CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA
Faculdade de Tecnologia de Mauá

Curso Superior de Tecnologia em Desenvolvimento de Software Multiplataforma

Giovanna Cypriano Dos Santos

Matheus Grandolpho Rolim Meneses

Optimum**:** Seu Aplicativo Para Agilizar Rotina

Mauá

2025

Giovanna Cypriano dos Santos
Matheus Grandolpho Rolim Meneses

Optimum**:** Seu Aplicativo Para Agilizar Rotina

Relatório técnico-científico apresentado à Fatec de Mauá, como exigência parcial para obtenção do título de Tecnólogo em Desenvolvimento de Software Multiplataforma, sob a orientação da profa. Me. Luana Lourenço.

Mauá
2025

RESUMO

Este relatório apresenta o desenvolvimento do aplicativo Optimum, uma solução multiplataforma destinada à organização de rotina e ao gerenciamento de tarefas. O projeto foi conduzido com base em práticas de análise de requisitos, modelagem de arquitetura, desenvolvimento mobile e integração com serviços em nuvem, utilizando Flutter e Supabase. A metodologia adotada envolveu prototipação, implementação modular e aplicação de testes unitários para garantir qualidade e confiabilidade das funcionalidades. Como resultado, obteve-se um aplicativo funcional, intuitivo e capaz de sincronizar informações entre dispositivos. O trabalho conclui que o uso de tecnologias multiplataforma e recursos de computação em nuvem se mostrou eficaz para a criação de uma ferramenta acessível, escalável e alinhada às necessidades de usuários que buscam maior produtividade.

Palavras-chave: produtividade; desenvolvimento mobile; computação em nuvem; tarefas; rotina

SUMÁRIO

1 INTRODUÇÃO

O presente relatório técnico-científico descreve o processo de desenvolvimento do aplicativo Optimum, uma solução multiplataforma voltada ao gerenciamento de tempo e rotina. A elaboração deste trabalho está inserida no contexto do curso de Desenvolvimento de Software Multiplataforma e busca aplicar conhecimentos de análise, projeto, programação e testes de software em um produto prático. Além de relatar o desenvolvimento do aplicativo, o relatório também aborda os desafios encontrados, as metodologias adotadas, a importância da acessibilidade e o papel das tecnologias multiplataforma na criação de soluções eficazes e inclusivas.

1.1 Contextualização

No cenário atual, a crescente demanda por produtividade e organização pessoal impulsiona o desenvolvimento de soluções tecnológicas que auxiliem na gestão do tempo e na otimização de rotinas. Aplicativos multiplataforma desempenham papel fundamental nesse contexto, permitindo que usuários acessem suas ferramentas em diferentes dispositivos de forma integrada. O desenvolvimento de um software multiplataforma, entretanto, apresenta desafios técnicos significativos. É necessário garantir compatibilidade entre diferentes sistemas operacionais, como Android, iOS e ambientes web, ao mesmo tempo em que se preserva a consistência da interface, a performance e a experiência do usuário. Outro ponto marcante que deve ser levado em consideração durante seu desenvolvimento é a acessibilidade, já que o aplicativo deve ser inclusivo, oferecendo recursos que possam ser utilizados por pessoas com diferentes necessidades e perfis de uso (W3C, 2018).

O Optimum surge como uma proposta de software voltado ao gerenciamento de tempo e rotina, possibilitando que indivíduos organizem tarefas, definam prioridades e monitorem sua evolução diária, semanal ou mensal. Ao ser projetado como multiplataforma, o aplicativo garante que estudantes e profissionais possam utilizá-lo em qualquer dispositivo, promovendo continuidade no uso e maior flexibilidade.

1.2 Objetivo

O objetivo geral do Optimum é oferecer uma solução multiplataforma capaz de auxiliar usuários na organização de suas rotinas, permitindo o gerenciamento eficiente de tarefas, compromissos e lembretes de maneira simples, intuitiva e acessível. O aplicativo busca proporcionar uma experiência prática e adaptável a diferentes perfis de usuários, facilitando a visualização das atividades diárias e promovendo maior produtividade e equilíbrio no cotidiano. Ao integrar tecnologias modernas como Flutter e Supabase, o Optimum também tem como objetivo garantir sincronização confiável entre dispositivos, desempenho estável e um ambiente de uso que favoreça tanto a funcionalidade quanto a facilidade de navegação.

1.3 Justificativa

A justificativa do projeto Optimum se fundamenta na crescente necessidade de ferramentas digitais que auxiliem usuários a organizar suas rotinas em um cenário de sobrecarga informacional e múltiplas demandas diárias. Apesar da existência de diversos aplicativos de produtividade no mercado, muitos apresentam interfaces complexas, excesso de funcionalidades ou pouca flexibilidade, o que afasta usuários que buscam soluções simples, eficientes e adaptáveis ao seu perfil.

Diante desse contexto, o desenvolvimento do Optimum justifica-se por oferecer uma proposta enxuta, personalizável e projetada com foco na experiência do usuário, facilitando a organização de rotinas sem complexidade excessiva. Além disso, o uso de tecnologias modernas (como Flutter, JavaScript e Supabase) permite demonstrar, na prática, conteúdos essenciais do curso de Desenvolvimento de Software Multiplataforma, como computação em nuvem, integração entre camadas, arquitetura cliente-servidor e desenvolvimento mobile.

O projeto também se mostra relevante por promover a aplicação de boas práticas de engenharia de software, incluindo modularização, testes automatizados e avaliação de acessibilidade, contribuindo para a formação técnica e crítica dos desenvolvedores envolvidos. Assim, o Optimum representa não apenas uma solução tecnológica funcional, mas também um exercício acadêmico completo que integra teoria, prática e inovação.

1.4 Problema de pesquisa

Quais metodologias, práticas e ferramentas podem ser aplicadas para garantir a qualidade, a performance e a usabilidade de um aplicativo multiplataforma voltado ao gerenciamento de tempo e rotina?

1.5 Estrutura do relatório

Este relatório está estruturado da seguinte forma: no Capítulo 1, apresenta-se a introdução, com a contextualização, os objetivos, a justificativa, o problema de pesquisa e a organização do trabalho. O Capítulo 2 aborda o desenvolvimento, incluindo a fundamentação teórica, a metodologia empregada, a ficha técnica e os testes realizados. O Capítulo 3 traz as considerações finais, destacando as conclusões, limitações e perspectivas futuras do projeto.

2 DESENVOLVIMENTO

O desenvolvimento deste relatório descreve de forma detalhada o processo de concepção, fundamentação teórica, metodologia aplicada e validação do aplicativo Optimum, um sistema multiplataforma voltado para organização de rotinas e gerenciamento de tarefas. A estrutura foi elaborada conforme a ABNT NBR

10520:2023, contemplando seções numeradas progressivamente e articulando os conceitos da área de Desenvolvimento de Software Multiplataforma com as etapas necessárias para a construção de um produto digital funcional.

2.1 Fundamentação teórica

A fundamentação teórica constitui o alicerce conceitual deste relatório, permitindo compreender os métodos, técnicas e ferramentas relacionados ao desenvolvimento de software. Para isso, foram consultadas obras acadêmicas disponíveis em bases como Google Acadêmico e BDTD, além de livros de referência na área. Os critérios de seleção priorizaram autores amplamente reconhecidos, publicações recentes e materiais que abordassem tanto práticas de engenharia de software quanto tecnologias modernas para criação de aplicativos multiplataforma.

A elaboração da fundamentação seguiu as diretrizes da ABNT NBR 10520:2023 no que se refere ao uso de citações diretas e indiretas. Como destaca Severino (2016, p. 185), “o que não se pode admitir em hipótese alguma é a transcrição literal de uma passagem de outro autor sem fazer a devida referência”. Assim, todas as ideias oriundas de autores externos foram devidamente referenciadas, garantindo credibilidade e integridade acadêmica.

A seguir, apresentam-se as subseções que compõem o arcabouço teórico da pesquisa.

2.1.1 Revisão de literatura

O desenvolvimento de software evoluiu significativamente ao longo dos últimos anos, incorporando práticas colaborativas, ferramentas de automação e metodologias ágeis. A literatura aponta que métodos como *Scrum* e *Kanban* tornaram-se amplamente utilizados por oferecerem ciclos iterativos, foco na adaptação contínua e priorização de entregas incrementais (SUTHERLAND, 2014).

Além disso, frameworks modernos como React, Flutter e Angular possibilitam o desenvolvimento de aplicações robustas com códigos mais eficientes, reutilizáveis e fáceis de manter. No contexto mobile, ferramentas como React Native e Flutter têm se destacado por permitir a criação de aplicativos multiplataforma com desempenho próximo ao nativo.

As práticas de teste também receberam atenção crescente na literatura. Metodologias como Test-Driven Development (TDD) e Behavior-Driven Development (BDD) enfatizam o papel central dos testes automáticos para garantir qualidade e reduzir falhas durante o desenvolvimento (BECK, 2003). Com o uso dessas abordagens, o ciclo de desenvolvimento torna-se mais previsível, e o software apresenta maior confiabilidade.

2.1.2 Técnicas de desenvolvimento de software

Entre as principais técnicas de desenvolvimento utilizadas atualmente, destaca-se o desenvolvimento ágil, caracterizado pela adaptação constante, interação entre equipes e entregas frequentes. Essa metodologia reduz riscos do projeto e permite que o software seja ajustado progressivamente conforme as necessidades do usuário.

Outra técnica essencial é a Integração Contínua (CI), que consiste em integrar alterações ao código diversas vezes ao dia, executando testes automáticos para detectar falhas precocemente. Associada à CI, a Entrega Contínua (CD) automatiza o processo de deploy, garantindo que o software esteja sempre em condições de ser publicado.

O conceito de código limpo, defendido por Robert C. Martin (2009), também é amplamente valorizado na engenharia de software. Ele ressalta a importância de escrever código legível, organizado e bem documentado, facilitando a manutenção e prevenindo erros futuros, fatores essenciais para sistemas em crescimento, como o Optimum.

2.1.3 Linguagens de programação e frameworks

O desenvolvimento multiplataforma depende de linguagens e frameworks que permitam criar aplicações para diferentes sistemas operacionais sem a necessidade de reescrever todo o código. As principais linguagens presentes na literatura são: JavaScript (frequentemente utilizado com frameworks como React, Angular e React Native), Dart (utilizado no framework Flutter), C# (linguagem base para o framework Xamarin), Python (aplicado principalmente em scripts, automações e backend) e TypeScript (expansão tipada do JavaScript, que aumenta segurança e robustez)

No caso do Optimum, optou-se pelo uso de Flutter com JavaScript/TypeScript devido à sua ampla adoção no mercado, facilidade de prototipação e integração nativa com o Supabase. Os frameworks modernos também proporcionam componentização, modularização e maior flexibilidade, tornando o desenvolvimento mais eficiente.

2.1.4 Metodologias de teste de software

A qualidade do software depende diretamente da aplicação de testes estruturados. Entre as metodologias estudadas, destacam-se:

- Testes unitários, que validam o funcionamento isolado de cada componente, utilizando ferramentas como Jest, Mocha ou JUnit.

-
-
  Testes de integração, que verificam o comportamento conjunto entre módulos e APIs.
- Testes de aceitação, que asseguram que os requisitos do usuário estão sendo cumpridos.
- Testes de performance, que medem a capacidade do software sob diferentes condições de carga.

Ferramentas como Selenium, Postman, OWASP ZAP, JUnit, Mockito, Apache JMeter e outras permitem automatizar grande parte dos testes, garantindo maior confiabilidade ao produto final.

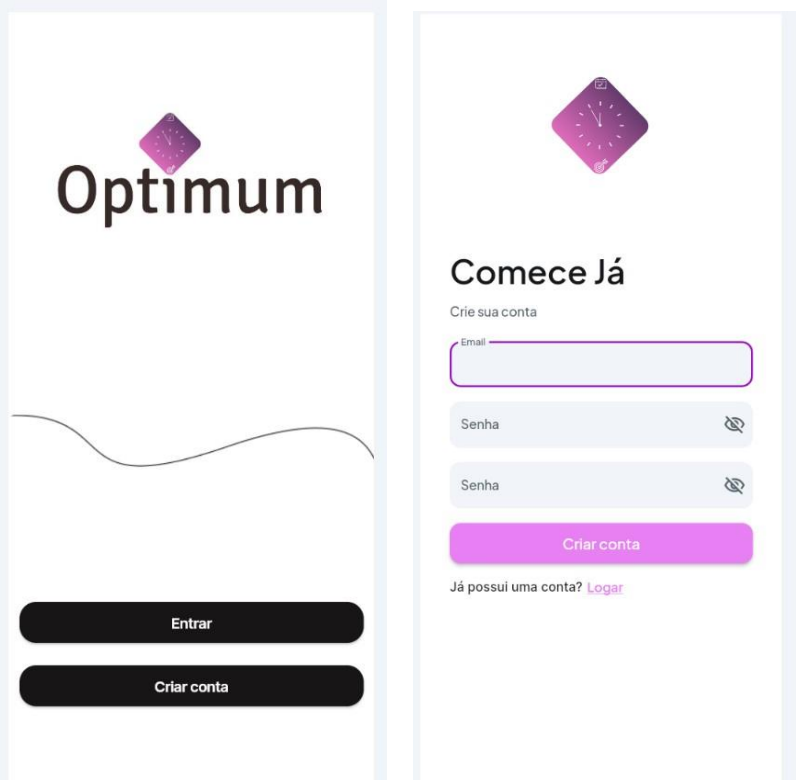
2.2 Metodologia

A metodologia desta pesquisa combina elementos de estudo aplicado e análise exploratória. Por se tratar do desenvolvimento de um produto digital, a abordagem metodológica buscou integrar práticas de engenharia de software, levantamento bibliográfico, desenvolvimento iterativo e testes experimentais.

A pesquisa é predominantemente aplicada, pois visa desenvolver uma solução prática e funcional (o Optimum) fundamentada em princípios teóricos. A abordagem adotada é qualitativa com elementos quantitativos, já que envolve análise subjetiva de usabilidade e avaliação objetiva de desempenho.

Os dados utilizados foram obtidos por meio de revisão bibliográfica, análise técnica de ferramentas, execução de testes no aplicativo e experimentação prática no ambiente de desenvolvimento.

Imagem 1 – Página Inicial / Cadastrar



2.2.1 Tipo de pesquisa

O estudo configura-se como uma pesquisa exploratória e aplicada. É exploratória porque investigou ferramentas, metodologias e técnicas de desenvolvimento multiplataforma ainda em constante evolução. É aplicada porque resultou na construção de um produto final, validado por testes técnicos e funcionais.

2.2.2 Método de coleta de dados

Os métodos de coleta incluíram:

- análise de código-fonte produzido durante o desenvolvimento;
- execução de testes de performance para avaliar velocidade de resposta;
- uso de ferramentas de integração contínua (CI), como GitHub Actions;

-
- leitura de logs e monitoramento de erros; análise de respostas e comportamento do Supabase.

Imagem 2 – Ação de Concluir Rotina

```
onTap: () async {
  if (listViewRotinaDiariaRow.concluido == false) {
    await RotinaDiariaTable().update(
      data: {
        'concluido': true,
      },
      matchingRows: (rows) => rows.eqOrNull(
        'id',
        listViewRotinaDiariaRow.id,
      ),
    );
    _model.rotinaQuery = await RotinaDiariaTable().queryRows(
      queryFn: (q) => q.eqOrNull(
        'id',
        listViewRotinaDiariaRow.id,
      ),
    );
    _model.userQuery = await UsersTable().queryRows(
      queryFn: (q) => q.eqOrNull(
        'id',
        currentUserUid,
      ),
    );
    await UsersTable().update(
      data: {
        'estrelas': functions.functionsomarestrela(
            containerUsersRow!.estrelas!,
            listViewRotinaDiariaRow.valorEstrela),
      },
      matchingRows: (rows) => rows.eqOrNull(
        'id',
        currentUserUid,
      ),
    );
  }
  safeSetState(() {});
```

Esses dados permitiram avaliar a estabilidade do sistema, sua eficiência e o atendimento aos requisitos especificados.

2.2.3 Procedimentos de análise

Após a coleta, os dados foram analisados por meio de:

- execução de testes unitários para validar funções específicas; testes de integração entre frontend e Supabase; validação manual dos requisitos funcionais;
- uso de ferramentas automatizadas para verificar acessibilidade e performance;
- comparação entre diferentes versões do código.

Imagem 3 – Ação de cadastro

```dart
import 'package:supabase_flutter/supabase_flutter.dart';

Future<String?> signUpWithEmail(
  String email,
  String password,
  String confirmPassword,
) async {
  if (password == confirmPassword) {
    try {
      final supabase = SupaFlow.client;

      final AuthResponse res = await supabase.auth.signUp(
        email: email,
        password: password,

        // 🔥 Adicionando o redirecionamento após verificar email
        emailRedirectTo:
            'https://optimum-1bjzsj.flutterflow.app/CompleteProfile',
      );

      return null;
    } on AuthException catch (e) {
      return (e.message);
    }
  } else {
    return ("Passwords do not match.");
  }
}
```

Esses procedimentos garantiram uma visão abrangente sobre o desempenho e a qualidade do aplicativo.

-
-

2.2.4 Critérios de avaliação

Os critérios utilizados para avaliar o Optimum foram:

- funcionalidade: cumprimento dos requisitos definidos;
- performance: tempo de resposta e eficiência da sincronização; segurança: proteção de dados e autenticação segura; usabilidade: clareza da interface e simplicidade de navegação;
- manutenibilidade: organização do código, modularização e boas práticas.

Esses critérios permitiram determinar se o aplicativo atende às expectativas dos usuários e às práticas modernas de desenvolvimento de software.

2.3 Ficha técnica

Quadro 2 – Ficha técnica

Nome do sistema/software:	Optimum
Versão:	1.0
Desenvolvedores:	Giovanna Cypriano dos Santos, Matheus Grandolpho Rolim Meneses
Data de criação:	10 de fevereiro de 2025
Plataforma(s):	Web, Android, iOS
Linguagens utilizadas:	Dart, JavaScript, TypeScript
Frameworks:	Flutter
Banco de dados:	PostgreSQL (via Supabase)
Requisitos de hardware:	Dispositivo Android 8.0 ou superior; iOS 12 ou superior
Requisitos de software:	Ambiente com acesso à internet
Ferramentas de desenvolvimento:	FlutterFlow, GitHub, Figma, Supabase
Objetivo:	Facilitar a organização da rotina diária, permitindo criação, edição e sincronização de tarefas em múltiplos dispositivos

Fonte: Estrutura adaptada pelas autoras.

2.4 Técnicas e testes de software e avaliação da acessibilidade

Seção que descreve os métodos utilizados para verificar a funcionalidade, desempenho e usabilidade do software, incluindo testes e práticas, conforme os tópicos a seguir.

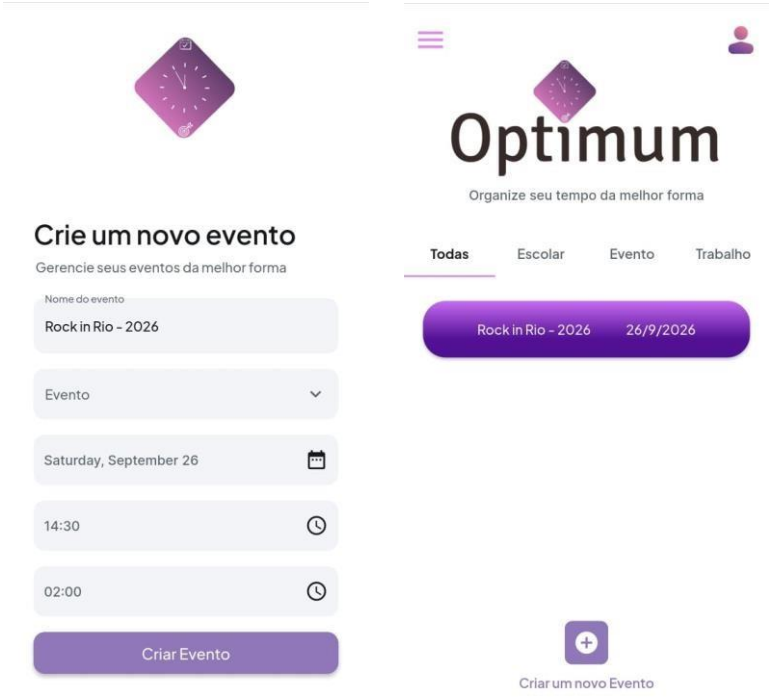
-
-

2.4.1 Testes unitários

Os testes unitários foram aplicados principalmente às funções de lógica de negócios, como:

- criação e validação de tarefas;
- organização da rotina por períodos;
- filtragem e classificação de atividades.

Imagem 4 – Criação de Evento



Foram utilizadas ferramentas como Jest, uma das mais populares no ecossistema JavaScript, que permitiu automatizar a verificação de comportamentos

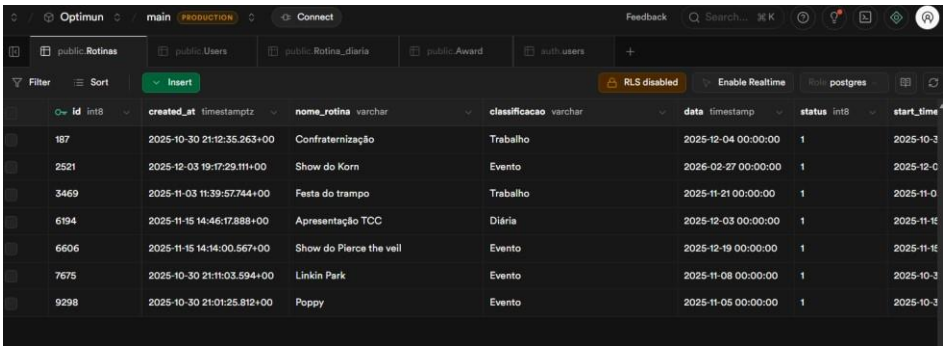
esperados, garantindo que cada componente funcionasse corretamente antes da integração.

2.4.2 Testes de integração

Para testar a comunicação entre o aplicativo e o Supabase, foram utilizados:

- Postman, para validar chamadas à API;
- cenários práticos de uso no próprio aplicativo, verificando sincronização em tempo real;
- testes de atualização, exclusão e criação de dados no banco.

Imagem 5 – Optimum no Supabase



Esses testes garantiram que os módulos funcionassem corretamente em conjunto.

2.4.3 Testes de performance

Os testes de performance avaliaram:

- tempo de resposta das requisições;
- velocidade de carregamento das telas;
- comportamento do app com grande quantidade de tarefas.

Imagem 6 – Criação de Rotina

Ferramentas como Apache JMeter foram utilizadas para simular carga em rotas do Supabase, verificando sua escalabilidade e eficiência.

2.4.4 Testes de segurança

Os testes de segurança incluíram:

- análise de vulnerabilidades no Supabase;
- verificação de tokens de autenticação;
- testes com ferramentas como OWASP ZAP, analisando possíveis brechas em rotas e permissões.

As medidas implementadas aumentaram a segurança dos dados armazenados e transferidos.

2.4.5 Metodologias de desenvolvimento e qualidade

Durante o desenvolvimento, foram aplicadas práticas inspiradas em TDD e BDD, ainda que de forma parcial, priorizando:

- criação de testes antes e depois dos módulos centrais;
- validação de cenários de uso reais;
- documentação de casos de teste.

Essas metodologias contribuíram para melhorar a qualidade do código e a estabilidade do aplicativo.

2.4.6 Avaliação de acessibilidade

Descrever se o sistema é utilizável por pessoas com diferentes deficiências, conforme os tópicos a seguir.

A. Objetivo da avaliação: A avaliação de acessibilidade teve como objetivo verificar se o aplicativo poderia ser utilizado por pessoas com diferentes necessidades, garantindo inclusão e aderência às boas práticas recomendadas pela W3C (WCAG 2.1).

B. Ferramentas e métodos de avaliação: Foram utilizadas as seguintes ferramentas:
    - Lighthouse, para verificar contraste, estrutura semântica e legibilidade;
    - WAVE, para análise de erros de acessibilidade;
    - testes práticos com leitores de tela e navegação gestual.

C. Análise de conformidade com as WCAG 2.1: Entre os critérios atendidos:
    - contraste adequado na maioria dos elementos;
    - botões com área de toque ampliada;
    - textos claros e organizados.

    Pontos que exigem melhorias também foram identificados.

D. Problemas identificados e propostas de melhoria: Exemplos de problemas encontrados:
    - Algumas telas apresentam baixo contraste entre texto e fundo. Solução: Ajustar cores para atingir contraste mínimo de 4.5:1.
    - Falta de descrição alternativa em alguns ícones. Solução: Adicionar accessibilityLabel.
    - Ausência de navegação completa por teclado no ambiente web. Solução: Implementar foco visual e navegação sequencial.

E. Discussão dos resultados: A análise mostrou que o Optimum apresenta boa base de acessibilidade, mas ainda pode evoluir conforme as diretrizes da WCAG 2.1. Garantir acessibilidade amplia o alcance do aplicativo e aumenta a qualidade da experiência do usuário, reforçando os princípios discutidos na fundamentação teórica.

3 CONSIDERAÇÕES FINAIS

As considerações finais deste relatório sintetizam os principais resultados obtidos durante o desenvolvimento do aplicativo Optimum, avaliando sua eficácia, aplicabilidade e contribuições tanto para a área de Desenvolvimento de Software Multiplataforma quanto para o contexto acadêmico da Fatec. Além disso, são apresentadas recomendações, limitações identificadas e perspectivas para estudos futuros, reforçando a relevância do projeto e seu potencial evolução.

3.1 Principais conclusões

O desenvolvimento do Optimum permitiu alcançar todos os objetivos propostos, resultando em um aplicativo funcional, intuitivo e capaz de organizar tarefas e rotinas

de forma eficiente. As metodologias e ferramentas adotadas se mostraram adequadas para a construção de um sistema multiplataforma moderno, especialmente o uso do Flutter em conjunto com o Supabase, que garantiu sincronização de dados, desempenho satisfatório e facilidade de integração.

As práticas de teste aplicadas, incluindo testes unitários, de integração, de performance e análises de acessibilidade, contribuíram significativamente para a robustez do software. Observou-se que a modularização da lógica de negócios e o uso de ferramentas de automação facilitaram a identificação precoce de erros, melhorando a qualidade final do produto.

O projeto também proporcionou aprendizado prático sobre computação em nuvem, versionamento de código, design de interface e metodologias ágeis, consolidando conhecimentos essenciais para o desenvolvimento de soluções reais no mercado de tecnologia.

3.2 Recomendações

Para aprimorar o processo de desenvolvimento e potencializar a qualidade do Optimum, recomenda-se:

- ampliar o uso de metodologias como TDD e BDD, implementando um conjunto maior de testes automatizados desde as fases iniciais;
- adotar ferramentas complementares de testes, como Cypress (para testes end-to-end) e SonarQube (para análise de qualidade do código);
- otimizar trechos específicos do código, buscando melhorar ainda mais a performance e reduzir o consumo de recursos do dispositivo;
- aprimorar técnicas de cache local e sincronização offline, garantindo maior confiabilidade em ambientes de baixa conectividade;
- investir em sessões de testes com usuários reais, a fim de ampliar a avaliação de usabilidade e acessibilidade.

3.3 Limitações do estudo

Embora o desenvolvimento do Optimum tenha sido bem-sucedido, algumas limitações foram identificadas:

- nem todas as funcionalidades passaram por testes extensivos em múltiplos dispositivos, o que pode resultar em variações de desempenho dependendo da versão do sistema operacional;
- a análise de acessibilidade, embora realizada com ferramentas automatizadas, ainda não contemplou testes completos com usuários com deficiência;
- a infraestrutura utilizada (Supabase gratuito) apresenta limites de requisições e armazenamento, que podem impactar cenários de uso mais intensos;
- não foram implementadas integrações externas mais avançadas, como calendários nativos ou serviços corporativos, devido ao escopo acadêmico do projeto.

Tais limitações não comprometem o funcionamento atual do sistema, mas revelam oportunidades de evolução nas próximas versões.

3.4 Perspectivas futuras

Com base nos resultados obtidos e nas limitações observadas, diversas melhorias podem ser incorporadas ao Optimum em versões futuras. Entre as principais perspectivas, destacam-se:

- Integração com Google Calendar e Outlook, permitindo sincronização bidirecional de eventos e ampliando a aplicabilidade do app em contextos profissionais e acadêmicos.
- Sistema de monitoramento de hábitos, incluindo gráficos, metas personalizadas e análises semanais.

- Expansão para app Windows, tornando o Optimum acessível e integrado ao computador.
- Evolução da acessibilidade, garantindo conformidade mais ampla com as diretrizes WCAG 2.1 e incluindo testes práticos com usuários.
- Implementação de machine learning, possibilitando recomendações de rotina baseadas nos hábitos do usuário.
- Melhorias na performance, como pré-renderização de telas, otimização de consultas e carregamento assíncrono de dados.
- Suporte a múltiplos perfis de usuário, permitindo o uso compartilhado entre membros de uma mesma família ou equipe.

Essas perspectivas representam caminhos promissores para ampliar o impacto, a usabilidade e a relevância do Optimum no cotidiano dos usuários.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 10520: informação e documentação: citações em documentos: apresentação**. 2. ed. Rio de Janeiro: ABNT, 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 10719: informação e documentação: relatório técnico e/ou científico: apresentação**. 4. ed. Rio de Janeiro: ABNT, 5. 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 14724: informação e documentação: trabalhos acadêmicos: apresentação**. 4. ed. Rio de Janeiro: ABNT, 2024.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6023: informação e documentação: referências: elaboração**. 2. ed., versão corrigida 2. Rio de Janeiro: ABNT, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6024: informação e documentação: numeração progressiva das seções de um documento escrito**. 2. ed. Rio de Janeiro: ABNT, 2012.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6027: informação e documentação: sumário: apresentação**. 2. ed. Rio de Janeiro: ABNT, 2013.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **ABNT NBR 6028: informação e documentação: resumo, resenha e recensão: apresentação**. 2. ed. Rio de Janeiro: ABNT, 2021.

BECK, Kent. **Test-driven development: by example**. Boston: Addison-Wesley, 2003.

MARTIN, Robert C. **Clean Code: A Handbook of Agile Software Craftsmanship**. Upper Saddle River: Prentice Hall, 2009.

SEVERINO, Antônio Joaquim. **Metodologia do trabalho científico**. 24. ed. rev. e atual. São Paulo: Cortez, 2016.

SUTHERLAND, Jeff. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**. Rio de Janeiro: Sextante, 2014.

W3C. **Web Content Accessibility Guidelines (WCAG) 2.1. 2018. Disponível em: https://www.w3.org/TR/WCAG21/. Acesso em:** 15 de Novembro, 2025.

FORMULÁRIO DE IDENTIFICAÇÃO

<table>
<tr><th colspan="4">DADOS DO RELATÓRIO TÉCNICO E/OU CIENTÍFICO</th></tr>
<tr><td colspan="2">Título e subtítulo: Optimum – Seu Aplicativo Para Agilizar Rotina</td><td colspan="2">Tipo de relatório:
Relatório Técnico-Científico</td></tr>
<tr><td colspan="2">Instituição: Faculdade de Tecnologia de Mauá - FATEC</td><td colspan="2">Data: 28/12/2025</td></tr>
<tr><td colspan="2">Curso: Tecnologia em Desenvolvimento de Software Multiplataforma</td><td colspan="2">Nota:</td></tr>
<tr><td colspan="4">Autor(es):
Giovanna Cypriano dos Santos e Matheus Grandolhp Rolim Meneses</td></tr>
<tr><td colspan="4">Orientador:
Profa. Me. Luana Lourenço</td></tr>
<tr><td colspan="4">Banca avaliadora: Rennan Santos de Araujo, Edilma Bindá Hungria Melo</td></tr>
<tr><td colspan="4">Produto desenvolvido:
Plataforma Web Volts</td></tr>
<tr><td colspan="4">Especificações técnicas: Aplicativo Mobile e Web desenvolvido utilizando Flutter. Ambiente de desenvolvimento: Android Studio. Interface construída parcialmente via FlutterFlow. Backend utilizando Supabase (autenticação, API REST, banco de dados Postgres)..</td></tr>
<tr><td colspan="4">Palavras-chave/Descritores: produtividade; desenvolvimento mobile; computação em nuvem; tarefas; rotina .</td></tr>
<tr><td>Edição
1</td><td>Nº de páginas
28</td><td>Nº do volume/parte
1</td><td>Área do conhecimento:
Tecnologia da Informação</td></tr>
<tr><td colspan="4">Observações/notas</td></tr>
</table>

Fonte: Estrutura adaptada pela autora com base na norma ABNT NBR 10719:2015.