



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

ALANIS OLIVEIRA SANTOS

UMA ABORDAGEM INTEGRADA DE ENGENHARIA ORIENTADA A MODELOS E
***LOW-CODE* PARA INOVAÇÃO EM SISTEMAS CORPORATIVOS**

RUSSAS

2026

ALANIS OLIVEIRA SANTOS

UMA ABORDAGEM INTEGRADA DE ENGENHARIA ORIENTADA A MODELOS E
LOW-CODE PARA INOVAÇÃO EM SISTEMAS CORPORATIVOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em ENGENHARIA DE SOFTWARE do CAMPUS DE RUSSAS da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em ENGENHARIA DE SOFTWARE.

Orientador: Prof. Dr. Patrícia Freitas Campos de Vasconcelos

RUSSAS

2026

ALANIS OLIVEIRA SANTOS

UMA ABORDAGEM INTEGRADA DE ENGENHARIA ORIENTADA A MODELOS E
LOW-CODE PARA INOVAÇÃO EM SISTEMAS CORPORATIVOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: 27 de Janeiro de 2026

BANCA EXAMINADORA

Prof^ª. Dra. Patrícia Freitas Campos de
Vasconcelos (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Mayrton Dias de Queiroz
Universidade Federal do Ceará (UFC)

Prof. Ms. Pitágoras Graça Martins
Universidade Federal do Ceará (UFC)

Para minha mãe, rainha da Banca 2 de Julho -
de bala em bala, entre doces, revistas e histó-
rias, formou 3 filhos. Seu amor é a base desta
conquista.

AGRADECIMENTOS

À minha orientadora, Profa. Dra. Patrícia Freitas Campos de Vasconcelos, pela excelência na condução deste trabalho, por ter me dado oportunidades ímpares e por ter acreditado no meu potencial.

Aos membros da banca examinadora, Prof. Ms. Pitágoras Graça Martins e Prof. Dr. Mayrton Dias de Queiroz, agradeço pelo tempo dedicado, pelas valiosas colaborações e pelas sugestões que contribuíram significativamente para o resultado deste trabalho.

À minha mãe, dedico esta conquista. Agradeço por ter feito o impossível para que esta caminhada acontecesse. Obrigado por suas batalhas diárias, de domingo a domingo, e por nunca ter desistido de lutar para me dar a oportunidade de concluir a graduação. Sua força é minha maior inspiração.

Aos meus irmãos, pelo apoio constante e por serem verdadeiros amigos. Agradeço pela parceria em todos os momentos, suportando juntos as distâncias e as longas jornadas entre Jacobina, Petrolina, Salvador, Fortaleza e Russas. A união de vocês foi fundamental para que eu seguisse na batalha até aqui.

Ao meu pai e aos meus demais familiares, pelo carinho, pela torcida constante e por sempre acreditarem em mim, servindo de base e refúgio sempre que precisei.

Aos meus amigos, de perto e de longe, em especial a Lucas e Evely, agradeço pelo apoio e pelo companheirismo. Obrigado por trazerem leveza aos dias tensos e por celebrarem comigo cada pequena vitória desta caminhada.

À empresa objeto deste estudo de caso, agradeço pela oportunidade e pela concessão de acesso às informações e ao incentivo financeiro, sem as quais a realização deste trabalho não seria possível.

RESUMO

Na atual era da transformação digital, a crescente demanda por software exige que as empresas adotem processos mais ágeis e adaptáveis para atender às exigências do mercado. Este trabalho propõe o uso do *framework* EasInnova como uma abordagem integrada para o desenvolvimento ágil de software, combinando Engenharia Orientada por Modelos (MDE) e plataformas *low-code*. Organizada em três etapas - AsIs (análise do cenário atual), Transformação (implementação de mudanças) e To-Be (estado otimizado) - e estruturada em três camadas de modelagem: camada de negócios (CIM), camada funcional (PIM) e camada de implementação (PSM), a solução busca superar o desalinhamento entre os negócios e a Tecnologia da Informação, transformando requisitos estratégicos em sistemas funcionais de forma eficiente. O estudo de caso reflexivo proposto investigou a aplicação deste método no desenvolvimento de um aplicativo de avaliação de riscos (Saúde e Segurança do Trabalho) em uma plataforma *low-code*, tendo como principal expectativa a demonstração de ganhos em agilidade e adaptabilidade no ambiente corporativo.

Palavras-chave: transformação digital; engenharia orientada a modelos; low-code; easinnova; saúde e segurança do trabalho;

ABSTRACT

In the current era of digital transformation, the growing demand for software requires companies to adopt more agile and adaptable processes to meet market demands. This work proposes the use of the EasInnova framework as an integrated approach for agile software development, combining Model-Driven Engineering (MDE) and low-code platforms. Organized into three stages - AsIs (current scenario analysis), Transformation (implementation of changes), and To-Be (optimized state) - and structured into three modeling layers: business layer (CIM), functional layer (PIM), and implementation layer (PSM), the solution seeks to overcome the misalignment between business and Information Technology, efficiently transforming strategic requirements into functional systems. The proposed reflective case study will investigate the application of this method in the development of a risk assessment application (Occupational Health and Safety) on a low-code platform, with the primary expectation of demonstrating gains in agility and adaptability within the corporate environment.

Keywords: digital transformation; model-driven engineering; low-code; easinnova; occupational health and safety;

LISTA DE FIGURAS

Figura 1 – Os quatro princípios da composabilidade, conforme propostos pelo Gartner .	15
Figura 2 – Exemplo de arquitetura componível com módulos integrados por APIs, destacando flexibilidade e adaptabilidade	16
Figura 3 – Passos da arquitetura MDA, com seus artefatos e relacionamentos	17
Figura 4 – Interface do FlutterFlow.	20
Figura 5 – Diagrama de transformação dos processos para maior eficiência: Modelagem do Estágio <i>AsIs</i>	29
Figura 6 – Diagrama de transformação dos processos para maior eficiência: Modelagem do Estágio <i>Transformation</i>	32
Figura 7 – Cronograma do Projeto - Diagrama de Gantt	33
Figura 8 – Fases do desenvolvimento de aplicativo mobile.	37
Figura 9 – Resultados da Telemetria: (a) CPU Time, (b) Memory Usage, (c) Thread Count, (d) Top Consumers.	39
Figura 10 – Resultados da Avaliação TAM - Dimensões de Utilidade.	41
Figura 11 – Resultados da Avaliação TAM - Dimensões de Facilidade.	42
Figura 12 – Distribuição do Tempo de Finalização dos Relatórios	43
Figura 13 – Screenshots do Protótipo Funcional Alpha Checklist.	44

LISTA DE TABELAS

Tabela 1 – Síntese dos Objetivos, Métodos e Evidências Produzidas	35
Tabela 2 – Comparativo de Ciclo de Vida: Padrão de Mercado vs. EasInnova	37

LISTA DE ABREVIACÕES E SIGLAS

<i>API</i>	Application Programming Interface
<i>BPMN</i>	Business Process Model and Notation
<i>CEA</i>	Composable Enterprise Architecture
<i>CIM</i>	Computation Independent Model
<i>ERP</i>	Enterprise Resource Planning
<i>FUP</i>	Facilidade de Uso Percebida
<i>KPI</i>	Key Performance Indicator
<i>LCNC</i>	Low-Code No-Code
<i>LLM</i>	Large Language Model
<i>MDA</i>	Model Driven Architecture
<i>MDE</i>	Model Drive Engineering
<i>MVP</i>	Minimum Viable Product
<i>OMG</i>	Object Management Group
<i>OPAAL</i>	Objeto, Processo, Ator, Atributos, Ligações
<i>PBC</i>	Package Business Capabilities
<i>PIM</i>	Platform Independent Model
<i>PSM</i>	Platform Specific Model
<i>RN</i>	Regra de Negócio
<i>SQL</i>	Structured Query Language
<i>SST</i>	Saúde e Segurança do Trabalho
<i>UCD</i>	User Case Diagram
<i>UI</i>	User Interface
<i>UX</i>	User Experience

SUMÁRIO

1	INTRODUÇÃO	11
2	OBJETIVOS	13
2.1	Objetivos Gerais	13
2.2	Objetivos Específicos	13
3	FUNDAMENTAÇÃO TEÓRICA	14
3.1	Arquitetura Componível	14
3.1.1	<i>Definição e Importância</i>	14
3.1.2	<i>Características Principais</i>	14
3.1.3	<i>Packaged Business Capabilities (PBCs)</i>	15
3.1.4	<i>Benefícios e Desafios</i>	15
3.2	<i>Model Driven Engineering and Model Driven Architecture</i>	16
3.3	<i>Low-code</i>	18
3.3.1	<i>Relação entre MDE, MDA e low-code</i>	21
3.4	<i>Framework EasInnova</i>	22
4	TRABALHOS RELACIONADOS	25
4.1	<i>Framework EasInnova e Metodologias MDE-low-code</i>	25
4.2	<i>Integração MDA-low-code e Abordagens Emergentes</i>	26
4.3	<i>Citizen Development e Democratização Tecnológica</i>	26
4.4	<i>Aplicações Específicas em SST e Domínios Críticos</i>	27
4.5	<i>Síntese e Contribuições</i>	27
5	METODOLOGIA	28
5.1	Caracterização do Contexto do Estudo de Caso	28
5.2	Aplicação Sistemática do Framework EasInnova	29
5.3	Processo de Desenvolvimento e Planejamento Temporal	30
5.4	Ferramentas e Tecnologias Utilizadas	34
5.5	Procedimentos de Coleta e Análise de Dados	34
5.6	Alinhamento entre Objetivos, Métodos e Evidências.	35
6	RESULTADOS	36
6.1	Rastreabilidade e Consistência dos Modelos	36
6.2	Análise de Produtividade e Métricas de Processo	36

6.3	Análise de Desempenho Técnico (Telemetria)	38
6.3.1	<i>Perfil de Processamento e CPU</i>	38
6.3.2	<i>Gestão de Memória e Estado da Aplicação</i>	38
6.3.3	<i>Concorrência e Multithreading</i>	39
6.4	Avaliação da Percepção dos Usuários (TAM)	40
6.4.1	<i>Utilidade Percebida (UP)</i>	40
6.4.2	<i>Facilidade de Uso Percebida (FUP)</i>	41
6.4.3	<i>Intenção de Uso e Autonomia do Usuário</i>	42
6.5	Análise de Desempenho Operacional (Logs)	43
6.6	Demonstração das Interfaces do Aplicativo (Artefato Final)	44
7	DISCUSSÃO	46
7.1	O Equilíbrio entre Abstração e Performance	46
7.2	Modularidade e a Arquitetura Componível	46
7.3	Trade-offs da Persistência Offline e Alinhamento de Negócio	46
7.4	Citizen Development e Democratização do Desenvolvimento	47
8	CONCLUSÃO	48
	REFERÊNCIAS	50
	ANEXOS	52
	ANEXO A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	
	UTILIZADO	53

1 INTRODUÇÃO

Na era da transformação digital, as empresas enfrentam o desafio de alinhar suas estratégias de negócio com soluções de Tecnologia da Informação (TI) que sejam ágeis e adaptáveis às mudanças do mercado (MARTINS *et al.*, 2021). A transformação digital, definida como "A integração de tecnologia digital em todas as áreas de um negócio, resultando em mudanças fundamentais em como as empresas operam e entregam valor aos clientes (The Enterprisers Project, 2020)", é impulsionada pela necessidade de inovação contínua e adaptação às expectativas dos consumidores. Além disso, envolve uma mudança cultural que exige que as organizações desafiem continuamente os seus padrões, experimentem frequentemente ideias e aceitem o fracasso como parte do processo de aprendizado (The Enterprisers Project, 2020). Estudos acadêmicos corroboram essa visão, destacando que a transformação digital não é apenas sobre adoção de tecnologias, mas também sobre uma mudança profunda na cultura organizacional e nos modelos de negócio (KRAUS *et al.*, 2021; VERHOEF *et al.*, 2019). A literatura também enfatiza que a transformação digital é essencial para a sobrevivência e competitividade das empresas em um ambiente cada vez mais volátil (VIAL, 2019).

De acordo com (Gartner, 2020), a arquitetura empresarial componível é fundamental para alcançar adaptabilidade em tempo real e resiliência. Essa abordagem contrasta com os sistemas tradicionais de *Enterprise Resource Planning* (ERP), que, embora versáteis, podem ser menos flexíveis devido à sua natureza monolítica. Enquanto os ERPs oferecem módulos customizáveis, as arquiteturas componíveis permitem uma modularidade mais avançada, com componentes independentes que podem ser facilmente substituídos ou expandidos sem afetar o sistema como um todo. Essa flexibilidade é crucial para empresas que precisam responder rapidamente a mudanças no mercado ou em suas estratégias de negócio (GARTNER, 2021). A adoção de arquiteturas componíveis, no entanto, apresenta desafios, como a complexidade de integrar diferentes componentes e alinhar requisitos de negócio com implementações técnicas.

Nesse contexto, a Engenharia Orientada por Modelos (MDE) e as plataformas *low-code* têm ganhado destaque como abordagens que podem facilitar o desenvolvimento de software corporativo. A MDE propõe o uso de modelos como artefatos principais no processo de desenvolvimento, abstraindo a lógica de negócio da tecnologia de implementação (BRAMBILLA *et al.*, 2017). Já as plataformas *low-code* permitem a criação de aplicações com mínimo de codificação manual, utilizando interfaces visuais e ferramentas de arrastar e soltar (SAHAY *et al.*, 2020). A combinação dessas duas abordagens pode oferecer uma solução poderosa para

alinhar requisitos de negócio com implementações técnicas de forma ágil e eficiente, reduzindo o tempo de desenvolvimento e permitindo maior participação de equipes não técnicas.

Para enfrentar essa lacuna, o *framework* EasInnova introduz uma abordagem que combina *Model-Driven Engineering* (MDE) e *low-code* (Pouco código). Desenvolvido por MISSIKOFF, o *framework* organiza o processo de desenvolvimento em três estágios (*AsIs*, *Transformation*, *ToBe*) e três camadas de modelagem (*CIM* - Modelo Independente de Computação, *PIM* - Modelo Independente de Plataforma, *PSM* - Modelo Específico de Plataforma), permitindo a tradução de modelos de negócio em aplicações funcionais. A metodologia facilita a transformação digital ao envolver não técnicos na modelagem de processos, promovendo um alinhamento mais próximo entre estratégias de negócio e soluções tecnológicas (MISSIKOFF, 2020).

Apesar do potencial das abordagens baseadas em MDE e plataformas low-code para apoiar a transformação digital e o alinhamento entre negócio e TI, ainda são limitados os estudos empíricos que avaliam, de forma sistemática, a eficácia de *frameworks* que integram essas abordagens em contextos reais de desenvolvimento de soluções corporativas. Nesse sentido, torna-se relevante investigar como o *framework* EasInnova contribui para a agilidade do processo de desenvolvimento, para a rastreabilidade entre requisitos e implementação e para o alinhamento estratégico entre áreas técnicas e de negócio. Assim, este trabalho propõe-se a analisar a aplicação do EasInnova em um estudo de caso concreto, buscando evidenciar seus benefícios, limitações e impactos no desenvolvimento de aplicações corporativas.

2 OBJETIVOS

Esta seção apresenta o objetivo geral que norteia a pesquisa, bem como os objetivos específicos que detalham as etapas necessárias para alcançá-lo.

2.1 Objetivos Gerais

O objetivo geral deste trabalho é avaliar a eficácia do *framework* EasInnova como uma abordagem integrada do MDE e plataformas *low-code* no desenvolvimento de soluções corporativas, analisando seus impactos na agilidade do processo e no aprimoramento do alinhamento entre as corporações, negócios e TI.

2.2 Objetivos Específicos

Para alcançar o objetivo geral, foram definidos os seguintes objetivos específicos:

- Modelar um processo de negócio real utilizando as etapas do framework EasInnova (*AsIs*, *Transformation* e *ToBe*) para traduzir os requisitos em artefatos nas camadas CIM e PIM.
- Implementar a solução proposta - o Alpha Checklist, aplicativo mobile - em uma plataforma *low-code* (*FlutterFlow*), a partir da transformação do modelo PIM em um PSM, materializando os artefatos em uma aplicação funcional.
- Avaliar, por meio de uma análise quali-quantitativa, os resultados da aplicação do *framework* no desenvolvimento do aplicativo, considerando métricas de processo (como tempo de desenvolvimento, número de defeitos e usabilidade) e a percepção dos envolvidos sobre os benefícios e desafios da abordagem.
- Analisar a rastreabilidade e a consistência entre os requisitos de negócio, os modelos gerados e a implementação final, verificando a fidelidade da solução aos objetivos estratégicos definidos na etapa *AsIs*.

3 FUNDAMENTAÇÃO TEÓRICA

Esta seção visa contextualizar os conceitos fundamentais que servem de base para a abordagem proposta no presente trabalho. Serão explorados os paradigmas da Arquitetura Componível, Engenharia Orientada a Modelos e desenvolvimento *low-code*, culminando na apresentação do *framework* EasInnova, que integra essas abordagens para promover a inovação em sistemas corporativos e o alinhamento entre negócios e tecnologia da informação.

3.1 Arquitetura Componível

A Arquitetura Componível é um paradigma de desenvolvimento de software que se contrapõe às abordagens monolíticas, priorizando a construção de sistemas a partir de módulos independentes e interoperáveis. Será detalhado os fundamentos deste modelo, abordando sua definição, seus princípios, o conceito de Packaged Business Capabilities (PBCs) e uma análise de seus benefícios e desafios.

3.1.1 Definição e Importância

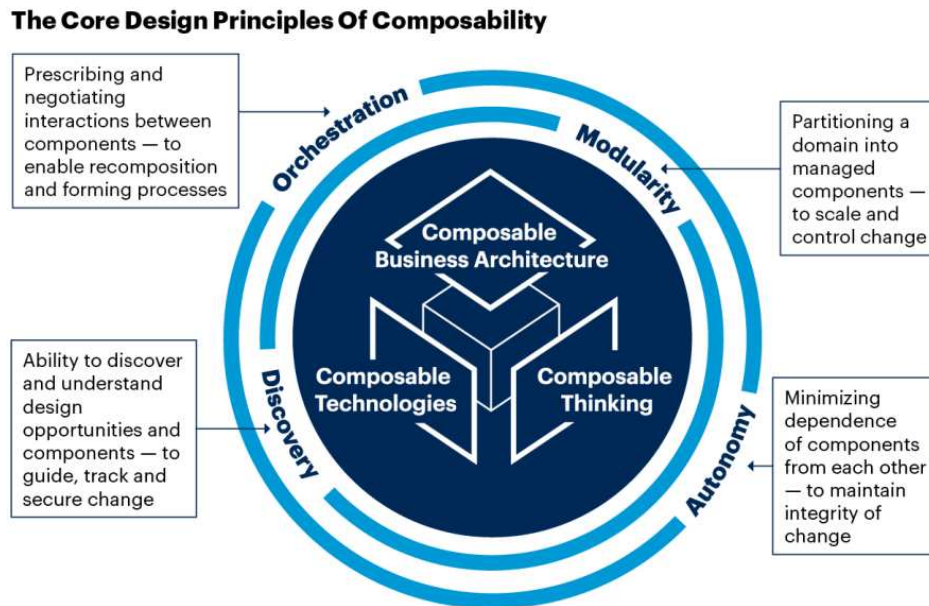
As arquiteturas componíveis, também conhecidas como *Composable Enterprise Architecture* (CEA), representam um paradigma moderno no desenvolvimento de sistemas corporativos, caracterizado pela construção de soluções a partir de módulos independentes e interoperáveis. Esses módulos, denominados *Packaged Business Capabilities* (PBCs), podem ser combinados, substituídos ou expandidos conforme as demandas do negócio, promovendo flexibilidade e agilidade organizacional (IVAS, 2024). Diferentemente das arquiteturas monolíticas tradicionais, como os sistemas ERP, que apresentam forte acoplamento entre componentes e dificuldades de adaptação (XLEDGER, 2025), as arquiteturas componíveis permitem respostas rápidas às mudanças do mercado e às necessidades estratégicas.

3.1.2 Características Principais

As arquiteturas componíveis baseiam-se em quatro características fundamentais, como mostrado na Figura 1, que as distinguem de abordagens tradicionais (Gigaspace, 2024):

- **Modularidade:** Cada componente opera de forma independente, permitindo desenvolvimento, substituição ou reutilização sem impactar o sistema como um todo.

Figura 1 – Os quatro princípios da composabilidade, conforme propostos pelo Gartner



Fonte: Adaptada de (Gartner, 2020).

- **Autonomia:** Os módulos executam operações de forma independente, favorecendo respostas rápidas às mudanças.
- **Orquestração:** A integração entre módulos é coordenada por APIs padronizadas, garantindo interoperabilidade.
- **Descoberta e Reuso:** Componentes podem ser reutilizados, otimizando recursos.

3.1.3 *Packaged Business Capabilities (PBCs)*

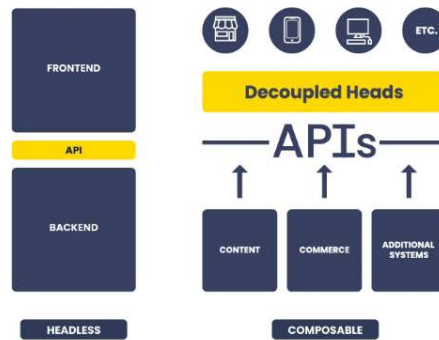
As *Packaged Business Capabilities* (PBCs) são componentes de software que encapsulam capacidades de negócio bem definidas, combinando esquemas de dados, APIs e canais de eventos em uma única entidade (NATIS *et al.*, 2019). Diferentemente de microserviços, que focam em funcionalidades técnicas, as PBCs priorizam capacidades de negócio completas.

3.1.4 *Benefícios e Desafios*

As arquiteturas componíveis oferecem benefícios significativos, como ilustrado na Figura 2, incluindo maior agilidade, redução do *time-to-market* e flexibilidade na adaptação a mudanças (N-iX, 2024).

Entretanto, a implementação dessas arquiteturas enfrenta desafios, como a complexidade de gestão, que exige governança robusta para coordenar múltiplos componentes, e a mudança cultural, que demanda colaboração entre equipes técnicas e de negócio (Contentstack,

Figura 2 – Exemplo de arquitetura componível com módulos integrados por APIs, destacando flexibilidade e adaptabilidade



Fonte: Adaptada de ButterCMS (ButterCMS, 2023).

2023).

3.2 *Model Driven Engineering and Model Driven Architecture*

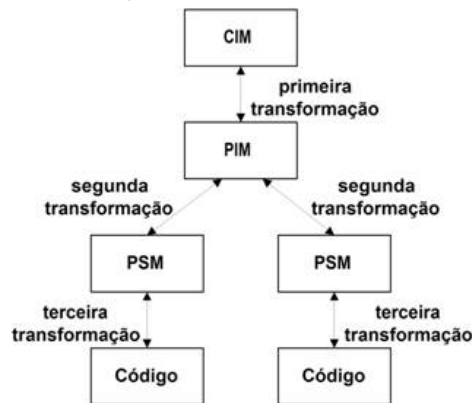
A Engenharia Orientada a Modelos (*Model-Driven Engineering - MDE*) representa um paradigma fundamental no desenvolvimento de software que coloca os modelos no centro do processo de criação, manutenção e evolução de sistemas (BRAMBILLA *et al.*, 2017). Diferentemente das abordagens tradicionais centradas em código, a MDE estabelece os modelos como artefatos de primeira classe, utilizados para especificar, testar, simular, verificar, modernizar, manter, compreender e gerar código para sistemas computacionais (SCHMIDT, 2006). Este paradigma emerge da necessidade de lidar com a crescente complexidade dos sistemas técnicos, que apresentam um número cada vez maior de componentes, funcionalidades e envolvimento de diferentes disciplinas (FRANCE; RUMPE, 2007).

A MDE fundamenta-se no princípio de que a abstração constitui um mecanismo poderoso para dominar a complexidade durante o processo de desenvolvimento. Assim como as linguagens de programação abstraem detalhes de baixo nível da arquitetura de *hardware*, a MDE promove um nível adicional de abstração, permitindo que desenvolvedores se concentrem nos aspectos essenciais do domínio de aplicação ao invés de se preocuparem com detalhes específicos de implementação. Esta abordagem visa aumentar a produtividade através da maximização da compatibilidade entre sistemas via reutilização de modelos padronizados, simplificação do processo de *design* mediante modelos de padrões recorrentes no domínio da aplicação, e promoção da comunicação entre indivíduos e equipes através da padronização de terminologia e melhores práticas (BRAMBILLA *et al.*, 2017).

Um exemplo paradigmático da MDE é a *Model-Driven Architecture* (MDA), proposta pelo ((OMG), 2003) em 2001. A MDA representa uma iniciativa abrangente para especificação de sistemas e interoperabilidade baseada no uso de modelos formais, constituindo um marco evolutivo significativo na forma como a OMG define padrões de interoperabilidade. Tradicionalmente, a interoperabilidade baseava-se largamente em padrões CORBA e serviços associados, onde sistemas heterogêneos operavam ao nível de interfaces de componentes padrão. A MDA, por outro lado, coloca modelos formais de sistema no centro do problema de interoperabilidade.

A arquitetura MDA estrutura-se fundamentalmente na separação de preocupações entre a lógica de negócios e a implementação tecnológica. Esta separação é operacionalizada através de três camadas arquiteturais distintas, cada uma representando diferentes níveis de abstração e independência tecnológica, ilustrada na Figura 3.

Figura 3 – Passos da arquitetura MDA, com seus artefatos e relacionamentos



Fonte: (SOUZA, 2008).

O *Computation Independent Model* (CIM) constitui a camada mais abstrata, descrevendo o sistema do ponto de vista conceitual utilizando terminologia familiar aos praticantes do domínio do sistema. Esta camada captura os requisitos e o ambiente do sistema sem referência a aspectos computacionais específicos ((OMG), 2003).

O *Platform Independent Model* (PIM) representa a camada intermediária, descrevendo aspectos computacionais do sistema de forma suficientemente detalhada e formal, porém sem especificar detalhes específicos de plataforma de implementação ((OMG), 2003). Os PIMs capturam a lógica de negócios utilizando linguagens de modelagem independentes de plataforma, frequentemente incluindo diagramas *BPMN* (*Business Process Model and Notation*) para representar fluxos de processos organizacionais ((OMG), 2011). A BPMN, desenvolvida originalmente pela *Business Process Management Initiative* (BPMI) e posteriormente mantida pela OMG, fornece uma notação gráfica padronizada para especificação de processos de negócio,

sendo particularmente adequada para modelagem no nível *PIM* devido à sua capacidade de representar semânticas complexas de processos de forma intuitiva tanto para usuários técnicos quanto de negócios .

O *Platform Specific Model (PSM)* constitui a camada mais concreta, incluindo detalhes específicos da plataforma de implementação. Por plataforma, a MDA entende um conjunto coeso de subsistemas e tecnologias sobre os quais um sistema pode executar. O *PSM* incorpora as particularidades tecnológicas necessárias para implementação do sistema na plataforma específica escolhida.

3.3 *Low-code*

O desenvolvimento *low-code* representa uma evolução paradigmática na criação de software que visa facilitar e democratizar o processo de desenvolvimento através da redução significativa da necessidade de codificação manual (DEQODE, 2023). Esta abordagem emergiu como resposta às crescentes demandas organizacionais por agilidade digital e à persistente escassez de desenvolvedores qualificados no mercado global de tecnologia (IDEOSOFTWARE, 2023). As plataformas *low-code* utilizam interfaces visuais intuitivas, recursos de arrastar e soltar (*drag-and-drop*), e componentes pré-construídos para permitir que aplicações sejam desenvolvidas em um tempo significativamente menor comparado à programação tradicional (Mendix, 2023), tornando-se uma solução particularmente atrativa para empresas que buscam acelerar seus processos de transformação digital.

A fundamentação conceitual do desenvolvimento *low-code* baseia-se na abstração das complexidades técnicas inerentes à programação tradicional (DEQODE, 2023), permitindo que usuários com conhecimento técnico limitado participem ativamente do processo de criação de soluções empresariais. Uma revisão sistemática da literatura demonstrou que as plataformas *low-code/No-Code (LCNC)* estão transformando práticas de negócio ao deslocar o foco de abordagens tradicionais intensivas em código para modelagem centrada no negócio, incorporando capacidades de inteligência artificial e *machine learning* que automatizam aspectos complexos do desenvolvimento, promovendo inovação e colaboração entre equipes de TI e negócios (OZMAN, 2025). Este paradigma representa uma mudança fundamental na forma como organizações abordam o desenvolvimento de software, *transitioning* de metodologias exclusivamente técnicas para abordagens mais inclusivas e colaborativas.

O fenômeno do desenvolvimento *low-code* está intrinsecamente relacionado ao con-

ceito de *citizen development*, que se refere à prática de empoderar funcionários não-técnicos para criar aplicações de negócios utilizando tecnologias aprovadas pela equipe de TI. Pesquisas indicam que até 2025, 70% das novas aplicações desenvolvidas por organizações utilizarão tecnologias *low-code* ou *no-code*, um aumento significativo em relação aos menos de 25% registrados em 2020 (AIMULTIPLE, 2025). Esta tendência reflete uma democratização do desenvolvimento de software, onde as barreiras técnicas são substancialmente reduzidas, permitindo que especialistas de domínio traduzam diretamente suas necessidades de negócio em soluções funcionais.

Os benefícios das plataformas *low-code* são multidimensionais e abrangem aspectos operacionais, estratégicos e econômicos. A redução significativa no tempo de desenvolvimento representa uma das vantagens mais tangíveis, com estudos demonstrando que essas plataformas têm o potencial de reduzir o tempo de desenvolvimento em até 90% comparado aos métodos tradicionais (DECERTO, 2025). A maior produtividade das equipes resulta da eliminação de tarefas repetitivas de codificação, permitindo que desenvolvedores se concentrem em aspectos mais estratégicos e complexos dos projetos. A facilitação da colaboração entre as equipes emerge como um benefício crucial, pois as interfaces visuais e a linguagem comum proporcionada pelas plataformas *low-code* reduzem as barreiras de comunicação tradicionalmente existentes entre departamentos técnicos e não-técnicos.

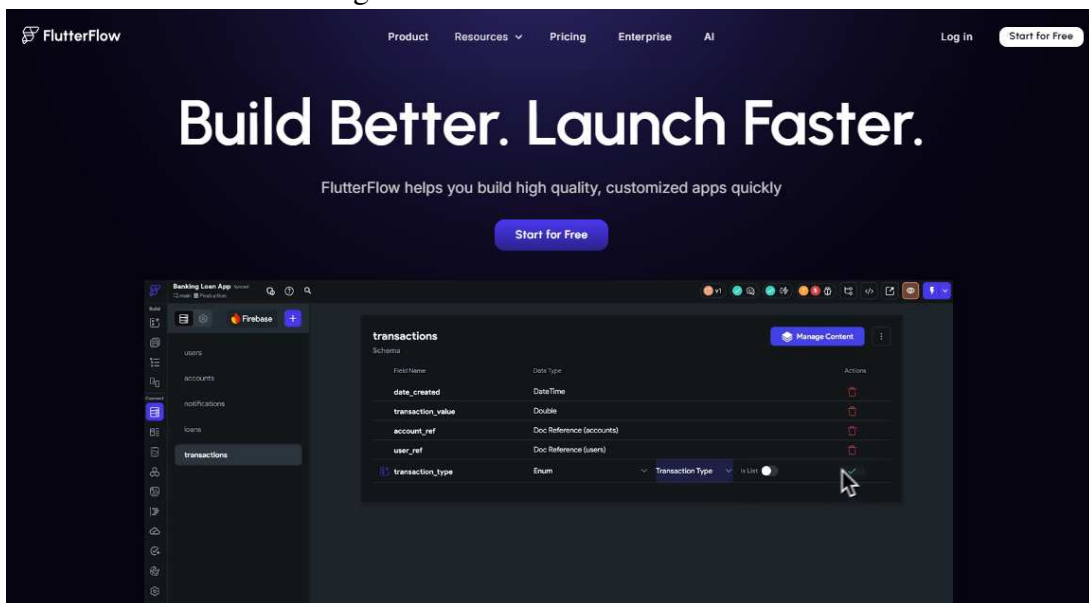
Contudo, as plataformas *low-code* também apresentam limitações e desafios que devem ser cuidadosamente considerados no processo de adoção organizacional. O *vendor lock-in* constitui uma preocupação fundamental, pois organizações podem tornar-se excessivamente dependentes de fornecedores específicos de plataforma, limitando flexibilidade e autonomia tecnológica. As limitações em customizações complexas representam outro desafio significativo, especialmente para organizações que requerem funcionalidades altamente especializadas ou integração com sistemas legados complexos. Preocupações com segurança e governança emergem como questões críticas, particularmente quando *citizen developers* criam aplicações sem supervisão adequada de TI, potencialmente introduzindo vulnerabilidades de segurança ou violações de *compliance*.

A escalabilidade limitada para aplicações enterprise críticas representa uma limitação técnica importante das plataformas *low-code* atuais. Embora essas plataformas sejam altamente eficazes para aplicações de pequeno a médio porte, podem apresentar limitações quando aplicadas a sistemas de grande escala que requerem performance extrema, alta disponibilidade, ou

processamento de volumes massivos de dados.

Dentre as ferramentas disponíveis no mercado, o *FlutterFlow* emerge como uma plataforma *low-code* especializada que demonstra a evolução e sofisticação crescente dessas soluções (FlutterFlow, 2023). O *FlutterFlow* é uma plataforma *low-code* especializada no desenvolvimento de aplicações móveis que se integra ao *framework Flutter* para gerar código nativo utilizando a linguagem de programação *Dart*, facilitando prototipagem rápida e desenvolvimento ágil (FlutterFlow, 2023). Esta plataforma exemplifica como soluções *low-code* podem combinar facilidade de uso com capacidades técnicas avançadas, oferecendo integração com sistemas de gerenciamento de banco de dados como *SQLite*, um sistema de gerenciamento relacional leve, embutido e sem servidor que armazena conteúdo em um único arquivo, suporta *SQL* completo e garante propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade). Tendo uma boa visualização de cada componente e funcionalidade, como é mostrado na Figura 4.

Figura 4 – Interface do FlutterFlow.



A capacidade do *FlutterFlow* de suportar funcionamento online através de bancos como o Firebase ou *offline* através da integração com *SQLite* representa uma funcionalidade crucial para aplicações corporativas, especialmente em cenários onde conectividade de rede pode ser intermitente ou onde requisitos de performance demandam acesso local a dados. A plataforma facilita prototipagem ágil e customização através de custom code na linguagem de programação *Dart*, sendo particularmente adequada para projetos com prazos curtos e recursos limitados. Esta combinação de capacidades demonstra como plataformas *low-code* modernas podem oferecer funcionalidades sofisticadas mantendo a simplicidade de uso que caracteriza

esta abordagem de desenvolvimento.

3.3.1 *Relação entre MDE, MDA e low-code*

A relação simbiótica entre Engenharia Orientada a Modelos (MDE), *Model-Driven Architecture* (MDA) e desenvolvimento *low-code* representa uma convergência natural de paradigmas que enfatizam abstração e automação no desenvolvimento de software. A MDE posiciona modelos como artefatos centrais no processo de desenvolvimento, enquanto a MDA estrutura esses modelos em camadas hierárquicas (*CIM, PIM, PSM*) para separar lógica de negócios da implementação técnica específica. As plataformas *low-code* complementam essas abordagens ao permitir a transformação rápida de modelos conceituais em aplicações funcionais utilizando interfaces visuais e geração automática de código.

Esta integração promove agilidade no desenvolvimento através da redução de ciclos de feedback entre modelagem conceitual e implementação prática, mantém fidelidade aos requisitos através da preservação da intenção de negócio desde a concepção até a implementação, e reduz dependência de codificação manual através da automação de aspectos repetitivos do desenvolvimento. A combinação dessas abordagens cria uma ponte eficiente entre teoria e execução, permitindo que organizações traduzam estratégias de negócio em soluções tecnológicas de forma mais direta e eficiente.

O desenvolvimento *low-code* representa, portanto, uma evolução significativa no paradigma de criação de software que combina acessibilidade, eficiência e capacidades técnicas avançadas. Através da democratização do desenvolvimento e da redução de barreiras técnicas, essas plataformas permitem que organizações respondam mais rapidamente às demandas de mercado, otimizem recursos de TI. Embora desafios relacionados à governança, segurança e escalabilidade persistam, a evolução contínua dessas plataformas e o desenvolvimento de melhores práticas organizacionais sugerem que o desenvolvimento *low-code* continuará desempenhando um papel crescente na transformação digital empresarial, oferecendo uma abordagem equilibrada entre simplicidade de uso e sofisticação técnica necessária para atender às complexas demandas do ambiente de negócios contemporâneo.

3.4 *Framework EasInnova*

O *framework* EasInnova, desenvolvido por (MISSIKOFF, 2020), é uma metodologia integrada que combina MDE e *low-code* para facilitar a inovação de processos de negócios e o desenvolvimento ágil de aplicações corporativas. Sua estrutura é organizada em três estágios (*AsIs*, *Transformation*, *ToBe*) e três camadas de modelagem (*CIM*, *PIM*, *PSM*), visando traduzir requisitos em soluções técnicas funcionais de forma eficaz (MISSIKOFF, 2020).

Estágio *AsIs* (Cenário Atual) O estágio *AsIs* é a fase inicial, focada em compreender e documentar o cenário atual da organização. Na camada *CIM*, o objetivo é modelar o estado atual dos negócios, descrevendo textualmente os processos e identificando os componentes principais como Objetos, Processos, Atores, Atributos e Ligações (OPAAL), que permite mapear um domínio indicando o que existe (objetos), como funciona (processos), quem interage (atores), com que características (atributos) e de que modo se relacionam (ligações). Isso inclui a criação de diagramas preliminares, a análise de problemas e oportunidades, e a identificação dos *stakeholders* envolvidos. Na camada *PIM*, o foco é detalhar os modelos do *AsIs*, utilizando diagramas *BPMN* para mapear os processos atuais, além de modelar entidades, relacionamentos e fluxos de dados. Por fim, na camada *PSM*, realiza-se uma análise dos sistemas existentes, incluindo a organização de bancos de dados, inventário de dados legados, avaliação da arquitetura técnica e identificação de restrições e interfaces. Essa etapa estabelece uma base sólida para entender as limitações e oportunidades do cenário atual.

Estágio *Transformation* (Transformação) O estágio *Transformation* é a ponte entre o cenário atual e o desejado, concentrando-se na definição de estratégias para a inovação. Na camada *CIM*, o objetivo é elaborar estratégias de inovação, formulando declarações de inovação e desenvolvendo estratégias para problemas identificados, como a análise de soluções alternativas e seus prós e contras (utilizando métodos como SWOT, TOWS ou SOAR). Também se seleciona a solução candidata e identificam-se todos os atores envolvidos. Na camada *PIM*, o foco é redesenhar os processos técnicos, modelando a solução candidata e definindo novas organizações, papéis e competências, com a construção progressiva dos modelos *ToBe*. Isso inclui a análise de viabilidade técnica e a reestruturação de responsabilidades. Na camada *PSM*, define-se a estratégia de implementação, escolhendo a plataforma de baixo código a ser adotada, planejando a migração de dados, definindo a requisição de infraestrutura e selecionando tecnologias adequadas para a transformação organizacional. Essa etapa é crucial para garantir que a transição do *AsIs* para o *ToBe* seja viável e bem planejada.

Estágio *ToBe* (Cenário Futuro) O estágio *ToBe* representa o cenário futuro desejado, com a consolidação das mudanças propostas. Na camada CIM, o objetivo é definir o novo cenário, elaborando sua descrição textual, o novo léxico *OPAAL*, diagramas preliminares e validando conceitualmente com os *stakeholders*, alinhando os objetivos de negócios. Na camada PIM, finalizam-se os modelos técnicos detalhados, incluindo diagramas *BPMN* e UML atualizados, diagramas de casos de uso e validação técnica dos novos fluxos e processos, garantindo consistência com os requisitos. Por fim, na camada PSM, os modelos *ToBe* são transferidos para a plataforma de baixo código escolhida, configurando o *BPMN* na plataforma, desenvolvendo interfaces e sistemas externos, configurando regras de negócio e realizando testes para gerar a aplicação final. Essa etapa assegura que a inovação seja implementada de forma prática e funcional, alinhada às necessidades da organização. A Tabela 1 ilustra o *framework* Easinnova, suas etapas e camadas.

Quadro 1 – EasInnova *Framework*: Etapas e Camadas MDA

Camadas MDA	AsIs (Cenário Atual)	Transformation (Transformação)	ToBe (Cenário Futuro)
CIM (Modelo Independente de Computação)	<p>Objetivo: Modelar o estado atual do processo de negócios</p> <ul style="list-style-type: none"> - Descrição textual do cenário de negócios atual - Análise OPAAL (Objeto, Processo, Ator, Atributos, Ligações) - Desenvolvimento de léxico de terminologia - Diagrama de classes UML preliminar - Análise de problemas e oportunidades - Identificação de <i>stakeholders</i> 	<p>Objetivo: Elaborar estratégias para inovação</p> <ul style="list-style-type: none"> - Formulação de declarações de inovação - Desenvolvimento de estratégias para resolver problemas - Elaboração de soluções alternativas - Análise de prós e contras - Métodos de avaliação (<i>SWOT, TOWS, SOAR</i>) - Seleção da solução candidata 	<p>Objetivo: Definir o novo cenário conceitual</p> <ul style="list-style-type: none"> - Descrição textual do novo cenário - Elaboração do novo léxico OPAAL - Diagrama de classes UML preliminar do novo modelo - Validação conceitual com <i>stakeholders</i> - Alinhamento com objetivos de negócios
PIM (Modelo Independente de Plataforma)	<p>Objetivo: Representar processos em modelo independente de plataforma</p> <ul style="list-style-type: none"> - Modelagem BPMN do processo atual - Desenvolvimento de diagramas UML - Modelagem de entidades e relacionamentos - Definição completa dos fluxos de dados - Identificação de interações entre atores 	<p>Objetivo: Redesenhar processos tecnicamente</p> <ul style="list-style-type: none"> - Modelagem <i>BPMN</i> da solução candidata - Refinamento técnico das propostas - Análise de viabilidade técnica - Definição de novos fluxos de trabalho - Reestruturação de responsabilidades 	<p>Objetivo: Finalizar modelos</p> <ul style="list-style-type: none"> - <i>BPMN</i> detalhado do novo processo - Diagramas UML atualizados - Validação técnica dos modelos - Verificação de consistência
PSM (Modelo Específico de Plataforma)	<p>Objetivo: Analisar sistemas existentes</p> <ul style="list-style-type: none"> - Inventário de dados e sistemas legados - Avaliação da arquitetura técnica atual - Identificação de restrições técnicas - Mapeamento de interfaces - Documentação de dependências 	<p>Objetivo: Definir estratégia de implementação</p> <ul style="list-style-type: none"> - <i>Benchmarking</i> de plataformas <i>low-code</i> - Estabelecimento de requisitos de dados e infraestrutura - Seleção de tecnologias - Prototipagem rápida 	<p>Objetivo: Implementar na plataforma <i>low-code</i></p> <ul style="list-style-type: none"> - Configuração do <i>BPMN</i> na plataforma - Desenvolvimento de interfaces - Integração com sistemas externos - Configuração de regras de negócios - Testes e geração da aplicação

Fonte: Elaborado pela autora (2025).

4 TRABALHOS RELACIONADOS

Esta seção apresenta uma revisão dos principais trabalhos que fundamentam e contextualizam o presente estudo. Os trabalhos relacionados foram organizados em temas que abordam o *framework* EasInnova, a integração da MDE e plataformas *low-code*, o movimento de *citizen development*, a transformação digital facilitada por *low-code* e aplicações específicas em Saúde e Segurança do Trabalho(SST). Essa estrutura permite posicionar o trabalho dentro do estado da arte, evidenciando sua relevância e contribuições no contexto da inovação em sistemas corporativos.

4.1 *Framework* EasInnova e Metodologias MDE-*low-code*

A complexidade no desenvolvimento de sistemas corporativos frequentemente resulta em um distanciamento entre as necessidades de negócios e as entregas tecnológicas. Para mitigar esse problema, torna-se necessário adotar abordagens que unifiquem a visão estratégica e a execução técnica.

Nesse contexto, o *framework* EasInnova, proposto por (MISSIKOFF, 2020), constitui a base conceitual fundamental deste trabalho. Ele integra MDE e plataformas *low-code*, que organiza o processo de inovação em três estágios - *AsIs*, *Transformation* e *ToBe* - alinhados às camadas da Arquitetura Orientada a Modelos (MDA: CIM, PIM, PSM). Desenvolvido para capacitar pessoas sem formação técnica a colaborarem com projetos de inovação e processos de negócios, o *framework* reduz a dependência de especialistas em desenvolvimento e promove um alinhamento mais estreito entre as áreas de negócios e tecnologia.

Complementarmente, (MISSIKOFF, 2021) apresenta uma abordagem baseada em conhecimento que detalha a construção progressiva de uma base organizacional por meio de seis artefatos estruturados, culminando em uma ontologia de processos de negócio. Essa metodologia propõe uma transição de descrições textuais intuitivas para representações formais, mantendo os usuários de negócios no centro do processo. Tal abordagem é particularmente relevante para este trabalho, pois facilita a análise inicial de processos (estágio *AsIs*) e a identificação de oportunidades de inovação em SST.

4.2 Integração *MDA-low-code* e Abordagens Emergentes

Enquanto o EasInnova fornece o método, a viabilidade técnica deste projeto apoia-se na convergência entre os princípios da MDA e as ferramentas de desenvolvimento rápido. A literatura recente aponta que essa integração deixou de ser apenas teórica para se tornar uma prática de mercado.

Corroborando essa visão, (MENDIX, 2025) demonstra que os princípios abstratos da MDA se concretizam operacionalmente nas plataformas *low-code* modernas. Estas oferecem linguagens visuais que permitem a colaboração direta entre partes interessadas, essencial para o Alpha Checklist, onde a validação das regras de segurança deve ser feita por especialistas da área, e não apenas por programadores.

Além da facilidade de desenvolvimento, a sustentabilidade da solução a longo prazo é uma preocupação crítica em projetos corporativos. Nesse sentido, (ALFONSO *et al.*, 2024) abordam a questão da interoperabilidade e o risco de aprisionamento tecnológico (*vendor lock-in*). Os autores propõem uma estratégia para migração semi-automática de modelos entre ferramentas distintas, utilizando o *framework* BESSER e Inteligência Artificial (*LLMs*). Embora o foco deste TCC seja a implementação em uma plataforma específica, a discussão levantada por (ALFONSO *et al.*, 2024) valida a escolha de uma abordagem baseada em modelos (MDA), pois ela assegura que a lógica de negócios do Alpha Checklist possa ser preservada e migrada futuramente, independentemente da ferramenta *low-code* utilizada.

4.3 Citizen Development e Democratização Tecnológica

A aplicação prática do EasInnova pressupõe um perfil ativo de usuário capaz de criar soluções, conceito este conhecido como *citizen development*. Este movimento alinha-se diretamente aos objetivos deste trabalho ao promover a democratização do desenvolvimento de software.

Segundo (KPMG, 2025), *citizen development* é a capacitação de funcionários não técnicos para criar aplicações via plataformas *low-code/no-code* (LCNC). O estudo destaca que 41% das empresas já possuem iniciativas ativas nessa área, dado que valida a relevância de mercado da abordagem adotada para a criação do Alpha Checklist.

No entanto, para que essa autonomia não gere riscos de segurança ou despadronização (*shadow IT*), a governança é essencial. (PMI, 2025) contribui com um *framework*

educacional que enfatiza a colaboração entre a TI e os desenvolvedores cidadãos. Reforçando essa estrutura, (PRISM, 2024) detalha que o uso de ferramentas de ‘arrastar e soltar’ torna o desenvolvimento acessível, mas requer definição clara de papéis. Para este Trabalho de Conclusão de Curso (TCC), essas referências fundamentam a postura do autor como um desenvolvedor cidadão que, guiado por uma metodologia (EasInnova), e com auxílio dos técnicos da empresa-alvo desenvolvem uma solução segura e padronizada.

4.4 Aplicações Específicas em SST e Domínios Críticos

Embora a literatura sobre SST com abordagens integradas de MDA e *low-code* seja limitada, casos práticos oferecem insights valiosos. A empresa VeilSun desenvolveu soluções *low-code* para segurança na construção civil, incluindo inspeções digitais e relatórios em tempo real, alinhando-se ao propósito do Alpha Checklist (VeilSun, 2026). A empresa Emerge Apps, utilizando a plataforma Caspio, criou ferramentas de conformidade e segurança, como o OSHALogs.com, que automatiza relatórios OSHA, funcionando como um exemplo inspirador para este TCC (Caspio, 2026; OSHALogs, 2026).

4.5 Síntese e Contribuições

Os trabalhos relacionados fornecem uma base teórica e prática para este trabalho. Apesar da relevância dos trabalhos apresentados, observa-se uma lacuna específica na literatura acadêmica: a falta de um estudo de caso que aplique uma metodologia formal e integrada como a *MDE-low-code* para desenvolver, de ponta a ponta, uma solução específica para o domínio de Saúde e Segurança do Trabalho (SST). Enquanto os estudos de caso existentes demonstram a aplicação prática de plataformas *low-code*, eles não detalham a aplicação de um *framework* metodológico estruturado como o EasInnova. O presente trabalho busca, portanto, preencher essa lacuna, ao documentar e avaliar a aplicação do EasInnova no desenvolvimento da aplicação.

5 METODOLOGIA

Este trabalho caracteriza-se como um estudo de caso avaliativo, de natureza aplicada, com abordagem quali-quantitativa, no qual a pesquisadora atua como participante no processo de desenvolvimento da solução. O objetivo da metodologia é investigar a eficácia do *framework* EasInnova como uma abordagem integrada de Engenharia Orientada a Modelos (MDE) e plataformas *low-code* no desenvolvimento de soluções corporativas, considerando aspectos de processo, produto e alinhamento entre negócio e Tecnologia da Informação.

Serão considerados Critérios de Avaliação da Eficácia do *framework*: Redução de tempo, Rastreabilidade, Alinhamento negócio-TI e Aceitação dos usuários. A metodologia está organizada em cinco etapas principais: (i) caracterização do contexto do estudo de caso; (ii) aplicação sistemática do *framework* EasInnova; (iii) processo de desenvolvimento e planejamento temporal; (iv) ferramentas e tecnologias utilizadas; e (v) procedimentos de coleta e análise de dados.

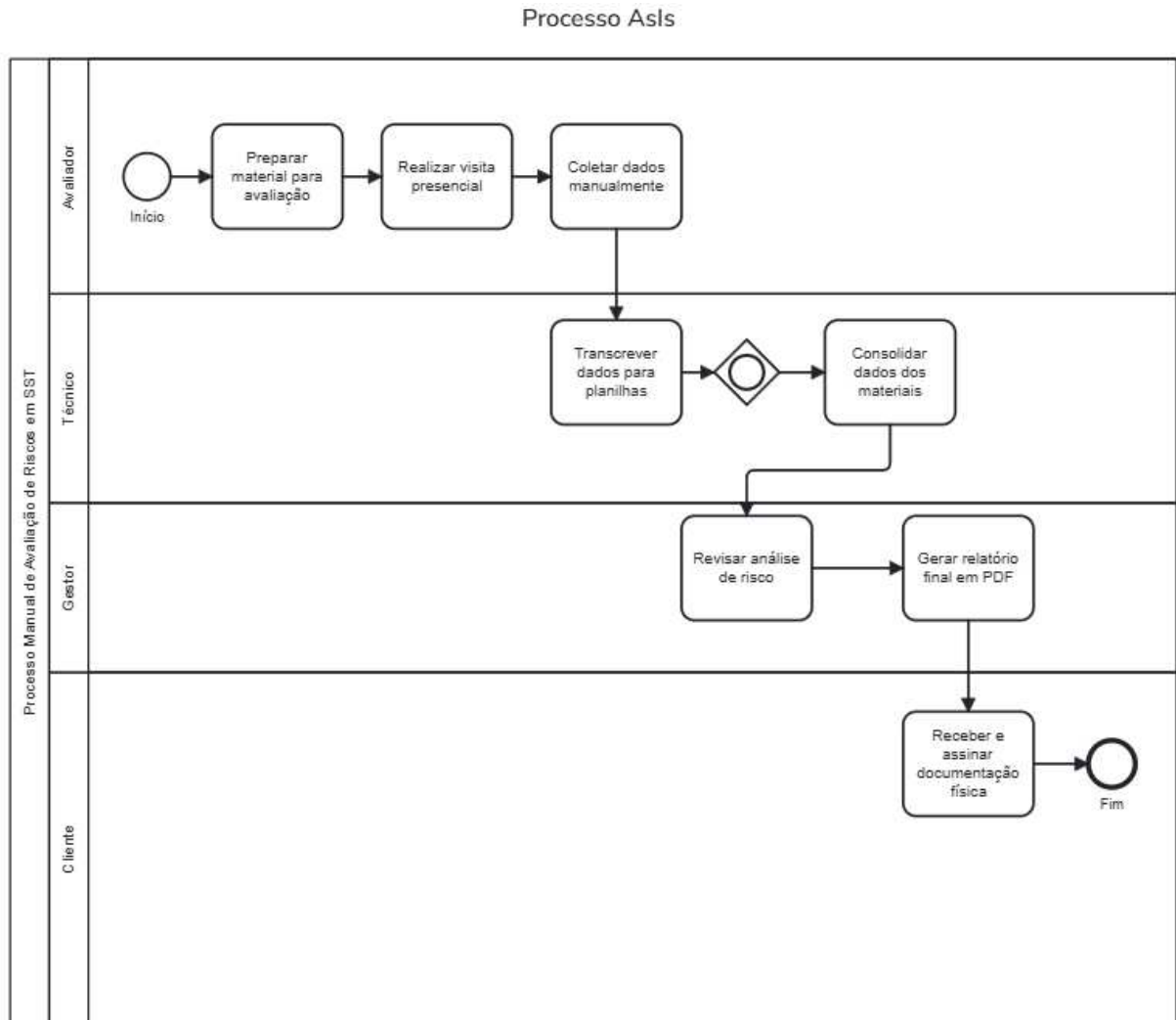
5.1 Caracterização do Contexto do Estudo de Caso

A empresa-alvo, especializada em engenharia civil e saúde e segurança do trabalho, enfrentava desafios na gestão de riscos ocupacionais. O processo *AsIs*, ilustrado na Figura 5, era manual, baseado em inspeções presenciais, coleta de dados em planilhas e geração de relatórios em PDF sem padronização.

Esse processo gerava ineficiências, como atrasos de até 7 dias na consolidação de dados, perda de informações durante transcrições manuais, ausência de automação na coleta e análise de dados, e falta de um método padronizado para relatórios mensais ou inclusão de assinaturas digitais, essenciais para conformidade e rastreabilidade. A empresa demandava uma solução que automatizasse a coleta de dados, gerasse relatórios PDF com funcionalidade *offline* e incorporasse assinaturas digitais para atender às exigências regulatórias, promovendo eficiência e escalabilidade.

A pesquisadora atuou como participante ativa no processo de desenvolvimento, sendo responsável pela modelagem, implementação e avaliação da solução, caracterizando o estudo como um estudo de caso participativo de natureza avaliativa.

Figura 5 – Diagrama de transformação dos processos para maior eficiência: Modelagem do Estágio *AsIs*.



Fonte: Elaborado pela autora (2025).

5.2 Aplicação Sistemática do *Framework* EasInnova

O *framework* foi utilizado para orientar o desenvolvimento do Alpha Checklist. A utilização do *framework* EasInnova sugere uma separação entre lógica de negócios e implementação técnica, facilitando o desenvolvimento rápido e alinhado.

- **Estágio *AsIs* (Análise do Cenário Atual):** No estágio *As-Is*, buscou-se compreender e documentar o funcionamento atual do processo de avaliação de riscos.
 - **CIM:** Identificação dos problemas do processo manual (atrasos, perda de dados) e das oportunidades de automação.
 - **PIM:** Modelagem do fluxo de trabalho existente com um diagrama BPMN (Figura 5) para visualizar as ineficiências.
 - **PSM:** Análise da arquitetura tecnológica atual, baseada em planilhas.

Essa etapa estabeleceu uma base sólida para compreender o estado atual do processo e subsidiar as decisões de transformação.

- **Estágio *Transformation* (Definição da Estratégia):** O estágio Transformation teve como objetivo definir a estratégia de inovação que permitisse superar as limitações identificadas no cenário atual.
 - **CIM:** Definição da solução candidata: um aplicativo móvel com funcionalidade *offline*, geração de relatórios e assinatura digital.
 - **PIM:** Desenho do novo processo automatizado em um diagrama BPMN (Figura 6) e detalhamento dos requisitos técnicos.
 - **PSM:** Seleção estratégica da plataforma *low-code* (FlutterFlow) e das tecnologias de suporte (Dart, SQLite).

Essa etapa atuou como ponte entre a compreensão do problema e a implementação da solução tecnológica.

- **Estágio *ToBe* (Implementação do Cenário Futuro):** No estágio ToBe as mudanças propostas foram consolidadas e implementadas.
 - **CIM:** Validação do novo fluxo de trabalho conceitual com os *stakeholders*.
 - **PIM:** Finalização dos modelos técnicos detalhados para servirem como "plano de construção" da aplicação.
 - **PSM:** Implementação da solução no FlutterFlow: desenvolvimento das interfaces, configuração do banco de dados local e codificação das regras de negócio.

Assim, o desenvolvedor conseguiu traduzir sistematicamente os requisitos de negócios em uma aplicação funcional, mantendo uma distinção clara entre lógica de negócios e implementação técnica ao longo do processo. Ao final desse estágio, obteve-se uma aplicação funcional, alinhada aos modelos definidos nas etapas anteriores.

5.3 Processo de Desenvolvimento e Planejamento Temporal

Utilizando o *framework* EasInnova, o desenvolvimento do aplicativo evoluiu do mapeamento *AsIs* até a implementação *ToBe* em apenas oito semanas. O cronograma do projeto foi documentado por meio de um diagrama de Gantt (Figura 7), evidenciando a aplicação prática do *framework* no gerenciamento do desenvolvimento. O uso da metodologia garantiu uma transição eficiente entre as etapas do projeto, otimizando recursos e tempo.

No estágio *AsIs* (semanas 1-2), o foco foi documentar o processo manual de avaliação

de riscos. No nível CIM, uma análise OPAAL (Quadro 2) identificou gargalos, como atrasos na consolidação de relatórios e perda de dados durante transcrições manuais. O diagrama *BPMN* (Figura 5) mapeou o fluxo existente, destacando etapas críticas como coleta presencial de dados e dependência de planilhas. No nível PIM, modelos de entidades e relacionamentos representaram os dados envolvidos, enquanto o nível PSM identificou o acúmulo de papéis e dependências de sistemas de terceiros que não funcionavam de forma *offline*, sendo importante na coleta de dados em campo.

Quadro 2 – Análise OPAAL

Categoria	Descrição
Objetos	relatórios de risco, planilhas Excel, dados brutos, papéis acumulados e bancos de dados
Processos	Preparar material, visita presencial, coletar dados, transcrever dados, consolidar dados, Revisar análise, gerar relatório PDF e receber assinatura
Atores	Equipe de campo, técnicos, clientes e sistemas de terceiros
Atributos	Tempo de consolidação, precisão dos dados, conectividade, dependência, e acúmulo de papéis
Ligações	Dados brutos alimentam planilhas, planilhas conectam dados aos relatórios, relatórios dependem de dados presenciais e assinatura valida relatório

Fonte: Elaborado pela autora (2025).

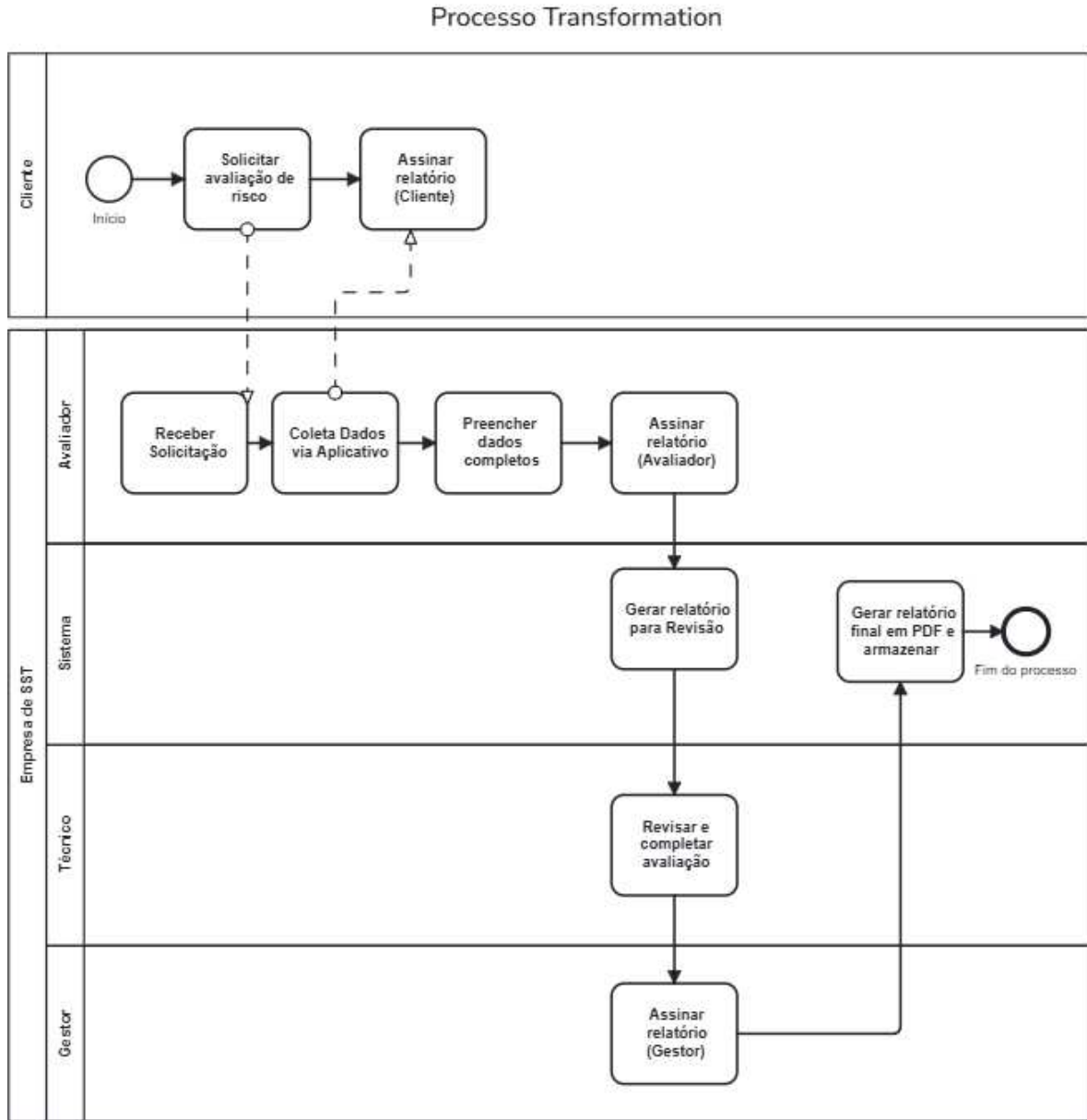
No estágio *Transformation* (semanas 3-5), foi definida a estratégia do novo sistema, com foco na superação das ineficiências do processo manual de avaliações de risco. No nível CIM, regras de negócio foram modeladas e categorizadas como detalhado no Quadro 3, priorizando funcionalidades como operação *offline*, geração automatizada de relatórios em PDF e aderência a normas de SST.

Quadro 3 – Regras de Negócio Identificadas no Estágio *Transformation*

ID	Descrição	Categoria
RN01	O sistema deve funcionar completamente offline em áreas remotas	Funcionalidade
RN02	Os relatórios devem ser gerados automaticamente em formato PDF	Desempenho
RN03	Todos os relatórios devem incorporar assinaturas digitais validadas para atender às exigências regulatórias	Conformidade
RN04	O tempo total entre coleta de dados e geração do relatório final não deve exceder 2 dias	Desempenho
RN05	Os dados coletados precisam seguir rigorosamente as normas de Saúde e Segurança do Trabalho vigentes	Conformidade
RN06	Deve ser possível anexar evidências fotográficas às avaliações de risco	Funcionalidade
RN07	As avaliações de risco devem seguir uma metodologia padronizada	Conformidade
RN08	O sistema deve implementar diferentes níveis de acesso	Segurança

Fonte: Elaborado pela autora (2025).

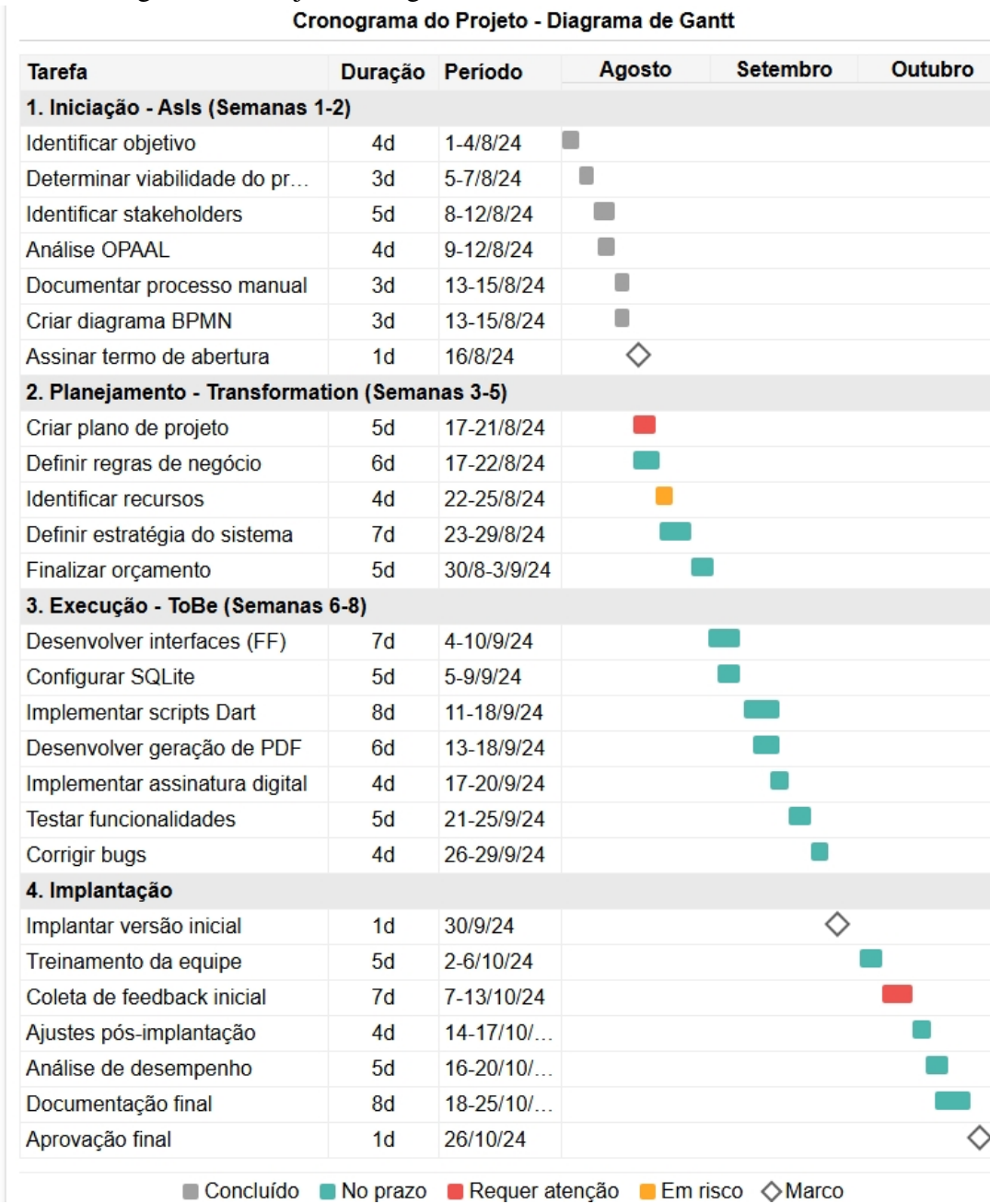
Figura 6 – Diagrama de transformação dos processos para maior eficiência: Modelagem do Estágio *Transformation*



Fonte: Elaborado pela autora (2025).

No estágio *ToBe* (semanas 6-8), os modelos conceituais foram convertidos em uma aplicação funcional usando *FlutterFlow*. No nível PSM, interfaces de usuário foram construídas com componentes *drag-and-drop*, integradas ao *SQLite* para armazenamento local. A implementação no nível PSM com o *FlutterFlow* permitiu uma materialização ágil dos requisitos definidos, enquanto a utilização do *SQLite* assegurou a funcionalidade *offline*, crucial para as inspeções em campo (RN01).

Figura 7 – Cronograma do Projeto - Diagrama de Gantt



Fonte: Elaborado pela autora (2025).

Ademais, essa abordagem integrada no *PSM* foi crucial para a eficiente concretização da aplicação. A plataforma de desenvolvimento visual proporcionou a agilidade necessária para a montagem da estrutura da interface, enquanto a solução de armazenamento de dados local forneceu a autonomia e a resiliência operacional em campo.

5.4 Ferramentas e Tecnologias Utilizadas

Para a implementação do Alpha Checklist, foram empregadas as seguintes ferramentas e tecnologias:

- **FlutterFlow:** Plataforma *low-code* utilizada para o desenvolvimento da interface do usuário e lógica de navegação.
- **SQLite:** Banco de dados local para armazenamento de dados coletados em campo, garantindo funcionalidade *offline*.
- **Dart:** Linguagem de programação usada para implementar lógicas de negócio complexas.
- **BPMN.io:** Ferramenta de modelagem para criar diagramas BPMN no estágio *AsIs*.
- **GitHub:** Plataforma de controle de versão para gerenciar o código-fonte.

5.5 Procedimentos de Coleta e Análise de Dados

Para a validação da eficácia do *framework* EasInnova e da solução Alpha Checklist foi realizada por meio de procedimentos quantitativos e qualitativos.

A coleta de dados abrangeu um período de operação contínua, totalizando uma amostra de $N = 242$ relatórios. A análise focou em três indicadores chave de desempenho (KPIs) alinhados às Regras de Negócio (RNs):

- **Eficiência Temporal (Cycle Time):** Medição do tempo decorrido entre o início da coleta e a disponibilização da informação.
- **Taxa de Finalização Imediata (First-Time Right):** Porcentagem de relatórios concluídos e validados em um único ciclo de trabalho.
- **Volume de Produção:** Quantificação do uso real da ferramenta para demonstrar robustez.

Além da análise de logs, adotou-se uma abordagem qualitativa complementar para validar a aceitação tecnológica (modelo TAM) junto aos usuários finais. A análise qualitativa teve caráter exploratório, buscando compreender a percepção dos usuários quanto aos benefícios e limitações da abordagem adotada.

Os participantes do estudo responderam um TCLE (Termo de consentimento livre e esclarecido) que trata dos aspectos éticos envolvidos na pesquisa.

5.6 Alinhamento entre Objetivos, Métodos e Evidências.

A tabela 1 evidencia o alinhamento direto entre os objetivos definidos para o trabalho, os métodos adotados e as evidências empíricas produzidas ao longo do estudo de caso, reforçando a coerência metodológica e a adequação da abordagem escolhida para avaliar a eficácia do framework EasInnova.

Tabela 1 – Síntese dos Objetivos, Métodos e Evidências Produzidas

Objetivo	Método / Procedimento Metodológico	Evidências Produzidas
Avaliar a eficácia do <i>framework</i> EasInnova como uma abordagem integrada de MDE e plataformas <i>low-code</i> no desenvolvimento de soluções corporativas	Estudo de caso avaliativo, com abordagem quali-quantitativa, aplicando integralmente os estágios (<i>As-Is</i> , <i>Transformation</i> , <i>To-Be</i>) e camadas (CIM, PIM, PSM) do <i>framework</i> EasInnova	Resultados consolidados nos Capítulos 6 e 7, incluindo métricas de processo, análise de rastreabilidade, desempenho técnico e avaliação de aceitação (TAM)
Modelar um processo de negócio real utilizando as etapas do <i>framework</i> EasInnova para traduzir os requisitos em artefatos nas camadas CIM e PIM	Modelagem do processo <i>As-Is</i> por meio de análise OPAAL e diagramas BPMN; redesenho do processo no estágio <i>Transformation</i> com BPMN e definição de regras de negócio	Diagramas BPMN (<i>As-Is</i> e <i>Transformation</i>), análise OPAAL, identificação de requisitos e regras de negócio documentadas
Implementar a solução proposta (Alpha Checklist) em uma plataforma <i>low-code</i> , a partir da transformação do modelo PIM em um PSM	Implementação da aplicação móvel utilizando FlutterFlow, SQLite e scripts em Dart, seguindo os modelos definidos nas camadas CIM e PIM	Aplicação funcional Alpha Checklist, interfaces implementadas, banco de dados configurado, geração automática de relatórios e suporte <i>offline</i>
Avaliar os resultados da aplicação do <i>framework</i> considerando métricas de processo e a percepção dos envolvidos	Coleta e análise quantitativa de logs da aplicação (N = 242 relatórios) e aplicação de questionário baseado no TAM com usuários finais	Métricas de tempo de ciclo, taxa de finalização, volume de uso; resultados do TAM (utilidade percebida, facilidade de uso e intenção de uso)
Analisar a rastreabilidade e a consistência entre requisitos de negócio, modelos gerados e implementação final	Construção de matriz de rastreabilidade entre regras de negócio (CIM) e funcionalidades implementadas (PSM)	Matriz de rastreabilidade CIM × PSM, evidenciando a materialização dos requisitos estratégicos na solução final

Fonte: Elaborado pela autora (2025).

6 RESULTADOS

Este capítulo apresenta os dados obtidos a partir da operação assistida do Alpha Checklist, consolidados para responder aos objetivos de avaliação de processo e produto definidos neste trabalho.

6.1 Rastreabilidade e Consistência dos Modelos

Para verificar a fidelidade da solução aos objetivos estratégicos (Objetivo Específico 4), realizou-se o mapeamento entre os requisitos definidos na camada CIM e a implementação final na camada PSM.

Quadro 4 – Matriz de Rastreabilidade: Requisitos de Negócio vs. Solução

ID	Requisito (CIM)	Status PSM
RN01	Funcionalidade Offline	Atendido (SQLite)
RN03	Assinatura Digital	Atendido (Widget)
RN04	Tempo de Ciclo Reduzido	Atendido (Logs)
RN05	Validação de Normas SST	Atendido (Lógica)

Fonte: Elaborado pela autora (2025).

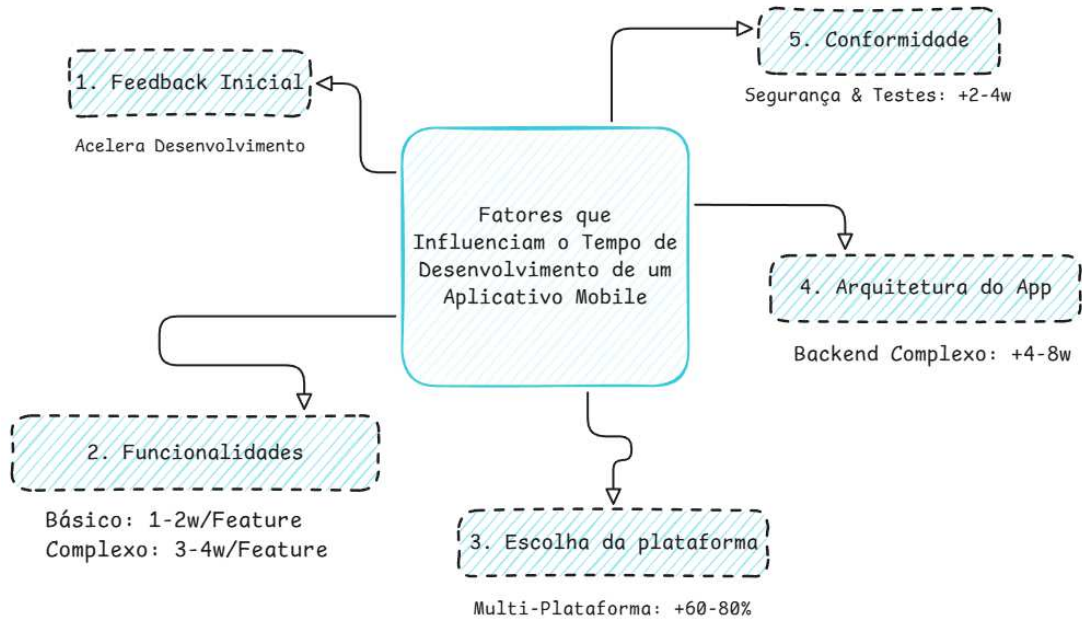
O Quadro 4 demonstra que as regras de negócio críticas modeladas no estágio *Transformation* foram materializadas na aplicação final. Esses resultados indicam que o *framework* EasInnova favorece a manutenção da rastreabilidade entre requisitos de negócio e implementação técnica ao longo do ciclo de desenvolvimento.

6.2 Análise de Produtividade e Métricas de Processo

Para mensurar o impacto do framework EasInnova na agilidade de entrega, utilizou-se a métrica de *Time-to-Market*, comparando o ciclo real do projeto com benchmarks de mercado fornecidos por empresas especializadas em desenvolvimento de software.

Segundo dados da Leanware (2024), o cronograma típico para o desenvolvimento de um aplicativo móvel de média complexidade (MVP) varia entre 3 a 5 meses (aproximadamente 12 a 20 semanas). O processo tradicional é fragmentado em etapas e fatores, como mostra a Figura 9, suas etapas são sequenciais de Descoberta, Design, Desenvolvimento e Testes, que consomem recursos significativos de coordenação.

Figura 8 – Fases do desenvolvimento de aplicativo mobile.



Fonte: Adaptado de Leanware (2024).

Em contraste, a aplicação da abordagem EasInnova (MDE + *low-code*) permitiu a conclusão da aplicação em apenas 8 semanas, posicionando o projeto abaixo do limite mínimo estimado pelo mercado.

A Tabela 2 apresenta o comparativo detalhado por etapas, alinhando o ciclo de vida da LeanWare com as fases do EasInnova.

Tabela 2 – Comparativo de Ciclo de Vida: Padrão de Mercado vs. EasInnova

Fase de Mercado (LeanWare)	Fase EasInnova	Tempo Mercado	Tempo Real
Discovery & Strategy	As-Is / CIM	2 – 4 semanas	1 semana
Design (UI/UX)	Transformation / PIM	3 – 4 semanas	2 semanas
Desenvolvimento	To-Be / PSM	8 – 10 semanas	3 semanas
Teste & Deploy	Validação	2 – 3 semanas	2 semanas
Tempo Total	–	15 – 21 Semanas	8 Semanas

Fonte: Dados de mercado baseados em LeanWare (2024) para MVPs de média complexidade.

A análise da Tabela 2 evidencia uma redução de aproximadamente 50% a 60% no tempo total de entrega em relação à média de mercado. O ganho mais expressivo ocorre na fase de Desenvolvimento (referente à camada PSM), onde o uso do FlutterFlow reduziu o esforço de codificação de ≈ 9 semanas (média de mercado) para apenas 3 semanas.

Isso valida a hipótese de que a integração de modelos visuais (PIM) com plataformas de desenvolvimento acelerado (*low-code*) elimina a latência da programação manual tradicional.

Esses resultados sugerem que a redução do tempo de desenvolvimento não está

associada exclusivamente ao uso da plataforma low-code, mas à integração desta com modelos bem definidos nas camadas CIM e PIM, conforme proposto pelo *framework* EasInnova.

6.3 Análise de Desempenho Técnico (Telemetria)

Para validar a robustez do artefato gerado pelo Flutterflow, realizou-se um perfilamento de baixo nível utilizando a ferramenta *Android Visual Studio* e *Prefetto* para captura do *trace*. A captura dos dados (*trace*) ocorreu durante o cenário de maior estresse do aplicativo: a consolidação de dados e exportação do relatório técnico em PDF.

6.3.1 Perfil de Processamento e CPU

A análise do consumo de tempo de CPU por processo (Figura 9) revela uma distribuição com forte assimetria positiva. Enquanto a maioria dos processos de segundo plano do Android opera em intervalos curtos (0-5s), o processo com `mycompany.alphachecklist` destaca-se como o principal consumidor, com aproximadamente 35 segundos de processamento contínuo.

Análise Técnica: Este pico de 35s é perfeitamente condizente com a execução da lógica de negócio complexa, onde o motor do Flutter processa a árvore de componentes gerada para consolidar os dados da vistoria e gerar o arquivo PDF.

6.3.2 Gestão de Memória e Estado da Aplicação

O consumo de memória RAM atingiu um pico de aproximadamente 663 MB, valor identificado como um *outlier* no gráfico de dispersão (*scatter plot*) de processos do sistema.

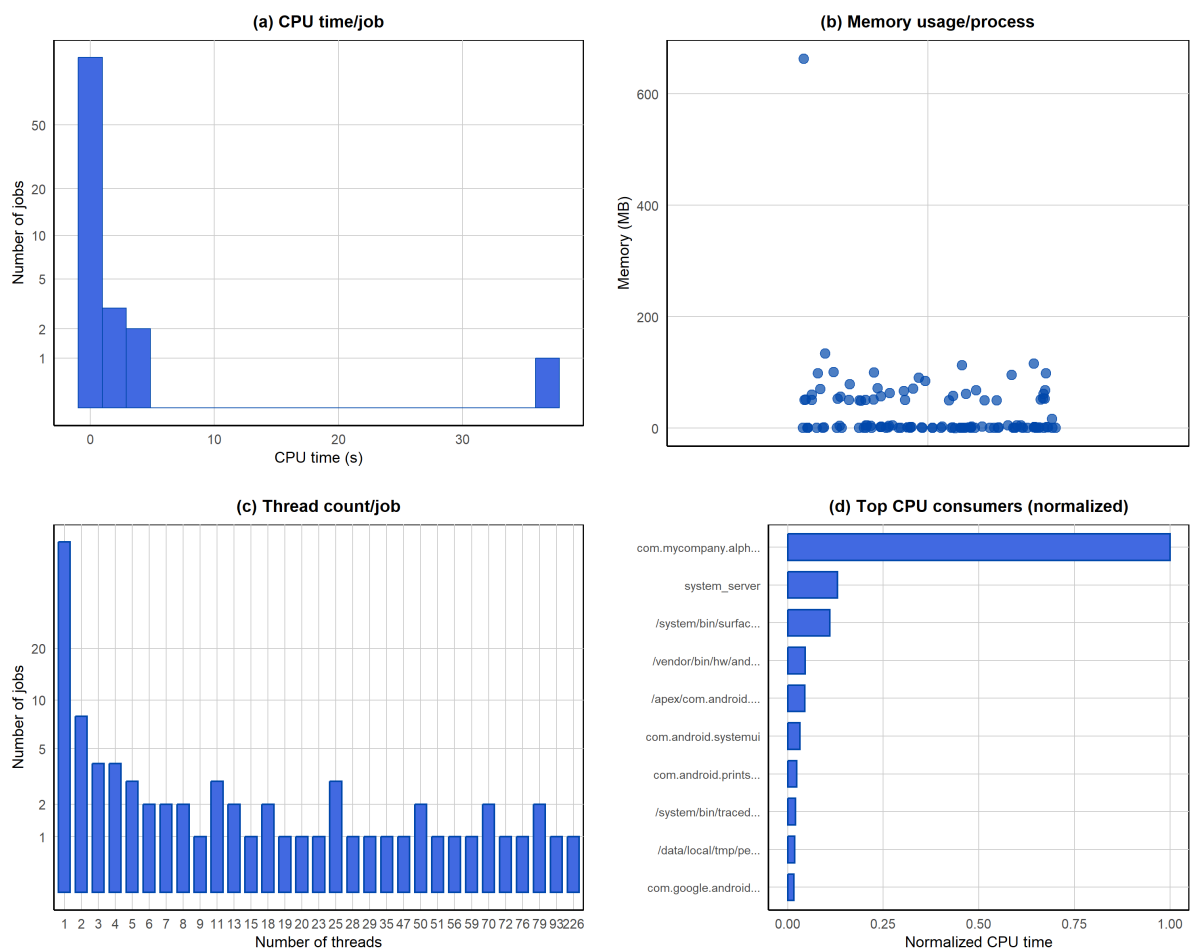
- **Justificativa Técnica:** Este volume de alocação de memória correlaciona-se diretamente com a arquitetura de dados do *Alpha Checklist*. Para viabilizar a geração do relatório PDF de forma totalmente *offline*, o sistema armazena o conjunto integral de dados da vistoria no *App State*.
- **Análise de Desempenho:** O pico de memória ocorre no momento da exportação. Embora elevado, o gerenciamento de memória mostrou-se eficaz, visto que não houve o encerramento forçado do processo pelo sistema operacional (*Out of Memory - OOM*), comprovando que a plataforma *low-code* gerou um código capaz de lidar com grandes volumes de dados em cache.

6.3.3 Concorrência e Multithreading

O histograma de contagem de threads (*Thread count/job*) indicou que o aplicativo operou com um número de threads significativamente superior à média do sistema (superior a 50 threads em picos).

- **Interpretação:** Esse paralelismo é resultado da engine do Flutter, que isola a thread de interface (*UI Thread*) das threads de processamento de dados e I/O. Isso explica por que, apesar do alto consumo de CPU (35s) e Memória (663 MB), a percepção de facilidade de uso e fluidez (validada no teste TAM) permaneceu positiva, sem travamentos visíveis na interface.

Figura 9 – Resultados da Telemetria: (a) CPU Time, (b) Memory Usage, (c) Thread Count, (d) Top Consumers.



Fonte: Elaborado pela autora via Perfetto Trace (2026).

A análise de telemetria evidencia que o uso de plataformas low-code não compromete o desempenho técnico da aplicação, mesmo em operações críticas como a consolidação de dados

e a geração de relatórios em PDF. Apesar dos picos pontuais de consumo de CPU e memória, o sistema manteve estabilidade e fluidez na interação com o usuário. Esses resultados indicam que, quando associada a uma modelagem estruturada nas camadas CIM e PIM, conforme proposto pelo framework EasInnova, a abordagem low-code é capaz de sustentar soluções tecnicamente robustas. Assim, a análise de desempenho reforça a viabilidade do EasInnova em contextos corporativos críticos, nos quais confiabilidade e desempenho são requisitos essenciais.

6.4 Avaliação da Percepção dos Usuários (TAM)

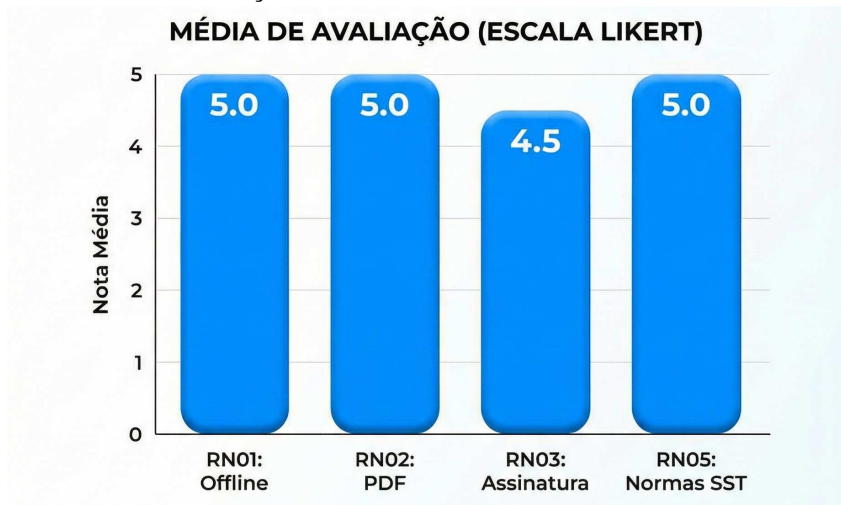
Para avaliar a aceitação da tecnologia, utilizou-se como base o Modelo de Aceitação de Tecnologia (*Technology Acceptance Model* - TAM), adaptado para medir a Utilidade Percebida, a Facilidade de Uso e a Intenção de Uso. A avaliação foi realizada via Google Forms e respondidas por dois participantes (N=2), o avaliador que realiza o trabalho de campo e pelo gestor.

6.4.1 Utilidade Percebida (UP)

A utilidade percebida refere-se ao quanto o usuário acredita que o uso do aplicativo melhora seu desempenho profissional. Os dados coletados e demonstrados na Figura 10 mostram um alto índice de satisfação:

- **Eficiência e Redução de Retrabalho:** 100% dos respondentes atribuíram nota máxima (5) para a agilidade na finalização das vistorias e para a eliminação de retrabalho, como a transcrição posterior de dados.
- **Conformidade com Requisitos:** Os Requisitos de Negócio (RNs), como o funcionamento *offline* (RN01) e a geração automática de relatórios em PDF (RN02), foram validados com o nível "Concordo Totalmente".
- **Impacto no Fluxo de Trabalho:** Quanto aos benefícios percebidos, os usuários destacaram a autonomia e a eliminação de tarefas burocráticas. O Respondente 1 da pesquisa descreveu como: "Praticidade, não preciso ficar revisando várias vezes para consolidar os dados, atualmente o técnico consegue ter mais autonomia, o app já coleta a minha assinatura então não preciso ficar assinando várias vezes, uma mão na roda"

Figura 10 – Resultados da Avaliação TAM - Dimensões de Utilidade.



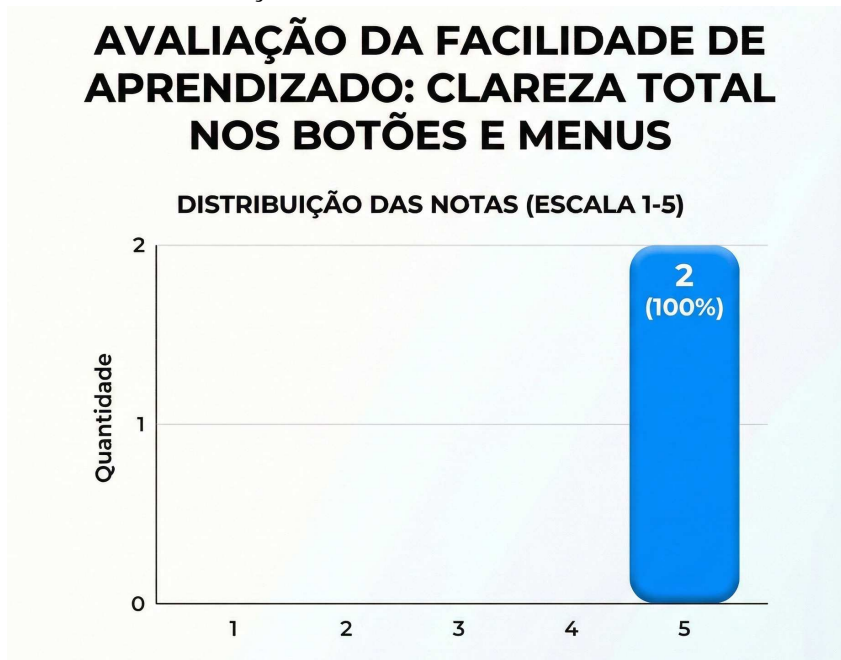
Fonte: Elaborado pela autora (2025).

6.4.2 Facilidade de Uso Percebida (FUP)

Este tópico avalia o esforço necessário para aprender e operar o sistema.

- **Curva de Aprendizado:** Ambos os usuários atribuíram nota 5 para a facilidade de aprender a utilizar o aplicativo, como mostra na Figura 11, ressaltando que botões e menus são claros e intuitivos.
- **Experiência do Usuário:** O feedback dos usuários apresentou-se como uma ótima solução adaptada às necessidades da área, o único ponto negativo foi a adaptação dos clientes à assinatura digital, onde eles precisam ainda se adaptarem, algo que no papel - como já estão acostumados - é mais fácil para eles.

Figura 11 – Resultados da Avaliação TAM - Dimensões de Facilidade.



Fonte: Elaborado pela autora (2025).

6.4.3 Intenção de Uso e Autonomia do Usuário

A validação final confirmou que a abordagem integrada de MDE e *low-code* resultou em uma ferramenta com alta intenção de adoção e, principalmente, potencial de evolução contínua pela própria organização.

- **Preferência e Eficácia:** Todos os usuários afirmaram preferir o uso do aplicativo em detrimento do método anterior (manual ou planilhas), destacando a superioridade da solução digital frente aos processos legados.
- **Capacidade de Evolução (Citizen Development):** Um diferencial crítico identificado na avaliação qualitativa foi a percepção de autonomia técnica. O Respondente 2 ressaltou a facilidade de intervir na solução caso surjam novas necessidades de negócio, afirmando afirmando “consigo também acessar facilmente a base do app no flutterflow e adicionar algum outro pela facilidade da plataforma, gerar um backup no app e conseguir gerar a 'build' do app no flutterflow” (RESPONDENTE 2, 2026).

Essa observação é fundamental, pois comprova que o uso do framework *EasInnova* democratiza o desenvolvimento, permitindo que especialistas de domínio - e não apenas desenvolvedores especializados em mobile - possam realizar pequenas modificações e manutenções ágeis.

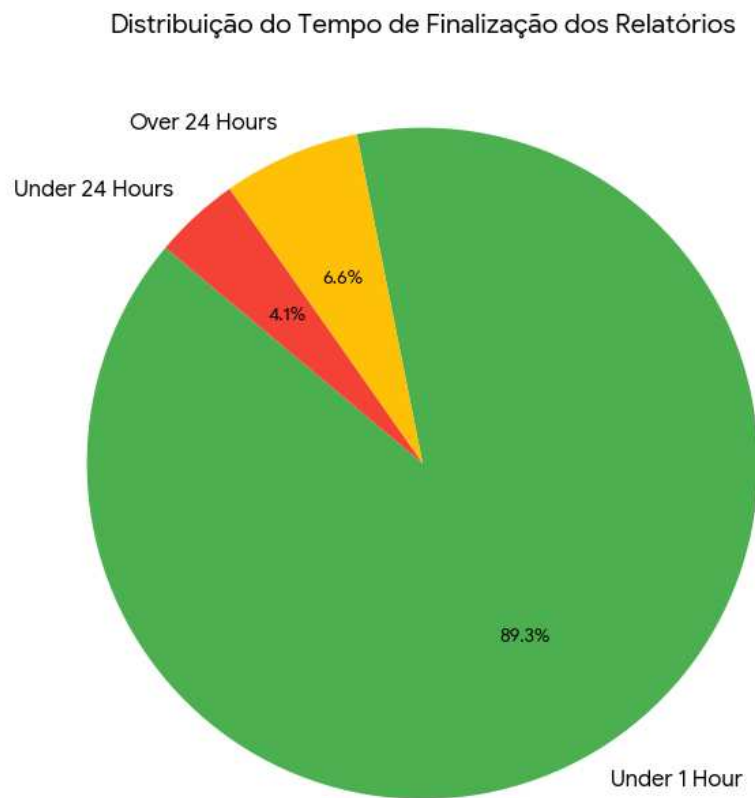
- **Observações de Desempenho:** Como ponto de melhoria para versões futuras, notou-se

que em dispositivos com *hardware* limitado o aplicativo apresentou lentidão pontual na geração de múltiplos agentes, servindo como insumo técnico para otimizações na camada de persistência local.

6.5 Análise de Desempenho Operacional (Logs)

A mineração de dados na base de produção (242 relatórios) revelou a eficiência da solução em campo.

Figura 12 – Distribuição do Tempo de Finalização dos Relatórios



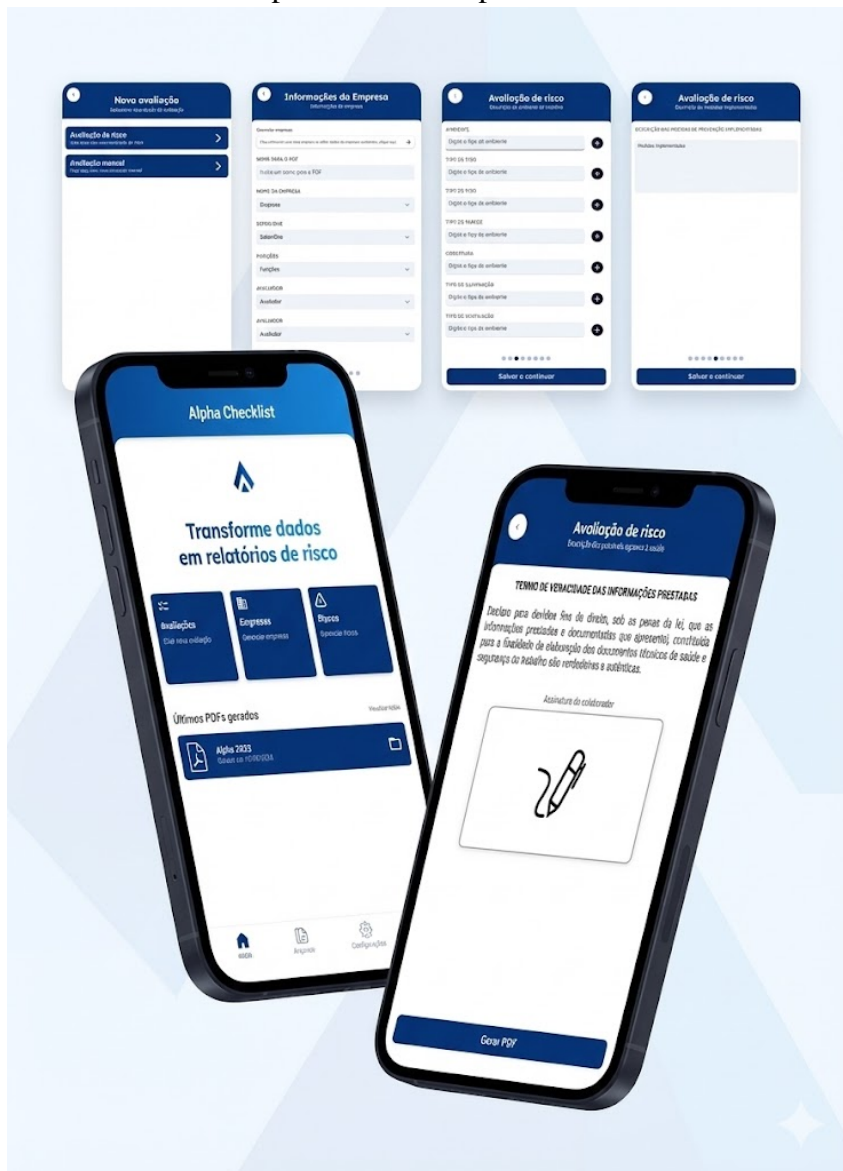
Fonte: Elaborado pela autora (2025).

Observa-se que 89,3% dos relatórios são finalizados em menos de 1 hora, comprovando a eficácia operacional.

6.6 Demonstração das Interfaces do Aplicativo (Artefato Final)

Nesta seção, apresentam-se as interfaces do aplicativo *Alpha Checklist*, que representam a concretização das transformações de modelo (CIM → PIM → PSM) propostas pelo framework *EasInnova*. As telas foram desenvolvidas utilizando a plataforma *FlutterFlow*, focando na usabilidade para o usuário em campo.

Figura 13 – Screenshots do Protótipo Funcional Alpha Checklist.



Fonte: Elaborado pela autora (2025).

A Figura 13 ilustra a tela de entrada, onde a lógica de negócio (CIM) define a filtragem por unidade e também exibe o módulo de assinatura, ponto crítico identificado na análise inicial para construção do aplicativo.

De forma geral, os resultados obtidos indicam que a aplicação do framework EasInnova contribuiu para a redução do tempo de desenvolvimento, para a manutenção da rastreabilidade entre requisitos e implementação e para a aceitação positiva da solução pelos usuários finais. Esses achados fornecem evidências empíricas da eficácia da abordagem integrada de MDE e low-code proposta, sendo discutidos de forma mais aprofundada no capítulo seguinte.

7 DISCUSSÃO

A análise integrada dos dados quantitativos e qualitativos permite validar a eficácia do framework *EasInnova*. A discussão fundamenta-se na triangulação entre a percepção de utilidade, a eficiência técnica observada nos *traces* de sistema e os paradigmas de arquitetura moderna discutidos na fundamentação teórica.

7.1 O Equilíbrio entre Abstração e Performance

Os resultados da telemetria, que indicaram picos de CPU (35s) e memória (663 MB), não comprometeram a experiência do usuário, que atribuiu nota máxima à facilidade de uso. Este fenômeno explica-se pela arquitetura gerada via *FlutterFlow*, que isola processos pesados em threads secundárias. Para a Engenharia de Software, isso comprova que o uso de *low-code*, quando guiado por uma modelagem robusta (MDE), é capaz de produzir sistemas corporativos que suportam cargas de trabalho reais sem sacrificar a usabilidade. Diante disso, observa-se que as decisões de modelagem adotadas permitiram mitigar os impactos normalmente associados à abstração, preservando o comportamento esperado da solução em cenários de uso intensivo.

7.2 Modularidade e a Arquitetura Componível

A implementação PSM do *Alpha Checklist* demonstrou na prática os princípios da Arquitetura Componível e das PBCs. Os módulos de Assinatura Digital e o Motor de Geração de PDF foram desenvolvidos de forma independente e encapsulada. Essa modularidade permite que tais capacidades de negócio sejam reutilizadas em outros fluxos de inspeção da organização com mínimo esforço de reconfiguração, garantindo a agilidade e a resiliência operacional preconizadas pelo Gartner. Nesse sentido, a estrutura da solução evidencia que a separação entre responsabilidades conceituais e técnicas favorece a evolução do sistema sem a necessidade de intervenções extensivas na implementação.

7.3 Trade-offs da Persistência Offline e Alinhamento de Negócio

Um ponto crítico discutido é o uso intensivo do *App State*. A decisão de projeto de manter os dados da vistoria em cache local para permitir a geração de PDF *offline* (Requisito RN01) gerou o *outlier* de memória detectado.

Contudo, este *trade-off* mostrou-se acertado, pois a "autonomia" e a "eliminação de retrabalho" foram os benefícios mais citados pelos técnicos no TAM. Conclui-se que o alinhamento entre Negócio e TI foi preservado: a necessidade estratégica de campo (CIM) sobrepôs-se à economia estrita de recursos de hardware (PSM), resultando em uma solução fiel aos objetivos de negócio da empresa. Assim, os compromissos assumidos no projeto refletem escolhas conscientes entre requisitos técnicos e operacionais, coerentes com as prioridades estabelecidas ao longo do processo de transformação.

7.4 Citizen Development e Democratização do Desenvolvimento

A validação qualitativa trouxe evidências do potencial para o Citizen Development. O feedback de que é possível "acessar facilmente a base do app no FlutterFlow e adicionar outros itens" confirma que o framework *EasInnova* reduz a barreira técnica de entrada. Ao permitir que especialistas de domínio participem ativamente da evolução do software, a abordagem reduz a dependência histórica de departamentos de TI centralizados e promove uma transformação digital mais inclusiva e colaborativa. Dessa forma, a ampliação do papel dos usuários de negócio ocorre de maneira controlada, mantendo-se o equilíbrio entre autonomia, consistência e responsabilidade técnica.

De forma geral, a discussão dos resultados evidencia que a integração entre MDE e plataformas low-code, operacionalizada pelo framework *EasInnova*, contribui para reduzir o distanciamento entre modelos conceituais e implementações técnicas, sem comprometer desempenho ou qualidade da solução. Os achados reforçam que a eficácia do *EasInnova* não se limita à agilidade de desenvolvimento, mas se estende à manutenção da rastreabilidade, ao suporte à modularidade e à viabilidade de adoção em contextos corporativos críticos. Essas evidências consolidam o framework como uma abordagem metodológica adequada para apoiar iniciativas de transformação digital orientadas por processos e modelos de negócio.

8 CONCLUSÃO

Este trabalho teve como objetivo central avaliar a eficácia do framework EasInnova como uma abordagem integrada entre a Engenharia Dirigida por Modelos e as plataformas low-code. Através do desenvolvimento e validação do aplicativo Alpha Checklist, foi possível observar como essa sinergia acelera a inovação em sistemas corporativos, mantendo o alinhamento rigoroso entre as necessidades de negócio e a implementação técnica. A pesquisa demonstrou que a utilização das camadas de abstração da MDA serviu como uma ponte de tradução eficiente, permitindo que os objetivos específicos fossem plenamente alcançados. Observou-se uma redução de aproximadamente 60% no time-to-market em comparação aos benchmarks de mercado, o que validou a eficiência do FlutterFlow como camada de execução PSM. No âmbito técnico, o perfilamento de baixo nível via Perfetto comprovou que, apesar do consumo elevado de memória de 663 MB durante a geração de PDFs, a arquitetura multithreading do Flutter manteve a fluidez do sistema e a estabilidade necessária para o ambiente de produção. Além disso, os altos índices de Utilidade e Facilidade de Uso obtidos no modelo TAM confirmaram que o sistema agrega valor real ao dia a dia do técnico de campo, eliminando retrabalhos e aumentando a autonomia operacional.

Para além da entrega de um artefato funcional, este trabalho contribuiu para a validação de paradigmas modernos da computação, como a Arquitetura Componível. A implementação modular do sistema comprovou que funcionalidades críticas podem ser tratadas como Packaged Business Capabilities, garantindo a modularidade e o reuso de componentes. Outro ponto relevante foi a observação do potencial para o Citizen Development, uma vez que a transparência visual da plataforma low-code permitiu que especialistas de domínio participassem da validação e sugerissem evoluções no design, estreitando o abismo histórico entre o departamento de TI e as áreas finalísticas da organização. Contudo, apesar dos resultados positivos, o estudo apresentou limitações que devem ser consideradas, como a amostra restrita de usuários para a validação qualitativa e o impacto de hardware em dispositivos de baixa performance durante a renderização de documentos pesados no App State.

Como continuidade desta pesquisa e para mitigar as limitações identificadas, sugerem-se trabalhos futuros focados na otimização de performance, especificamente investigando métodos de compressão de dados em memória para reduzir o pico de RAM detectado na telemetria. Também é recomendada a exploração de métodos de gestão de mudança digital, como o uso de biometria para reduzir as barreiras culturais identificadas na coleta de assinaturas digitais. Por

fim, vislumbra-se a integração de uma camada de automação de testes unitários ao framework EasInnova, gerados diretamente a partir dos modelos PIM para elevar a confiabilidade do código. Conclui-se, portanto, que a integração entre MDE e Low-code apresenta-se como uma evolução necessária para organizações que buscam resiliência e agilidade na era da transformação digital, permitindo que o software evolua na mesma velocidade em que o negócio se transforma.

REFERÊNCIAS

- AIMULTIPLE. **30+ Low-Code/ No-Code Statistics in 2025**. 2025. Online; accessed 21 jul. 2025. Disponível em: <<https://research.aimultiple.com/low-code-statistics/>>.
- ALFONSO, M. *et al.* Interoperability between low-code platforms: A better-based approach. In: **Proceedings of the International Conference on Software Engineering**. [s.n.], 2024. p. 56–67. Acesso em: 18 jul. 2025. Disponível em: <<https://arxiv.org/abs/2412.05075v1>>.
- BRAMBILLA, M.; CABOT, J.; WIMMER, M. **Model-Driven Software Engineering in Practice**. Morgan & Claypool, 2017. ISBN 9781627057080. Disponível em: <<https://doi.org/10.2200/S00751ED2V01Y201701SWE004>>.
- ButterCMS. **Explicação da arquitetura combinável: definição e benefícios**. 2023. Online; accessed 18 jul. 2025. Disponível em: <<https://buttercms.com/blog/what-is-composable-architecture/>>.
- Caspio. **Automating Workforce Compliance With Low Code**. 2026. Disponível em: <<https://www.caspio.com/customer-interview/automating-workforce-compliance-low-code/>>.
- Contentstack. **Navigating the challenges of implementing composable commerce**. 2023. Online; accessed 18 jul. 2025. Disponível em: <<https://www.contentstack.com/blog/composable/navigating-the-challenges-of-implementing-composable-commerce>>.
- DECERTO. **Cost Savings and ROI of Low-Code Platforms for Insurance Companies**. 2025. Online; accessed 21 jul. 2025. Disponível em: <<https://www.decerto.com/post/cost-savings-and-roi-of-low-code-platforms-for-insurance-companies>>.
- DEQODE. **Low-Code and No-Code: A Paradigm Shift in Application Development**. 2023. Online; accessed 21 jul. 2025. Disponível em: <<https://deqode.com/blog/2023/10/20/low-code-and-no-code-a-paradigm-shift-in-application-development/>>.
- FlutterFlow. **FlutterFlow: Plataforma de Desenvolvimento Visual de Aplicações Flutter**. 2023. <<https://flutterflow.io>>.
- FRANCE, R. B.; RUMPE, B. Model-driven development of complex software: A research roadmap. **2007 Future of Software Engineering**, IEEE Computer Society, p. 37–54, 2007.
- Gartner. **Gartner Keynote: The Future of Business Is Composable**. 2020. Online; accessed 21 jul. 2025. Disponível em: <<https://www.gartner.com/smarterwithgartner/gartner-keynote-the-future-of-business-is-composable>>.
- GARTNER. **Gartner Survey of Over 2,000 CIOs Reveals the Need for Enterprises to Embrace Business Composability in 2022**. 2021. Press release. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2021-10-18-gartner-survey-of-over-2000-cios-reveals-the-need-for-enterprises-to-embrace-business-composability-in-2022>>. Acesso em: 18 jul. 2025.
- Gigaspace. **Composable Architecture: Building Blocks for Business Acceleration**. 2024. Online; accessed 17 jul. 2025. Disponível em: <<https://www.gigaspace.com/blog/composable-architecture-building-blocks-for-business-acceleration>>.

IDEOSOFTWARE. **LOW-CODE Accelerates Global Digital Transformation / Digitization blog | Project-based Software Development**. 2023. Online; accessed 21 jul. 2025. Disponível em: <<https://www.ideosoftware.com/blog/low-code-accelerates-global-digital-transformation,218.html>>.

IVAS, I. Implementation of composable enterprise in an evolutionary way through holistic business—it delivery of business initiatives: Real industry use case. In: **Proceedings of the 26th International Conference on Enterprise Information Systems (ICEIS 2024)**. [S.l.]: SCITEPRESS – Science and Technology Publications, Lda., 2024. v. 1, p. 397–408.

KPMG. **Transforming Business Value Creation with Citizen Development**. 2025. Disponível em: <<https://kpmg.com/xx/en/home/insights/2025/01/citizen-development.html>>. Acesso em: 18 jul. 2025.

KRAUS, S.; BREIER, M.; DASÍ-RODRÍGUEZ, S.; SCHEDLITZKI, D.; GIONES, F.; CHI-HYUNG, L.; WOOD, E.; AL. et. Digital transformation: An overview of the current state of research. **International Journal of Innovation Management**, World Scientific, v. 25, n. 3, p. 2140001, 2021. Disponível em: <<https://doi.org/10.1142/S1363919621400011>>.

Leanware. **Quanto tempo leva para criar um aplicativo sob medida?** 2024. Disponível em: <<https://www.goappy.net/tempo-criar-aplicativo-sob-medida/>>. Acesso em: 21 jan. 2026.

MARTINS, G. M.; NORONHA, T.; BRAGA, A.; ROMANO, D. Business-it alignment and digital transformation: Setting a research agenda. In: **Proceedings of the International Conference on Information Systems Development (ISD)**. Valência: AIS Electronic Library (AISeL), 2021. Acesso em: 18 jul. 2025. Disponível em: <<https://aisel.aisnet.org/isd2014/proceedings2021/currenttopics/7/>>.

Mendix. **5 Benefits of Low-Code Application Development**. 2023. Online; accessed 21 jul. 2025. Disponível em: <<https://www.mendix.com/blog/benefits-low-code-development/>>.

MENDIX. **Model-Driven Development: The Foundation of Low-Code**. 2025. Disponível em: <<https://www.mendix.com/blog/model-driven-development-the-foundation-of-low-code/>>. Acesso em: 18 jul. 2025.

MISSIKOFF, M. A simple methodology for model-driven business innovation and low code implementation. **arXiv preprint arXiv:2010.11611**, 2020.

MISSIKOFF, M. A knowledge-based approach to business analysis for process innovation. In: **CEUR Workshop Proceedings**. [s.n.], 2021. Acesso em: 18 jul. 2025. Disponível em: <<http://ceur-ws.org/>>.

N-iX. **Composable architecture: A guide to modular and agile solutions**. 2024. Online; accessed 16 jul. 2025. Disponível em: <<https://www.n-ix.com/composable-architecture/>>.

NATIS, Y.; GAUGHAN, D.; O'NEILL, M.; LHEUREUX, B.; PEZZINI, M. **Innovation Insight for Packaged Business Capabilities and Their Role in the Future Composable Enterprise**. [S.l.], 2019. Disponível em: <<https://www.gartner.com/en/documents/3976170>>.

(OMG), O. M. G. **Model Driven Architecture (MDA) Guide Version 1.0.1**. 2003. <<https://www.omg.org/cgi-bin/doc?omg/03-06-01>>. Acesso em: 30 jul. 2025.

(OMG), O. M. G. **Business Process Model and Notation (BPMN) Version 2.0**. 2011. <<https://www.omg.org/spec/BPMN/2.0>>. Acesso em: 30 jul. 2025.

OSHALogs. **OSHA Recordkeeping Made Unbelievably Simple**. 2026. Disponível em: <<https://www.oshalogs.com>>.

OZMAN, F. M. A systematic literature review on current developments of low code-no code solutions in the it sector. **World Journal of Advanced Engineering Technology and Sciences**, v. 14, n. 03, p. 162–169, 2025. Disponível em: <<https://doi.org/10.30574/wjaets.2025.14.3.0072>>.

PMI. **PMI Citizen Developer**. 2025. Disponível em: <<https://www.pmi.org/citizen-developer>>. Acesso em: 18 jul. 2025.

PRISM, B. **The Role of Citizen Developers in Automation**. 2024. Disponível em: <<https://www.blueprism.com/resources/blog/the-role-of-citizen-developers-in-automation/>>. Acesso em: 18 jul. 2025.

SAHAY, A.; INDAMUTSA, A.; RUSCIO, D. D.; PIERANTONIO, A. Supporting the understanding and comparison of low-code development platforms. In: **Proceedings of the 25th International Conference on Evaluation and Assessment in Software Engineering (EASE 2020)**. ACM, 2020. p. 15:1–15:10. Disponível em: <<https://doi.org/10.1145/3375174.3375259>>.

SCHMIDT, D. C. Model-driven engineering. **IEEE Computer**, v. 39, n. 2, p. 25–31, 2006.

SOUZA, T. S. de. **Model Driven Architecture – Conceitos Fundamentais**. 2008. Online; accessed 21 jul. 2025. Disponível em: <<http://www.linhadecodigo.com.br/artigo/1953/model-driven-architecture-conceitos-fundamentais.aspx>>.

The Enterprisers Project. **What is digital transformation?** 2020. Acesso em: 18 jul. 2025. Disponível em: <<https://enterprisesproject.com/what-is-digital-transformation>>.

VeilSun. **Construction Safety Solutions | Custom Low-Code Apps**. 2026. Disponível em: <<https://www.veilsun.com/construction-safety-solutions>>.

VERHOEF, P. C.; BROEKHUIZEN, T.; BART, Y.; BHATTACHARYA, A.; DONG, J.; FABIAN, N.; HAENLEIN, M. Digital transformation: A multidisciplinary reflection and research agenda. **Journal of Business Research**, Elsevier, v. 122, p. 889–901, 2019. Disponível em: <<https://doi.org/10.1016/j.jbusres.2019.09.022>>.

VIAL, G. Understanding digital transformation: A review and a research agenda. **Journal of Strategic Information Systems**, Elsevier, v. 28, n. 2, p. 118–144, 2019. Disponível em: <<https://doi.org/10.1016/j.jsis.2019.01.003>>.

XLEDGER. **Composable ERP vs Traditional ERP: Why Your Choice Is Clear**. 2025. Online; accessed 21 jul. 2025. Disponível em: <<https://xledger.com/blog/composable-erp-vs-traditional-erp-why-your-choice-is-clear/>>.

ANEXO A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO UTILIZADO

O TCLE a seguir refere-se à pesquisa intitulada “Uma Abordagem Integrada de Engenharia Orientada a Modelos e Low-code para Inovação em Sistemas Corporativos”.

O documento descreve os objetivos do estudo, destacando a proposta de avaliação da eficácia do framework EasInnova no desenvolvimento do aplicativo Alpha Checklist, uma solução voltada para otimizar a avaliação de riscos em Saúde e Segurança do Trabalho (SST). Além disso, esclarece os procedimentos da pesquisa (uso do aplicativo e questionário de aceitação), os possíveis riscos (cansaço leve) e benefícios (melhoria da ferramenta e processos), a voluntariedade da participação, as garantias de sigilo e os contatos para dúvidas. O termo assegura ainda a assistência aos participantes e prevê a publicação dos resultados de forma anônima e agregada.



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

O (A) Senhor (a) está sendo convidado (a) a participar de uma pesquisa. Por favor, leia este documento com bastante atenção antes de assiná-lo. Caso haja alguma palavra ou frase que o (a) senhor (a) não consiga entender, converse com o pesquisador responsável pelo estudo ou com um membro da equipe desta pesquisa para esclarecê-los.

A proposta deste termo de consentimento livre e esclarecido (TCLE) é explicar tudo sobre o estudo e solicitar a sua permissão para participar do mesmo.

O objetivo desta pesquisa é avaliar a eficácia do framework EasInnova no desenvolvimento do aplicativo Alpha Checklist, uma solução voltada para a avaliação de riscos em Saúde e Segurança do Trabalho (SST). A justificativa do estudo é compreender como a integração de metodologias modernas pode agilizar o desenvolvimento de software corporativo, garantindo qualidade técnica e facilidade de uso para os profissionais da área.

Se o(a) Sr.(a) aceitar participar da pesquisa, os procedimentos envolvidos em sua participação são os seguintes: a utilização do aplicativo Alpha Checklist para simulação ou realização de avaliações de risco e/ou a resposta a um questionário eletrônico (baseado no modelo TAM - Modelo de Aceitação Tecnológica) para avaliar a utilidade e a facilidade de uso da ferramenta. O tempo estimado para participação é de 15 a 20 minutos.

Sua participação na pesquisa é totalmente voluntária, ou seja, não é obrigatória. Caso o(a) Sr.(a) decida não participar, ou ainda, desistir de participar e retirar seu consentimento durante a pesquisa, não haverá nenhum prejuízo ao seu vínculo institucional com a empresa ou com a Universidade Federal do Ceará.

Solicitamos também sua autorização para apresentar os resultados deste estudo em



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS

eventos acadêmicos e publicar em revista científica nacional e/ou internacional. Por ocasião da publicação dos resultados, seu nome será mantido em sigilo absoluto, bem como em todas as fases da pesquisa, sendo os dados tratados de forma anônima e agregada.

É assegurada a assistência durante toda pesquisa, bem como é garantido ao Sr.(a), o livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, enfim, tudo o que o(a) Sr.(a) queira saber antes, durante e depois da sua participação.

Endereço do(s) responsável (is) pela pesquisa:

Pesquisador Responsável: Alanis Oliveira Santos

Instituição: Universidade Federal do Ceará

Endereço: Tv. Raimundo Maciel Pereira, 41, Russas (CE), Cep: 62906-090.

Telefones para contato: (74) 99195-8862

E-mail: alanisoliveira39@gmail.com

ATENÇÃO: Se você tiver alguma consideração ou dúvida sobre a sua participação na pesquisa entre em contato com o Comitê de Ética em Pesquisa da UFC – Rua Coronel Nunes de Melo, 1000 Rodolfo Teófilo fone: 3366-8346.

O abaixo assinado _____, _____ anos, RG: _____.declara que é de livre e espontânea vontade que está participando da pesquisa. Eu declaro que li cuidadosamente este Termo de Consentimento Livre e Esclarecido e que, após sua leitura tive a oportunidade de fazer perguntas sobre o seu conteúdo, como também sobre a pesquisa e recebi explicações que responderam por completo minhas dúvidas. E declaro ainda estar recebendo uma cópia assinada deste termo e que minha participação é de caráter voluntário e não serei remunerado.

Pesquisador

Responsável: _____

Data: __/__/__