Todas as funcionalidades planejadas foram implementadas com sucesso. A Tabela 2 apresenta um resumo das funcionalidades desenvolvidas e seu respectivo status.

Tabela 2 – Funcionalidades implementadas.

Funcionalidade	Descrição	Status
Autenticação	Login com email/senha e Google	Concluído
Gestão de Provas	Cadastro e edição de provas	Concluído
Realização de Provas	Sistema de avaliação com cálculo automático	Concluído
Geração de Certificados	Emissão de certificados em PDF	Concluído
Validação de Certificados	Hash SHA-256 e código de validação	Concluído
Gerenciamento de Usuários	CRUD completo de usuários	Concluído
Edição de Certificados	Configuração visual de templates	Concluído

Fonte: Autor (2025).

5.1.1 COMPARAÇÃO COM OS OBJETIVOS

Os resultados alcançados demonstram que todos os objetivos específicos definidos no Capítulo 1 foram atendidos, conforme descrito a seguir:

- a) Implementação de um sistema de autenticação seguro;
- b) Desenvolvimento de interface administrativa para cadastro e gerenciamento de provas;
- c) Implementação de sistema de avaliação com cálculo automático de pontuação;
- d) Geração automatizada de certificados digitais em formato PDF;
- e) Implementação de mecanismo de validação por meio de hash criptográfico;
- f) Disponibilização de interface responsiva para múltiplas plataformas;
- g) Controle de acesso conforme perfil de usuário;
- h) Implementação de funcionalidades de gerenciamento de usuários.

5.2 PLATAFORMAS SUPORTADAS

O sistema foi desenvolvido para operar de forma multiplataforma e testado nos seguintes ambientes:

- a) **Web:** Testado nos navegadores Google Chrome, Microsoft Edge e Mozilla Firefox;
- b) **Android:** Testado em dispositivos Android com diferentes versões do sistema operacional;
- c) **iOS:** Preparado para compilação, não sendo testado em dispositivos físicos.

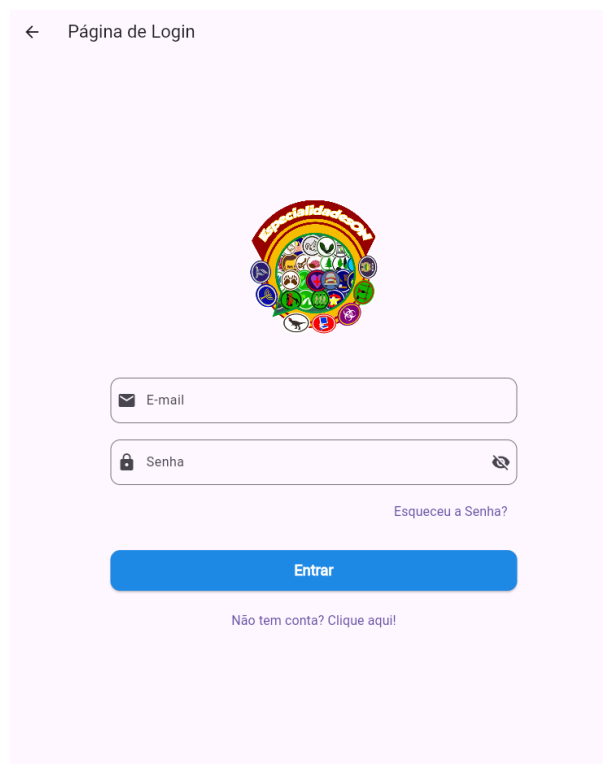
Os testes realizados indicaram funcionamento consistente da aplicação nas plataformas avaliadas, mantendo uniformidade de interface e funcionalidades.

5.3 INTERFACE DO SISTEMA

A interface do sistema foi desenvolvida com foco na usabilidade, seguindo princípios do Material Design, a fim de proporcionar uma experiência intuitiva e padronizada aos usuários.

A Figura 5 apresenta a tela de autenticação do sistema, permitindo o acesso por meio de credenciais cadastradas ou autenticação via conta Google.

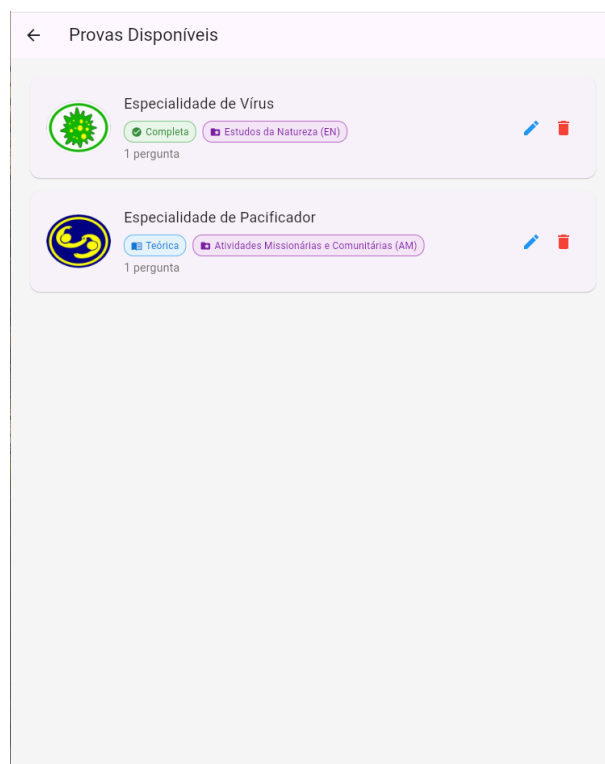
Figura 5 – Tela de login do sistema



Fonte: Autor (2025).

A Figura 6 ilustra a tela de listagem de especialidades disponíveis, na qual os desbravadores podem visualizar e selecionar as especialidades para realização das avaliações.

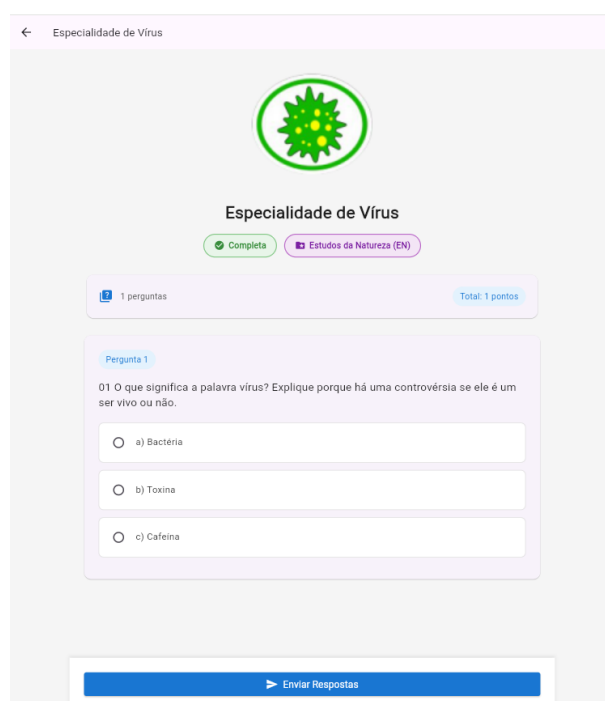
Figura 6 – Tela de listagem de especialidades disponíveis



Fonte: Autor (2025).

A Figura 7 demonstra a interface destinada à realização das provas, onde os usuários respondem às questões de múltipla escolha conforme a especialidade selecionada.

Figura 7 – Tela de realização de prova



Fonte: Autor (2025).

A Figura 8 apresenta um exemplo de certificado digital gerado automaticamente pelo sistema após a aprovação do desbravador em uma especialidade.

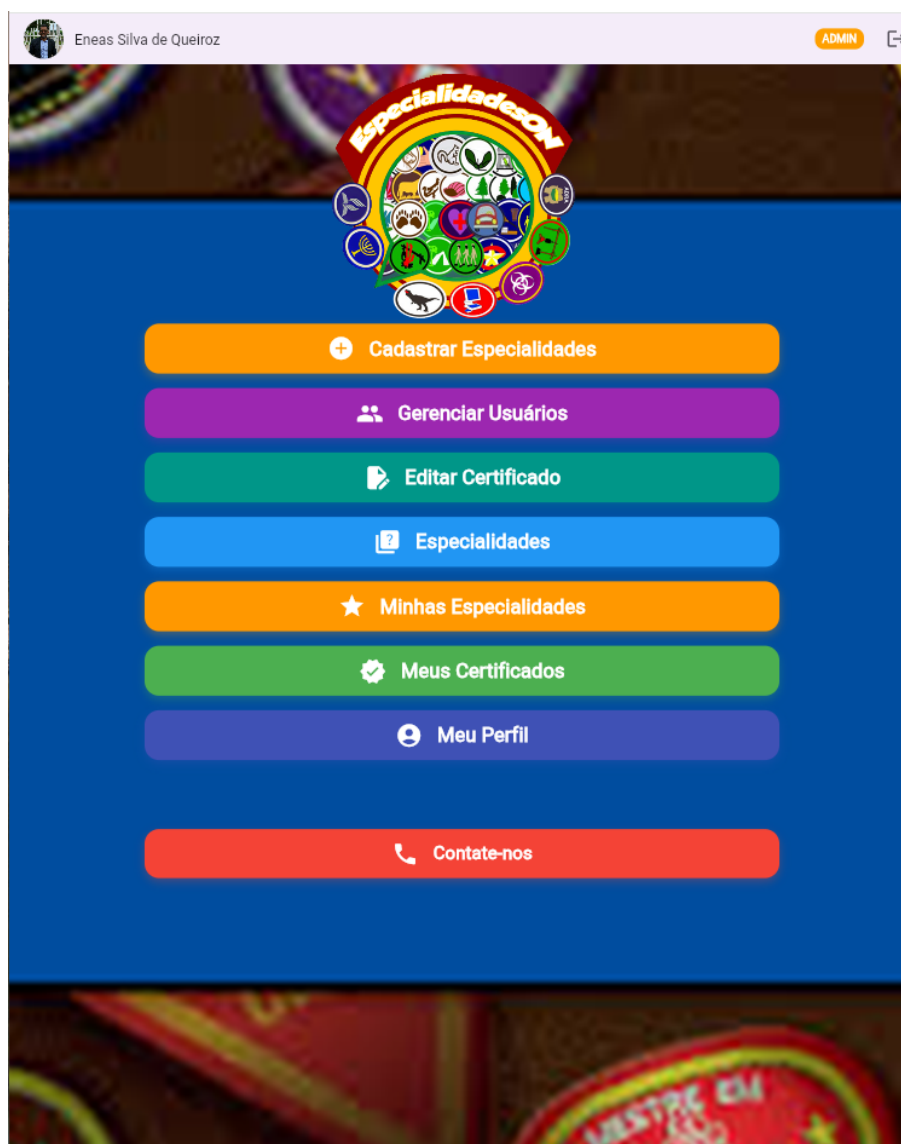
Figura 8 – Exemplo de certificado gerado pelo sistema



Fonte: Autor (2025).

Por fim, a Figura 9 ilustra o painel administrativo do sistema, no qual os instrutores podem gerenciar provas, usuários e visualizar informações relevantes para o acompanhamento das atividades.

Figura 9 – Dashboard administrativo do sistema



Fonte: Autor (2025).

5.4 TESTES REALIZADOS

5.4.1 TESTES FUNCIONAIS

Foram realizados testes funcionais com o objetivo de validar o correto funcionamento das funcionalidades implementadas.

- Autenticação de usuários, incluindo login, registro, recuperação de senha e logout;
- Cadastro, edição e realização de provas;
- Cálculo automático de pontuação e verificação de aprovação;
- Geração, visualização e validação de certificados digitais.

Os testes funcionais apresentaram resultados satisfatórios, indicando o correto funcionamento do sistema.

5.4.2 TESTES DE SEGURANÇA

Os testes de segurança verificaram as regras de acesso e proteção de dados, assegurando que apenas usuários autorizados possam acessar informações sensíveis.

- a) Validação das regras de acesso ao Firestore e Storage;
- b) Controle de acesso por perfil de usuário;
- c) Proteção de dados sensíveis.

Os resultados confirmaram a adequação das regras de segurança implementadas.

5.4.3 TESTES DE USABILIDADE

Foram realizados testes de usabilidade visando garantir facilidade de uso e navegação intuitiva.

- a) Clareza na navegação entre telas;
- b) Feedback visual para ações do usuário;
- c) Mensagens de erro claras e informativas.

Os testes indicaram boa aceitação da interface pelos usuários.

5.4.4 TESTES DE PERFORMANCE

Os testes de performance avaliaram o comportamento do sistema sob condições normais de uso.

- a) Carregamento rápido das telas;
- b) Sincronização eficiente dos dados;
- c) Geração ágil de certificados em PDF.

5.5 DESEMPENHO

O sistema apresentou desempenho satisfatório em todas as operações avaliadas, com tempos adequados de resposta e estabilidade durante a execução das funcionalidades.

5.6 LIMITAÇÕES IDENTIFICADAS

Apesar dos resultados positivos, foram identificadas algumas limitações relacionadas ao escopo do projeto.

- a) Ausência de testes em dispositivos iOS físicos;

- b) Inexistência de limite de tentativas por prova;
- c) Ausência de histórico detalhado de tentativas;
- d) Inexistência de notificações push.

Essas limitações não comprometem o funcionamento do sistema, mas indicam possibilidades de aprimoramento futuro.

5.7 MÉTRICAS DO PROJETO

O desenvolvimento do sistema resultou nos seguintes indicadores:

- a) Aproximadamente 15.000 linhas de código;
- b) Mais de 20 arquivos Dart;
- c) 15 telas implementadas;
- d) 10 widgets reutilizáveis;
- e) 5 coleções principais no Firestore;
- f) Mais de 20 regras de segurança;
- g) Cerca de 30 validações implementadas.

Essas métricas evidenciam a complexidade e o escopo do projeto desenvolvido.

6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta as considerações finais do trabalho, incluindo as conclusões, contribuições, trabalhos futuros e reflexões sobre o desenvolvimento do sistema.

6.1 CONCLUSÃO

O sistema *EspecialidadesOn* foi desenvolvido com sucesso, atendendo a todos os objetivos propostos. A solução desenvolvida oferece uma alternativa moderna e eficiente para digitalizar, atualizar e melhorar o processo de avaliação de especialidades do Clube de Desbravadores, utilizando tecnologias consolidadas no mercado, como Flutter e Firebase.

O sistema implementa todas as funcionalidades planejadas, incluindo autenticação segura, gestão de provas, realização de avaliações, geração de certificados digitais com validação criptográfica e gerenciamento de usuários. A arquitetura MVC adotada garantiu organização e manutenibilidade do código, facilitando o desenvolvimento e a evolução do sistema.

A utilização do Flutter permitiu criar uma aplicação multiplataforma com código único, reduzindo custos de desenvolvimento e manutenção. O Firebase ofereceu uma infraestrutura robusta e escalável, com serviços integrados que facilitaram o desenvolvimento e reduziram a complexidade da arquitetura.

O sistema de validação de certificados através de hash SHA-256 garante a autenticidade e integridade dos documentos emitidos, oferecendo confiabilidade e credibilidade ao processo de certificação. A geração automática de certificados em PDF padroniza a apresentação e facilita a distribuição.

O principal benefício alcançado foi a eliminação da necessidade de os instrutores gastarem horas criando provas e gerando certificados manualmente. Agora, os instrutores podem se concentrar na orientação dos desbravadores e utilizar a plataforma para avaliação das especialidades, otimizando significativamente o tempo e melhorando a qualidade do processo de avaliação.

Os testes realizados confirmaram que o sistema está funcionando corretamente em todas as plataformas testadas (Web e Android), com boa performance e interface intuitiva. As regras de segurança implementadas garantem a proteção adequada dos dados e o controle de acesso por perfil.

6.2 CONTRIBUIÇÕES

Este trabalho contribui para a área de desenvolvimento de sistemas multiplataforma e gestão digital de processos educacionais e de certificação, com foco específico no Clube de Desbravadores. As principais contribuições são:

- a) **Solução específica:** Desenvolvimento de um sistema completo e funcional para digitalizar o processo de avaliação de especialidades do Clube de Desbravadores;
- b) **Otimização de tempo:** Eliminação da necessidade de instrutores gastarem horas criando provas e certificados manualmente;
- c) **Automatização:** Automação completa do fluxo de avaliação e certificação;
- d) **Arquitetura:** Demonstração prática da aplicação do padrão MVC em Flutter;
- e) **Segurança:** Implementação de sistema de validação de certificados com hash criptográfico;
- f) **Multiplataforma:** Uso eficiente do Flutter para criar uma solução completa e acessível em diferentes dispositivos;
- g) **Integração:** Integração eficaz entre Flutter e Firebase;
- h) **Documentação:** Documentação detalhada do processo de desenvolvimento e do sistema.

O sistema desenvolvido pode ser utilizado por clubes de desbravadores em todo o mundo, oferecendo uma alternativa moderna e eficiente aos processos tradicionais. A solução também pode ser adaptada para outras organizações que necessitam gerenciar processos de avaliação e certificação digital.

6.3 TRABALHOS FUTUROS

Durante o desenvolvimento, foram identificadas diversas oportunidades de melhoria e funcionalidades adicionais que podem ser implementadas em trabalhos futuros.

6.3.1 FUNCIONALIDADES PLANEJADAS

- a) Sistema de *roles* mais flexível, permitindo níveis avançados de permissão;
- b) Histórico completo de tentativas de prova;
- c) Limite de tentativas por prova;
- d) Notificações *push*;
- e) Dashboard administrativo avançado;
- f) Relatórios e estatísticas detalhadas;
- g) Validação online de certificados via código de verificação;
- h) Paginação de usuários;
- i) Filtros avançados (clube, data, área etc.);
- j) Exportação de dados (CSV/Excel);
- k) Sistema de backup automático;

- l) Modo offline com sincronização posterior;
- m) Suporte a múltiplos idiomas.

6.3.2 MELHORIAS TÉCNICAS

- a) Testes unitários automatizados;
- b) Testes de integração;
- c) Pipeline CI/CD para automação de *deploy*;
- d) Documentação mais detalhada de APIs;
- e) Monitoramento de erros (ex.: Sentry);
- f) Coleta de métricas de uso (*analytics*);
- g) Otimizações de performance;
- h) Cache local de dados;
- i) Compressão automática de imagens.

6.4 CONSIDERAÇÕES FINAIS

O desenvolvimento do sistema *EspecialidadesON* possibilitou a aplicação prática dos conhecimentos adquiridos ao longo do curso, resultando em uma solução tecnológica voltada à modernização do processo de avaliação das especialidades do Clube de Desbravadores. O trabalho atingiu seu objetivo principal ao propor e implementar um sistema capaz de digitalizar e automatizar a aplicação de provas e a emissão de certificados, contribuindo para maior eficiência, padronização e segurança do processo.

A solução desenvolvida demonstrou-se funcional, estável e adequada às necessidades identificadas, oferecendo uma alternativa moderna aos métodos manuais tradicionalmente utilizados. O sistema proporciona benefícios diretos aos instrutores e desbravadores, reduzindo o tempo gasto em atividades repetitivas e ampliando a acessibilidade às avaliações por meio de uma plataforma digital.

Apesar dos resultados positivos, o trabalho apresenta limitações inerentes ao seu escopo e ao tempo disponível para desenvolvimento. Nesse sentido, melhorias e novas funcionalidades podem ser incorporadas futuramente, conforme discutido no capítulo de Trabalhos Futuros, visando à ampliação das capacidades do sistema e ao seu aprimoramento contínuo.

Conclui-se, portanto, que o sistema *EspecialidadesON* cumpre sua proposta e representa uma contribuição relevante para o processo educacional do Clube de Desbravadores, além de servir como base para o desenvolvimento de futuras soluções tecnológicas voltadas à gestão de avaliações e certificações digitais.

REFERÊNCIAS

BARBOSA, Carlos. Certificação Digital e Autenticidade Documental. **Revista de Segurança da Informação**, 2019.

CHACON, Scott; STRAUB, Ben. **Pro Git**. [S.l.: s.n.], 2014. Disponível em: <https://git-scm.com/book/en/v2>.

GIL, Antônio Carlos. **Métodos e Técnicas de Pesquisa Social**. 6. ed. [S.l.]: Atlas, 2019.

GITHUB Documentation. [S.l.: s.n.], 2024. Disponível em: <https://docs.github.com>.

GOOGLE FIREBASE. **Firestore Documentation**. [S.l.], 2024. Acesso em: 04 dez. 2025.

GOOGLE FLUTTER TEAM. **Flutter Documentation**. [S.l.], 2024. Acesso em: 04 dez. 2025.

GOOGLE MATERIAL DESIGN TEAM. **Material Design Guidelines**. [S.l.], 2024. Acesso em: 04 dez. 2025.

HARDT, D. **The OAuth 2.0 Authorization Framework**. [S.l.: s.n.], 2012. RFC 6749. <https://www.rfc-editor.org/rfc/rfc6749>, Acesso em: 04 dez. 2025.

IASD, Divisão Sul-Americana da. **Manual Administrativo dos Desbravadores**. [S.l.: s.n.], 2022. Acesso em: 12 fev. 2025. Disponível em: <https://www.adventistas.org>.

JURIC, Bojan; KOVACIC, Ana. Digital Assessment Systems in Online Learning Environments. **Journal of Educational Technology**, v. 17, n. 2, p. 45–58, 2020.

KRASNER, Glenn; POPE, Stephen. A description of the Model-View-Controller design pattern. **Journal of Object-Oriented Programming**, 1988.

MANIFESTO para Desenvolvimento Ágil de Software. [S.l.: s.n.], 2001. Disponível em: <https://agilemanifesto.org>.

MARCONI MARINA; LAKATOS, Eva. **Fundamentos de Metodologia Científica**. 8. ed. [S.l.]: Atlas, 2021.

OWASP FOUNDATION. **OWASP Mobile Application Security Verification Standard (MASVS)**. [S.l.: s.n.], 2023. <https://owasp.org/www-project-mobile-security/>. Acesso em: 04 dez. 2025.

PUB-DEV Package Repository. [S.l.: s.n.], 2024. Disponível em: <https://pub.dev>.

REENSKAUG, Trygve. **The Model-View-Controller (MVC) Pattern**. [S.l.: s.n.], 1979. <https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>. Acesso em: 04 dez. 2025.

SANDHU, Ravi; COYNE, Edward; FEINSTEIN, Hal; YOUMAN, Charles. Role-Based Access Control Models. In: PROCEEDINGS of the IEEE Symposium on Research in Security and Privacy. [S.l.: s.n.], 1996. p. 238–249.

SANTOS, Paulo Henrique; OLIVEIRA, Marta. Transformação digital na educação: desafios e oportunidades. **Revista Brasileira de Educação**, 2020.

SCHWABER, Ken; SUTHERLAND, Jeff. **Scrum Guide**. [S.l.], 2020. Disponível em: <https://scrumguides.org>.

STALLINGS, William. **Cryptography and Network Security**. 7. ed. [S.l.]: Pearson, 2017.

WHITE, Ellen G. **História e Missão dos Desbravadores**. [S.l.]: Casa Publicadora Brasileira, 2015.

APÊNDICE A – DESCRIÇÃO

Este apêndice apresenta informações complementares relacionadas ao desenvolvimento do sistema *EspecialidadesOn*, oferecendo detalhes técnicos e descritivos que aprofundam o entendimento das funcionalidades, arquitetura e processos do sistema.

A.1 ESTRUTURA GERAL DO SISTEMA

O sistema foi projetado seguindo o padrão arquitetural MVC (Model–View–Controller), permitindo modularidade, facilidade de manutenção e separação clara entre interface, lógica e dados. A organização interna do sistema reflete a divisão de responsabilidades entre camadas, o que favorece a escalabilidade e futuras expansões.

A.2 FLUXO DE USO DO SISTEMA

O fluxo básico de uso pelos usuários é resumido a seguir:

- a) O usuário realiza login utilizando email/senha ou autenticação Google;
- b) O desbravador acessa a lista de especialidades disponíveis;
- c) O instrutor administra provas, usuários e certificados via dashboard;
- d) Após a realização da prova, o sistema calcula automaticamente a pontuação;
- e) Caso aprovado, o certificado é gerado em PDF e armazenado no Firebase Storage;
- f) Um código de validação é criado para permitir verificação futura.

A.3 EXEMPLO DE ESTRUTURA DE DOCUMENTO JSON

A seguir, apresenta-se um exemplo da estrutura interna de um documento da coleção **provas** no Firestore:

```
{
  "titulo": "Primeiros Socorros",
  "area": "Saúde",
  "tipo": "Teórica",
  "imagem": "url_da_imagem.png",
  "perguntas": [
    {
      "enunciado": "O que é hemorragia?",
      "respostas": [
        {"texto": "Sangramento excessivo", "pontos": 10, "correta": true},
```

```
        {"texto": "Dor de cabeça", "pontos": 0, "correta": false}
      ]
    }
  ],
  "percentualAprovacao": 70,
  "totalPontos": 10
}
```

A.4 ESTRUTURA DO CERTIFICADO

A estrutura de configuração do certificado no Firestore é baseada nos seguintes atributos:

- a) **posições**: coordenadas X e Y de cada elemento (nome, data, título etc.);
- b) **cores**: valores em hexadecimal para textos e títulos;
- c) **fontes**: tamanhos ajustáveis conforme o template;
- d) **templateBase**: imagem de fundo utilizada para gerar o PDF.

A.5 PROCESSO DE GERAÇÃO DO HASH SHA-256

O hash utilizado para validar cada certificado é gerado com base em:

- a) ID do usuário que realizou a prova;
- b) ID da prova realizada;
- c) Data e hora da aprovação;
- d) Título da especialidade.

Esse conjunto de dados garante que cada certificado possua um hash único e verificável.

A.6 CONSIDERAÇÕES FINAIS

Este apêndice apresenta detalhes complementares que fortalecem o entendimento técnico do sistema *EspecialidadesOn* e contribuem para uma documentação mais completa e padronizada conforme exigências acadêmicas.

APÊNDICE B – REGRAS DE SEGURANÇA DO FIRESTORE E STORAGE

Este apêndice apresenta as regras de segurança utilizadas no sistema *EspecialidadesOn*, responsáveis pelo controle de acesso às coleções do Firestore e aos arquivos armazenados no Firebase Storage. As regras têm o objetivo de garantir integridade, autenticidade e confidencialidade dos dados dos usuários.

B.1 REGRAS DO FIRESTORE

```
// =====
// Regras de Segurança Firestore
// =====

rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    // ----- USERS -----
    match /users/{userId} {
      allow read: if request.auth.uid == userId || isAdmin();
      allow create: if request.auth != null;
      allow update: if request.auth.uid == userId || isAdmin();
      allow delete: if isAdmin();
    }

    // ----- PROVAS -----
    match /provas/{provaId} {
      allow read: if request.auth != null;
      allow create, update, delete: if isAdmin();
    }

    // ----- PROVAS APROVADAS -----
    match /provasaprovadas/{docId} {
      allow read: if request.auth.uid == resource.data.userId;
      allow create: if request.auth.uid == request.resource.data.userId;
    }

    // ----- CERTIFICADOS -----
    match /certificados/{certId} {
```

```
    allow read: if request.auth.uid == resource.data.userId;
    allow create: if request.auth.uid == request.resource.data.userId;
  }

  // ----- CONFIGURAÇÃO DE CERTIFICADO -----
  match /certificadoConfig/{configId} {
    allow read: if request.auth != null;
    allow write: if isAdmin();
  }

  function isAdmin() {
    return request.auth.token.admin == true;
  }
}
}
```

B.2 REGRAS DO STORAGE

```
// =====
// Regras de Segurança Storage
// =====

rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {

    // --- Imagens de Provas ---
    match /provas/{fileName} {
      allow read: if request.auth != null;
      allow write: if request.auth.token.admin == true;
    }

    // --- Fotos de Perfil ---
    match /perfil/{userId}/{fileName} {
      allow read: if request.auth.uid == userId;
      allow write: if request.auth.uid == userId;
    }

    // --- Limitações ---
    allow write: if request.resource.size < 5 * 1024 * 1024 // Max 5MB
```

```
        && request.resource.contentType.matches('image/*');  
    }  
}
```

As regras apresentadas garantem o controle de acesso adequado e a proteção dos dados armazenados no sistema.

APÊNDICE C – ESTRUTURA DAS COLEÇÕES DO FIRESTORE

Este apêndice apresenta a estrutura completa das coleções utilizadas no banco de dados Cloud Firestore do sistema *EspecialidadesOn*. A organização segue um modelo NoSQL, baseado em documentos e coleções.

C.1 COLEÇÃO: USERS

Armazena informações dos usuários do sistema.

```
users (collection)
  userId (document)
    name: string
    email: string
    clube: string
    admin: boolean
    dataNascimento: timestamp
    fotoUrl: string
    criadoEm: timestamp
    atualizadoEm: timestamp
```

C.2 COLEÇÃO: PROVAS

Armazena as provas cadastradas pelos instrutores.

```
provas (collection)
  provaId (document)
    titulo: string
    area: string
    tipo: string // Teórica ou Completa
    imagemUrl: string
    aprovadoCom: number // Percentual (70)
    perguntas: array
      pergunta {
        texto: string
        respostas: array
          resposta {
            texto: string
            pontos: number
          }
        }
      }
    }
```

criadoEm: timestamp

C.3 COLEÇÃO: PROVASAPROVADAS

Registra cada aprovação de prova realizada pelos usuários.

```
provasaprovadas (collection)
  aprovacaoId (document)
    userId: string
    provaId: string
    tituloProva: string
    pontosObtidos: number
    totalPontos: number
    percentual: number
    area: string
    tipo: string
    dataAprovacao: timestamp
```

C.4 COLEÇÃO: CERTIFICADOS

Armazena os certificados emitidos.

```
certificados (collection)
  certId (document)
    userId: string
    provaId: string
    titulo: string
    area: string
    tipo: string
    codigoValidacao: string
    dataEmissao: timestamp
    reemitido: boolean
    criadoEm: timestamp
```

C.5 COLEÇÃO: CERTIFICADOCONFIG

Armazena configurações visuais de cada template de certificado.

```
certificadoConfig (collection)
  configId (document)
    posNome: {x, y, tamanhoFonte, cor}
```

```
posTitulo: {x, y, tamanhoFonte, cor}  
posArea: {x, y, tamanhoFonte, cor}  
posData: {x, y, tamanhoFonte, cor}  
posCodigo: {x, y, tamanhoFonte, cor}  
template: string // Caminho da imagem do template
```

Este anexo detalha toda a modelagem de dados utilizada no Firestore, fundamental para o funcionamento do sistema.