

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE SOFTWARE**

JOÃO PEDRO ODORIZZIO DE SOUZA

PRINCIPAIS ATIVIDADES DE UM DESENVOLVEDOR MOBILE

**CORNÉLIO PROCÓPIO
2025**

JOÃO PEDRO ODORIZZIO DE SOUZA

PRINCIPAIS ATIVIDADES DE UM DESENVOLVEDOR MOBILE

MAIN ACTIVITIES OF A MOBILE DEVELOPER

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Software, da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Gabriel Canhadas Genvigir

CORNÉLIO PROCÓPIO

2025



Esta licença permite download e compartilhamento do trabalho desde que sejam atribuídos créditos ao(s) autor(es), sem a possibilidade de alterá-lo ou utilizá-lo para fins comerciais. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

JOÃO PEDRO ODORIZZIO DE SOUZA

PRINCIPAIS ATIVIDADES DO DESENVOLVEDOR MOBILE

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Software, da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 03/dezembro/2025

Gabriel Canhadas Genvigir
Especialização
Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio

André Luiz Przybysz
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio

Paulo Augusto Nardi
Doutorado
Universidade Tecnológica Federal do Paraná - Campus Cornélio Procópio

CORNÉLIO PROCÓPIO

2025

RESUMO

O presente trabalho tem como objetivo apresentar as principais atividades realizadas por um desenvolvedor *mobile* no contexto do *framework Scrum*, com foco no desenvolvimento de aplicações *Flutter* na empresa Forlogic Software. O documento descreve práticas, ferramentas e processos empregados no ciclo de desenvolvimento, abrangendo desde a implementação e correção de funcionalidades até a participação ativa nas cerimônias ágeis e no processo de entrega contínua. São detalhadas as contribuições técnicas e metodológicas do autor no período de maio a novembro de 2025, evidenciando a importância da atuação colaborativa dentro de um time ágil para o sucesso do desenvolvimento de *software*.

Palavras-chave: desenvolvimento *mobile*; *flutter*; *scrum*; Forlogic Software.

ABSTRACT

This work aims to present the main activities performed by a mobile developer within the Scrum framework, focusing on Flutter application development at Forlogic Software. The document describes the practices, tools, and processes used in the software development cycle, from feature implementation and bug fixing to active participation in agile ceremonies and continuous product improvement. It details the technical and methodological contributions of the author during the period from May to November 2025, highlighting the importance of collaborative work within an agile team for software development success.

Keywords: mobile development; flutter; scrum; Forlogic Software.

LISTA DE ILUSTRAÇÕES

Figura 1 - Time Forlogic.....	14
Figura 2 - Ciclo de desenvolvimento <i>scrum</i>	15
Figura 3 - Camadas da arquitetura limpa	18
Figura 4 - Estrutura em camadas do <i>pipeline</i> de processamento	19
Gráfico 1 - Gráfico de Burndown	19
Figura 5 - <i>Planning Poker</i>	20
Figura 6 - <i>Bugs</i> corrigidos.....	21
Figura 7 - Cards de melhorias	22

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Forlogic Software	13
1.2	O papel do desenvolvedor <i>mobile</i>	14
2	REFERENCIAL TEÓRICO.....	15
2.1	Gerenciamento de projetos e métodos ágeis	15
2.2	Desenvolvimento <i>mobile</i> e multiplataforma	16
2.3	Boas práticas de engenharia aplicadas	16
3	MATERIAS E MÉTODOS	17
3.1	Ferramentas e ambientes	17
3.2	Arquitetura e padrões	17
3.3	Processos e <i>pipeline</i>	18
3.4	Métricas da documentação	19
4	ATIVIDADES DESENVOLVIDAS	20
4.1	Planejamento e organização das <i>sprints</i>	20
4.2	Desenvolvimento de funcionalidades	20
4.3	Correções, refatorações e suporte	20
5	RESULTADOS E DISCUSSÃO	22
5.1	Melhorias técnicas	22
5.2	Impacto no processo ágil	22
6	CONCLUSÃO	24
7	REFERÊNCIAS.....	25

1 INTRODUÇÃO

O gerenciamento de projetos de *software* tem como finalidade assegurar o cumprimento do escopo, prazos, custos e a entrega de produtos com a qualidade esperada. Em ambientes de rápida evolução tecnológica, metodologias ágeis surgem como alternativas para aumentar a adaptabilidade dos times e reduzir o tempo entre a concepção e a entrega de valor.

Este trabalho descreve as atividades desenvolvidas por João Pedro Odorizzio de Souza no período de maio a novembro de 2025, atuando como Desenvolvedor *Mobile* na Forlogic Software. O autor participou de atividades técnicas (desenvolvimento em *Flutter*, integrações e testes), bem como de práticas colaborativas do *Scrum* (*planning*, *daily*, *review* e *retrospective*), contribuindo para a evolução do produto e da maturidade do time.

1.1 Forlogic Software

A Forlogic Software é uma empresa brasileira fundada em 2004, no município de Cornélio Procopio, Paraná, inicialmente formada por um grupo de ex-alunos da UTFPR que buscavam transformar projetos acadêmicos em soluções tecnológicas aplicáveis ao mercado. Desde sua origem, a empresa direcionou seus esforços para o desenvolvimento de *software* voltado à gestão da qualidade, apoiando organizações na padronização de processos, no controle documental e na melhoria contínua. Ao longo dos anos, a *Forlogic* expandiu de uma pequena equipe de desenvolvimento para uma estrutura mais robusta, consolidando-se como fornecedora de produtos e serviços especializados na área da qualidade e metrologia.

Com o crescimento de sua base de clientes, a empresa evoluiu de projetos sob demanda para a criação de plataformas próprias, como o Qualiex, que se tornou uma de suas soluções mais reconhecidas no mercado. Essa transformação exigiu a adoção de práticas modernas de engenharia de *software*, incluindo arquitetura modular, desenvolvimento orientado a serviços e automação de processos de entrega. A maturidade técnica alcançada permitiu à *Forlogic* atender empresas de diferentes portes e segmentos, assim como expandir sua atuação para outros países da América Latina.

Atualmente, os produtos da *Forlogic* são desenvolvidos por *squads* multidisciplinares compostos por *Product Owners*, *Tech Leads*, desenvolvedores,

designers e profissionais de QA. Essa estrutura favorece a colaboração contínua, a experimentação e a entrega incremental de funcionalidades. A governança dos projetos é sustentada por práticas de integração e entrega contínua (CI/CD), testes automatizados, versionamento estruturado e monitoramento ativo dos sistemas. Ferramentas como o *Azure DevOps* apoiam todo o ciclo de vida do desenvolvimento, desde o planejamento das demandas até a implantação, garantindo maior qualidade, rastreabilidade e previsibilidade nas entregas.

Com essa trajetória, a Forlogic Software consolidou-se como uma empresa referência em soluções de gestão da qualidade, mantendo como diferencial a combinação entre inovação tecnológica, forte cultura organizacional e compromisso com a melhoria contínua dos processos de seus clientes.

Figura 1 - Time Forlogic



1.2 O papel do desenvolvedor *mobile*

O desenvolvedor *mobile* tem responsabilidade por projetar, implementar e manter funcionalidades do aplicativo, garantindo boa experiência ao usuário e qualidade técnica. Isso inclui: definição de arquitetura, implementação de interfaces, integração com serviços *backend*, tratamento de exceções, testes automatizados e participação nas cerimônias ágeis. Ao longo do período, utilizou-se *Flutter/Dart*, aplicando padrões como injeção de dependência e arquitetura em camadas.

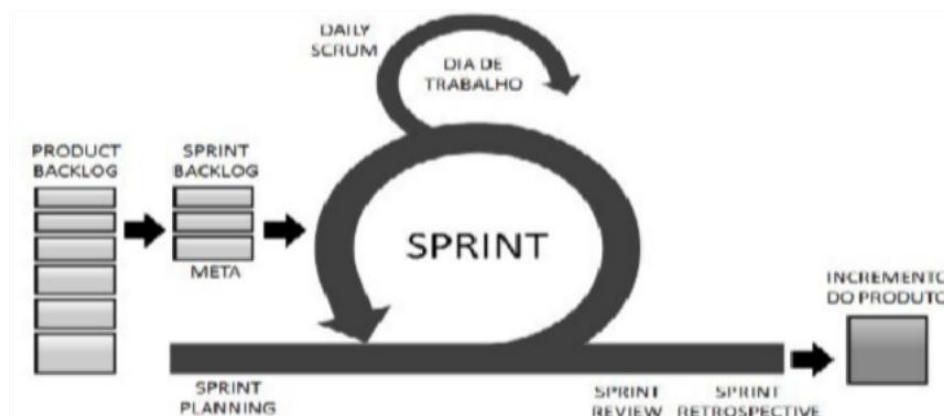
2 REFERENCIAL TEÓRICO

A construção deste projeto apoia-se em um conjunto de conceitos, metodologias e tecnologias que fornecem a base necessária para compreender as estratégias adotadas durante o desenvolvimento, alinhando-se aos fundamentos da engenharia de *software* (PRESSMAN, 2011; SOMMERVILLE, 2016). Os tópicos apresentados nesta seção oferecem suporte teórico às escolhas técnicas e organizacionais realizadas, destacando a relevância dos métodos ágeis (BECK et al., 2001; SCHWABER; SUTHERLAND, 2020), das abordagens de desenvolvimento *mobile* multiplataforma (GOOGLE, 2025) e das boas práticas de engenharia e interface (BARBOSA; SILVA, 2010) no contexto atual da indústria.

2.1 Gerenciamento de projetos e métodos ágeis

O gerenciamento de projetos de *software* envolve planejamento, execução e controle das atividades necessárias para alcançar os objetivos do projeto (PRESSMAN, 2011). Métodos ágeis, representados pelo Manifesto Ágil, priorizam a entrega incremental, colaboração entre equipes e resposta a mudanças (BECK et al., 2001). O *Scrum* é um dos *frameworks* ágeis mais difundidos, definindo papéis (*Product Owner, Scrum Master, Development Team*), eventos (*sprint, planning, daily, review, retrospective*) e artefatos (*product backlog, sprint backlog, increment*) conforme estabelecido no Guia do *Scrum* (SCHWABER; SUTHERLAND, 2020).

Figura 2 - Ciclo de desenvolvimento *scrum*



Fonte: Adaptado de Schwaber e Sutherland (2020).

2.2 Desenvolvimento *mobile* e multiplataforma

O desenvolvimento *mobile* enfrenta desafios como fragmentação de plataformas, restrições de *hardware* e necessidade de interfaces intuitivas. *Frameworks* multiplataforma, como *Flutter*, permitem reduzir a duplicação de esforço ao reutilizar uma base de código. O *Flutter* fornece um conjunto rico de *widgets*, renderização própria e integração com APIs nativas quando necessário, tornando-o adequado para aplicações corporativas que demandam performance e manutenção simplificada.

2.3 Boas práticas de engenharia aplicadas

Boas práticas de engenharia incluem a modularização do código, injeção de dependência para facilitar testes, testes automatizados (unitários e de integração), *CI/CD* para validação constante e monitoramento de qualidade. Em aplicações móveis, o uso de banco local (*SQLite*) para cenários *offline*, juntamente com mecanismos de sincronização robustos, é essencial para garantir a confiabilidade do produto.

3 MATERIAS E MÉTODOS

A condução do projeto apoiou-se em um conjunto de ferramentas, padrões de arquitetura e práticas de desenvolvimento que permitiram organizar as etapas de construção do *software* e garantir qualidade ao longo de todo o processo. Os materiais utilizados e os métodos adotados foram selecionados de acordo com as necessidades do projeto, visando eficiência, escalabilidade e integração contínua entre as atividades da equipe.

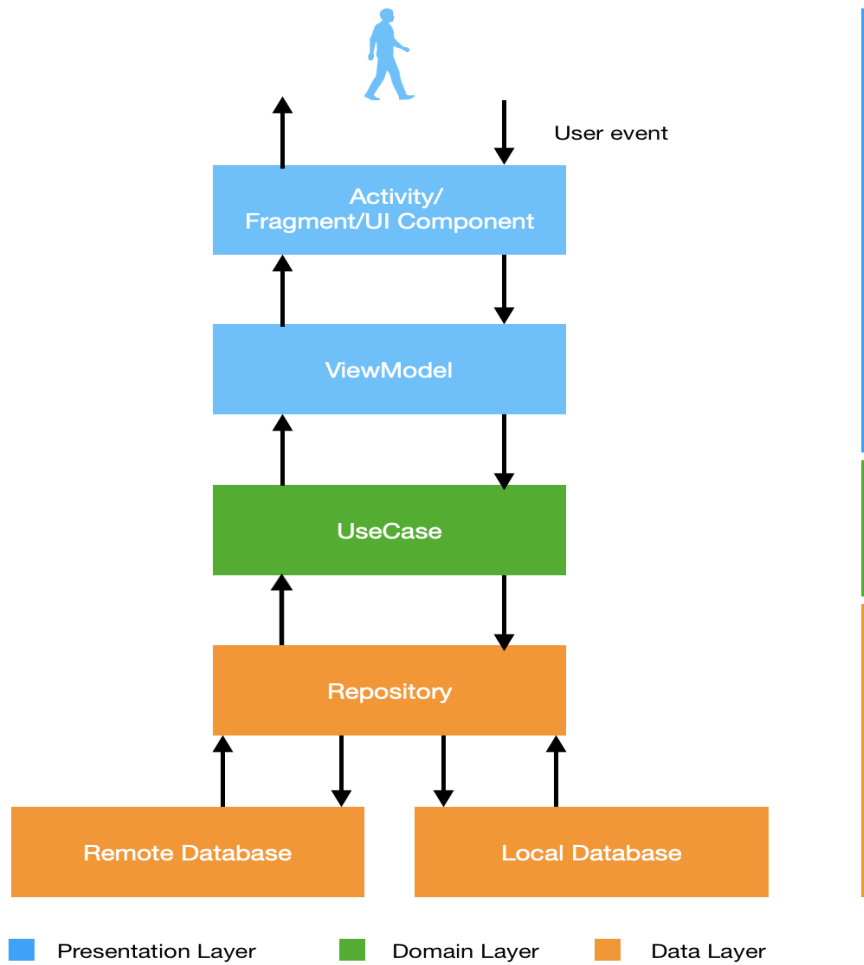
3.1 Ferramentas e ambientes

O ambiente de desenvolvimento compreende: *Flutter (Dart)* para desenvolvimento *cross-platform*; *Visual Studio Code* como *IDE* principal; *Android Studio/Xcode* para depuração e *builds* nativos; *Git* para controle de versão com repositórios hospedados no *Azure DevOps*; *SQLite* para armazenamento local; *Postman* para testes de *API*; e *Azure DevOps* para gerenciamento de *boards*, *pipelines CI/CD* e revisões de código.

3.2 Arquitetura e padrões

A arquitetura adotada segue separação em camadas (*presentation/domain/data*), com uso do padrão *Repository* para abstração de acesso a dados. A injeção de dependência foi implementada via *GetIt*, favorecendo testes e desacoplamento. O gerenciamento de estado foi desenvolvido com *ValueListenableBuilder* em cenários onde a reatividade simples era suficiente, reservando abordagens mais complexas para fluxos com maior necessidade.

Figura 3 - Camadas da arquitetura limpa

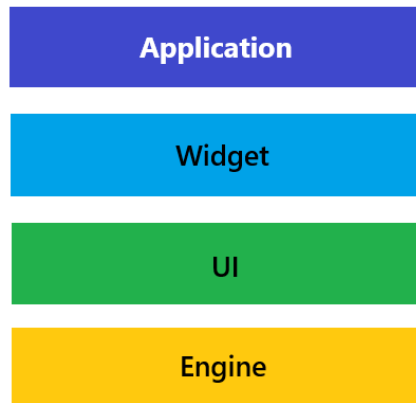


Fonte: Objective (2020).

3.3 Processos e *pipeline*

O processo de desenvolvimento segue *Scrum* com *sprints* quinzenais. O *Azure DevOps* é utilizado como plataforma única para gerenciamento de *backlog*, *boards* e execução de *pipelines* de *CI/CD*. Cada *merge* passa por *build* automático e bateria de testes básicos para garantir estabilidade.

Figura 4 - Estrutura em camadas do *pipeline* de processamento



Fonte: Autoria própria

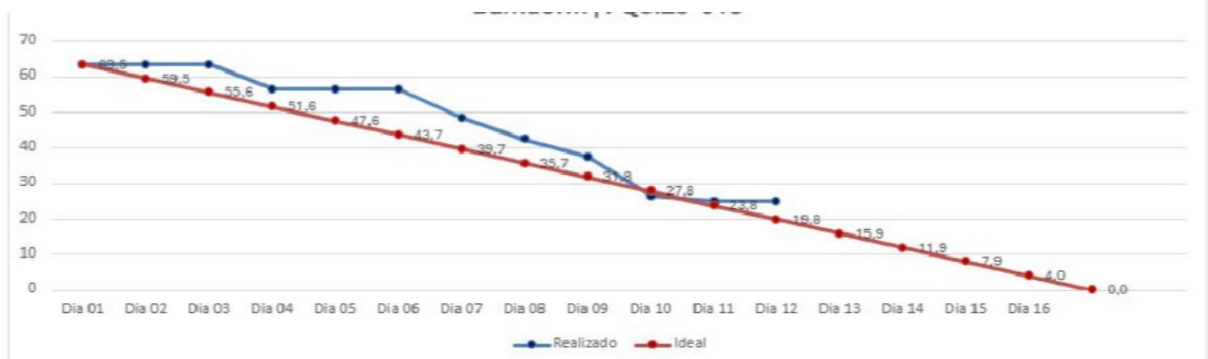
3.4 Métricas da documentação

Para avaliar a saúde do projeto e garantir o controle das atividades, foram utilizadas métricas de acompanhamento visual, fundamentais para o monitoramento do progresso em engenharia de *software* (PRESSMAN, 2011). Entre elas, destaca-se o uso do *burndown chart* para comparar o esforço planejado versus o realizado.

O **Gráfico 1** apresenta o *burndown* da *Sprint* PQ3.25-010. A linha vermelha representa a queima ideal de tarefas, enquanto a linha azul indica o trabalho efetivamente realizado pela equipe. Observa-se que, nos primeiros três dias (Dia 01 a 03), o progresso manteve-se estável, indicando uma fase inicial de planejamento ou impedimentos. Contudo, entre o **Dia 06 e o Dia 09**, houve uma redução acentuada no esforço restante, demonstrando uma recuperação no ritmo de trabalho que permitiu à equipe alinhar-se novamente à meta ideal planejada.

Além do gráfico, a documentação dos eventos do *Scrum*, como o controle de horas e planilhas de pendências, foi mantida no Google *Workspace* para histórico e compartilhamento com os *stakeholders*.

Gráfico 1 - Gráfico de Burndown



Fonte: Azure DevOps.

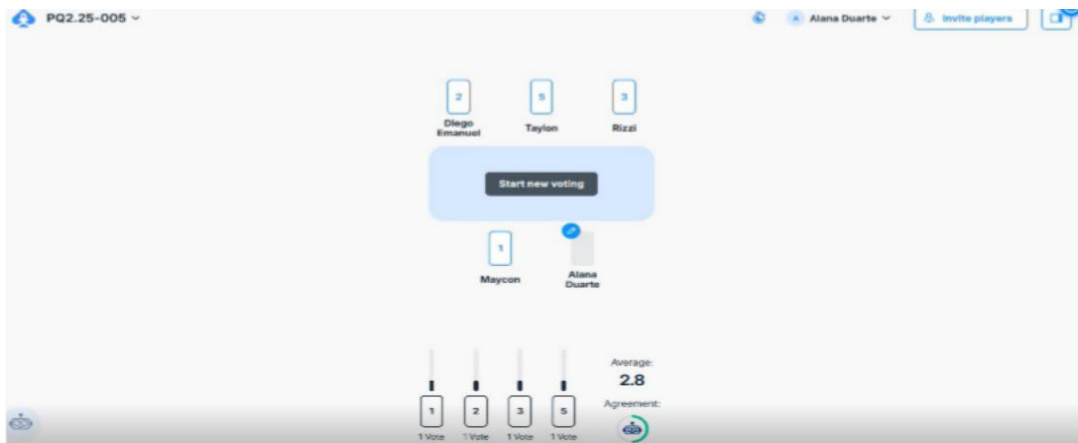
4 ATIVIDADES DESENVOLVIDAS

Este capítulo detalha as atividades realizadas na *Forlogic Software*, organizadas em planejamento, desenvolvimento, correções e integração. As descrições a seguir sintetizam as entregas mais relevantes e os procedimentos adotados no período de maio a novembro de 2025.

4.1 Planejamento e organização das *sprints*

O planejamento das *sprints* foi realizado em sessões de *Sprint Planning*, onde itens do *backlog* eram priorizados pelo *Product Owner* e estimados pelo time. Houve uma participação ativa do refinamento das histórias, definição de critérios de aceite e estimativas por *Planning Poker*. As *dailies* serviram para sincronização rápida e remoção de impedimentos, e as *retrospectives* promoveram melhoria contínua.

Figura 5 - *Planning Poker*



Fonte: Planning Poker

4.2 Desenvolvimento de funcionalidades

As funcionalidades desenvolvidas abrangeram múltiplos módulos do produto *mobile*, incluindo Auditorias, Ocorrências, Inspeções, Documentos e Metrologia. O trabalho envolveu: criação de componentes de *UI* reutilizáveis, implementação de regras de negócio no domínio, sincronização de dados (*offline-first*), e integração com *APIs RESTful* e serviços nativos quando necessário.

4.3 Correções, refatorações e suporte

Houve atuação contínua na resolução de *bugs* reportados por *QA* e usuários, análise de causas raiz, e aplicação de refatorações para melhoria da

manutenibilidade. Foram priorizadas correções de sincronização, ajustes em rotas de navegação e prevenção de vazamentos de memória.

Figura 6 - Bugs corrigidos

ID	Work Item Ty...	Title	Assigned To	State	Tags
69475	Bug	Auditorias (aplicativo) Quando Auditor...	Joao Pedro Odorizzo ...	Done	APK
69774	Bug	Auditorias (aplicativo) Específico Aplic...	Joao Pedro Odorizzo ...	Done	
84569	Bug	Qualiex (aplicativo) Ocorrências Ajust...	Joao Pedro Odorizzo ...	Done	
87991	Bug	Qualiex (aplicativo) Documentos Ao c...	Joao Pedro Odorizzo ...	Done	
88152	Bug	Auditorias (App) Geral Ao sincronizar ...	Joao Pedro Odorizzo ...	Done	
88339	Bug	Qualiex (aplicativo) Na tela de Docume...	Joao Pedro Odorizzo ...	Done	
90130	Bug	Auditorias (Aplicativo) Filtros do planej...	Joao Pedro Odorizzo ...	Done	
90300	Bug	Qualiex(Aplicativos) Ocorrências Nom...	Joao Pedro Odorizzo ...	Done	
93310	Bug	Qualiex (Aplicativo) Ocorrências [Cad...	Joao Pedro Odorizzo ...	Done	
96385	Bug	Qualiex (Aplicativo) - Cores no módulos...	Joao Pedro Odorizzo ...	Done	
98745	Bug	Qualiex/Metroex (aplicativo) Metrologi...	Joao Pedro Odorizzo ...	Done	
100199	Bug	Qualiex (aplicativo) Ocorrências Inputs ...	Joao Pedro Odorizzo ...	Done	
100295	Bug	Qualiex (aplicativo) Ocorrências Cam...	Joao Pedro Odorizzo ...	Done	
100296	Bug	Qualiex (aplicativo) Campos dropdown...	Joao Pedro Odorizzo ...	Done	
101111	Bug	Auditorias (Aplicativo) A listagem de a...	Joao Pedro Odorizzo ...	Done	
101283	Bug	Qualiex/Metroex Minhas Ações - Label...	Joao Pedro Odorizzo ...	Done	
102482	Bug	Qualiex/Metroex (aplicativo) Ao editar u...	Joao Pedro Odorizzo ...	Done	
102557	Bug	Auditorias (aplicativo) Visualização de o...	Joao Pedro Odorizzo ...	Done	
103410	Bug	Auditorias (aplicativo) Na listagem de ...	Joao Pedro Odorizzo ...	Done	

Fonte: Azure DevOps

5 RESULTADOS E DISCUSSÃO

As ações descritas geraram resultados mensuráveis na qualidade e previsibilidade das entregas. A utilização de métricas de *sprint*, com acompanhamento de *burndown* e *velocity*, permitiu ajustar estimativas e planejar melhor as capacidades do time.

5.1 Melhorias técnicas

As refatorações e implementação de padrões (*Repository*, injeção de dependência) melhoraram a cobertura de testes e reduziram a incidência de regressões. Ajustes de sincronização *offline* aumentaram a confiabilidade do *app* em condições de conectividade instável.

Figura 7 – Cards de melhorias

ID	Work Item Ty...	Title	Assigned To	State	Tags
82417	Management	* Qualiex (Aplicativo) Atas e Decisões P...	Joao Pedro Odorizzo ...	Done	
87770	Management	* Qualiex (Aplicativo) Documentos Pes...	Joao Pedro Odorizzo ...	Done	
87916	Management	* Qualiex (Aplicativo) Documentos Pes...	Joao Pedro Odorizzo ...	Done	
88082	Management	* Qualiex (Aplicativo) Ocorrências Melh...	Joao Pedro Odorizzo ...	Done	
88230	Management	* Qualiex(aplicativo) Substituir no aplicat...	Joao Pedro Odorizzo ...	Done	
90521	Management	* Auditorias (aplicativo) Adicionar Fireba...	Joao Pedro Odorizzo ...	Done	
92213	Management	* Qualiex (aplicativo) Inserir eventos Mixp...	Joao Pedro Odorizzo ...	Done	
92214	Management	* Qualiex (aplicativo) Inserir eventos Mixp...	Joao Pedro Odorizzo ...	Done	
92215	Management	* Qualiex (aplicativo) Inserir eventos Mixp...	Joao Pedro Odorizzo ...	Done	
96306	Management	* Inspeções Testes unitários e widgets	Joao Pedro Odorizzo ...	Done	Inspeções
96471	Management	* App Coletor Xamarin Atualizar o app p...	Joao Pedro Odorizzo ...	Done	
97388	Management	* Inspeções Criar o arquivo app_theme.d...	Joao Pedro Odorizzo ...	Done	Inspeções
97524	Management	* Inspeções Implementar o salvamento lo...	Joao Pedro Odorizzo ...	Done	Inspeções
99079	Management	* Atualização de Nível da API para Googl...	Joao Pedro Odorizzo ...	Done	
99961	Management	* Gerar IPA do aplicativo VerifiQ no Test F...	Joao Pedro Odorizzo ...	Done	
100467	Management	* Qualiex (aplicativo) Liberar Metrologia e...	Joao Pedro Odorizzo ...	Done	
102480	Management	* Qualiex (aplicativo) Liberar nova versão ...	Joao Pedro Odorizzo ...	Done	
103327	Management	* App Docs - Xamarin Atualizar o app p...	Joao Pedro Odorizzo ...	Done	
103328	Management	* App Indicators - Xamarin Atualizar o a...	Joao Pedro Odorizzo ...	Done	

Fonte: Azure DevOps

5.2 Impacto no processo ágil

A consolidação das cerimônias do *Scrum* e a disciplina na atualização dos quadros de tarefas no *Azure DevOps* resultaram em um aumento significativo na transparência e na previsibilidade do projeto. A visibilidade do progresso permitiu que

a equipe realizasse inspeções frequentes e adaptações rápidas, alinhando-se aos pilares empíricos de transparência, inspeção e adaptação defendidos por **Schwaber e Sutherland (2020)**.

Além disso, o processo de revisão de código (*code review*) não apenas difundiu o conhecimento técnico entre os membros do time, mas também atuou como uma estratégia eficaz de garantia da qualidade de *software* (**PRESSMAN, 2011**). Essa prática elevou o padrão de entrega, reduzindo a incidência de *bugs* em produção e reforçando a modularização e manutenibilidade do código desenvolvido.

6 CONCLUSÃO

O presente trabalho evidenciou a aplicação integrada de conceitos fundamentais de engenharia de *software* e práticas ágeis no contexto de desenvolvimento de aplicações móveis. Foi possível constatar que a utilização do *framework Flutter*, aliada a uma arquitetura bem definida, proporcionou a criação de interfaces responsivas e de alto desempenho, validando as escolhas técnicas discutidas no referencial teórico.

Do ponto de vista metodológico, a adoção do *Scrum* e o uso de ferramentas de *ALM* (como o *Azure DevOps*) foram determinantes para a organização do fluxo de trabalho. A análise das métricas, como o *burndown chart*, demonstrou que a disciplina no gerenciamento de tarefas impacta diretamente na previsibilidade e na capacidade de entrega da equipe.

A experiência prática contribuiu significativamente para a formação profissional, consolidando competências técnicas em desenvolvimento multiplataforma e *soft skills* relacionadas à comunicação e gestão do tempo. Como resultados práticos, observou-se uma melhoria tangível na qualidade do produto final e uma maior estabilidade nas versões publicadas.

Para trabalhos futuros, recomenda-se a ampliação da cobertura de testes automatizados (unitários e de integração) para garantir ainda mais robustez ao ciclo de CI/CD, bem como a exploração de arquiteturas de estado mais complexas para cenários de maior escala.

7 REFERÊNCIAS

BARBOSA, Simone; SILVA, Bruno. **Interação humano-computador**. Rio de Janeiro: Elsevier, 2010.

BECK, Kent; et al. **Manifesto para o desenvolvimento ágil de software**. 2001. Disponível em: <https://agilemanifesto.org>. Acesso em: 17 out. 2025.

GOOGLE. **Flutter documentation**.

Disponível em: <https://flutter.dev/docs>. Acesso em 17 out. 2025.

MICROSOFT. Azure DevOps – **Burndown Chart do Sprint**. Disponível em:

https://dev.azure.com/forlogic/ForLogic/_sprints/analytics/Power%20Rangers/ForLogic/DevOps%20-%20Projetos/@Power%20Rangers/PQ4.25/PQ4.25-010%20Melhorar%20a%20experi%C3%Aancia%20de%20uso%20do%20aplicativo%20de%20Metrologia,%20garantindo%20maior%20praticidade%20na%20execu%C3%A7%C3%A3o%20das%20atividades%20e%20suporte%20ao%20trabalho%20offline.

Acesso em: 04 dez. 2025.

MICROSOFT. **Azure DevOps documentation**.

Disponível em: <https://learn.microsoft.com/en-us/azure/devops/>. Acesso em 17 out. 2025.

MICROSOFT. Azure DevOps – **Query de Cards**. Disponível em:

https://dev.azure.com/forlogic/ForLogic/_queries/query-edit/?newQuery=true&parentId=8d6cd293-5ed9-4155-9c27-49f15acc9b0e.

Acesso em: 04 dez. 2025.

OBJECTIVE. **Clean Architecture com MVVM**. Disponível em:

<https://www.objective.com.br/insights/clean-architecture-com-mvvm/>. Acesso em: 04 dez. 2025.

PLANNING POKER ONLINE. **Planning Poker Online**. Disponível em:

<https://planningpokeronline.com/>. Acesso em: 04 dez. 2025.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. Ed. Porto Alegre: AMGH, 2011.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide: the definitive guide to Scrum**. 2020.

Disponível em: <https://scrumguides.org>. Acesso em: 17 out. 2025.

SOMMERVILLE, Ian. **Engenharia de software**. 10. Ed. São Paulo: Pearson, 2016.