



DISSERTAÇÃO DE MESTRADO PROFISSIONAL

**Identificação de fatores de riscos associados ao
processo de desenvolvimento de softwares
provedores dos programas sociais: o caso do
Programa Social Brasileiro Alfa**

Rodrigo Franco de Souza

Programa de Pós-Graduação Profissional em Engenharia Elétrica

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Elétrica

DISSERTAÇÃO DE MESTRADO PROFISSIONAL

Identificação de fatores de riscos associados ao processo de desenvolvimento de softwares provedores dos programas sociais: o caso do Programa Social Brasileiro Alfa

Rodrigo Franco de Souza

Dissertação de Mestrado Profissional submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica

Banca Examinadora

Prof. Carlos André de Melo Alves., Dr., UnB/ADM/FACE _____

Orientador

Prof^a. Fabiana Freitas Mendes, Dr^a., UnB/GAMA _____

Examinadora interna

Prof. Rosalvo Ermes Streit, Dr., UCB _____

Examinador externo

Prof. Rafael Rabelo Nunes, Dr., UnB/ADM/FACE _____

Suplente

FICHA CATALOGRÁFICA

FRANCO DE SOUZA, RODRIGO

Identificação de fatores de riscos associados ao processo de desenvolvimento de softwares provedores dos programas sociais: o caso do programa social brasileiro Alfa, [Distrito Federal] 2024. Xvi 179 p., 210 x 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2024).

Dissertação de Mestrado Profissional - Universidade de Brasília, Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. Desenvolvimento de software

2. Gestão de riscos

3. Taxonomia SEI

4. Programas Sociais

5. Estudo de caso

I. ENE/FT/UnB

II. Título (série)

PUBLICAÇÃO: PPEE.MP.068

REFERÊNCIA BIBLIOGRÁFICA

FRANCO, R. (2024). IDENTIFICAÇÃO DE FATORES DE RISCOS ASSOCIADOS AO PROCESSO DE DESENVOLVIMENTO DE SOFTWARES PROVEDORES DOS PROGRAMAS SOCIAIS: O CASO DO PROGRAMA SOCIAL BRASILEIRO ALFA. Dissertação de Mestrado Profissional, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 179 p.

CESSÃO DE DIREITOS

AUTOR: Rodrigo Franco de Souza

TÍTULO: Identificação de fatores de riscos associados ao processo de desenvolvimento de softwares provedores dos programas sociais: O Caso do Programa Social Brasileiro Alfa.

GRAU: Mestre em Engenharia Elétrica ANO: 2024

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Do mesmo modo, a Universidade de Brasília tem permissão para divulgar este documento em biblioteca virtual, em formato que permita o acesso via redes de comunicação e a reprodução de cópias, desde que protegida a integridade do conteúdo dessas cópias e proibido o acesso a partes isoladas desse conteúdo. O autor reserva outros direitos de publicação e nenhuma parte deste documento pode ser reproduzida sem a autorização por escrito do autor.

Rodrigo Franco de Souza

Depto. de Engenharia Elétrica (ENE) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro CEP 70919-970 - Brasília - DF - Brasil

AGRADECIMENTOS

À minha esposa, Maira Vieira, cujo incentivo e parceria incansáveis foram fundamentais para superar cada desafio desta jornada. Sua presença constante e apoio foram a força que me impulsionou a seguir em frente.

Aos meus filhos, Felipe, Gabriel e Caio, cuja compreensão e apoio incondicional me proporcionaram a tranquilidade e motivação necessárias para me dedicar a este trabalho. Vocês são minha maior inspiração e motivo de todo esforço. Papai ama vocês.

À minha mãe, Antonia Marques, e ao meu pai, Preto Rezende, cujo amor, fé e apoio foram o alicerce para a construção dos meus sonhos. Vocês não apenas acreditaram em mim em cada etapa desta jornada, mas também sonharam comigo esse sonho, tornando as esperanças realidade. Mãe, Pai, com imenso orgulho e profunda gratidão, compartilho com vocês esta conquista.

Ao meu orientador, Prof. Carlos André, agradeço imensamente por ter aceitado orientar este trabalho já em andamento, demonstrando paciência, dedicação e um compromisso excepcional. Suas orientações precisas foram cruciais para a realização deste estudo.

Aos professores do Programa de Pós-Graduação Profissional em Engenharia Elétrica da Universidade de Brasília, pela excelência acadêmica e pelo suporte oferecido ao longo do curso, e aos professores da banca de qualificação, pela leitura atenta e pelas valiosas ponderações sobre o estudo. O conhecimento transmitido por vocês foi vital para meu desenvolvimento profissional e pessoal.

A Empresa BETA, por ter fornecido acesso aos entrevistados e permitido a coleta de dados essenciais para a realização deste estudo. A colaboração e o suporte oferecidos foram indispensáveis para o sucesso da minha pesquisa. O compromisso da Empresa BETA com a educação e o desenvolvimento profissional não apenas enriqueceu este trabalho, mas também contribuiu significativamente para meu crescimento pessoal e profissional.

Por fim, aos meus amigos, colegas de trabalho e às Equipes de Nossa Senhora, pela amizade, apoio e pelos momentos de descontração e fé que me ajudaram a manter o equilíbrio nos períodos mais intensos.

Agradeço a todos por fazerem parte desta caminhada, compartilhando comigo os desafios e as alegrias. Este trabalho também celebra nossa jornada juntos.

RESUMO

O desenvolvimento de software em contextos governamentais, particularmente para programas sociais, enfrenta desafios que enfatizam a importância da gestão de riscos para melhor assegurar o sucesso e a entrega efetiva de serviços à população. O objetivo geral desta pesquisa é identificar os principais fatores de riscos associados às etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa. A pesquisa, de natureza descritiva e aplicada, adota uma abordagem qualitativa, empregando como estratégia metodológica o estudo de caso. Os dados coletados abrangem entrevistas com desenvolvedores, arquitetos, gestores de projeto, complementados por coleta de documentos internos da Empresa BETA. O tratamento dos dados abrangeu análise documental e análise de conteúdo, considerando 64 atributos, 13 elementos de classe e 3 classes descritas na taxonomia de risco hierárquica proposta pelo *Software Engineering Institute* - SEI. Após detalhar as fases de desenvolvimento de software provedor do referido programa social, os principais resultados das análises permitiram a identificação de 39 atributos, sendo 17 vinculados à classe 'Engenharia de Produto' 12 vinculados à classe 'Ambiente de Desenvolvimento' e 10 vinculados à classe 'Restrições do Programa'. O exame dos atributos identificados enfatizou a classe 'Restrições do Programa', mostrando a influência de fatores externos no processo de desenvolvimento de softwares, em consonância com achados de estudos anteriores. Adicionalmente, as análises permitiram detalhar os atributos segundo elementos de classe, destacando-se o atributo 'Cronograma', vinculado ao elemento de classe 'Recursos'. Este trabalho contribui para o campo da gestão de riscos e da engenharia de software, oferecendo evidências sobre o uso de atributos que podem ajudar a identificar riscos no processo de desenvolvimento de softwares de programas sociais e destacando a importância de uma identificação de riscos que considere as especificidades de cada etapa do desenvolvimento de software. Os resultados também podem contribuir para reflexões sobre a gestão de riscos em processos de desenvolvimento de software de programas sociais no País.

Palavras-chave: Desenvolvimento de software, Gestão de riscos, Taxonomia SEI, Programas Sociais, Estudo de caso.

ABSTRACT

The software development for governmental context, particularly for social programs, faces challenges that emphasize the importance of risk management to better ensure success and effective service delivery to the population. This research main goal is to identify the main risk factors associated with the stages of the software development process providers of the Brazilian Social Program Alpha. The research has descriptive and applied nature, and adopts a qualitative approach by employing the research strategy case study. The collected data encompass interviews with developers, architects, project managers, complemented by the collection of internal documents from Company BETA. Data treatment included documentary analysis and content analysis, considering 64 attributes, 13 class elements, and 3 classes described in the hierarchical risk taxonomy proposed by the Software Engineering Institute - SEI. After detailing the phases of software development provider of the aforementioned social program, the main results of the analyzes allowed the identification of 39 attributes, with 17 linked to the 'Product Engineering' class, 12 linked to the 'Development Environment' class, and 10 linked to the 'Program Constraints' class. The examination of the identified attributes emphasized the 'Program Constraints' class, showing the influence of external factors on the software development process, in line with findings from previous studies. Additionally, the analyzes allowed detailing the attributes according to class elements, highlighting the attribute 'Schedule', linked to the 'Resources' class element. This work contributes to the field of risk management and software engineering, offering evidence on the use of attributes that can help identify risks in the software development process of social programs and highlighting the importance of risk identification that considers the specificities of each stage of software development. The results can also contribute to reflections on risk management in software development processes of social programs in the country.

Keywords: Software development, Risk management, SEI Taxonomy, Social Programs, Case study.

LISTA DE ABREVIações

SIGLA	Nome
ABNT	Associação Brasileira de Normas Técnicas
ACs	Áreas de Conhecimento
ADR	Análise e Decisão e Resolução
AE1	Programa Social Brasileiro Alfa-1
AE2	Programa Social Brasileiro Alfa 2
BFP	Bolsa Família
BPC	Benefício de Prestação Continuada
CADUNICO	Cadastro Único para Programas Sociais
CEF	Caixa Econômica Federal
CMM	Capability Maturity Model
CNIS	Cadastro Nacional de Informações Sociais
COBIT	Control Objectives for Information and Related Technologies
CyBOK	Cyber Security Body of Knowledge
DAD	Disciplined Agile Delivery
DEVOPS	Desenvolvimento e Operações
DRE	Desenvolvimento de Requisitos
DRU	Desenvolvimento para a Reutilização
DSDM	Dynamic Software Development Method
GLP	Gás Liquefeito de Petróleo
GR	Gestão de Riscos
GRI	Gerência de Riscos
IBGC	Instituto Brasileiro de Governança Corporativa
IBGE	Instituto Brasileiro de Geografia e Estatística
INSS	Instituto Nacional do Seguro Social
ITP	Integração do Produto
KANBAN	Kanban
LOAS	Lei Orgânica da Assistência Social
MC	Ministério da Cidadania
ME	Ministério da Economia
MEI	Microempreendedores Individuais
MGI	Ministério da Gestão e Inovação
MCTIC	Ministério da Ciência, Tecnologia, Inovações e Comunicações
MPS	Ministério da Previdência Social
MPS–BR	Melhoria do Processo de Software Brasileiro
MSS	Multilevel Security Spiral
OKR	Objective and Key Results
OMS	Organização Mundial da Saúde
OTG	Testing Guide
OWASP	Projeto de Segurança de Aplicações Web Abertas
PCP	Projeto e Construção do Produto
PMBOK	Project Management Body of Knowledge
PRONAF	Programa Nacional de Fortalecimento da Agricultura Familiar

RUP	Rational Unified Process
SDLC	Software Development Life Cycle
SEBRAE	Serviço Brasileiro de Apoio às Micro e Pequenas Empresas
SEI	Software Engineering Institute
SOFTEX	Associação para Promoção da Excelência do Software Brasileiro
SWEBOK	Software Engineering Body of Knowledge
TBQ	Taxonomy-Based Questionnaire
TD	Test Drive Development
UI	Interface do Usuário
UML	Linguagem de Modelagem Unificada
VAL	Validação
VER	Verificação

LISTA DE FIGURAS

Figura 1 - Ciclo de vida padrão de desenvolvimento de software.	25
Figura 2 - Processos da NBR ISO/IEC 12207.....	30
Figura 3 - Etapas da Gestão de Risco.	44
Figura 4 - As 21 área do conhecimento no âmbito do CyBOK.....	49
Figura 5 - Modelo para Identificação de Riscos com Taxonomias – MIRT.	56
Figura 6 - Modelo de Gerenciamento de Risco.....	58
Figura 7 - Taxonomia de Riscos no Desenvolvimento de Software.	59
Figura 8 - Classificação da Pesquisa	73
Figura 9 - Organograma Empresa BETA.	76
Figura 10 - Nuvem de palavras dos termos mais apresentados nas entrevistas	129

LISTA DE QUADROS

Quadro 1 - Características das metodologias ágeis.....	28
Quadro 2 - Normas internacionais e modelos de qualidade de software.	31
Quadro 3 - Níveis de maturidade do CMMI.....	32
Quadro 4 - Níveis de maturidade do MPS.BR.....	33
Quadro 5 - Conceitos de Risco.	40
Quadro 6 - Termos e Definições.	40
Quadro 7 – Frameworks de gerenciamento de risco e seus objetivos e pilares.....	46
Quadro 8 - Atividades sequenciais para aplicação do TBQ.	55
Quadro 9- Atividades para Gerenciamento de Riscos Baseado em Taxonomia do CMMI.....	55
Quadro 10 – Proposta de uso de Taxonomia de Riscos no Desenvolvimento de Software.....	60
Quadro 11 - Principais 10 fatores de risco - todas as classes.....	61
Quadro 12 – Principais Programas Sociais Brasileiros	65
Quadro 13 – Síntese dos programas sociais.....	68
Quadro 14 - Síntese do total de beneficiários por continente.....	69
Quadro 15 – Total de funcionários por diretoria DIT e DPS.	77
Quadro 16 – Visão de atuação predominante por classe.....	78
Quadro 17 - Ordem para realização das entrevistas.....	79
Quadro 18 - Relação entre objetivos, coleta e análise dos dados.....	84
Quadro 19 – Atividades do processo de desenvolvimento de software	86
Quadro 20 - Ponto de saturação teórica	90
Quadro 21 - Atributos identificados durante a análise das entrevistas.....	92
Quadro 22 – Elementos da classe Engenharia de produto	131
Quadro 23 - Elementos da classe Ambiente de Desenvolvimento.....	132
Quadro 24 - Elementos da classe Restrições do Programa.....	133
Quadro 25 – Análise a partir das classes presentes na Taxonomia SEI.....	135
Quadro 26 - Visão geral das medidas de proteção social por componente e país..	162

LISTA DE TABELAS

Tabela 1 - Bases pesquisadas.	22
------------------------------------	----

SUMÁRIO

FICHA CATALOGRÁFICA	I
REFERÊNCIA BIBLIOGRÁFICA.....	II
FICHA CATALOGRÁFICA	III
REFERÊNCIA BIBLIOGRÁFICA.....	III
CESSÃO DE DIREITOS	III
AGRADECIMENTOS	IV
RESUMO.....	V
ABSTRACT	VI
LISTA DE ABREVIACÕES.....	VII
LISTA DE FIGURAS	IX
LISTA DE QUADROS	X
LISTA DE TABELAS	XI
1 INTRODUÇÃO	14
1.1 Contextualização.....	14
1.2 Definição do problema de pesquisa.....	16
1.3 Objetivos	18
1.3.1 Objetivo Geral.....	18
1.3.2 Objetivos Específicos.....	18
1.4 Justificativa	18
1.5 Publicações Resultantes desta Pesquisa	20
1.6 Estrutura	21
2 FUNDAMENTAÇÃO TEÓRICA	22
2.1 Desenvolvimento de software	22
2.2 Gestão de Riscos.....	38
2.3 Risco e Desenvolvimento de Software.....	50
2.4 Programas Sociais.....	63
3 MÉTODO.....	73
3.1 Caracterização/tipologia de pesquisa	73
3.2 Caracterização dos instrumentos de pesquisa	74
3.3 Caracterização da organização em estudo.....	76
3.4 Perfil dos participantes.....	76
3.5 Procedimentos de coleta de dados	80
3.6 Procedimentos de análise dos dados	82

4	DESCRIÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS.....	85
4.1	Descrição das etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro ALFA (Associado objetivo específico 1) .	85
4.2	Verificação dos principais fatores de riscos associados ao processo de desenvolvimento dos softwares, considerando os atributos da Taxonomia SEI (Associado objetivo específico 2)	89
4.3	Classificação dos fatores riscos de acordo com elementos de classe correspondentes da Taxonomia SEI (Associado ao objetivo específico 3)	131
4.4	Comparação da ocorrência dos fatores de risco verificados segundo as classes (Associado ao objetivo específico 4)	134
5	CONSIDERAÇÕES FINAIS.....	139
	REFERÊNCIAS.....	144
	APÊNDICE A - PEDIDO DE AUTORIZAÇÃO PARA REALIZAÇÃO DE PESQUISA DE CAMPO	157
	APÊNDICE B - ESTRUTURA DO ROTEIRO DE ENTREVISTAS	158
	APÊNDICE C – MODELO DE FICHA ANALÍTICA.....	159
	APÊNDICE D – E-MAIL CONVITE PARA ENTREVISTA	160
	APÊNDICE E – E-MAIL CONVITE PARA VALIDAÇÃO DO PRÉ-TESTE	161
	APÊNDICE F – MEDIDAS DE PROTEÇÃO SOCIAL	162
	APÊNDICE G – GLOSSÁRIO ROTEIRO DE ENTREVISTAS	171
	APÊNDICE H – ATIVIDADES DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE	173

1 INTRODUÇÃO

Nesta seção são apresentados a contextualização do tema investigado, a definição do problema de pesquisa, o objetivo geral, os objetivos específicos, a justificativa, as publicações resultantes desta pesquisa e a estrutura do trabalho.

1.1 Contextualização

O desenvolvimento de software é uma atividade em constante transformação e oferece oportunidades para investigações. O rápido avanço tecnológico e a evolução das demandas do mercado são fatores essenciais que impulsionam a diversidade e a multiplicação de pesquisas, além de incentivar a adoção de novas abordagens no estudo de fenômenos (Rocha, 2022). Esta diversidade se manifesta claramente no campo da gestão de riscos no desenvolvimento de software (Leopoldino, 2019).

Diante deste cenário, os projetos de desenvolvimento de software enfrentam crescentes exigências devido à constante evolução tecnológica e aos altos padrões de qualidade requeridos. Nesse contexto, é esperado que a gestão de projetos de software seja eficaz e orientada para os principais objetivos de cada projeto (Brito, 2021), sendo necessário adotar práticas e metodologias que permitam um gerenciamento adequado, garantindo a entrega de produtos de software que atendam às necessidades dos usuários e estejam alinhados com os requisitos estabelecidos.

O aumento no uso de produtos de software tem resultado em uma maior demanda por qualidade, o que tem levado os desenvolvedores a adotarem novos modelos para garantir padrões adequados (Garcia; Oliveira; Salviano, 2016). Nesse contexto, empresas e especialistas estão cada vez mais em busca de métodos, técnicas e ferramentas de desenvolvimento que auxiliem na obtenção de melhores requisitos de software, na comunicação eficaz com os clientes e na entrega de produtos de software com alta qualidade (Cezerino; Aparecida; Paes, 2016).

Assim, os softwares passam a desempenhar um papel crucial em diversos setores, inclusive nos programas sociais implementados pelo governo brasileiro. Os softwares provedores, especificamente, referem-se àquelas plataformas e sistemas que facilitam a entrega e gestão de serviços sociais, atuando como intermediários entre as entidades governamentais e a população beneficiária. Esses sistemas são fundamentais para a eficácia dos Programas Sociais Brasileiros, como o Programa Social Brasileiro Alfa, pois permitem a administração eficiente de recursos, a distribuição de benefícios e a coleta de dados para a melhoria contínua dos serviços.

Na busca por melhores resultados no âmbito do desenvolvimento de software, a gestão de risco tem se destacado significativamente. Conforme definido pela ABNT NBR 31000:2018, o risco é compreendido como o efeito da incerteza nos objetivos, traduzindo-se na possibilidade de desvios ou falhas em alcançar os objetivos estipulados devido às incertezas ou eventos inesperados. Este entendimento reforça a necessidade de uma abordagem sistemática e bem estruturada para a gestão de riscos, a fim de mitigar potenciais desafios que possam surgir durante o processo de desenvolvimento.

É neste contexto que a Taxonomia de Riscos SEI assume uma posição central, pois segundo Carr et al. (1993), como um marco no campo da gestão de riscos em projetos de software, essa taxonomia fornece uma base sólida para a identificação, análise e tratamento de riscos. Com sua estrutura dividida em classes, elementos de classe e atributos, ela oferece um guia detalhado e sistematizado que auxilia os gestores e desenvolvedores a desmembrar e compreender os riscos em um nível mais granular. Isso facilita não apenas a identificação de áreas críticas que necessitam de atenção, mas também a aplicação de medidas preventivas e corretivas de maneira mais eficaz.

Dada a necessidade crescente de uma gestão de riscos eficiente, observa-se que empresas desenvolvedoras de software estão cada vez mais adotando novas abordagens para assegurar o sucesso de seus projetos. Segundo Silva (2017), essas abordagens são fundamentadas em técnicas de gestão de risco robustas, que visam minimizar os problemas que podem afetar negativamente a produtividade e a qualidade dos softwares desenvolvidos. Nesse sentido, a aplicação da Taxonomia SEI, com sua estrutura hierárquica detalhada, torna-se um recurso indispensável para essas organizações na busca por melhores práticas de desenvolvimento e gestão de riscos.

Quando se trata de gestão de risco no desenvolvimento de software, é comum que os projetos enfrentem vários problemas. Esses problemas podem surgir de ameaças imprevistas, não planejadas ou simplesmente ignoradas. Ao aumentar a complexidade e a quantidade de sistemas, torna-se cada vez mais necessário usar ferramentas e técnicas eficientes para minimizar os efeitos de produtos de baixa qualidade (Fernandes, 2019).

Segundo Andrade et al. (2019), projetos que tratem no tempo certo as operacionalizações de risco ou parte delas podem ter maior chance de sucesso. Ainda segundo o autor é a natureza, o tamanho, a complexidade, o grau de inovação e as

etapas do projeto que vão determinar as melhores técnicas a se utilizar (Alberto, 2021).

Reafirmando essa premissa, a ISO 9001:2015 (NBR ISO 9001:2015) introduz o conceito de mentalidade de riscos que consiste na capacidade de uma organização identificar os fatores que podem desviar seus processos e sistema de gestão da qualidade dos resultados planejados. Ela enfatiza a implementação de controles preventivos para minimizar os efeitos negativos e maximizar o aproveitamento das oportunidades que possam surgir.

A percepção dessa estreita relação entre risco e qualidade estabelece um vínculo indispensável que atravessa todas as etapas do desenvolvimento de software, tornando-se um elemento fundamental na busca pela qualidade (Andrade et al., 2019), assegurando a segurança e o êxito dos projetos em um ambiente complexo e dinâmico (Abioye et al., 2020) onde, a abordagem proativa e orientada para riscos contribui para o aprimoramento contínuo e o sucesso das organizações, proporcionando um maior controle sobre suas operações e promovendo a satisfação dos clientes.

É importante ressaltar que a preocupação com a gestão de risco e a aplicação das melhores práticas também se reflete no âmbito governamental. O reconhecimento da gestão de riscos como um processo permanente demonstra a compreensão de que ela não pode ser tratada de forma isolada (Brasil, 2018). Nesse contexto, a gestão de riscos no desenvolvimento de software assume um papel abrangente e essencial para o sucesso do desenvolvimento de software público.

A partir desta contextualização, e para melhor apoiar no desenvolvimento desta pesquisa, serão considerados alguns conceitos essenciais, como o conceito de ciclo de vida para descrever etapas do desenvolvimento de software (Swebok, 2014), o gerenciamento de risco como a prática de avaliar e controlar quaisquer perigos que possam afetar projetos, processos e produtos de software (Hall, 1998), e o risco será compreendido como o efeito da incerteza nos objetivos, podendo ser positivo ou negativo ISO/IEC 16085 (2021).

1.2 Definição do problema de pesquisa

A pandemia de COVID-19 teve início em dezembro de 2019 em Wuhan, China, e se espalhou pelo mundo, resultando em uma crise de saúde global sem precedentes (Heymann e Shindo, 2020). A doença, causada pelo coronavírus SARS-CoV-2, causou milhares de mortes em vários países, levando a Organização Mundial da

Saúde - OMS a declarar em março de 2020 uma pandemia de nível global WHO (2020). Após três anos, em maio de 2023, a OMS determinou que a COVID-19 passou a ser reconhecida como uma questão de saúde estabelecida e contínua, deixando de ser uma emergência de saúde pública de preocupação internacional WHO (2023).

No Brasil, os primeiros registros da COVID-19 foram confirmados logo antes da oficialização da pandemia, o que levou à adoção imediata de medidas críticas para mitigar a disseminação do vírus, conforme apontado por Cardoso (2020). Durante esse período, milhões de brasileiros se viram em situações econômicas precárias, uma circunstância detalhada por Costa (2021), pressionando os governos estaduais, municipais e o federal a desenvolverem programas sociais e de auxílio financeiro. Estes programas foram essenciais para sustentar as famílias afetadas pela crise. Em resposta aos desafios impostos pelas restrições à mobilidade, tais iniciativas foram implementadas através de plataformas digitais. Um exemplo foi o Programa Social Brasileiro Alfa, que se destacou pelo uso intensivo de tecnologias digitais para contornar os obstáculos à circulação física, como destacado por Marins et al. (2021).

A empresa, cujo nome fictício é BETA, foi escolhida para desenvolver os softwares necessários para a viabilização do programa social Alfa. Esses softwares deveriam ser capazes de identificar os cidadãos que tinham direito aos benefícios por meio do cruzamento de informações encontradas em bases de dados governamentais (Cardoso, 2020).

É importante destacar que a administração pública está passando por uma transição do mundo analógico para o digital, buscando oferecer serviços de forma mais eficiente e ágil. Ao mesmo tempo, a sociedade busca utilizar esses serviços para acessar seus direitos (Corvalán, 2017). Assim, frente aos desafios do desenvolvimento de software e a pressão exercida pela sociedade em busca de softwares que entreguem resultados com a maior qualidade possível e no menor prazo, mostra-se imperativa a necessidade de compressão do processo de gestão de risco no desenvolvimento de software na Empresa BETA e de modo particular no desenvolvimento dos softwares que viabilizaram a execução do Programa Social Brasileiro Alfa.

É fundamental ressaltar que uma organização governamental especializada na gestão de dados, aqui referida como a Empresa BETA, desempenha um papel crucial como a entidade gestora de um importante repositório de dados. Este repositório, conhecido sob o nome de Registro de Informações, abriga uma vasta quantidade de registros, incluindo mais de 36 bilhões de entradas relacionadas a indivíduos e entidades corporativas, englobando informações relevantes sobre seu público-alvo. A

capacidade dessa instituição de administrar tal volume de dados não apenas reforça sua autoridade na facilitação de políticas públicas governamentais e no suporte à população, mas também a posiciona como um possível modelo para o desenvolvimento de estratégias de gestão de risco em outras entidades públicas Empresa BETA .

Assim, com base nas informações e conceitos inicialmente apresentados durante a contextualização, torna-se relevante os aspectos relacionados a gestão de risco para o êxito no desenvolvimento de softwares. Dessa maneira, emerge a necessidade de investigar mais a fundo esta temática, levando à proposição do seguinte problema de pesquisa: **Quais foram os principais fatores de riscos associados às etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa?**

1.3 Objetivos

1.3.1 Objetivo Geral

Identificar os principais fatores de riscos associados às etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa.

1.3.2 Objetivos Específicos

1. Descrever as etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro ALFA;
2. Verificar os principais fatores de riscos associados ao processo de desenvolvimento de software previamente descrito, considerando os atributos da taxonomia de risco hierárquica do SEI;
3. Classificar os fatores de riscos verificados de acordo com elementos de classe da referida taxonomia;
4. Comparar a ocorrência dos fatores de riscos verificados segundo as classes previamente referidas, com base na literatura.

1.4 Justificativa

A justificativa desta dissertação está organizada em três aspectos fundamentais: teórico, prático e social. Cada um desses aspectos aborda lacunas específicas de estudo relacionadas à gestão de risco no desenvolvimento de software, especialmente no contexto do desenvolvimento dos softwares responsáveis pelo provimento do Programa Social Brasileiro ALFA. A compreensão dessas lacunas e a

busca por soluções eficazes para mitigar os riscos envolvidos nesse processo são essenciais para aprimorar a qualidade dos serviços oferecidos, garantir a segurança da informação e promover a eficiência e a transparência das operações governamentais (Andrade et al., 2019).

A fim de explorar e analisar a temática da gestão de riscos no desenvolvimento de software, foram conduzidas pesquisas em diversas bases de dados, como Web of Science, SciELO, SAGE, SCOPUS, SpringerLink e Wiley Online Library. Com base na revisão da literatura deste estudo, foi percebido que várias investigações já abordaram o gerenciamento de risco e o desenvolvimento de software em separado, no entanto, nenhuma pesquisa focou na Gestão de Risco voltada ao desenvolvimento de Software no Governo e tampouco nos softwares provedores do Programa Social Brasileiro ALFA.

Assim, no aspecto teórico, observa-se uma carência de estudos direcionados à aplicação da gestão de risco de maneira mais estruturada em projetos de desenvolvimento de software. Embora existam pesquisas relevantes nessa área, a literatura acadêmica ainda é incipiente e escassa nesse campo específico. Portanto, esta pesquisa busca preencher essa lacuna, fornecendo ideias sobre a gestão de risco nas organizações públicas e nos softwares provedores do programa social alfa, contribuindo para o avanço do conhecimento nessa área (Fan; Yu, 2004).

Embora exista uma literatura que trate da Gestão de Risco no Governo, permeando Universidades Federais (Furlan; Pacheco, 2021), a Presidência da República, por meio de Instrução Normativa Brasil (2020), considerando o contexto do Programa Social Brasileiro ALFA, observa-se uma lacuna na literatura acadêmica em relação à aplicação da gestão de risco nas etapas de desenvolvimento de softwares.

Conforme Oliveira et al. (2020), a Gestão de Riscos no desenvolvimento de software é um tópico recente em pesquisas acadêmicas e principalmente no ambiente corporativo, mostrando-se como um desafio constante para organizações. No aspecto prático, a ausência de estudos específicos sobre a gestão de risco no contexto do Programa Social Brasileiro ALFA e do Governo ressalta a importância de investigar e compreender os desafios enfrentados ao longo das etapas de desenvolvimento de software.

O estudo de Miranda (2017) expõe um recorte destacando as instruções normativas, boas práticas, políticas de gestão de risco e, porém, da mesma forma, explicita uma lacuna para estudos abordando a gestão de risco aplicada ao desenvolvimento de software.

O trabalho de Sousa Nunes et al. (2020) destaca a importância do da gestão de riscos no desenvolvimento de software, porém, revela uma baixa maturidade da pesquisa, com um número limitado de publicações em periódicos especializados. Diante dessa situação, este estudo assume uma relevância ao ampliar o conhecimento sobre a gestão de risco além de fornecer subsídios para a adoção de

estratégias eficazes de gestão de risco no processo de desenvolvimento de software. Isso contribui para melhorar a governança, a segurança e o desempenho das organizações, preenchendo a lacuna existente e podendo contribuir para promover a qualidade, segurança e transparência dos serviços oferecidos pelo Governo.

Além disso, a pesquisa tem um impacto social, pois estudos sobre a gestão de risco no desenvolvimento de software podem auxiliar a produção de softwares de melhor qualidade, mais seguros e fomentar uma eficiente destinação de recursos públicos no desenvolvimento desses softwares, contribuindo para um gasto responsável dos recursos públicos (Marins et al., 2021).

Numa perspectiva prática, este trabalho pode fornecer subsídios para as instituições do Governo, incluindo gestores e profissionais envolvidos na gestão de risco dentro das etapas de desenvolvimento de software, nos ministérios, institutos e na própria Empresa BETA. Essas instituições e indivíduos poderão beneficiar-se em suas atividades dos achados fornecidas neste estudo.

A pesquisa tem relevância prática para pesquisadores, acadêmicos e a sociedade em geral interessados em ampliar o conhecimento sobre gestão de risco no desenvolvimento de software no contexto governamental. Desta forma, este trabalho se apresenta como uma oportunidade de disseminar conhecimentos e recomendações práticas que podem ser aplicadas no contexto Governamental, além de servir como base para pesquisas futuras nessa área.

O trabalho enfatiza um campo de estudo sobre a gestão de riscos em projetos de desenvolvimento de software para programas sociais governamentais. Através da análise baseada na Taxonomia SEI, esta pesquisa pode contribuir para investigar a dinâmica de riscos contexto o desenvolvimento de software voltado para a execução e o apoio de programas sociais.

1.5 Publicações Resultantes desta Pesquisa

Nesta pesquisa, foram alcançados marcos importantes que contribuíram para o campo da gestão de riscos no desenvolvimento de software. Um desses avanços foi a aceitação do artigo intitulado "Gestão de Riscos e Desenvolvimento de Software: estudo sobre a distribuição de artigos científicos em bases de dados internacionais de 2019 a 2023" para publicação na Revista Ibérica de Sistemas e Tecnologias de Informação - RISTI. O artigo foi incluído em uma edição especial da revista, destacando sua relevância e a contribuição para o conhecimento acadêmico e prático na área.

Adicionalmente, o estudo foi apresentado no 7th International Conference on Information Technology & Systems - ICITS'24 , realizado na Universidad de La Frontera, em Temuco, Chile, entre os dias 24 e 26 de janeiro de 2024. A apresentação

forneceu uma plataforma para a disseminação das descobertas e propiciou discussões com outros especialistas.

1.6 Estrutura

Este trabalho está organizado em cinco capítulos, detalhando diversos aspectos da pesquisa sobre gestão de riscos no desenvolvimento de software para programas sociais. O primeiro capítulo introduz o tema, problema, objetivos e justificativa da pesquisa. O segundo explora conceitos fundamentais sobre gestão de risco, desenvolvimento de sistemas de informação e programas sociais, estabelecendo uma base teórica. O terceiro capítulo descreve a metodologia de pesquisa utilizada, incluindo procedimentos e técnicas de coleta e análise de dados. O quarto capítulo aborda um estudo de caso sobre gestão de riscos, e o quinto apresenta as conclusões, resultados, contribuições e sugestões para pesquisas futuras. Essa estrutura facilita uma análise sistemática e completa do tema.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta o referencial teórico que foi utilizado para a construção deste trabalho, incluindo os conceitos fundamentais, a literatura utilizada como guia e a legislação pertinente ao tema abordado na pesquisa acadêmica. Na seção 2.1 será apresentada uma revisão sobre Desenvolvimento de Software, na Seção 2.2 Gestão de Risco, na Seção 2.3 Risco e Desenvolvimento de Software e na Seção 2.4 Programas Sociais.

Conforme Tabela 1, apresenta as bases de dados que foram utilizadas para realizar investigações sobre o tema da gestão de riscos no desenvolvimento de software. Os filtros utilizados na busca incluíram os termos "Risk Management and Development Software" e foram aplicados filtros específicos, como área de assunto relacionada a "Computação" ou "Engenharia", e tipo de documento limitado a artigos de pesquisa. O período de análise abrangeu os anos de publicação de 2019 a 2023.

Tabela 1 - Bases pesquisadas.

Bases de Pesquisa	Ano	Quantidade
SCOPUS (Elsevier)	2019 a 2023	08
Scientific Electronic Library Online (SciELO)		04
SAGE Journals Online		01
SpringerLink		15
Web of Science		12
Wiley Online Library		01
Biblioteca Digital Brasileira de Teses e Dissertações (BDTD)		04
Total		451

Fonte: o autor, a partir de dados da pesquisa.

Após a seleção dos artigos, foi analisado o índice de impacto de acordo com o Qualis CAPES A1 a A4, em periódicos aderentes das Engenharias IV com o objetivo de privilegiar a utilização de estudos publicados em periódicos de estratos superiores.

2.1 Desenvolvimento de software

A Engenharia de Software é uma área da computação que se concentra na especificação, desenvolvimento e manutenção de sistemas de software. Seu principal objetivo é organizar os processos do projeto e aumentar a produtividade na criação de software. Isso é alcançado por meio da aplicação de práticas de engenharia de que

fornecem uma estrutura sólida para a construção de sistemas de alta qualidade (Sbrocco e Macedo, 2012).

Conforme destacado por o software pode ser descrito como um conjunto de instruções que, quando executadas, produzem os resultados desejados e os comportamentos esperados. Além disso, ele possui estruturas de dados que permitem a manipulação de informações. É importante ressaltar, como mencionado por ZUCULE (2017), o software não se limita apenas ao programa em si, mas também abrange toda a documentação e configuração necessárias para o seu funcionamento adequado. Desta forma, o software, como qualquer outra criação humana, passa por um processo evolutivo contínuo, que engloba as práticas de desenvolvimento adotadas pelo mercado e as tecnologias utilizadas em sua construção.

De acordo com Davenport (1993), processo é uma sequência específica de atividades no tempo e espaço, com começo e um fim, onde podem ser visivelmente identificadas as contribuições e os resultados produzidos. Por conseguinte, o processo de software geralmente segue uma estrutura comum, composta por um pequeno número de atividades que são aplicáveis a todos os projetos de software, independentemente do tamanho ou da complexidade. Essa estrutura proporciona consistência e orientação para alcançar os objetivos do projeto de forma eficiente e eficaz.

O mesmo raciocínio do parágrafo anterior é corroborado por Costa (2022), definindo que o processo de desenvolvimento de software é composto por um conjunto de atividades, ferramentas e modelos que permitem a criação de um software. Cada modelo é implementado através de uma metodologia, que determina a sequência das atividades, métodos e ferramentas, bem como as relações entre eles. É importante citar que cada modelo pode ter várias metodologias associadas a ele.

Conforme destaca a ISO/IEC 12207-2017, o processo de desenvolvimento de software é um conjunto de atividades inter-relacionadas ou interativas que transforma entradas em saídas. O ciclo de vida por sua vez é a evolução de um sistema, produto, serviço, projeto ou outra entidade feita por humanos, desde a concepção até a desativação. Já o modelo de ciclo de vida é um framework de processos e atividades relacionados com o ciclo de vida, que pode ser organizado em etapas, atuando como uma referência comum para comunicação e entendimento (ISO, 2017).

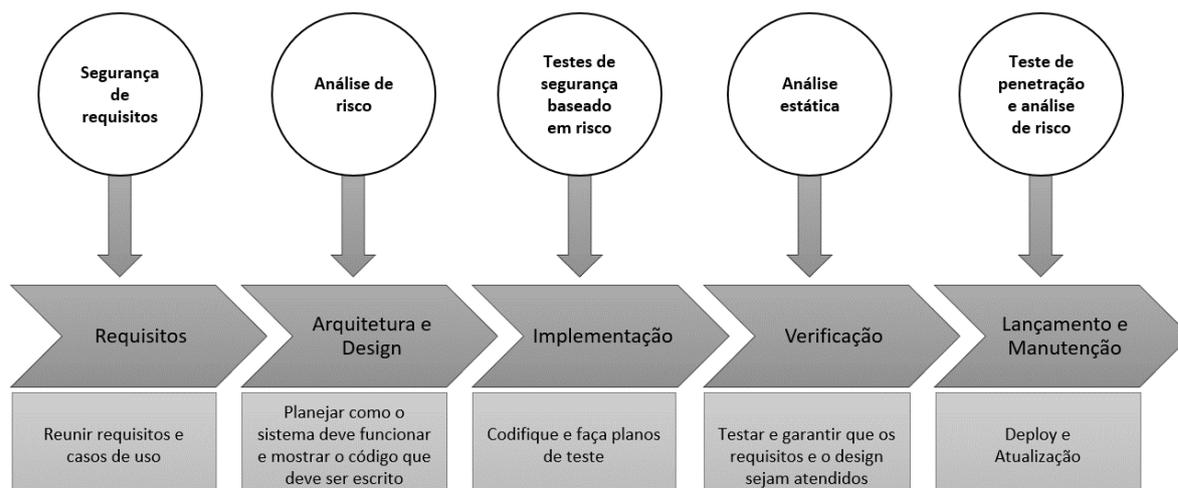
Assim, o processo de desenvolvimento de software, é representado pelo ciclo de vida do desenvolvimento de software que é um conceito fundamental, já apresentado fundamentação teórica, na engenharia de software, utilizado para

descrever o processo de planejamento, criação, codificação, teste e implementação de uma especificação de requisitos do usuário (Sommerville, 2019).

Esses princípios delineados na ISO/IEC 12207-2017 ressaltam a importância de um planejamento proativo e eficiente no desenvolvimento de software. Como discutido por Menasce e Almeida (2002), um planejamento proativo, especialmente em relação à capacidade, é crucial para garantir o desempenho eficiente dos sistemas de software. Isso permite que eles atendam a demandas variáveis sem comprometer a qualidade ou a funcionalidade. Dentro das etapas de planejamento, um conceito importante que vem sendo introduzido é o 'overengineering'. Embora interpretado como o uso excessivo de recursos, pode ser entendido como uma estratégia defensiva prudente para garantir que os sistemas possam suportar cargas inesperadamente altas. Essa abordagem reflete a relevância de um planejamento de capacidade bem fundamentado, que equilibra eficiência de recursos com preparação para cenários de pico de carga, sendo assim, parte integrante do ciclo de planejamento e desenvolvimento de software.

O ciclo de vida do desenvolvimento de software, conforme estabelecido pela norma ISO/IEC/NBR 12207, abrange diversas fases essenciais. Inicialmente, na fase de Concepção e Definição de Requisitos, os requisitos do software são identificados e documentados de acordo com as necessidades dos usuários e stakeholders. Em seguida, na fase de Design e Arquitetura, a estrutura e a arquitetura do software são projetadas, considerando a funcionalidade desejada e os requisitos estabelecidos. Posteriormente, durante a fase de Implementação, o código-fonte é desenvolvido e os componentes do sistema são construídos conforme o design previamente estabelecido. Após a implementação, ocorre a fase de Testes e Verificação, na qual são realizados testes para garantir que o software atenda aos requisitos especificados e funcione conforme o esperado. Por fim, na fase de Implantação e Manutenção, o software é entregue aos usuários finais e colocado em operação, enquanto atividades de manutenção são realizadas para garantir seu funcionamento contínuo e corrigir eventuais defeitos ao longo do tempo (ISO, 2017).

Figura 1 - Ciclo de vida padrão de desenvolvimento de software.



Fonte: o autor, adaptado de (ISO, 2017).

Buscando uma melhor compreensão sobre processo de desenvolvimento de software e sua evolução ao longo da história, o **Erro! Fonte de referência não encontrada.** apresenta um resumo dos marcos históricos na evolução do desenvolvimento de software. De forma muito objetiva, Bezerra (2020) apresenta a evolução da história do software a partir da perspectiva de vários autores.

Entre os anos de 1950 e 1968, predominava o desenvolvimento ad hoc de software, caracterizado pela ausência de métodos consistentes e padrões estabelecidos. Cada empresa seguia suas próprias práticas, resultando em projetos muitas vezes sem planejamento adequado (Pressman; Maxim, 2016).

Na década de 1968, ocorreu a "crise do software", onde a demanda crescente e a falta de metodologias consolidadas resultaram em diversos problemas, como custos e prazos ultrapassados, elevados custos de manutenção e insatisfação dos usuários (Pressman; Maxim, 2016).

Em 1970, Winston Royce propôs o modelo cascata, um método sequencial e linear de desenvolvimento de software, onde as atividades são realizadas em etapas sucessivas (Royce, 1987).

No ano de 1986, Barry Boehm introduziu o modelo em espiral, enfatizando a importância de um ciclo de desenvolvimento iterativo e organizado em espiral, onde cada iteração revisita todas as fases do desenvolvimento (Wazlawick, 2019).

Em 1996, foi lançada a Linguagem de Modelagem Unificada - UML, uma linguagem gráfica desenvolvida para estruturar projetos de software, unindo os esforços de Grady Booch, James Rumbaugh e Ivar Jacobson (Booch et al., 2006; Silva et al., 2017).

O ano de 2001 marcou o surgimento das metodologias ágeis, com o Manifesto Ágil estabelecendo os princípios e valores fundamentais para o desenvolvimento de software ágil. Essas metodologias enfatizam entregas rápidas e contínuas, permitindo que o software evolua ao longo do tempo em resposta às mudanças do cliente (Pontes; Arthaud, 2018).

Ao longo da história, o desenvolvimento de software evoluiu significativamente, impulsionado por técnicas e métodos que surgiram como suporte para a área. Esses avanços transformaram completamente a forma como o software era produzido, comercializado e utilizado pelas empresas e clientes, essa transformação, em parte, foi impulsionado pelo surgimento dos métodos ágeis (Bezerra, 2020).

Os métodos ágeis são um conjunto de processos aplicados ao desenvolvimento de software que são iterativos, incrementais, auto-organizados e emergentes. Entre os métodos ágeis mais populares estão Scrum, Kanban, Extreme Programming - XP, Dynamic Software Development Method – DSDM , Test Drive Development -TD - e Disciplined Agile Delivery - DAD (Tavares et al., 2021).

De acordo com Amaral et al. (2011) o conceito 'ágil' pode ser visto como uma abordagem fundamentada em um conjunto de princípios, cujo objetivo é tornar o processo de gerenciamento de projetos mais simples, flexível e iterativo, de forma a obter melhores resultados em desempenho (tempo, custo e qualidade), menos esforço em gerenciamento e maiores níveis de inovação e agregação de valor para o cliente.

Por sua vez, Pressman (2011) leva a concluir que os métodos de desenvolvimento ágeis surgiram como uma resposta às limitações dos métodos de desenvolvimento convencionais, isso considerando que é extremamente difícil prever como um sistema de computador irá evoluir. O mercado está sujeito a mudanças constantes, as necessidades dos usuários evoluem ao longo do tempo e novos concorrentes surgem com frequência. Por essa razão, é altamente improvável que todos os requisitos de um projeto possam ser completamente definidos desde o início. Nesse contexto, é crucial adotar uma abordagem ágil, capaz de se adaptar às mudanças (Hüttermann, 2012).

Em 2008, na conferência Ágil de em Toronto, Patrick Debois, um administrador de sistemas, apresentou uma palestra intitulada "Agile Operations and Infrastructure: How Infra-gile are you?" Ele compartilhou suas experiências sobre a incorporação do Scrum nas operações de TI. Debois estava lidando com a migração de ambientes de um data center e sentia a necessidade de uma abordagem mais ágil, pois alternava entre o desenvolvimento de novas funcionalidades e a resolução de problemas

operacionais. Ele se uniu a Andrew Clay Shafer, que também estava interessado em infraestrutura ágil, para discutir o assunto e dando início ao conceito Desenvolvimento e Operação, que posteriormente passa a ser representado pela sigla DevOps Development and Operations, conforme Davis e Daniels (2016).

De acordo com Hüttermann (2012) Development and Operations - DevOps é a combinação de desenvolvimento, representando os desenvolvedores de software, e garantia de qualidade e operações, representando aqueles que colocam o software em infraestrutura de produção, como administradores de sistema, administradores de banco de dados e técnicos de rede, com o objetivo de agilizar o processo de entrega de software, reduzindo o tempo desde o início do ciclo até o seu final. Sob o mesmo ponto de vista, Bass et al. (2015) definem DevOps como um conjunto de práticas que visam reduzir o tempo entre a confirmação de uma mudança em um sistema e a sua implantação em produção, garantindo alta qualidade.

Para tanto, quatro aspectos essenciais na prática DevOps fazem-se necessários: cultura, automatização, medição e compartilhamento. A cultura do DevOps prioriza as pessoas em relação aos processos e ferramentas, reconhecendo que os softwares são feitos por pessoas e para pessoas. A automatização é vista como uma característica essencial para obter ganhos significativos com o DevOps. A medição é fundamental para avaliar a qualidade e o desempenho do processo de entrega de software. O compartilhamento promovido pelo DevOps cria uma cultura onde as pessoas compartilham ideias, processos e ferramentas (Hüttermann, 2012).

Sob o mesmo ponto de vista, Bass et al. (2015) destacam cinco práticas importantes para a cultura DevOps:

1. Integrar as operações ao time de desenvolvimento garantindo que os requisitos de logs compreensíveis sejam considerados desde o início.
2. Tornar os desenvolvedores mais responsáveis pelo tratamento de incidentes que visa reduzir o tempo entre a descoberta e a resolução de falhas.
3. O uso do mesmo processo de implantação para todos que visa facilitar a identificação do histórico do processo além de promover a alta qualidade de implantação.
4. A adoção da metodologia de implantação contínua capaz de encurtar o tempo entre o código desenvolvido e o código implantado, enfatizando o uso de testes automatizados para garantir a qualidade das mudanças.

5. Transformar a infraestrutura em código, por meio de scripts de implantação, garantindo a qualidade das aplicações implantadas e assegurando que a implantação ocorra conforme planejado, para tanto é importante aplicar práticas de controle de qualidade aos scripts de implantação para garantir que suas especificações estejam corretas.

O Quadro 1 reúne as principais características relacionadas as metodologias ágeis atualmente em uso, essas características combinadas promovem a agilidade e a capacidade de resposta, contribuindo para que as equipes de projeto se adaptem às mudanças e entreguem produtos de alta qualidade de forma mais eficiente.

Quadro 1 - Características das metodologias ágeis.

Característica	Descrição
Desenvolvimento incremental	As metodologias ágeis implementam a melhoria contínua do sistema, adicionando funcionalidades incrementais ao longo do ciclo de desenvolvimento.
Cooperação entre Equipe e Cliente	Existe uma colaboração constante entre a equipe de desenvolvimento e o cliente, com ênfase nos feedbacks e na comunicação efetiva.
Entregas Rápidas de Alta Qualidade	As funcionalidades são entregues de forma rápida, graças à organização da equipe e à ênfase na qualidade do produto. O ciclo de vida do projeto é dividido em entregas frequentes, normalmente a cada 2 a 4 semanas.
Flexibilidade de Escopo do Projeto	O escopo do projeto pode ser ajustado a qualquer momento, permitindo a adaptação às necessidades e prioridades em constante mudança.
Criação de Valor Progressivo	O cliente percebe valor no produto desde o início do projeto, uma vez que são entregues funcionalidades parciais que agregam valor ao sistema.
Adaptabilidade às Mudanças	As mudanças no projeto são esperadas e bem recebidas, com as equipes ágeis se adaptando com facilidade às novas demandas e requisitos.

Fonte: o autor, a partir de Bezerra (2020).

O desenvolvimento de software requer uma abordagem contínua de evolução, onde as boas metodologias de desenvolvimento apresentam elementos intangíveis que desempenham um papel fundamental na construção e evolução de um produto de qualidade. Embora não sejam físicas, essas práticas conceituais têm um impacto significativo na qualidade do software em desenvolvimento. Segundo Vetorazzo (2018) as falhas mais comuns na adoção de uma metodologia podem ser atribuídas aos seguintes fatores:

- Falta de suporte gerencial;
- Pouca experiência ou treinamento insuficiente;
- Boicote e falta de comprometimento da própria equipe;

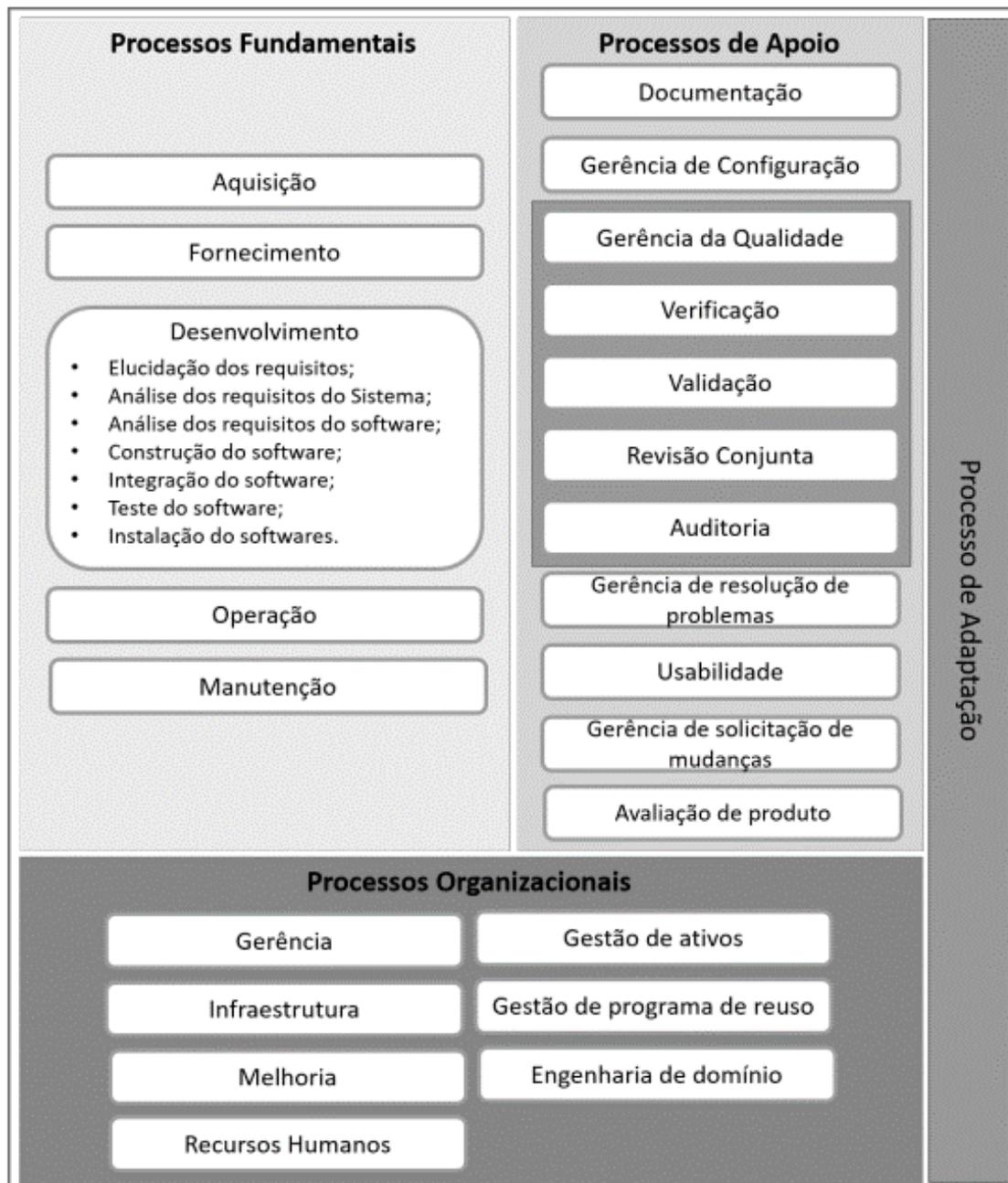
- Falta de reconhecimento da necessidade de melhorias.

O desenvolvimento de software é composto por um conjunto de atividades, ferramentas e modelos utilizados para criar um software. Cada modelo pode ser aplicado usando diferentes metodologias, que definem a sequência das atividades, dos métodos e das ferramentas, além de estabelecer como elas devem se relacionar entre si. Conforme Vetorazzo (2018), essas metodologias são essenciais para orientar e organizar o processo de desenvolvimento, garantindo a eficiência e a qualidade do software resultante.

Um projeto de desenvolvimento de software pode seguir uma das seguintes metodologias estabelecidas, tais como o Project Management Body of Knowledge - PMBOK, o Organizational Project Management Maturity Model - OPM3, o Capability Maturity Model – CMM, o Control Objectives for Information and Related Technologies - COBIT, o Rational Unified Process- RUP e a Unified Modeling Language - UML. O PMBOK define o gerenciamento de projetos como a aplicação de conhecimentos, habilidades, ferramentas e técnicas para as atividades do projeto, visando atender aos requisitos estabelecidos. Essas metodologias fornecem diretrizes e práticas para garantir o sucesso do projeto de software e o cumprimento de seus objetivos (Martins ,2010).

Complementarmente, a NRB ISO/IEC 12207 estabelece um conjunto de diretrizes para o ciclo de vida do software. Ela inclui processos e atividades a serem realizados durante o desenvolvimento, configuração e manutenção do software, permitindo que as organizações definam seus próprios processos de ciclo de vida do software, esses processos são agrupados de acordo com seus objetivos principais no ciclo de vida do software. Essa categorização resulta em quatro classes/grupos de processos: processos fundamentais, processos de apoio, processos organizacionais e processos de adaptação Costa (2022), conforme Figura 2:

Figura 2 - Processos da NBR ISO/IEC 12207.



Fonte: o autor, adaptado de ISO (2017).

Os processos fundamentais, citados na Figura 2, são os que iniciam o ciclo de vida de um produto ou serviço e são basicamente todas as atividades que a empresa executa nos serviços de desenvolvimento, manutenção ou operação de software. Esses processos comandam a execução de todos os outros processos (Júnior et al., 2009). Os processos de apoio são os que auxiliam outro processo, garantem a qualidade. Processos organizacionais, são os que assessoram na organização e gerência e os processos de adaptação são os que definem e determinam as atividades necessárias para que sejam realizadas adaptações necessárias, podendo ser

aplicadas a projetos, organizações, cultura, métodos, técnicas ou padrões (Vacari, Prikladnicki,2014).

A NRB ISO/IEC 12207 (2017) destaca três importantes conceitos relacionados a ciclo de vida, modelo de desenvolvimento e processo de desenvolvimento, segundo ISO/IEC. O ciclo de vida descreve as fases do desenvolvimento do software, enquanto o modelo de desenvolvimento define a abordagem ou estratégia para organizar essas fases. O modelo de desenvolvimento de software, também, estabelece a estrutura geral para o desenvolvimento do software, enquanto o processo de desenvolvimento detalha as atividades específicas a serem realizadas em cada fase do ciclo de vida, independentemente do modelo utilizado.

Segundo Sommerville (2019), há uma ligação entre padrões de produto e de processo. Os padrões de produtos aplicam-se à saída do processo de software, e os padrões de processo incluem atividades específicas de processos que asseguram que os padrões sejam seguidos. Ainda segundo o estudo citado neste parágrafo, existe um conjunto de normas técnicas e modelos de qualidade de software estabelecidos internacionalmente que podem apoiar a organização no aprimoramento dos processos de desenvolvimento de software.

No Quadro 2 é possível verificar um recorte das principais normas e modelos, que abrangem diversos aspectos relacionados à qualidade do software. Essas referências são importantes para orientar e direcionar as práticas de desenvolvimento de software de forma a garantir a conformidade e a excelência dos produtos desenvolvidos.

Quadro 2 - Normas internacionais e modelos de qualidade de software.

Norma	Descrição	Tipo
ISO 12207	Processos de ciclo de vida do software	Processo
ISO 90003	Diretrizes para aplicação da norma ISO 9001 ao desenvolvimento, fornecimento e manutenção de software	Processo
ISO 15504	Projeto da ISO/IEC para avaliação dos processos de desenvolvimento de software	Processo
CMMI	Capability Maturity Model Integrated. Modelo do SEI que estende o CMM para avaliação de processos de software	Processo
MPS-BR	Modelo Brasileiro de qualidade de processo de software baseado nas normas ISO 12207 e 15504 e no modelo CMMI	Processo
ISO 9126	Norma para qualidade de produtos de software	Produto
ISO 14598	Guias para avaliação de produtos de software	Produto

Fonte: o autor adaptado de Vettorazzo (2018).

O modelo Capability Maturity Model Integration - CMMI foi criado pelo Software Engineering Institute SEI que é um centro de pesquisa e desenvolvimento patrocinado pelo Departamento de Defesa dos Estados Unidos que provê uma prática avançada

de engenharia de software qualificando graus de qualidade de software e tem foco na capacidade de maturidade de processos de software, ou seja, serve de guia para a melhoria de processos. O CMMI Institute apresenta cinco níveis de maturidade citados no Quadro 3, que estabelecem patamares de evolução de processos, caracterizando estágios de melhoria da implementação de processos:

Quadro 3 - Níveis de maturidade do CMMI.

Nível	Descrição
Nível 1 – Inicial	Os processos são caóticos e a organização não apresenta ambiente estável de desenvolvimento. Se existem padrões, eles não são seguidos, e os projetos, geralmente, apresentam problemas de execução dos projetos no que diz respeito a prazos e custos.
Nível 2 - Gerenciado	Os projetos possuem requisitos gerenciados e processos planejados. Os requisitos, processos e serviços são gerenciados e há uma preocupação em seguir o que está determinado nos planos.
Nível 3 – Definido	Os processos são caracterizados e bem entendidos. Existe a padronização dos processos, o que possibilita que os produtos gerados sejam consistentes. Nesse estágio, os procedimentos são padronizados e devem prever a aplicação em diferentes projetos.
Nível 4 - Gerenciamento da qualidade	Os processos selecionados contribuem para o desempenho geral dos demais processos. Eles são controlados com base em estatísticas e técnicas quantitativas.
Nível 5 - Otimizado	os processos são continuamente avaliados e melhorados. Os objetivos quantitativos são estabelecidos e revisados com frequência.

Fonte: o autor, adaptado de CMMI Institute (2018).

Uma alternativa ao modelo criado pelo SEI é o Guia de Melhoria do Processo do Software Brasileiro– MPS.BR que é um programa mobilizador, de longo prazo, criado em dezembro de 2003 e coordenado pela Associação para Promoção da Excelência do Software Brasileiro - SOFTEX, que teve o apoio do Ministério da Ciência, Tecnologia, Inovações e Comunicações - MCTIC, Financiadora de Estudos e Projetos - FINEP, Serviço Brasileiro de Apoio às Micro e Pequenas Empresas - SEBRAE e Banco Interamericano de Desenvolvimento - BID/FUMIN (MPS et al., 2021).

Segundo Cec et al. (2006), o modelo de Melhoria do Processo de Software Brasileiro - MPS–BR é uma metodologia voltada à área de desenvolvimento de sistemas. A exemplo do modelo CMMI, o MPS-BR também apresenta níveis de maturidade, conforme ilustrado no Quadro 4.

Ambos os modelos são referências amplamente utilizados na área de desenvolvimento de software, com o objetivo de melhorar a qualidade e a eficácia dos processos organizacionais, fornecendo um referencial sólido para as organizações avaliarem, aprimorarem e certificarem seus processos de software. Eles auxiliam na identificação de áreas de melhoria, na definição de metas e na implantação de práticas que visam aumentar a eficiência, a qualidade e a produtividade das atividades de desenvolvimento de software.

Quadro 4 - Níveis de maturidade do MPS.BR.

Nível	Descrição
Nível A	Em otimização: um processo nesse nível é otimizado por meio da realização de mudanças e adaptações de forma ordenada, com foco na efetividade para atender mudanças nos objetivos do negócio.
Nível B	Gerenciado quantitativamente: a organização passa a ter uma visão quantitativa quanto ao desempenho dos seus processos.
Nível C	Definido: são implementados três novos processos com a mesma capacidade dos processos já implantados: Análise e decisão e resolução - ADR; Desenvolvimento para a reutilização - DRU; Gerência de riscos - GRI.
Nível D	largamente definido: estar nesse nível implica apenas a definição e a implementação de cinco novos processos com a mesma capacidade dos processos já implantados: Desenvolvimento de requisitos - DRE; Integração do produto - ITP; Projeto e construção do produto - PCP; Validação - VAL; Verificação - VER.
Nível E	Parcialmente definido: o foco principal é a padronização dos processos da organização, que devem ser definidos a partir dos processos e das melhores práticas já existentes.
Nível F	Gerenciado: existe o apoio à gestão do projeto referente à garantia da qualidade e medição. Esses processos possibilitam maior visibilidade de como os artefatos são produzidos.
Nível G	Parcialmente gerenciado: a implementação deve ser executada com cautela devido à mudança de cultura organizacional e à definição do conceito acerca do que é projeto para a organização

Fonte: o autor, adaptado de Mps et al. (2021).

A busca incessante pela melhoria contínua, impulsionada pelo crescente aumento da demanda por tecnologia, levou à necessidade de se estabelecer padrões e desenvolver métodos adequados. Assim, com o intuito de atender a essa demanda, foi criado o Software Engineering Body of Knowledge - SWEBOK, que é um guia que reúne e aplica as melhores práticas de engenharia de software coletadas ao longo de décadas e em diversos países, onde o principal objetivo consiste em estabelecer um conjunto de normas e critérios para orientar as atividades de engenharia de software (IEEE, 2014).

Além das iniciativas relacionadas a qualidade, padronização e eficiência, o processo de desenvolvimento de software busca diuturnamente introduzir práticas relacionadas à segurança, por se tratar de uma medida necessária para proteger dados sensíveis, garantir a privacidade dos usuários e prevenir potenciais ataques cibernéticos. Um software vulnerável pode resultar em violações de segurança, perdas financeiras, danos à reputação da empresa e impactos negativos para os usuários.

No ciclo de desenvolvimento de software, a segurança é um conceito importante, e é por isso que os desenvolvedores de software devem integrar a

segurança desde o início do processo de desenvolvimento de software. Ao longo do processo, a segurança deve ser levada em consideração para garantir que o produto de software tenha uma segurança ideal, como discutido por Mohaddes Deylami et al. (2015). O monitoramento e a avaliação contínuos podem levar à detecção precoce de erros ou problemas, implementação rápida de ações apropriadas e, o mais importante, produção do produto de software seguro.

O estudo de Devanbu e Stubblebine (2000) destaca a importância de se manter o foco em questões de segurança em todas as fases do desenvolvimento de software. Para tanto, os autores delinham como focar nos processos de requisitos e projeto de software, que são os dois níveis iniciais de desenvolvimento. Segundo Alenezi e Almuairfi (2019b), enfatizar essa importância visa reduzir e eliminar os riscos e erros tanto quanto possível. O projeto de software é um processo sistemático em que a avaliação de riscos de segurança deve ser integrada em cada fase

Sob o mesmo ponto de vista, Gupta (2003), discutiu estratégias de gerenciamento de riscos nas fases iniciais do desenvolvimento de software e informou que o gerenciamento tardio de riscos indiretamente representa maiores ameaças ao desenvolvimento de software seguro.

No mundo do software, é essencial que os profissionais compreendam os riscos de segurança associados ao sistema de software em desenvolvimento Abunadi e Alenezi (2016). A segurança de software é uma filosofia abrangente que engloba conceitos, listas de verificação, ferramentas, processos e métodos, os quais desempenham papéis importantes na arquitetura, projeto, codificação, construção, verificação e teste de sistemas de software (McGraw, 2016).

Os objetivos fundamentais de segurança, como confidencialidade, integridade e disponibilidade, conhecidos coletivamente como a tríade CIA, bem como o controle e monitoramento de riscos de segurança e conceitos como ativos, ameaças e vulnerabilidades, são aspectos cruciais a serem considerados como observador pelos autores Al Hamed e Alenezi (2016). Mesmo quando medidas de segurança são implementadas, é importante estar ciente de que existem riscos adicionais que um software pode enfrentar e que requerem atenção e soluções apropriadas (Boehm, 1991).

Conforme apontado por Sommerville (2019) o Software Development Life Cycle – SDLC é um processo que define as etapas para o desenvolvimento de software, desde a concepção até a implantação e a manutenção. A segurança precisa ser considerada em cada fase do SDLC para evitar complexidade, problemas e garantir a proteção do software contra ameaças (Abdul Karim et al., 2016) ilustram que a

segurança é essencial para fornecer autenticação, integridade e disponibilidade no SDLC.

Uma vez que erros na segurança de software são comuns e as ameaças têm aumentado, é necessário que a segurança seja considerada na fase inicial do ciclo de desenvolvimento de software e que os princípios de segurança sejam aplicados como um elemento padrão do ciclo, de tal modo que a adoção de práticas que reduzem defeitos de software leve à minimização do risco potencial decorrente da falta de atenção à segurança ao longo do processo de desenvolvimento.

Vários modelos de risco foram desenvolvidos com o objetivo de melhorar o nível de segurança do SDLC. Neves e Silva Neves e Silva (2016) discutiram que, para projetos de software, a análise de informações relacionadas ao risco pode levar a uma melhoria no processo de tomada de decisão dos gerentes.

A gestão de riscos de segurança no ciclo de desenvolvimento de software envolve o uso de vários modelos, entre eles o Waterfall, Spiral, V-Model e Multilevel Security Spiral - MSS. O modelo Waterfall é uma metodologia de desenvolvimento que envolve um processo contínuo, no qual o resultado de uma determinada fase de desenvolvimento se torna a entrada para as demais, conforme mostrado em Pawar (2015). O modelo Spiral é caracterizado como um modelo orientado pelo risco e é usado para orientar o processo de construção de sistemas. O modelo V-Model, também conhecido como modelo em "V", é usado para descrever a série de etapas no ciclo de desenvolvimento. A parte esquerda do "V" representa a decomposição dos requisitos e a formação das especificações do sistema, enquanto a parte direita representa a integração e a verificação das partes. A forma "V" ilustra como cada fase de design está associada a uma contraparte correspondente na fase de teste, conforme explicado em Pezzotta et al. (2012).

Essas medidas, que incluem a implementação metodológica de fases bem definidas e a correlação entre desenvolvimento e testes, contribuem significativamente para a construção de softwares seguros e confiáveis, abordando efetivamente os riscos de segurança desde as fases iniciais do ciclo de desenvolvimento. Na visão de Hanif et al. (2021) a detecção de vulnerabilidades em software está ganhando rapidamente popularidade na comunidade de pesquisa e na indústria, especialmente entre aqueles que atuam na área de segurança da computação. O aumento do interesse por esse tema se deve ao aumento dos ataques de segurança cibernética nos últimos anos, que demandam mais pesquisas na detecção de vulnerabilidades em software.

Ainda segundo o autor, diante do crescente número de ciberataques na última década que decorre de 2011 a 2021, os sistemas web têm se tornado alvos frequentes de exploração, acarretando consequências desastrosas para as empresas. Infelizmente, muitos projetos de desenvolvimento web são concebidos sem uma devida consideração à segurança, o que aumenta significativamente o risco de perda de ativos, interrupção dos serviços, exposição de informações sensíveis e comprometimento do sistema como um todo.

Neste contexto de crescente preocupação com a segurança em desenvolvimento web, destaca-se a contribuição do Projeto de Segurança de Aplicações Web Abertas (Open Web Application Security Project - OWASP). Este projeto oferece um relatório detalhado que visa proporcionar aos desenvolvedores e profissionais da área uma compreensão detalhada das dez principais vulnerabilidades presentes em softwares web, e fornece diretrizes e ações práticas para mitigar essas fragilidades durante o processo de desenvolvimento de software, conforme documentado por Alenezi e Almuairfi (2019a).

O OWASP é uma organização sem fins lucrativos que se concentra em melhorar a segurança de software Ghozali et al. (2019). As diretrizes da OWASP são aplicadas ao longo das fases do ciclo de vida do SDLC no desenvolvimento de aplicações, que incluem o planejamento do sistema, análise do sistema, design do sistema, implementação e teste (Sedek et al. , 2009).

A OWASP fornece várias ferramentas, guias e metodologias de teste para segurança cibernética sob uma licença de código aberto, o OWASP Testing Guide - OTG (Dirgahayu et al. , 2015). O OTG é uma ferramenta valiosa para avaliação e aprimoramento da segurança de aplicações web. Ele é composto por três componentes principais: o framework de teste OWASP, a metodologia de teste de aplicações web e o relatório de avaliação do sistema. O framework de teste OWASP é utilizado para auxiliar os desenvolvedores no processo de construção de aplicações web com foco na proteção e segurança. A metodologia de teste de aplicações web pode ser aplicada de forma independente ou como parte integrante do framework de teste. Utilizando o método de teste de penetração, os desenvolvedores podem avaliar a segurança do sistema da aplicação web desenvolvida, garantindo que os aspectos de proteção e segurança sejam adequadamente abordados. Com o auxílio do OTG, os desenvolvedores podem tomar medidas proativas para identificar e corrigir vulnerabilidades antes que sejam exploradas, fortalecendo assim a segurança do sistema e protegendo os dados sensíveis da aplicação web de acordo com os estudos de Pratama e Wiradarma (2019).

A OWASP é conhecida por seu 'Top 10 de Vulnerabilidades de Aplicações Web', que é uma lista das dez vulnerabilidades mais críticas em aplicações web. A lista é atualizada periodicamente e é amplamente usada por desenvolvedores e profissionais de segurança para identificar e corrigir vulnerabilidades em seus aplicativos, sendo, segundo a OWASP (2021) um ponto de partida para cuidados relacionados ao desenvolvimento de software.

A metodologia adotada para a elaboração da Top 10 do OWASP demonstra um compromisso em utilizar dados concretos para identificar e classificar as principais vulnerabilidades de segurança de aplicativos web. Ao combinar dados de contribuição e a perspectiva da comunidade de especialistas em segurança, obteve-se uma visão abrangente e atualizada sobre os riscos mais relevantes no cenário de segurança de aplicativos, conforme estabelecido pelo OWASP (2021).

A estrutura das categorias da Top 10 do OWASP foi construída com uma abordagem mais focada na causa raiz das vulnerabilidades, fornecendo orientações claras para a identificação e correção dos problemas. A coleta e análise de dados seguiram um processo transparente e colaborativo, envolvendo organizações do setor de segurança, desenvolvedores de ferramentas de teste e programas de recompensas por identificação de falhas. Os fatores dos dados, como taxa de incidência, cobertura, peso de explorabilidade e impacto ponderado, forneceram uma métrica objetiva para classificar as principais vulnerabilidades listadas pelo OWASP com base em seu risco potencial, como documentado por Wiradarma e Sasmita (2019).

Na visão de OWASP (2021) a preocupação com a segurança é relevante, uma vez que estudos indicam que aproximadamente 70% das vulnerabilidades de segurança são introduzidas antes mesmo da fase de codificação. Isso ocorre principalmente devido à prática comum de desenvolvedores de aplicativos compartilharem e reutilizarem software legado de terceiros, presumindo que seja seguro e confiável. Consoante com Fong e Okun (2007) e Bisht et al. (2017), essas vulnerabilidades surgem principalmente devido a decisões iniciais de design ou desenvolvimento, destacando a importância de considerar a segurança desde as fases iniciais do projeto.

É importante destacar que o desenvolvimento de software abrange uma série de práticas e metodologias que visam à criação e manutenção de sistemas de software eficazes. Highsmith (2009) e Brooks (1995) destacam a natureza adaptativa e iterativa do desenvolvimento de software, que exige flexibilidade e responsividade às mudanças. Essa abordagem ressalta a importância de ajustes e adaptações

constantes ao longo do ciclo de vida do desenvolvimento. Gomaa (2011) reforça a importância do design no processo de desenvolvimento de software, argumentando que um design bem elaborado é crucial para a manutenção, a extensibilidade e a qualidade geral do produto de software. Myers, Sandler e Badgett (2011) complementam essa visão ao enfatizar a importância dos testes unitários e da codificação como etapas críticas que asseguram a funcionalidade e a confiabilidade do software. Em conjunto, esses autores oferecem uma visão ampla do desenvolvimento de software, destacando a necessidade de abordagens adaptativas, design robusto e testes rigorosos.

2.2 Gestão de Riscos

De acordo com Renn (1998a) é comum compreender o risco como a possibilidade de algo não dar certo, estando associado à probabilidade de ocorrência de situações adversas, seja por eventos naturais ou por atividades humanas. Nesse contexto, é válido considerar que o risco está presente em todas as áreas da vida, pessoal, profissional e organizacional, podendo resultar tanto em perdas quanto em oportunidades (IBGC, 2007).

Seguindo a origem etimológica, a palavra "risco" tem sua raiz no italiano "risicare", que por sua vez deriva das palavras latinas "risicu" ou "riscu", que significam "ousar" ou "to dare" em inglês, essa etimologia sugere que o risco não é um caminho predestinado, mas sim uma escolha audaciosa, de acordo com o próprio significado do termo "ousar" (IBGC, 2007).

Há evidências históricas que remontam à Grécia Clássica e ao Império Romano, sugerindo que a preocupação com os riscos é uma questão antiga. De acordo com Simkins e Ramirez (2008) há indícios do surgimento dos princípios de preocupação com os riscos por volta de 2000 a.C, tanto que já existiam práticas de mitigação de riscos na antiga Babilônia, como indenizações em caso de perdas por roubos e inundações (Hubbard, 2009).

O risco como ciência teve suas origens no século XVII, durante o período do Renascimento. Foi nessa época que Blaise Pascal desenvolveu a "Teoria da Probabilidade" em 1654, enquanto buscava compreender os jogos de azar. Pascal também criou o "Triângulo de Pascal", que permite determinar a probabilidade de ocorrência de diferentes resultados em um determinado número de tentativas (Bernstein, 1997).

No entanto, foi a partir de 1975 que o tema do risco ganhou maior destaque com a atribuição de responsabilidades à alta administração em relação às políticas de gestão de riscos. Esse marco foi impulsionado pelo desenvolvimento de abordagens mais sistemáticas e estruturadas para a gestão de riscos, como a publicação do "Orange Book" pelo Ministério da Defesa do Reino Unido em 1975, que estabeleceu diretrizes para a gestão de riscos em projetos governamentais. Essa iniciativa ajudou a estabelecer uma base para a adoção de práticas de gestão de riscos em diferentes setores e organizações, fortalecendo a importância do tema e incentivando sua incorporação nas estratégias empresariais (Fraser Quail, 2021).

Durante o século XX, a gestão de riscos foi difundida, estudada e aplicada em diversos setores, incluindo saúde, finanças, seguros de vida, entre outros. Para essas organizações, a gestão de riscos não é considerada algo negativo, pelo contrário, é fundamental para o sucesso do negócio. Em iniciativas desses setores, a abordagem de riscos é útil, uma vez que os lucros dependem da identificação e aproveitamento de oportunidades atraentes, sempre balanceadas por meio de uma avaliação cuidadosa e calculada dos riscos envolvidos. Dessa forma, a gestão de riscos se torna um componente para auxiliar uma tomada de decisão eficiente e eficaz, podendo resultar em benefícios financeiros e proteção contra eventos adversos.

A partir dessas abordagens conceituais previamente citadas sobre risco, percebe-se que ao longo do tempo surgiram definições mais específicas do termo risco. Assim, com o intuito de aprofundar o estudo desse conceito no Quadro 5 são apresentadas as principais definições encontradas durante a análise da literatura para esse trabalho.

É importante destacar que o Quadro 5 apresentado não abrange todas as definições de risco disponíveis na literatura. No entanto, ao analisar a origem etimológica, é possível identificar uma certa convergência em relação ao termo "risco", embora não haja consenso sobre sua definição, tanto na literatura científica quanto no entendimento geral, não sendo possível estabelecer um único e preciso conceito de risco (Renn, 1998b).

Para fins deste trabalho será utilizado o conceito de risco trazido por Boehm (1991) onde o risco é entendido como um resultado negativo do exercício de uma vulnerabilidade, levando em consideração tanto a probabilidade quanto o impacto da ocorrência.

Quadro 5 - Conceitos de Risco.

Autor	Definição
Boehm (1991)	Risco é entendido como um resultado negativo do exercício de uma vulnerabilidade, levando em consideração tanto a probabilidade quanto o impacto da ocorrência.
Chittister e Haimes (1993)	The probability and consequence of not achieving some defined program goal-such as cost, schedule, or technical performance.
Jorion (1998)	Risco é a probabilidade de acontecerem resultados inesperados.
RENN (1998b)	Risks refer to the possibility that human actions or events lead to consequences that affects of what humans value.
Chittister et al. (1992)	Risk is the net negative of the exercise of a vulnerability, considering both the probability and the impact of the occurrence.
COSO ERM (2004)	A possibilidade de eventos ocorrerem e impactarem negativamente a conquista dos objetivos da organização
OFFICE OF GOVERNMENT COMMERCE (2009)	Risco é um evento imprevisível ou um conjunto de eventos que, se ocorrerem, terão um impacto no atingimento dos objetivos.
ISO 31000 (2018)	Risco é o efeito da incerteza nos objetivos, onde um efeito é um desvio em relação ao esperado, podendo ser positivo, negativo ou ambos, e pode abordar, criar ou resultar em oportunidades e ameaças.
IRGC (2000)	A possibilidade de consequências prejudiciais ou perdas, juntamente com sua probabilidade

Fonte: o autor, a partir de dados da pesquisa.

Além de compreender o conceito de risco, é igualmente importante considerar outros termos conforme definido no Quadro 6. Esses termos auxiliam na compreensão do campo e promovem uma base sólida para o estudo e a aplicação da gestão de riscos.

Quadro 6 - Termos e Definições.

Autor	Termo	Definição
ISO 31000:2018	Gestão de riscos	Atividades coordenadas para dirigir e controlar uma organização no que se refere a riscos
	Parte interessada	Pessoa ou organização que pode afetar, ser afetada ou perceber-se afetada por uma decisão ou atividade
	Fonte de risco	elemento que, individualmente ou combinado, tem o potencial para dar origem ao risco
	Evento	ocorrência ou mudança em um conjunto específico de circunstâncias
	Consequência	Uma consequência pode ser certa ou incerta e pode ter efeitos positivos ou negativos, diretos ou indiretos, nos objetivos.
	Probabilidade	Chance de algo acontecer
	Controle	Medida que mantém e/ou modifica o risco
PMBOK (2017)	Ameaça	Possível evento adverso que pode afetar negativamente os objetivos
	Oportunidade	Possibilidade favorável que pode ser aproveitada para melhorar o desempenho, a eficiência ou a competitividade

Fonte: o autor, a partir de dados da pesquisa.

Igualmente, a categorização dos riscos também não segue uma tipologia específica, devido às diversas perspectivas dos autores em relação ao conceito de risco. A literatura classifica diversos tipos de riscos, levando em consideração as diferentes formas de ocorrência de perdas e como elas impactam as organizações. É importante salientar que é comum haver subclassificações dentro dessas categorias (Ferro, 2015).

A compreensão dos conceitos de risco, suas categorias, os principais termos e definições é essencial para lidar de forma adequada com as incertezas e as oportunidades que podem afetar tanto a esfera pessoal quanto a profissional e organizacional.

Neste contexto, apresentam-se, na sequência, os principais tipos e categorias de riscos existentes e como eles podem ser compreendidos a partir das diversas perspectivas dos autores. Essa abordagem busca permitir uma melhor compreensão dos aspectos específicos de cada tipo de risco.

Os autores Jorion (1998) e Marshall (2002) apresentam diferentes categorias de risco. Jorion (1998) propõe três categorias: operacional, estratégico e financeiro. Por sua vez, Marshall (2002) destaca os riscos de mercado, crédito, estratégicos e operacionais, focando especificamente nas instituições financeiras. Baraldi (2004) classifica os riscos em mercado, crédito e operacional, enquanto Simkins e Ramirez (2008) destacam os riscos operacionais, de mercado e de crédito. Zonatto e Beuren (2010) expandem a classificação para incluir riscos estratégicos, de liquidez, legais, de imagem, ambientais, de revisão tarifária, P&D e de produtos.

Um critério alternativo para a classificação dos riscos envolve dois componentes interrelacionados: pessoas e reputação. Em função da inexistência de uma classificação de riscos consensual e aplicável a todas as organizações, a classificação deve ser adaptada às características específicas de cada organização, considerando sua indústria, mercado e setor de atuação (IBGC, 2007).

Ao longo dos anos, pesquisadores categorizaram empiricamente os tipos de riscos em projetos (Addison e Vallabh, 2002; Barki et al., 1993; Boehm, 1991; Han e Huang, 2007; McFarlan, 1982; Schmidt et al., 2001; Sharma et al., 2011; Wallace et al., 2004). Esses estudos sobre riscos em projetos focaram principalmente no aspecto técnico do gerenciamento de riscos, abordando listas de verificação, processos, ferramentas e estruturas (Singh, 2021).

Buscando promover a criação de um padrão, o Instituto Brasileiro de Governança Corporativa - IBGC classifica os riscos em relação à sua origem e natureza.

Quanto à origem, os riscos são categorizados como externos e internos. Os riscos externos são provenientes do ambiente macroeconômico, político, social, natural ou setorial, podendo causar impactos na organização. Já os riscos internos têm origem nos processos, quadro funcional e tecnologia utilizada pela própria organização.

Quanto à natureza, os riscos são classificados como estratégicos, operacionais e financeiros. Os riscos estratégicos estão relacionados às decisões tomadas pela alta administração e podem resultar em grandes perdas econômicas (IBGC, 2007). Da mesma forma, o risco operacional é definido como a possibilidade de ocorrência de perdas resultantes de falha, deficiência ou inadequação de processos internos, pessoas e sistemas ou de eventos externos (Brasil, 2006).

Segundo Haidabrus (2022), sem uma metodologia clara de gerenciamento de riscos, existe uma probabilidade de serem ignorados riscos importantes. Portanto, para alcançar os objetivos e o sucesso do projeto, é recomendado implementar um processo efetivo de gerenciamento de riscos.

Uma abordagem viável para lidar com os riscos é utilizar uma taxonomia de risco, que oferece uma estrutura para organizar dados e analisar diversos aspectos do risco em projetos de software. A taxonomia de riscos desenvolvida pelo SEI é uma opção a ser considerada, pois abrange os processos típicos do ciclo de vida do software e inclui as áreas gerais de risco do projeto relacionadas às características do software, ao ambiente e aos processos de desenvolvimento, bem como às limitações do projeto, de tal modo que utilizá-la pode ser uma alternativa para lidar com os riscos no contexto do desenvolvimento de software. A Taxonomia do SEI será melhor detalhada na sessão 2.3.

A partir do conhecimento do conceito de risco e suas formas de classificação, torna-se possível adentrar o processo de gestão de risco, que perpassa a identificação dos fatores de risco quanto a análise da probabilidade e impacto potencial de cada fator de risco, a priorização dos fatores de risco e desenvolvimento de estratégias de mitigação para reduzir a probabilidade e minimizar o impacto negativo caso um fator de risco torne-se um problema. Métodos de avaliação de risco (por exemplo, julgamento de especialistas, dados históricos, árvores de decisão e simulações de processos) podem ser usados para identificar e avaliar os fatores de risco (IEEE , 2014).

A gestão de riscos deve ser compreendida como um conjunto de atividades coordenadas que visam dirigir e controlar uma organização em relação aos riscos, com o propósito de criar e proteger valor. Essa abordagem deve estar integrada a todas as atividades da organização, sendo capaz de antecipar, reconhecer e responder de maneira adequada aos riscos. O objetivo é garantir que a gestão de riscos seja uma parte essencial do funcionamento da organização, promovendo uma cultura de risco e contribuindo para a obtenção de resultados positivos (Eduardo, 2019), sendo melhorada continuamente por meio do aprendizado e experiência ABNT (NBR 31000:2018).

Desta forma, a gestão de riscos tem como propósito apoiar a organização na integração dos processos de gestão de riscos em atividades e funções relevantes, criando um processo iterativo que auxilia a estabelecer estratégias, alcançar objetivos a partir de decisões fundamentadas, onde a eficácia da gestão de riscos depende de sua integração adequada na governança e em todas as atividades da organização (ABNT, 2018).

É adequado lembrar que riscos e incertezas são inerentes aos projetos, pois, por definição, um projeto é único e, portanto, enfrenta fatores desconhecidos. No entanto, vários autores afirmam que há uma infinidade de estudos focados na gestão de riscos em detrimento da gestão da incerteza (Meyer, Loch e Pich, 2002; Pich, Loch e Meyer, 2002; Atkinson, Crawford e Ward, 2006; Perminova, Gustafsson e Wikstrom, 2008; Cleden, 2009), e que estratégias para gerenciar riscos e incertezas podem exigir abordagens diferentes De Carvalho e Rabechini Junior (2015).

As organizações têm seguido o processo de gestão de riscos, uma abordagem sistemática de identificação, análise e resposta aos riscos Kutsch (2008) para reduzir a exposição a potenciais perdas (Brown et al., 2000) (DU et al., 2010) (PMBOK, 2017).

Vários estudos têm sido realizados para compreender o impacto da gestão de riscos de projetos no sucesso dos mesmos, embora a maioria deles enfatize o aspecto técnico, recomendando que as organizações desenvolvam processos, técnicas e ferramentas de gestão de riscos (De Carvalho e Rabechini Junior, 2015). A gestão de riscos atualmente tem um baixo investimento nas organizações quando comparada a outras áreas da gestão de projetos. Em muitos casos, a gestão de riscos é apenas uma adição padronizada de um valor percentual incluído no custo do produto final (Filippetto et al., 2021).

De acordo com a ABNT (2018), uma estrutura de gestão de riscos deve levar em consideração os seguintes elementos:

- a) uma compreensão do contexto da organização;

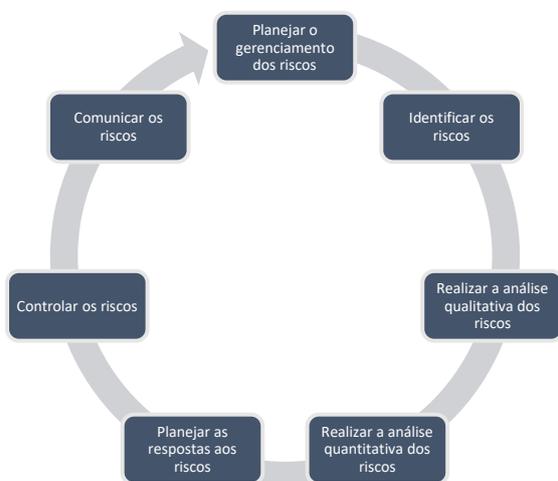
b) a criação de uma política de gestão de riscos;
c) responsabilização;
d) integração com os processos organizacionais;
d) recursos necessários; e
e) a criação de mecanismos de comunicação e reporte internos e externos à organização (D. Jocasta, 2016).

Neste contexto, a implementação da gestão de riscos exige certos cuidados, e, portanto, a empresa deve seguir uma série de etapas para garantir um início adequado e manter o processo ao longo de todas as fases. Hillson (2006), IBGC (2007) e Protiviti (2006) reforçam que, para implantar um modelo de gestão de riscos, deve-se pensar em uma forma de facilitar o gerenciamento do risco, o que proporciona benefícios para a organização.

Existem diversas metodologias de gestão de riscos disponíveis, contudo, independentemente da metodologia algumas etapas básicas são essenciais para identificar e gerenciar os riscos. Essas etapas incluem o planejamento, identificação, análise, avaliação, tratamento e monitoramento dos riscos (Dionne, 2013; Fadun, 2013; Kutsch e Hall, 2009; NBR ISO 31000:2018).

A Figura 3 fornece uma visão geral do processo de Gerenciamento dos Risco (PMBOK, 2017) e (OGC, 2009). De acordo com Neves; Silva; Salomon; Silva e Sotomonte (2014) as etapas apresentadas são semelhantes às etapas encontradas em outras abordagens de gestão de riscos, havendo um consenso entre as principais atividades que compõem esse processo.

Figura 3 - Etapas da Gestão de Risco.



Fonte: o autor, adaptado de PMBOK (2017).

Em cada uma dessas etapas é possível utilizar um conjunto de técnicas, ferramentas ou conceitos, conforme já destacado por (IEEE, 2014), os quais constituem uma estrutura de trabalho com funções pré-estabelecidas, adaptáveis à situação e à organização. Essas práticas podem incluir a aplicação de Canvas¹, Business Plans², Matriz SWOT³, Objective and Key Results – OKR⁴, ROADMAP⁵, Matriz Eisenhower⁶ entre outros, sendo extremamente úteis para estabelecer as estratégias iniciais de uma organização (dDinamica, 2018).

A ABNT fornece orientações sobre a seleção e aplicação de técnicas sistemáticas para o processo de avaliação de riscos. Em meio as principais técnicas destacam-se a técnica Delphi que é um procedimento para obter um consenso de um grupo de especialistas, sendo a lista de verificação uma ferramenta que pode ser utilizada para identificar perigos e riscos ou para avaliar a eficácia de controles em qualquer estágio do ciclo de vida de um produto, processo ou sistema especialistas. Ainda segundo o autor, outro método de coletas de um amplo conjunto de ideias é o brainstorming, que pode ser estimulado através de instruções ou por técnicas de entrevista.

Existe dois tipos de entrevistas segundo , a entrevista estruturada, na qual os entrevistados são individualmente solicitados a responder a um conjunto de questões pré-elaboradas constantes em um roteiro de instruções. Este método incentiva o entrevistado a considerar uma situação a partir de uma perspectiva diferente, ajudando, assim, a identificar riscos sob este novo ponto de vista. Por outro lado, há a entrevista semiestruturada, que é similar à estruturada, porém oferece mais liberdade para o entrevistador conduzir uma conversa que explore questões emergentes. Esta técnica, inclusive, foi utilizada nesta pesquisa, permitindo a obtenção de informações sobre atributos indicativos de riscos no processo de desenvolvimento de software.

A implementação do processo de gestão de risco, juntamente com as ferramentas, técnicas e conceitos, é uma etapa empregada para uma abordagem eficaz dessa gestão. Essa implementação pode ser guiada por uma norma ou padrão,

¹ O Canvas é um modelo visual usado para descrever o modelo de negócios de uma empresa de forma concisa. Foi desenvolvido por Alexander Osterwalder e Yves Pigneur em seu livro "Business Model Generation" (Osterwalder; Pigneur, 2010).

² Os Business Plans são documentos detalhados que descrevem as metas, estratégias e finanças de uma empresa. São utilizados como guias para o crescimento do negócio. Autores renomados na área incluem Tim Berry e Eric Ries (Berry, 2014).

³ A Matriz SWOT é uma ferramenta de análise que identifica pontos fortes, pontos fracos, oportunidades e ameaças de uma empresa. Foi desenvolvida por Albert Humphrey e é amplamente utilizada na gestão estratégica (Humphrey, 2017).

⁴ O OKR é uma metodologia de definição e acompanhamento de metas. Foi popularizado por John Doerr, que o introduziu na Google e em outras empresas de tecnologia (Doerr, 2018).

⁵ O Roadmap é uma representação visual que mostra a visão de longo prazo de um projeto ou produto. É uma ferramenta comumente usada em gerenciamento de projetos e planejamento estratégico.

⁶ A Matriz Eisenhower é uma técnica de gerenciamento de tempo que prioriza tarefas com base em sua importância e urgência. É atribuída ao ex-presidente dos Estados Unidos, Dwight D. Eisenhower, que a utilizava para tomar decisões eficazes em relação às suas responsabilidades.

que tem como objetivo fornecer suporte à tomada de decisão. Dependendo das necessidades e preferências da organização, é possível optar por utilizar um único padrão ou combinar vários deles (Boehm, 1991).

Segundo Hillson (2009) o gerenciamento de riscos não é uma tarefa difícil, ele nos oferece uma forma estruturada para pensarmos sobre o risco e como lidar com eles intuitivamente. Desta forma, e pretendendo estruturar o conhecimento sobre as principais normas que tratam da gestão de risco, foi elaborado o Quadro 7, que traz os principais frameworks relacionados ao gerenciamento de risco com seus objetivos e pilares.

Quadro 7 – Frameworks de gerenciamento de risco e seus objetivos e pilares.

Framework	Objetivo	Principais Pilares
COSO/ERM Gjerdrum e Peter (2011)	Oferecer um framework abrangente para o gerenciamento de riscos corporativos	<ul style="list-style-type: none"> - Ambiente interno; - Estabelecimento de objetivos; - Identificação de eventos; - Avaliação de riscos; - Resposta a riscos; - Atividades de controle; - Informação e comunicação; - Monitoramento
Orange Book HM Treasury (2020)	Fornecer diretrizes para a segurança de sistemas de informação	<ul style="list-style-type: none"> - Governança e liderança; - Integração; - Colaborativo e informado; - Identificação e avaliação de riscos; - Tratamento de riscos; - Monitoramento de riscos; - Reporte de riscos; - Continuamente melhorado;
ISO/IEC 31.000:2018 Gjerdrum e Peter (2011)	Estabelecer princípios e diretrizes para o gerenciamento de riscos em geral	<ul style="list-style-type: none"> - Comunicação e consulta; - Estabelecimento do contexto; - Processo de avaliação de riscos; - Tratamento de riscos; - Monitoramento e análise crítica; - Melhoria contínua
ISO/IEC 27001:2013 (ISO, 2013)	Estabelecer requisitos para a implementação de um Sistema de Gestão da Segurança da Informação (SGSI)	<ul style="list-style-type: none"> - Políticas de segurança da informação; - Organização da segurança da informação; - Gerenciamento de ativos; - Controle de acesso; - Criptografia; - Segurança física e ambiental; - Gestão de incidentes de segurança da informação; - Continuidade de negócios; - Conformidade
Três linhas de Defesa The Institute of Internal Auditors - IIA (2020)	Estruturar a governança corporativa e o gerenciamento de riscos em três níveis distintos	<p>1ª Linha de Defesa</p> <ul style="list-style-type: none"> - Controle da Gestão; - Medidas de Controle Interno; <p>2ª Linha de Defesa</p> <ul style="list-style-type: none"> - Funções que supervisionam ou se especializam em gerenciamento de riscos; <p>3ª Linha de Defesa</p> <ul style="list-style-type: none"> - Auditoria Interna;

PMBOK Eduardo (2019)	Oferecer um guia para o gerenciamento de projetos	<ul style="list-style-type: none"> - Integração; - Escopo; - Cronograma; - Custos; - Qualidade; - Recursos; - Comunicação; - Riscos; - Aquisições; - Partes interessadas
IBGC (2007)	Trazer ao mercado informações práticas que contribuam para o processo da governança corporativa	<ul style="list-style-type: none"> - Identificação e Classificação dos Riscos; - Avaliação dos Riscos; - Mensuração dos Riscos; - Tratamento dos Riscos; - Monitoramento dos Riscos; - Informação e Comunicação;
ISO 27005(2011)	Fornecer diretrizes para o estabelecimento de uma abordagem sistemática para o gerenciamento de riscos de Segurança da Informação, que é necessária para identificar as necessidades organizacionais em relação aos requisitos de segurança da informação e para criar um sistema de gerenciamento de segurança da informação eficaz.	<ul style="list-style-type: none"> - Comunicação do Risco; - Definição de Contexto; - Identificação de Risco; - Estimativa de Risco; - Avaliação de Risco; - Tratamento de Risco; - Aceitação do Risco; - Monitoramento e Análise Crítica do Risco;
COBIT (2019) Carr (2023)	É um framework de gerenciamento de TI usado por empresas para desenvolver, organizar e implementar estratégias de gestão de informação e governança	<ul style="list-style-type: none"> - Domínio Planejar e Organizar; - Processo Avalia e gerencia os riscos de TI; - Processo Gerencia os projetos; - Domínio Aquisição e Implementação; - Domínio Entregar e Suportar; - Domínio Monitoramento;

Fonte: o autor, a partir de dados da pesquisa.

Conforme o Quadro 7 existem diversos frameworks que podem auxiliar no gerenciamento de riscos. A pesquisa realizada por Oliveira (2020) mostrou que nos últimos 5 anos os estudos na área de gestão de riscos e projetos de TI possuem como principal abordagem os fatores de sucesso na implantação e condução da gestão de riscos nas organizações e seu efeito positivo nos projetos de TI em busca de agregação de valor.

No contexto atual, em que o uso de frameworks se mostra crucial para as organizações obterem valor por meio da tecnologia da informação, é importante destacar a necessidade de um modelo abrangente que demonstre de maneira detalhada como realizar essa tarefa e quais ferramentas utilizar. Embora os modelos existentes apresentem algumas técnicas e sugestões, muitas vezes não conseguem

estabelecer uma conexão completa com a complexidade do contexto, organização e processos (Oliveira et al., 2020).

Os frameworks podem ser entendidos como instruções das quais o setor pode se basear e seguir, levando em consideração as boas práticas, a fim de trazer mais resultados e valor para a organização. De acordo com Shehabuddeen et al. (1999) o propósito do uso de um framework é transmitir ideias e descobertas a uma ampla comunidade, seja esta, acadêmica e/ou industrial; realizar a paridade entre situações e procedimentos; definir o domínio ou os limites de uma situação; descrever o contexto ou argumentar a validade de uma descoberta; e suportar o desenvolvimento de procedimentos, técnicas ou métodos e ferramentas.

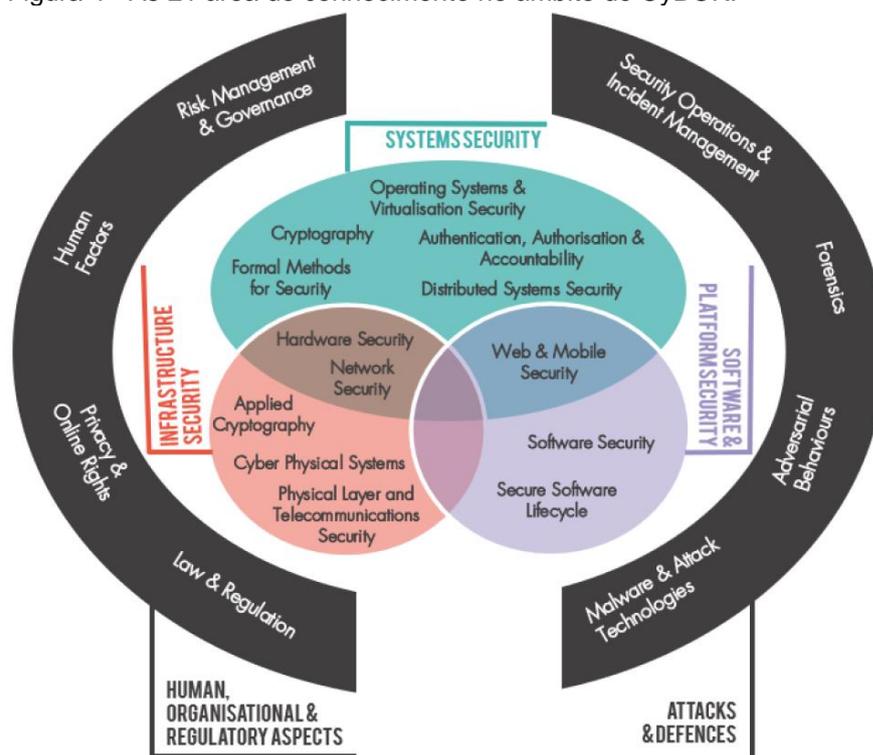
Porém, conforme pôde ser observado no Quadro 7 a quantidade de frameworks que tratam a temática da gestão de risco é volumosa, assim, observa-se que o conhecimento nessa área está sendo desenvolvido de forma fragmentada, o que dificulta tanto para os alunos quanto para os professores traçarem um caminho coerente de progressão nesse assunto (Rashid et al., 2018).

Diante desse cenário, é adequado buscar uma abordagem integrada que aborde tanto a implementação de frameworks de gestão de risco eficientes quanto a formação sólida em cibersegurança, garantindo assim uma gestão eficaz da tecnologia e a proteção adequada dos dados e sistemas Rashid et al. (2018).

Delineando minimizar os impactos dessa fragmentação do conhecimento, uma importante alternativa é o projeto Cyber Security Body of Knowledge - CyBOK que tem como objetivo a organização do conhecimento fundamental e amplamente reconhecido em cibersegurança. Assim como o Software Engineering Body of Knowledge - SWEBOK⁷, o CyBOK é concebido como um guia de conhecimento, abrangendo a literatura existente, como livros didáticos, artigos de pesquisa acadêmica, relatórios técnicos, white papers e padrões. O foco do projeto é mapear o conhecimento estabelecido, em vez de replicar integralmente tudo o que já foi escrito sobre o assunto. Para melhor organização o CyBOK é dividido em 21 Áreas de Conhecimento ACs de alto nível e agrupadas em cinco categorias amplas conforme Figura 4 de Rashid et al. (2019).

⁷ O SWEBOK (Software Engineering Body of Knowledge) é um guia que estabelece o conhecimento fundamental e essencial para a área de Engenharia de Software. Ele define as áreas de conhecimento, os tópicos e as práticas que são considerados parte do domínio da Engenharia de Software IEEE (2014)

Figura 4 - As 21 áreas do conhecimento no âmbito do CyBOK.



Fonte: Rashid et al. (2019).

Dentre as 21 áreas do Cybok, observa-se um relevante alinhamento desta dissertação com as seguintes áreas: Aspectos Humanos, Organizacionais e Regulatórios, que incluem o Gerenciamento de Risco, a Governança e a abordagem de erros humanos. Na área de Segurança de Sistemas, destaca-se a Autenticação, Autorização e Responsabilidade. Já na área de Segurança de Software e Plataforma, abordam-se a Segurança de Software e a Segurança no Ciclo de Vida do Software.

Acredita-se que todos os sistemas conectados serão atacados mais cedo ou mais tarde. Embora seja impossível alcançar uma solução 100% segura, é essencial realizar uma avaliação avançada de riscos e adotar medidas de mitigação adequadas. Nesse contexto, contar com um guia capaz de mapear e codificar o conhecimento fundamental e estabelecido em cibersegurança, abrangendo diversas áreas relacionadas à segurança de sistemas e informações, mostra-se extremamente relevante para o aprimoramento das práticas de gestão de riscos e segurança cibernética (Ebert, 2017), a apresenta o constructo para essa pesquisa.

Assim, definimos que uma taxonomia de risco é um método para identificação sistemática e repetível de riscos em desenvolvimento de software por meio um framework básico para elicitare e organizar a variedade de riscos abrangendo aspectos técnicos e não técnicos, lançando-se de um questionário para validação Carr et al. (1993). Posteriormente são selecionados os fatores que tiveram maior recorrência em

ambientes de projetos de desenvolvimento de software, sendo aplicados ao projeto do Programa Social Brasileiro Alfa.

Desta forma, a gestão de risco no contexto de desenvolvimento de software se mostra como um processo crítico que visa a identificar, avaliar e mitigar riscos potenciais que podem comprometer o sucesso do projeto. Kerzner (2013) discute que a gestão de riscos deve ser um processo contínuo que abrange a liderança, a comunicação e a resolução de conflitos dentro do projeto. Isso sublinha a importância de uma gestão proativa para evitar ou minimizar problemas antes que se tornem críticos. Humphrey (1989) argumenta que a adoção de processos de desenvolvimento bem definidos e sistemáticos pode diminuir os riscos inerentes ao projeto de software, destacando a importância de processos estruturados na mitigação de riscos. PMI (2017) reforça essa visão ao enfatizar que a gestão eficaz dos recursos é um aspecto fundamental na gestão de riscos, sugerindo que a otimização dos recursos disponíveis é crucial para o sucesso do projeto e para a minimização dos riscos.

2.3 Risco e Desenvolvimento de Software

A área de Engenharia de Software tem sido eficiente na criação de modelos de melhoria, processos, métodos e ferramentas para aumentar a probabilidade de sucesso em projetos de software (Somerville 1996). No entanto, apesar de todos esses esforços, a literatura relata que falhas em projetos é algo comum (Levando-se em consideração esses aspectos, depreende-se que os riscos têm o potencial de impactar negativamente os objetivos do projeto de software em termos de escopo, tempo e custo, o que pode levar ao atraso na conclusão de atividades planejadas do projeto (Singh, 2021).

O gerenciamento de riscos em projetos de software é uma abordagem sistemática de identificação, análise e resposta a riscos (Kutsch, 2008) para tomada de decisões proativas (Brown et al., 2000) com o objetivo de reduzir a exposição a perdas potenciais (Alberts e Dorofee, 2010; Brown et al., 2000; Du et al., 2007; Project Management Institute, 2017). Ao longo dos anos, diversos frameworks e modelos de gerenciamento de riscos em projetos de software foram propostos (Alberts e Dorofee, 2010; Boehm et al., 1995; Carbone e Tippett, 2004; Kontio, 2001; Manalif, 2013), levando em consideração o ciclo de vida do desenvolvimento de software. Assim, quando os objetivos do projeto são identificados e acordados, os requisitos tendem a ser estáveis e reduzem os riscos gerais do projeto. Qualquer redução de risco significa

menos tempo gasto em ações corretivas, melhorando assim o gerenciamento de riscos e o desempenho do projeto (Chiang e Mookerjee, 2004) (Singh, 2021).

Em seu estudo, Boehm (1988) apresentou uma representação sistemática do risco no setor de software, baseada na ideia de que ele deve ser incremental e focado na análise de riscos. Atualmente, a área de gerenciamento de riscos da engenharia de software abrange todos os processos do ciclo de vida do software (Lima2019), desta forma, a gestão de risco em projetos de software pode ser conceituada como sendo um processo de gerenciamento proativo que visa explorar oportunidades e tratar ameaças para garantir objetivos acordados, definidos e divulgados em um projeto, no contexto específico do desenvolvimento de software, conforme nos apresenta Chapman (2016) a gestão de risco desempenha um papel crucial na mitigação de potenciais ameaças e na garantia da qualidade e sucesso dos projetos.

O estudo da gestão de risco aplicado às etapas de desenvolvimento de softwares é de relevância no atual cenário tecnológico. Diversas pesquisas têm sido realizadas nessa área, evidenciando a importância de compreender e mitigar os riscos envolvidos no desenvolvimento de software (Costa, 2021). No entanto, os estudos sobre esse tema, em especial no Desenvolvimento de Software para o Governo, ainda são incipientes e escassos (Furlan e Pacheco2021).

Porém, essa escassez não se limita ao Governo, o estudo realizado por Fernandes e Rabechini Jr. (2021) revela que a relação entre a gestão de riscos e abordagens ágeis, por exemplo, é um fenômeno recente, com a maioria das pesquisas publicadas nos últimos cinco anos e uma quantidade limitada de estudos empíricos relacionando esses conceitos. Além disso, a análise revela que muitos dos principais estudos foram publicados em conferências e congressos, indicando que são pesquisas em estágio inicial, com o objetivo principal de propor modelos para a gestão de riscos em projetos gerenciados por abordagens ágeis (Jose et al., 2023).

Constata-se que os métodos ou abordagens ágeis, citados no parágrafo anterior, não incorporam atividades específicas para gerenciar riscos. No entanto, alega-se que os métodos ágeis reduzem riscos por meio de múltiplas iterações ou sprints (Albadarneh et al., 2015). Os mecanismos de feedback dos métodos ágeis, como a opinião do cliente sobre o incremento entregue no final de cada sprint, ajudam a equipe do projeto a responder às constantes mudanças em requisitos, necessidades, novas oportunidades e riscos (Tavares e Eduardo, 2019).

Em que pese o disposto no parágrafo anterior, existem poucas diretrizes sobre como esses mecanismos podem ser implementados de maneira mais eficaz. Além disso, existem ameaças em nível de projeto que podem não ser identificadas por

iteração, criando uma lacuna na capacidade dessas abordagens de lidar com riscos. Portanto, há uma necessidade de incorporar conceitos de gerenciamento de riscos nos métodos ágeis para abordar essa questão (Alharbi e Qureshi, 2014).

Sugere-se que os praticantes de métodos ágeis lidem com riscos da mesma forma que em projetos tradicionais. No entanto, abordagens ágeis e a abordagem convencional de gerenciamento de riscos parecem estar um pouco em conflito (Siddique e Hussein, 2014). O gerenciamento de riscos geralmente implementa uma abordagem de alto custo, enquanto os métodos ágeis suportam apenas uma abordagem leve.

A pesquisa de Tavares et al. (2017) foi um estudo para propor uma seleção de práticas de gerenciamento de riscos que poderiam ser aplicadas ao Scrum. No entanto, pesquisas adicionais são necessárias para desenvolver uma abordagem sistemática e abrangente para o gerenciamento de riscos que não esteja vinculada a um método ágil específico e que seja leve o suficiente para não violar o princípio da agilidade. Considerando esse estudo, o uso de práticas de gerenciamento de riscos em métodos ágeis deve estar de acordo com os princípios do Manifesto Ágil (Beck et al., 2001).

Os métodos ágeis vêm ganhando popularidade, uso e aceitação na comunidade de desenvolvimento de software. Da mesma forma, frameworks como o Scrum, que originalmente foi recomendado para uso por equipes pequenas, vêm ganhando cada vez mais adoção em grandes projetos de software (Oliveira, 2019), desta forma, viabilizar metodologias de gestão de risco que sejam viáveis independente do framework de desenvolvimento é o principal desafio.

Durante a pesquisa bibliográfica para elaboração da fundamentação teórica, foram identificados estudos como por exemplo o: *Uncertainty in information system development: Causes, effects, and coping mechanisms* dos autores Toni Taipalus, Ville Seppänen, Maritta Pirhonen da Universidade de Jyväskylä, Finlândia e o *Risk factors in software development projects: a systematic literature review* dos autores Júlio Menezes Jr, Cristine Gusmão e Hermano Moura da Universidade Federal de Pernambuco – Brasil, que de alguma forma, tentaram criar um roteiro a partir do ponto de vista de profissionais e estudantes estabelecendo uma taxonomia capaz de traduzir as causas e incertezas em projetos de desenvolvimento de software.

O trabalho realizado por Tavares e Eduardo (2019) identificou 127 práticas de gerenciamento de riscos que foram classificadas de acordo com os cinco componentes e 48 subcomponentes. Essa classificação visa facilitar a hierarquização

das práticas de gerenciamento de riscos, priorizando as mais importantes, que de alguma forma, aumentam as chances de sucesso dos projetos (Sádaba et al., 2014).

Já o estudo realizado por Taipalus et al. (2020) criou uma metodologia com coleta de dados que podem ser descritas como um processo linear de três etapas. Primeiramente, por meio de uma pesquisa online, os estudantes universitários foram questionados sobre o que causa incerteza em um projeto de Desenvolvimento de Software e com base nos resultados, foi formulada a estrutura para as entrevistas semiestruturadas. Em seguida, foram selecionados e entrevistados onze profissionais da indústria de software com diferentes formações. Por fim, utilizando a análise de conteúdo, as entrevistas foram analisadas três vezes, uma para cada pergunta de pesquisa.

Os resultados mostraram uma lista de 24 temas relacionados à incerteza, que foram categorizados em oito categorias, tais como canais de comunicação inadequados, falta de comprometimento do cliente e avaliação de tecnologia. Além disso, os resultados revelaram seis efeitos da incerteza, a saber: disforia⁸, trabalho desnecessário, cultura de trabalho tóxica, inovação, autodesenvolvimento e motivação. Por fim, foram identificados quatro mecanismos de enfrentamento da incerteza, como mudança de atitude, enfatização de papéis, abertura na comunicação e envolvimento com o cliente, todos em linha com pesquisas anteriores com profissionais da indústria, mas diferindo das percepções dos estudantes de desenvolvimento de software.

A identificação de riscos ou de fatores de risco é considerada a atividade mais influente do gerenciamento de riscos (De Bakker et al., 2010; López e Salmeron, 2012) e é utilizada tanto em métodos ágeis quanto em métodos tradicionais de desenvolvimento de software (Neves et al., 2014). Um fator de risco é definido como uma condição que pode representar uma séria ameaça à conclusão bem-sucedida de um projeto de desenvolvimento de software (March e Shapira, 1987). A ineficiência no processo de identificação de riscos no desenvolvimento de sistemas complexos é considerada uma das principais causas de falhas em projetos (Reeves et al., 2013).

Vários estudos indicam que existe uma relação direta entre o gerenciamento de riscos e o sucesso ou melhoria do desempenho de projetos de desenvolvimento de software (Jiang e Klein, 2000; Jiang et al., 2001; Raz et al., 2002; Wallace e Keil, 2004; Wallace et al., 2004a; De Bakker et al., 2010; Han e Huang, 2007). Essas pesquisas evidenciam que, no mínimo, os fatores de risco devem ser identificados e

⁸ Termo geralmente usados no contexto da saúde mental para descrever um estado de desconforto ou insatisfação profunda, que pode ser emocional, mental ou psicológica

bem controlados para que os projetos alcancem seus objetivos. Assim, a identificação de fatores de risco desempenha um papel crucial no sucesso e no desempenho de projetos de desenvolvimento de software onde, Menezes et al. (2019) ressalta que o resultado da identificação de riscos é uma lista que contém os riscos que foram identificados e sua categoria relacionada (Maniasi et al., 2006).

Corroborando com o mesmo princípio Odzaly e Greer (2017) ressaltam que a identificação de riscos é o primeiro passo no processo de gerenciamento de riscos. Essa etapa determina as possíveis ameaças em um projeto de software e avalia a efetividade do processo de gerenciamento de riscos. Como discutido no Capítulo 2.1, o uso de um método sistemático é essencial para aprimorar a identificação de riscos. Nesse contexto, uma abordagem promissora é a utilização de uma ontologia de identificação de riscos como base de conhecimento, que emprega uma taxonomia de riscos e se baseia nas diretrizes do CMMI como guia.

As taxonomias segundo DICIO (2023) são classificações ordenadas de elementos de acordo com seu relacionamento presumido; elas podem ser usadas como uma ferramenta muito útil em diferentes áreas da ciência e da indústria onde é necessário organizar e facilitar o acesso a um amplo conjunto de elementos relacionados (Maniasi et al 2006).

Podem ser lembrados dois modelos de gerenciamento de riscos baseados em taxonomia utilizados: o Questionário Baseado em Taxonomia do SEI - TBQ e o Gerenciamento de Riscos Baseado em Taxonomia do CMMI (Chrissis Konrad (2011)). Esses modelos adotam abordagens que envolvem a classificação estruturada dos riscos com base em uma taxonomia específica. O TBQ do SEI é um questionário estruturado que permite a identificação e avaliação dos riscos com base em uma taxonomia pré-definida (Pressman, 2014). . O Gerenciamento de Riscos Baseado em Taxonomia do CMMI fornece diretrizes e práticas para identificar, avaliar e gerenciar os riscos em projetos de desenvolvimento de software. Ele enfatiza a importância de mapear os riscos para as áreas de processo do CMMI, permitindo uma abordagem mais abrangente e sistemática para lidar com os riscos (Chrissis; Mike; Konrad (2011)).

A taxonomia de riscos baseada no SEI, é segundo Carr et al. (1993) um método que visa identificar riscos de forma sistemática e repetível Esse método fornece um framework básico para elicitar e organizar uma ampla gama de riscos no desenvolvimento de software, abrangendo aspectos técnicos e não técnicos. Em sua versão inicial, ele agrupa diferentes fontes de riscos em diversas categorias, oferecendo um questionário chamado TBQ que permite a realização de um processo sistemático de identificação. O questionário consiste em perguntas objetivas

relacionadas a cada atributo taxonômico, projetadas para elicitare a variedade de riscos e preocupações que podem potencialmente afetar o produto do projeto. A aplicação do questionário é semiestruturada, e tanto as perguntas quanto a sequência são usadas como um instrumento definidor, mas não limitante, para a aplicação de outras metodologias (Maniasi et al. ,2006). A aplicação do questionário consiste em quatro etapas, conforme Quadro 8:

Quadro 8 - Atividades sequenciais para aplicação do TBQ.

Etapa	Descrição
Comprometimento da Gestão	Antes que o questionário possa ser conduzido, é necessário ter a aceitação e o comprometimento executivo da empresa, além da seleção do projeto e a seleção dos participantes de diferentes áreas
Seleção e Treinamento da Equipe	A equipe inclui tanto o pessoal do projeto quanto os funcionários da empresa do cliente. Após a seleção da equipe, é necessário treinar o grupo na técnica a ser utilizada, responsabilidades dos papéis, protocolo de entrevistas a ser seguido etc.
Identificação de Riscos	Começa com uma apresentação para todos os participantes da identificação de riscos e do método a ser utilizado, e continua com entrevistas ao pessoal selecionado
Conclusão da Identificação	Para concluir a identificação de riscos, é necessário apresentar os resultados obtidos a todos os participantes.

Fonte: o autor, adaptado de Haidabrus (2022) e Carr et al. (1993).

Já o Gerenciamento de Riscos Baseado em Taxonomia do CMMI utiliza uma estrutura de classificação de riscos para auxiliar no processo de identificação e gerenciamento de riscos em projetos de desenvolvimento de software conforme Mary Beth Chrissis, Mike Konrad (2011), são apresentadas no Quadro 9:

Quadro 9- Atividades para Gerenciamento de Riscos Baseado em Taxonomia do CMMI.

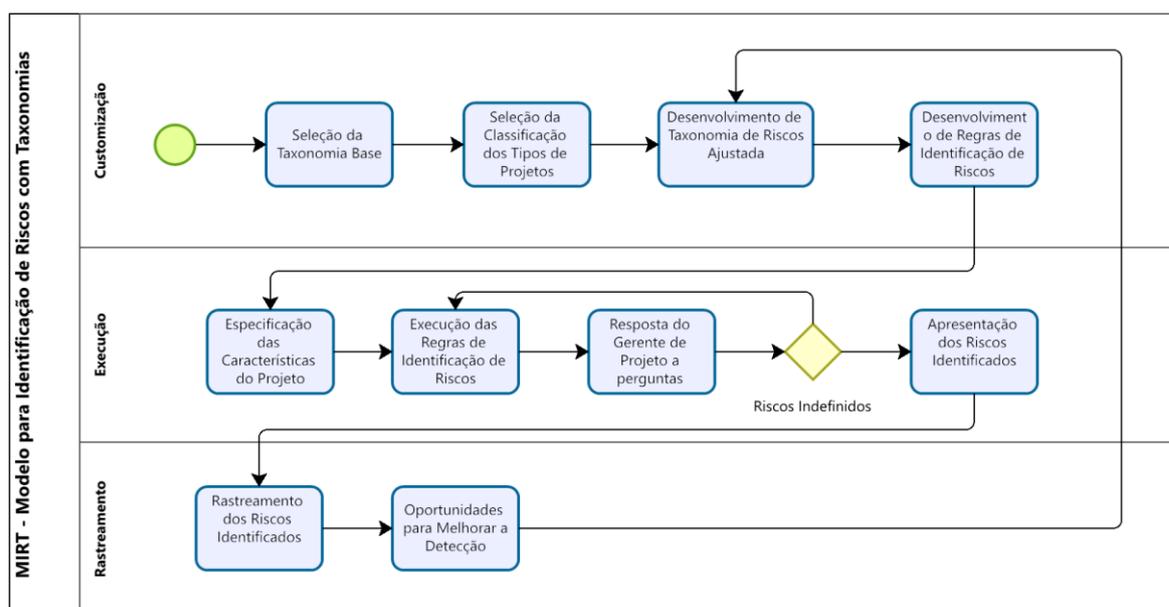
Etapa	Descrição
Preparação para o Gerenciamento de Riscos	Envolve determinar as fontes e categorias de riscos, definir parâmetros de risco e estabelecer uma estratégia de gerenciamento de riscos.
Identificação e Análise de Riscos	Envolve identificar os riscos e avaliá-los, categorizá-los e priorizá-los.
Mitigar Riscos	Requer desenvolver planos de mitigação de riscos e implementá-los.

Fonte: Do autor adaptado de Haidabrus (2022) e Carr et al. (1993).

Segundo Haidabrus (2022), tanto o questionário baseado em taxonomia do SEI quanto o gerenciamento de riscos baseado em taxonomia do CMMI fornecem uma estrutura sólida para auxiliar as organizações no entendimento e no tratamento dos riscos, visando melhorar a qualidade e a eficácia de seus processos de gerenciamento de riscos

O Modelo para Identificação de Riscos com Taxonomias – MIRT, desenvolvido por Maniasi et al. (2006) a partir de estudos de TeraQuest (1998) e Jones (1994), estabelece, padroniza e sistematiza práticas relacionadas à identificação de riscos para o gerenciamento de projetos de software. A Figura 5 apresenta o MIRT e sua estruturação em três etapas, customização, execução e rastreamento.

Figura 5 - Modelo para Identificação de Riscos com Taxonomias – MIRT.



Fonte: o autor, adaptado de Maniasi et al. (2006).

O MIRT é dividido em três fases: Customização, Execução e Rastreamento. Na fase de Customização, os principais componentes do modelo são adaptados e ajustados às características da organização. São realizadas quatro atividades, sendo duas na subfase de Inicialização (Seleção da Taxonomia Base e Seleção da Classificação dos Tipos de Projetos) e duas na subfase de Melhoria Contínua (Desenvolvimento da Taxonomia de Riscos Ajustada e Desenvolvimento de Regras de Identificação de Riscos) (Maniasi et al., 2006).

A fase de Execução tem como objetivo gerar uma lista de riscos candidatos em um projeto específico, levando em consideração suas características e utilizando a Taxonomia de Riscos Ajustada e as Regras de Identificação de Riscos obtidas na fase anterior. São realizadas quatro atividades, sendo duas executadas uma vez (Especificação das Características do Projeto e Apresentação dos Riscos Identificados) e duas realizadas de forma iterativa (Execução das Regras de Identificação de Riscos e Resposta do Gerente de Projeto a perguntas relacionadas a riscos) (Maniasi et al., 2006).

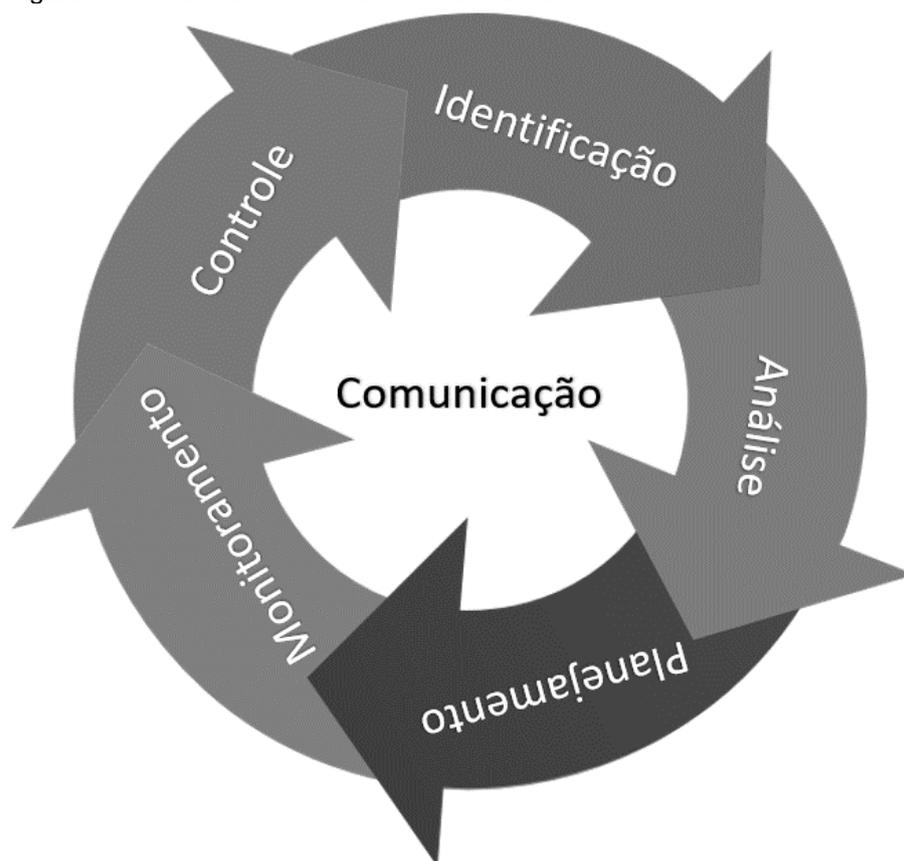
A fase de Rastreamento é a última etapa do MIRT e visa identificar oportunidades de melhoria na Taxonomia de Riscos Ajustada e nas Regras de Identificação de Riscos. São realizadas duas atividades recorrentes: Rastreamento dos Riscos Identificados e Detecção de Oportunidades de Melhoria. Nessa fase, é feita a validação dos riscos selecionados para o projeto e são utilizadas informações históricas para ajustar, adicionar, corrigir e produzir novas adaptações nas taxonomias e regras de identificação, visando melhorar o desempenho do MIRT em implementações futuras (Maniasi et al., 2006).

O método de identificação de riscos desenvolvido pelo SEI, é derivado da literatura publicada e da experiência anterior no desenvolvimento de software, foi testado em projetos de desenvolvimento de software financiados pelo governo, tanto na área de militar quanto na civil, visando sua utilidade e melhoria do próprio método (Carr et., 1993). Essa taxonomia pode ser parcialmente modificada levando em conta as especificidades de um projeto específico (Bayona-Oré Calvo-Manzano2014).

A chave para o planejamento de ações de risco reside em considerar as consequências futuras de uma decisão tomada hoje. Nesse contexto, o paradigma de gerenciamento de riscos do SEI apresenta as diferentes atividades que compõem o processo de gerenciamento de riscos no desenvolvimento de software Haidabrus (2022).

Conforme destacado por Hanif et al. (2021) esse modelo é representado como um círculo para ressaltar que o gerenciamento de riscos é um processo contínuo, e as setas indicam o fluxo lógico e temporal das informações entre as atividades de gerenciamento de riscos. A comunicação é colocada no centro do modelo, pois é tanto o canal pelo qual todas as informações fluem quanto, muitas vezes, o principal obstáculo ao gerenciamento de riscos. Em essência, o modelo funciona como um framework para o gerenciamento de riscos de software, permitindo que um projeto estruture uma prática de gerenciamento de riscos que se ajuste melhor à sua estrutura de gerenciamento de projeto (Haidabrus, 2022) conforme Figura 6.

Figura 6 - Modelo de Gerenciamento de Risco.



Fonte: o Autor, adaptado de Carr et al. (1993).

A taxonomia de risco do SEI possui uma estrutura hierárquica e sistematiza as fontes (áreas) de risco em três níveis:

- Classe;
- Elemento de classe;
- Atributo do elemento.

A classe define o campo de atividade da engenharia de software com o qual determinado risco pode estar associado. O elemento de classe indica uma área específica de risco no campo relevante de atividade. O atributo do elemento define um fator de risco em uma determinada área de risco, que pode estar associado a um evento indesejável, ação ou fato que é uma fonte de risco (Hofman e Grela, 2015).

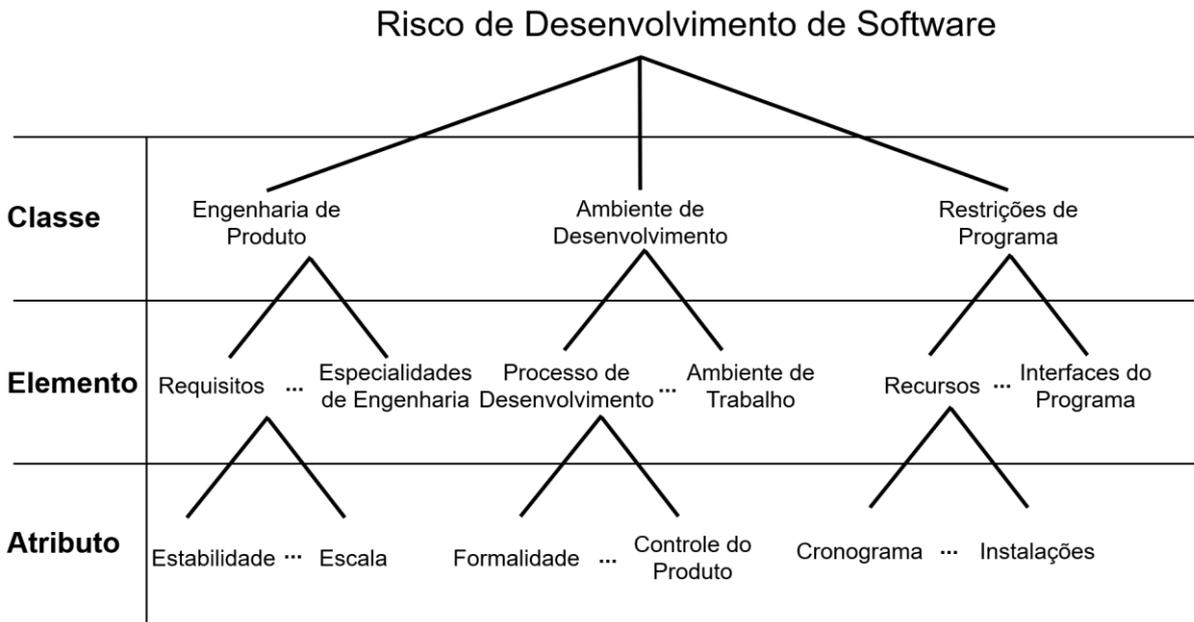
Conforme apresentado por Carr et al. (1993) a taxonomia de software está organizada em três classes principais:

- Engenharia de Produto: os aspectos técnicos do trabalho a ser realizado.
- Ambiente de Desenvolvimento: os métodos, procedimentos e ferramentas utilizados para produzir o produto.

- Restrições do Programa: os fatores contratuais, organizacionais e operacionais nos quais o software for desenvolvido, mas que geralmente estão fora do controle direto da gestão local.

Essas classes taxonômicas são subdivididas em elementos, e cada elemento é caracterizado por seus atributos, conforme Figura 7.

Figura 7 - Taxonomia de Riscos no Desenvolvimento de Software.



Fonte: Carr et al. (1993).

Esta figura 7 representa a ilustração mais significativa do referencial teórico adotado nesta dissertação. É essencial salientar que esta classificação foi deliberadamente selecionada como a taxonomia orientadora para a condução da etapa empírica deste estudo. A escolha desta taxonomia tem como propósito fundamental proporcionar uma base sólida para a análise e interpretação dos dados coletados, permitindo assim responder de forma eficaz aos objetivos delineados para esta pesquisa, enfatizando a importância desta estrutura conceitual não apenas como um fundamento teórico, mas também como um instrumento crucial no processo de investigação empírica.

A taxonomia pode ser usada para classificar diversos fatores associados ao desenvolvimento de sistemas dependentes de software, como tarefas de desenvolvimento, procedimentos de qualidade ou fontes e consequências de risco. As definições apresentadas nesta taxonomia são projetadas para facilitar a classificação dos próprios riscos, conforme associados ao processo de desenvolvimento trazidos por Carr et al. (1993).

Assim, o Quadro 10 apresenta a proposta de taxonomia de riscos no Desenvolvimento de Software, a partir de uma visão geral da Taxonomia SEI e de sua organização hierárquica. O Quadro 10 apresenta taxonomia com 3 classes, 13 elementos de classe e 64 Atributos. Esses níveis hierárquicos apresentados na taxonomia e descritos neste quadro são aproveitados na parte empírica deste estudo.

Quadro 10 – Proposta de uso de Taxonomia de Riscos no Desenvolvimento de Software.

	Classe (3)	Elemento (13)	Atributo (64)
Riscos no desenvolvimento do Software	Engenharia de Produto	Requisitos	a. Estabilidade; b. Completude; c. Clareza; d. Validade; e. Viabilidade; f. Precedente; g. Escala
		Design	a. Funcionalidade; b. Dificuldade; c. Interfaces; d. Desempenho; e. Testabilidade; f. Restrições de Hardware; g. Software não Desenvolvido
		Código e Teste Unitário	a. Viabilidade; b. Testes; c. Codificação/Implementação
		Integração e Teste	a. Ambiente; b. Produto; c. Sistema
		Especialidades de Engenharia	a. Manutenibilidade; b. Confiabilidade; c. Segurança; d. Segurança; e. Fatores Humanos; f. Especificações
	Ambiente de Desenvolvimento	Processo de Desenvolvimento	a. Formalidade; b. Adequação; c. Controle do Processo; d. Familiaridade; e. Controle do Produto
		Sistema de Desenvolvimento	a. Capacidade; b. Adequação; c. Usabilidade d. Familiaridade; e. Confiabilidade; f. Suporte do Sistema; g. Entregabilidade
		Processo de Gestão	a. Planejamento; b. Organização do Projeto; c. Experiência em Gestão; d. Interfaces de Programa
		Métodos de Gestão	a. Monitoramento; b. Gestão de Pessoal; c. Garantia de Qualidade; d. Gestão de Configuração
		Ambiente de Trabalho	a. Atitude em relação à Qualidade; b. Cooperação; c. Comunicação; d. Moral
	Restrições do Programa	Recursos	a. Cronograma; b. Equipe; c. Orçamento; d. Instalações
		Contrato	a. Tipo de Contrato; b. Restrições; c. Dependências
		Interfaces do Programa	a. Cliente; b. Contratantes Associados; c. Subcontratados; d. Contratante Principal; e. Gestão Corporativa; f. Fornecedores; g. Política

Fonte: o autor, adaptado de Carr et al. (1993).

A partir dessa base taxonômica do Quadro 10 um questionário foi aplicado a gerentes, designers, pessoal técnico, etc (Tabassi et al., 2019). Um questionário baseado em uma taxonomia de riscos é uma ferramenta que, ao ser utilizada, garante a cobertura de todas as áreas potenciais de risco devido à presença de perguntas relacionadas ao nível mais baixo da taxonomia de riscos – atributos (Hofman e Grela, 2015). O número e formato das perguntas podem variar dependendo das especificidades do projeto, do método escolhido de entrevista e o processamento de

seus resultados. Em qualquer caso, o foco deve ser na extração mais completa e eficaz do conhecimento dos membros do projeto (Haidabrus, 2022).

A pesquisa realizada por Menezes et al. (2019), teve como objetivo identificar e mapear os fatores de risco em ambientes de projetos de desenvolvimento de software por meio de uma revisão sistemática de literatura a partir de bases de dados científicas. Ao final, foram identificados 41 estudos nos quais foram extraídos e classificados os fatores de risco de acordo com a taxonomia de desenvolvimento de software desenvolvida pelo SEI.

Ainda segundo o autor, um total de 148 fatores de risco diferentes foram mapeados no trabalho. Isso representa uma média de aproximadamente 9 ocorrências para cada fator de risco. De acordo com a taxonomia de risco do SEI (Menezes et al., 2019), 51 fatores de risco pertencem à classe "Engenharia de Produto", 41 pertencem à classe "Ambiente de Desenvolvimento" e 56 pertencem à classe "Restrições do Programa". O Quadro 11 apresenta os 10 fatores de risco mais mencionados para todas as classes, durante a pesquisa realizada por Menezes et al. (2019):

Quadro 11 - Principais 10 fatores de risco - todas as classes.

Fator de Risco	Número de Ocorrências	Classe	Elemento
Equipe sem as habilidades necessárias	55	Restrições do Programa	Recursos
Ambiguidade de requisitos	44	Engenharia de Produto	Requisitos
Comprometimento inadequado do usuário/cliente	37	Restrições do Programa	Interfaces do Programa
Mudanças nos requisitos	32	Engenharia de Produto	Requisitos
Introdução de nova tecnologia	30	Engenharia de Produto	Requisitos
Ambiente organizacional instável	30	Restrições do Programa	Interfaces do Programa
Falhas em componentes fornecidos externamente/más interfaces	27	Engenharia de Produto	Design
Complexidade técnica	25	Engenharia de Produto	Requisitos
Ausência de planejamento ou planejamento inadequado	25	Ambiente de Desenvolvimento	Processo de Gerenciamento
Requisitos incompletos	24	Engenharia de Produto	Requisitos
Qualidade das especificações/documentação	24	Engenharia de Produto	Especialidades de Engenharia

Fonte: o autor, adaptado de Menezes et al. (2019).

Na análise dos estudos selecionados por Menezes et al. (2019), foi identificado que o fator de risco "Equipe sem as habilidades necessárias" ocorreu 55 vezes. Isso

significa que houve 55 ocorrências de fatores relacionados à falta de habilidades da equipe, que foram agrupados em um único fator. Alguns exemplos dessas ocorrências incluem falta de pessoal qualificado, falta de expertise técnica e ausência de habilidades especializadas necessárias para o projeto.

Todos os 41 estudos analisados apresentam listas de fatores de risco, os quais foram organizados de acordo com a Taxonomia do SEI (Menezes et al., 2019), com a identificação das classes e elementos correspondentes. Verificou-se uma predominância de fatores relacionados à classe "Engenharia de Produto", sendo que 5 desses fatores estavam associados ao elemento "Requisitos". O fator mais mencionado foi a falta de qualificação da equipe do projeto, citado em 55 ocasiões. Destaca-se que um total de 1414 ocorrências foram identificadas, das quais 353 (aproximadamente 25%) foram mencionadas no Quadro 11.

Na sequência Menezes et al. (2019) apresenta os 10 principais fatores de risco de cada classe. Na classe "Engenharia de Produto", o problema da falta de clareza ou ambiguidade dos requisitos é citado como o fator mais recorrente correspondendo a 53% do total de ocorrências da classe "Engenharia de Produto".

Ainda segundo o autor, existe uma certa uniformidade na distribuição dos fatores de risco da classe "Ambiente de Desenvolvimento", com uma maior inclinação para o fator associado a um planejamento inadequado do projeto representado aproximadamente 54% do total de ocorrências dessa classe. Já na classe "Restrições do Programa", os problemas relacionados à qualificação da equipe são os fatores de risco mais relevantes, destacando-se também os fatores de baixo comprometimento do usuário ou cliente, bem como as influências negativas do ambiente organizacional. Os elementos "Recursos" e "Interfaces do Programa" se destacam representando aproximadamente 51% do total de ocorrências da classe "Restrições do Produto".

É importante destacar que a pesquisa realizada por Menezes et al. (2019) demonstrou no resultados da análise de rigor-relevância a boa qualidade dos estudos selecionados. Apenas cinco dos artigos revisados utilizaram apenas revisão de literatura como método para identificar e/ou avaliar os fatores de risco. A pesquisa com profissionais da indústria é o método mais utilizado, sendo empregado por 25 estudos. Outros métodos mencionados incluem Delphi, estudos de caso, entrevistas, grupos de discussão e bancos de dados de projetos anteriores. Dessa forma, o autor afirmar que há evidências de que os fatores de risco encontrados, se combinados, são relevantes para a indústria e para os profissionais.

Assim, a intersecção entre risco e desenvolvimento de software envolve o entendimento de como os riscos específicos ao desenvolvimento de software podem

afetar a execução e o resultado do projeto. Menezes et al. (2019) abordam a importância de compreender e mitigar os riscos específicos do desenvolvimento de software, incluindo a ambiguidade e as mudanças de requisitos, bem como a introdução de novas tecnologias. Sommerville (2019) destaca a necessidade de gerenciamento eficaz dos requisitos e da qualidade, enfatizando que os requisitos claros e verificáveis são fundamentais para mitigar os riscos de desenvolvimento. Jones (1994) complementa essa perspectiva ao discutir o impacto das restrições do programa, como recursos e contratos, no desenvolvimento de software, ressaltando que fatores externos também representam riscos significativos que precisam ser gerenciados. A síntese dessas visões evidencia a necessidade de uma abordagem integrada que combine a gestão eficaz de requisitos, a atenção à qualidade do produto e a consideração das restrições externas como elementos fundamentais para a mitigação de riscos no desenvolvimento de software.

2.4 Programas Sociais

O contexto socioeconômico brasileiro no início dos anos 1970 trouxe à tona o fato de que a redução da desigualdade social não acompanhou o crescimento econômico. Grande parte deste problema seria derivado do processo histórico de formação social e econômico caracterizado pelo subdesenvolvimento. A economia brasileira foi baseada na produção agrícola em grandes latifúndios com uso de trabalho escravo e atividades consideradas exploratórias (Furtado, 1989).

Em função disso, as sociedades construíram ao longo de sua história, a partir da força do movimento social e das organizações políticas, uma série de mecanismos estatais que se destinam a diversos propósitos. Esses mecanismos ganham contorno de uma política social quando se destinam a proteger e promover seus membros. Não existe um padrão consensual de conjunto de políticas preestabelecido para se determinar o que seja uma política social (Abrah, 2012).

Apesar de reconhecer essas dificuldades e limitações, adota-se neste estudo entendimento da política social como sendo composta por um conjunto de programas e ações do Estado, que se manifestam em oferta de bens e serviços e transferências de renda, com o objetivo de atender as necessidades e os direitos sociais que afetam vários dos componentes das condições básicas de vida da população, até mesmo aqueles que dizem respeito à pobreza e à desigualdade (Castro, 2012).

Em função da pobreza conferida ao longo dos anos 1990 e 2000 no Brasil, as políticas sociais passaram por alterações em seu direcionamento, a fim de promover

medidas para combater a vulnerabilidade vivenciada por essa parcela da população. Uma das premissas dessas políticas consiste na consolidação do crescimento econômico como um componente crucial dentro do contexto de evolução social e a superação da pobreza como um resultado do desenvolvimento (Arbache2003); (Rocha2003); (Ferreira e Et al., 2006).

A proteção social dos cidadãos manifesta-se na seguridade social, que tem como ideia central a solidariedade - apesar do critério do seguro social ainda prevalecer em grande parte dos países - aos indivíduos, famílias e grupos em determinadas situações de dependência ou vulnerabilidade, entre as quais se podem citar: incapacidade de ganhar a vida por conta própria em decorrência de fatores externos, que independem da vontade individual; vulnerabilidade devido ao ciclo vital do ser humano - crianças e idosos, por exemplo; situações de risco, como em caso de acidentes - invalidez por acidente etc. (Teixeira (1992); (Fleury2005); (Vianna 1998); (Sposati2009).

A promoção social é entendida como a resultante da geração de igualdades, oportunidades e resultados para indivíduos e/ou grupos sociais. A ideia central é que a geração de igualdades está relacionada à expansão da oferta de bens e serviços sociais, enquanto bens equalizadores providos pelo poder público, principalmente a escolarização e o acesso à saúde como elementos centrais na geração de habilidades e capacidades em indivíduos e/ou grupo social (Sem, 2011) (Rawls, 2003).

A literatura nacional da área reflete a realidade atual da política social, uma vez que ela é também bastante setorial, tratando de políticas específicas e com predominância, em termos metodológicos, da análise voltada para descrição dos estágios alcançados e/ou deficiências reveladas (Castro, 2012).

O Estado brasileiro atualmente desenvolve um conjunto diversificado de políticas públicas, de tal modo que a política social brasileira, hoje, chama a atenção pela dimensão de seu conjunto: são centenas dos mais diferentes tipos de benefícios ofertados diariamente a dezenas de milhões de cidadãos atingidos pelas ações e programas implementadas pelas diversas políticas públicas de proteção e promoção social (Castro, 2012).

O Quadro 12 resume os principais programas sociais brasileiros extraídos a partir do Portal da Transparência⁹ e do Portal do Ministério do Desenvolvimento e Assistência Social, Família e Combate à Fome¹⁰, os benefícios são recursos financeiros transferidos diretamente da União para o cidadão que participa de

⁹ <https://portaldatransparencia.gov.br/>

¹⁰ <https://www.gov.br/mds/pt-br>

programas sociais específicos e são concedidos aos cidadãos inscritos nos respectivos programas sociais onde cada iniciativa possui regras específicas que precisam ser atendidas para que o valor seja concedido.

Quadro 12 – Principais Programas Sociais Brasileiros

Programa	Descrição	Ato Normativo
Auxílio Brasil	O Auxílio Brasil é um programa de transferência de renda às famílias em situação de pobreza e extrema pobreza. Além de assegurar uma renda básica a essas famílias, o programa objetiva simplificar a cesta de benefícios e estimular a sua emancipação BRASIL (2021b).	Lei nº 14.284/2021
Auxílio Gás dos Brasileiros	Subvenção econômica destinada a auxiliar as famílias de baixa renda na aquisição de gás liquefeito de petróleo – GLP BRASIL (2021a).	Lei nº 14.237/2021
Benefício de Prestação Continuada	O Benefício de Prestação Continuada - BPC, previsto na Lei Orgânica da Assistência Social - LOAS, é a garantia de um salário-mínimo mensal ao idoso com 65 anos ou mais e à pessoa com deficiência que comprovem não possuir meios de prover a própria manutenção, nem de tê-la provida por sua família. Em ambos os casos, é necessário que sejam comprovados os critérios de baixa renda por grupo familiar, independente de contribuição à Previdência Social. Para ter acesso ao BPC, a renda da família é considerada baixa quando a soma de todos os valores que cada familiar recebe, dividida pelo número de pessoas, é menor que 1/4 do salário-mínimo vigente BRASIL (1993).	Lei nº 8.742/1993 Decreto nº 6.214/2007
Bolsa Família	O Bolsa Família é o maior programa de transferência de renda do Brasil, reconhecido internacionalmente por já ter tirado milhões de famílias da fome. O Governo relançou o programa com mais proteção às famílias, com um modelo de benefício que considera o tamanho e as características familiares, aquelas com três ou mais pessoas passarão a receber mais do que uma pessoa que vive sozinha BRASIL (2023b).	MEDIDA PROVISÓRIA Nº 1.164, DE 2 DE MARÇO DE 2023
Carteira do Idoso	A carteira do Idoso é o instrumento de comprovação para o acesso ao benefício estabelecido pelo artigo 40 da Lei nº 10.741, de 1º de outubro de 2003 – Estatuto do Idoso. A carteira deve ser gerada apenas para pessoas idosas que não tem como comprovar renda igual ou inferior a 2 (dois) salários-mínimos BRASIL (2003a).	Lei nº 10.741/2003
Garantia-Safrá	É uma ação do Programa Nacional de Fortalecimento da Agricultura Familiar - PRONAF. Possui como beneficiários agricultores que têm renda familiar mensal de, no máximo, 1,5 (um e meio) salário-mínimo e que plantam entre 0,6 e 5 hectares de feijão, milho, arroz, mandioca e algodão. Uma vez aderidos ao programa, eles passam a receber o benefício quando o município em que moram comprova a perda de, pelo menos, 50% do conjunto dessas produções, ou de outras a serem definidas pelo órgão gestor do Fundo	Lei 10.420, de 10 de abril de 2002 Decreto Nº 4.962, de 22 de janeiro de 2004

	Garantia-Safra, em razão de estiagem ou excesso hídrico BRASIL (2002)BRASIL (2004).	
Minha Casa, Minha Vida	O Programa Minha Casa, Minha Vida tem por finalidade promover o direito à cidade e à moradia de famílias residentes em áreas urbanas e rurais, associado ao desenvolvimento urbano e econômico, à geração de trabalho e de renda e à elevação dos padrões de habitabilidade e de qualidade de vida da população BRASIL (2023a).	Medida Provisória (MP) 1.162/2023
Programa de Erradicação do Trabalho Infantil	Conjunto de ações que tem o objetivo de retirar crianças e adolescentes menores de 16 anos do trabalho precoce, exceto na condição de aprendiz a partir de 14 anos. O programa, além de assegurar transferência direta de renda a famílias, oferece a inclusão de crianças e jovens em serviços de orientação e de acompanhamento. A frequência à escola também é exigida BRASIL (1993).	LEI Nº 8.742/1993
Programa de Integração Social	Programa de Integração Social, destinado a promover a integração do empregado na vida e no desenvolvimento das empresas Brasil (1970).	Lei Complementar nº 7/1970
Seguro-Defeso	O seguro-desemprego do pescador artesanal é uma assistência financeira temporária concedida aos pescadores profissionais artesanais que, durante o período de defeso (aquele em que os animais protegidos normalmente se reproduzem), são obrigados a paralisar a atividade para preservação da espécie e sustentabilidade da atividade pesqueira na região. BRASIL (2003b).	LEI Nº 10.779/2023
Bolsa Capixaba - Espírito Santo	O Bolsa Capixaba é um benefício de transferência de renda do Governo do Espírito Santo para atender pessoas em situação de extrema pobreza que não recebem o Auxílio Brasil Lício et al. (2018).	LEI Nº 11.519/2021
Panela Cheia - Mato Grosso	Destinado a ações de transferência de renda com condicionalidades Lício et al. (2018).	LEI Nº 9.296/2009 LEI Nº 10.523/2017
Família Paranaense	O Família Paranaense é um programa estratégico que tem como atribuição primordial a articulação das políticas públicas de várias áreas dos governos estadual e municipal com outros diferentes setores da sociedade, com vistas ao protagonismo, à proteção e à emancipação das famílias que vivem em situação de maior vulnerabilidade social em todo o Estado Lício et al. (2018).	Lei Nº 17734/2013
Santa Renda Santa Catarina	Famílias do PBF com renda inferior à linha de extrema pobreza Lício et al. (2018).	Decreto nº 1.341/2021
DF Sem Miséria Distrito Federal	O Plano DF sem Miséria tem por objetivos reduzir as desigualdades sociais e a superação da extrema pobreza, elevar a qualidade de vida da população pobre e extremamente pobre e ofertar serviços públicos voltados às famílias pobres e extremamente pobres Lício et al. (2018).	Lei nº 4.737/2011
Adjunto da Solidariedade Acre	O Programa Adjunto da Solidariedade terá suas ações voltadas para as áreas de Educação e de Geração de Renda, com transferência direta de renda à população carente do Estado do Acre, mediante a concessão de Bolsas ou Auxílios definidos nesta lei Lício et al. (2018).	Lei Ordinária Nº 1363/2001

Bolsa Futuro Rondônia	Incentivo financeiro no valor de R\$ 1.200,00 destinado a estudantes de baixa renda concluintes da educação básica Lício et al. (2018).	Projeto de Lei nº 1942/2023
Renda Cidadã São Paulo	O programa atende famílias que vivenciam processo de vulnerabilidade em decorrência do desemprego ou subemprego, envolvendo fatores como a ausência de qualificação profissional, o analfabetismo, a situação precária de saúde, a inexistência de moradia ou sua precariedade, a dependência química, entre outros problemas e dificuldades Lício et al. (2018).	DECRETO Nº 46.550/2002
Renda Melhor Rio de Janeiro	O Programa Renda Melhor Jovem tem como objetivo incentivar o jovem que tenha ingressado no ensino médio com até 18 anos incompletos, em situação de extrema pobreza, integrantes de famílias beneficiários do Programa Renda Melhor, através de um incentivo financeiro que será depositado anualmente Lício et al. (2018).	DECRETO Nº 47.892/2021

Fonte: O autor, a partir de dados da pesquisa.

A implementação de uma política pública compreende o espaço temporal no qual seus resultados concretos são observáveis (Secchi, 2010). Semanalmente, Jeffrey Pressman e Wildavsky (1973) a descrevem como um processo de interação entre uma gama de objetivos e as ações definidas para atingi-los.

Considerando esse contexto e as previsões apresentadas relacionadas a pandemia da COVID-19, aproximadamente 151 países promoveram ações para implementação ou fortalecimento de políticas de mercado de trabalho, seguridade e assistência social (Gentilini et al., 2020). Dentre as medidas adotadas, aquelas relacionadas à assistência social receberam maior destaque, com a criação ou expansão de programas de transferência de renda.

Em sua última atualização datada de fevereiro de 2022, Gentilini et al. (2020) apresenta um total de 3.856 medidas de proteção social e trabalho que foram planejadas ou implementadas por 223 economias mundiais para conter os efeitos da pandemia. Segundo o autor, os programas de transferência de renda foram a intervenção de assistência social mais difundida: elas incluíram 962 medidas em 203 países, representando cerca de 25% do total de medidas de proteção social e trabalho e 41% da assistência social geral. Porém, devido à situação contínua da pandemia, ocorreram diversas extensões com alguns casos interessantes de evolução do valor e cobertura das transferências. Cerca de 204 programas de assistência em dinheiro em 108 países foram estendidos ou reintroduzidos devido ao impacto persistente e negativo da COVID-19 na vida e subsistência das pessoas.

O Programa Social Brasileiro Alfa representa um exemplo de resposta no Brasil a crises econômicas e sociais, ajustando-se em períodos diferentes para atender às

necessidades emergentes da população. Inicialmente, proporcionou apoio financeiro significativo, adaptando-se posteriormente com alterações nos valores e na cobertura para refletir as condições em mudança.

O Programa Social Brasileiro ALFA do Brasil este entre os maiores do mundo, em relação a quantidade de beneficiários. Países populosos como: Índia, Estados Unidos, Japão, Paquistão, China e Filipinas totalizaram cerca de 125.413,363 beneficiários em média.

O Quadro 13 sintetiza os programas de transferência de renda implementados em diferentes países e economias durante a pandemia da COVID-19. Nele, é possível verificar o número de beneficiários alcançados por cada um desses programas de transferência de renda (Gentilini et al., 2020).

Quadro 13 – Síntese dos programas sociais

Quantidade	Descrição
124	Programas de transferência de renda no mundo
124	Países que criaram algum tipo de transferência de renda
200.000.000	Número Máximo de beneficiários em um único programa
119	Número Mínimo de beneficiários em um único programa
772.739	Mediana de Beneficiários em todos os programas
10.940.490	Média de Beneficiários em todos os programas
1.356.620.798	Total de beneficiários no mundo
6.944.367	Total de beneficiários na Oceania
89.236.829	Total de beneficiários na África
97.715.915	Total de beneficiários na Europa
832.013.780	Total de beneficiários na Ásia
330.709.907	Total de beneficiários na América
140.554.593	Total de beneficiários na América do Sul
7º	Posição do Brasil em número de beneficiários no mundo
2º	Posição do Brasil em número de beneficiários na América

Fonte: o autor, adaptado de Gentilini et al. (2020)

Apesar dos desafios enfrentados em todo o mundo devido à pandemia da COVID-19, muitos países e economias adotaram programas de transferência de renda como parte de suas estratégias de apoio à população. Essas iniciativas buscaram fornecer suporte financeiro direto às famílias e indivíduos afetados pelos impactos econômicos da pandemia, garantindo um mínimo de segurança econômica durante aquele período desafiador. A abrangência desses programas demonstrou a importância do apoio social e da mitigação dos efeitos negativos causados pela crise global.

O Quadro 14 apresenta dados relacionados aos beneficiários de programas sociais em diferentes continentes: África, América, Ásia, Europa e Oceania. Nele, são

destacados os valores mínimos, médios, medianos e máximos de beneficiários desses programas em cada continente, oferecendo uma visão abrangente da extensão e da diversidade dessas iniciativas ao redor do mundo.

Quadro 14 - Síntese do total de beneficiários por continente

Continente	Mínimo	Médio	Mediano	Máximo
África	119	2.411.806	413.260	28.798.915
América	127	13.228.396	1.200.000	162.000.000
Ásia	6.509	21.793.579	5.086.080	116.520.000
Europa	404	3.758.304	182.410	25.242.145
Oceania	535	1.157.395	7.378	6.800.000

Fonte: o autor, adaptado de Gentilini et al. (2020)

Além das medidas relacionadas a transferência de renda, diversas outras medidas foram adotadas com o objetivo de minimizar os impactos da pandemia, em especial junto as famílias mais carentes. O Quadro 14 apresenta as economias mundiais que implementaram medidas relacionadas a transferências em renda, obras públicas, distribuição de alimentos (in natura/merenda escolar), apoio em serviços públicos e financeiros, licença remunerada/desemprego, apoio ao seguro de saúde, pensões e benefícios por incapacidade, isenção/subsídio de contribuições previdenciárias, subsídio salarial, treinamento, ajuste na regulamentação trabalhista e subsídio de redução do tempo de trabalho para aplicação durante a pandemia. Essas medidas estão relacionadas à assistência social, seguro social e mercado de trabalho Gentilini et al. (2020).

O Quadro 26 do apêndice F, apresenta uma visão geral das medidas de proteção social implementadas em diversos países, categorizadas em três componentes principais: Assistência Social, Seguro Social e Mercado de Trabalho. As medidas incluem transferências em dinheiro, trabalho público, alimentação escolar, utilidade e apoio financeiro, licença remunerada/desemprego, apoio ao seguro de saúde, pensões e benefícios por invalidez, contribuições para a segurança social (isenção/subsídio), subsídio salarial, ativação (treinamento), ajuste da regulamentação trabalhista, e subsídio por tempo de trabalho reduzido.

Através deste Quadro 26, é possível observar a diversidade de políticas adotadas pelos países para apoiar seus cidadãos em diferentes aspectos da segurança social e econômica. Por exemplo, enquanto algumas nações oferecem uma ampla gama de benefícios cobrindo quase todos os componentes listados, outras focam em áreas específicas. Alguns países, como a Armênia e o Azerbaijão, implementam medidas em quase todas as categorias, indicando um amplo sistema de proteção social. Em contraste, outros têm abordagens mais limitadas,

concentrando-se em poucos aspectos como transferências em dinheiro e alimentação escolar.

De forma mais específica, a partir do Quadro 26 do apêndice F é possível inferir, que em relação ao componente de assistência social, as transferências em dinheiro foram amplamente implementadas em 91% dos países analisados. Apenas os países independentes, como Liechtenstein, Nicarágua e Papua Nova Guiné, e os territórios autônomos, como Ilhas Faroe, Gibraltar, Groenlândia e Ilha de Man, não tiveram subsídios em dinheiro registrados. Além disso, 74% dos países criaram programas de alimentação escolar como forma de apoio durante o período analisado (Gentilini et al., 2020).

No que diz respeito ao seguro social, a licença remunerada/desemprego foi aplicada em 56% dos países, indicando que muitas nações ofereceram esse suporte aos trabalhadores afetados pela pandemia. Além disso, 28% dos países forneceram apoio ao seguro de saúde, bem como pensões e benefícios por invalidez. Essas medidas visavam garantir a segurança financeira dos indivíduos em meio à crise. O emprego no setor público também foi uma alternativa adotada por 23% dos países (Gentilini et al., 2020).

No mercado de trabalho, 52% dos países implementaram alguma medida subsídio salarial, adicionalmente, 35% dos países criaram programas de treinamento profissional com o objetivo de aprimorar suas habilidades e aumentar suas chances de emprego pós-pandemia. O subsídio por tempo de trabalho reduzido também foi oferecido em 40% países, enquanto 52% dos países criaram mecanismos de ajuste na regulamentação trabalhista (Gentilini et al., 2020).

A tecnologia da informação foi peça fundamental para superação de vários desafios durante a pandemia, como por exemplo na telemedicina (Kieling e Magnagnagno, 2021), no ensino e aprendizagem (Vaz e Pelicioni, 2021) e até no cuidado de idosos (Da et al., 2022).

A tecnologia vem trazendo mudanças significativas na vida das pessoas, facilitando o acesso à informação, produtos, serviços etc. Por conseguinte, as políticas sociais também acompanham os avanços tecnológicos. Contudo, sabemos que diante da desigualdade social existente no Brasil, nem todos conseguem navegar facilmente na internet ou possuem meios para tal, principalmente o público-alvo das políticas sociais, que são as pessoas em situação de vulnerabilidade social (Gomes, 2021).

Neste enquadramento, a crescente operacionalização do ciberespaço pela administração pública impulsionada pela COVID-19 deu origem a inovações, desafios, vulnerabilidades e lições (Medeiros, 2020). Visando responder às demandas impostas

pela pandemia, entidades públicas e privadas transferiram atividades para o ciberespaço. Contudo, a utilização dessa alternativa para atividades profissionais e educacionais, bem como para a coordenação de políticas públicas, potencializou as vulnerabilidades técnicas e sociais do ciberespaço. Segundo Sampaio (2016), enquanto foi necessário admitir as vantagens da participação da sociedade no ciberespaço, foi preciso reconhecer as limitações existentes.

Apesar das limitações, observamos nos últimos anos que o advento da informatização no setor público contribuiu com maior agilidade e eficiência na entrega de ações e serviços para a população. Nesse aspecto, destacamos a importância do uso das tecnologias da informação na implementação das políticas sociais, como é o caso do Cadastro Único para Programas Sociais - CADUNICO¹¹ do Governo definido como instrumento de identificação e caracterização socioeconômica das famílias brasileiras de baixa renda (Direito et al, 2016, p. 3), que por sua vez é constituído por sua base de dados, instrumentos, procedimentos e sistemas eletrônicos (BRASIL, 2007).

Para ser elegível ao auxílio, além de não ser beneficiário de outros programas sociais, exceto o Programa Bolsa Família, ou seguro-desemprego¹², a renda familiar mensal deveria ser inferior a meio salário-mínimo per capita ou três salários-mínimos no total, de acordo com as determinações estabelecidas.

As mães que viviam sozinhas e atendiam aos critérios de elegibilidade de renda tinham direito a receber o auxílio em dobro. Em complemento, no dia 16 de abril, a Câmara dos Deputados aprovou a extensão da medida para todas as famílias monoparentais¹³ e mães menores de idade.

A Empresa BETA teve papel importante na viabilização do Programa Social Brasileiro Alfa, ao realizar o cruzamento dos dados e informações necessárias. Como uma entidade pública focada em soluções de Tecnologia da Informação e Comunicação, seu objetivo principal é apoiar o aprimoramento e a execução de políticas sociais do Estado brasileiro. A atuação da Empresa BETA é fundamental para garantir a eficácia e a transparência na gestão de programas sociais,

¹¹ O Cadastro Único é um registro que permite ao governo saber quem são e como vivem as famílias de baixa renda no Brasil. Ele foi criado pelo Governo Federal, mas é operacionalizado e atualizado pelas prefeituras de forma gratuita. Ao se inscrever ou atualizar seus dados no Cadastro Único, você pode tentar participar de vários programas sociais. Cada programa tem uma exigência diferente, mas o primeiro passo é ter sempre seu cadastro atualizado.

¹² O Seguro-Desemprego é um benefício integrante da seguridade social, garantido pelo art.7º dos Direitos Sociais da Constituição Federal e tem por finalidade prover assistência financeira temporária ao trabalhador dispensado involuntariamente.

¹³ O termo "monoparentais" refere-se a famílias compostas por apenas um dos pais ou responsáveis legais, assumindo a responsabilidade principal pelo cuidado e criação dos filhos. Geralmente, esse tipo de família ocorre quando um dos pais é solteiro, divorciado, separado ou viúvo. Nas famílias monoparentais, uma única pessoa desempenha o papel de pai ou mãe, assumindo todas as responsabilidades relacionadas ao sustento, educação e bem-estar dos filhos, sem a presença de um parceiro ou cônjuge.

contribuindo para a identificação e mitigação de riscos associados à implementação dessas iniciativas.

A partir da publicação de decretos e normativos foram regulamentadas as competências de determinados órgãos na implementação do Programa Social Brasileiro Alfa, sendo ali estabelecida a autoridade competente para gerir o Programa Social Brasileiro Alfa. Ao mesmo passo, a Empresa foi definida como agente operador, responsável pelo processamento das informações necessárias para a geração da folha de pagamento do Programa Social Brasileiro Alfa (Cardoso, 2020), cabendo a ela realizar a identificação dos cidadãos que tinham direito ao benefício por meio do cruzamento das informações.

Com o propósito de facilitar e tornar viáveis as ferramentas necessárias, foram desenvolvidos 18 módulos tecnológicos no portal de consulta da Empresa de Processamento BETA, assim como diversas interfaces destinadas aos gestores de políticas, instituições de controle e fiscalização, além dos cidadãos. (Empresa de Processamento BETA, 2020)

Adicionalmente, visando assegurar o processamento adequado das informações, a Empresa de Processamento BETA criou um software avançado para executar a verificação dos dados durante o processo do Programa Social Brasileiro Alfa. Esse software foi projetado para analisar e comparar informações provenientes de 27 fontes de informações, garantindo a veracidade e a integridade dos dados utilizados para identificar os beneficiários elegíveis (Empresa de Processamento BETA, 2020). Complementarmente foram disponibilizados outros dois softwares desenvolvidos por parceiros.

Segundo dados oficiais, entre abril e agosto de 2020 68.243.993 cidadãos eram elegíveis ao recebimento do benefício, sendo 38.2 milhões a partir de novos cadastros no App.

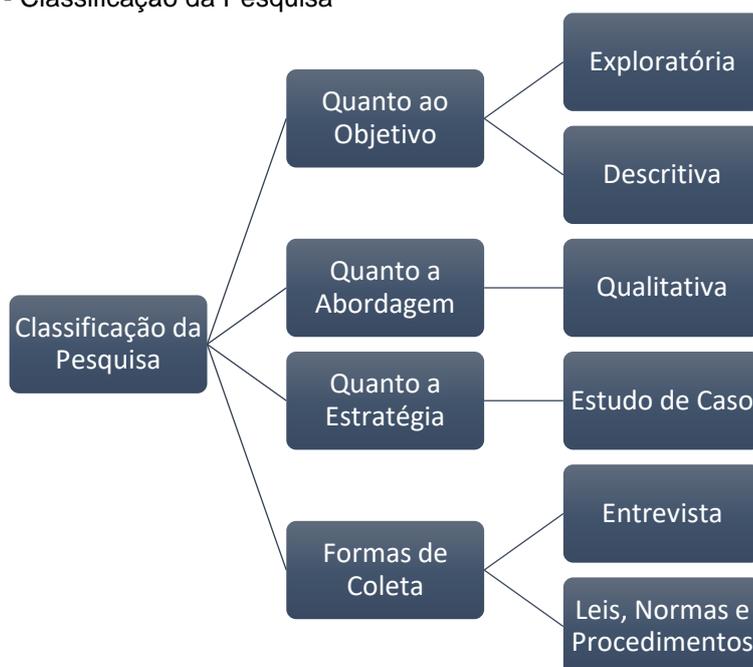
3 MÉTODO

Nesta sessão, é descrita a metodologia adotada nesta pesquisa. A Seção 3.1 exhibe a caracterização/tipologia de pesquisa, a Seção 3.2 trata sobre a caracterização dos instrumentos de pesquisa, a Seção 3.3 apresenta a caracterização da organização em estudo, a Seção 3.4 descreve os procedimentos de coleta de dados e a Seção 3.5 exhibe os procedimentos análise dos dados.

3.1 Caracterização/tipologia de pesquisa

Os tipos de pesquisa podem ser classificados de acordo com sua abordagem, natureza, objetivos, estratégia e forma de coleta dos dados. Neste sentido, a Figura 8 apresenta a classificação da pesquisa e métodos utilizados conforme Hernández Sampieri et al. (2013)

Figura 8 - Classificação da Pesquisa



Fonte: o autor, adaptado de Sampieri, Collado, Lucio (2013).

Baseado na Figura 8 e considerando as definições fornecidas por Sampieri; Collado; Lucio (2013), esta pesquisa pode ser classificada da forma a seguir descrita.

Quanto ao atingimento dos objetivos, esta pesquisa possui características descritivas, pois propõe a busca consiste em detalhar as características e perfis de grupos, processos ou objetos que podem ser submetidos a análises. Também, possui características de pesquisa exploratória, pois envolve um levantamento documental e bibliográfico com entrevistas com responsáveis pelo processo de desenvolvimento de software, além de realizar observações e análises de exemplos reais do processo.

Quanto à natureza, trata-se de uma pesquisa aplicada, pois visa gerar conhecimento acerca da problemática com aplicação prática na análise dos riscos do processo de desenvolvimento do software do Programa Social Brasileiro Alfa. O objeto de estudo é de importância para a organização, dada a relevância do produto.

No que se refere à abordagem, trata-se de uma pesquisa qualitativa, já que para Sampieri, Collado e Lucio (2013) por meio dessa abordagem, procura-se entender a perspectiva dos participantes sobre os eventos e casos que os cercam em um ambiente natural, aprofundando em suas experiências e pontos de vista, na maneira como enxergam sua realidade.

Quanto a estratégia, caracteriza-se por um estudo de caso. Segundo Yin (2015) o estudo de caso é a estratégia escolhida ao se examinarem acontecimentos contemporâneos, mas quando não se podem manipular comportamentos relevantes. Ainda segundo o autor, um estudo de caso é uma investigação empírica que investiga um fenômeno contemporâneo dentro de seu contexto da vida real, especialmente quando os limites entre o fenômeno e o contexto não estão claramente definidos.

A escolha deste estudo de caso específico, referente ao software provedor do Programa Social Alfa, foi motivada pela sua natureza singular durante um período pandêmico. O programa representa um exemplo único de resposta social e tecnológica frente a desafios enfrentados, o que o torna um caso especialmente relevante e rico para análise.

Segundo Yin (2015) os estudos de caso não precisam ficar limitados a uma única fonte de evidências. Na verdade, a maioria dos melhores estudos baseia-se em uma ampla variedade de fontes, permitindo assim que o pesquisador se dedique a uma ampla diversidade de questões históricas, comportamentais e de atitudes.

Esta pesquisa propõe a análise no lócus Empresa BETA. A caracterização da Empresa BETA está detalhada na Seção 3.3.

3.2 Caracterização dos instrumentos de pesquisa

Os instrumentos de pesquisa são fundamentais para que o pesquisador levante dados avalie o alcance das ações realizadas. Entre os instrumentos comumente utilizados estão as observações, os questionários, e as entrevistas (USP, 2012). Nesta pesquisa, será dada prioridade à utilização de fontes de dados secundários (documentos) e de dados primários (entrevistas), sendo descrita, nesta seção, a caracterização do roteiro de entrevistas usado neste estudo.

A entrevista caracteriza-se como uma das principais técnicas de coletas de dados e pode ser definida como conversa realizada face a face pelo pesquisador junto ao entrevistado, seguindo um método para se obter informações sobre determinado assunto (Cervo; Bervian, 2002). As entrevistas representam fontes de dados que neste trabalho, foram instrumentalizadas por meio de perguntas pré-estabelecidas em um roteiro semiestruturado.

Neste estudo, o roteiro de entrevistas semiestruturado contempla sete perguntas (Apêndice B). As cinco primeiras perguntas relacionadas ao que se pretende coletar de dados que subsidie a identificação de riscos em processo de desenvolvimento de software. Essas perguntas foram baseadas na fundamentação teórica, especialmente na Taxonomia SEI, descrita de forma geral na Figura 7 e detalhada no Quadro 10 da fundamentação teórica deste estudo. As duas últimas perguntas tratam sobre questões sociodemográficas com o objetivo de caracterizar o perfil dos entrevistados (Cavalcante et al. , 2018).

Com vistas a garantir a validade do instrumento de pesquisa, efetuou-se teste piloto do roteiro de entrevista. Segundo Yin (2015), o teste piloto pode ser considerado uma estratégia metodológica que auxilia o pesquisador a validar o instrumento de pesquisa desenhado, pois é aplicado antes dele entrar em contato com os sujeitos delimitados para o estudo.

Ainda segundo Canhota (2008), Mackey e Gass (2005), a realização de um estudo piloto permite testar, avaliar, revisar e aprimorar os instrumentos e procedimentos de pesquisa. Ao conduzir um estudo piloto, busca-se identificar pontos fracos e possíveis problemas, a fim de resolvê-los antes da implementação efetiva da pesquisa. Essa abordagem visa assegurar a qualidade e a eficácia da pesquisa.

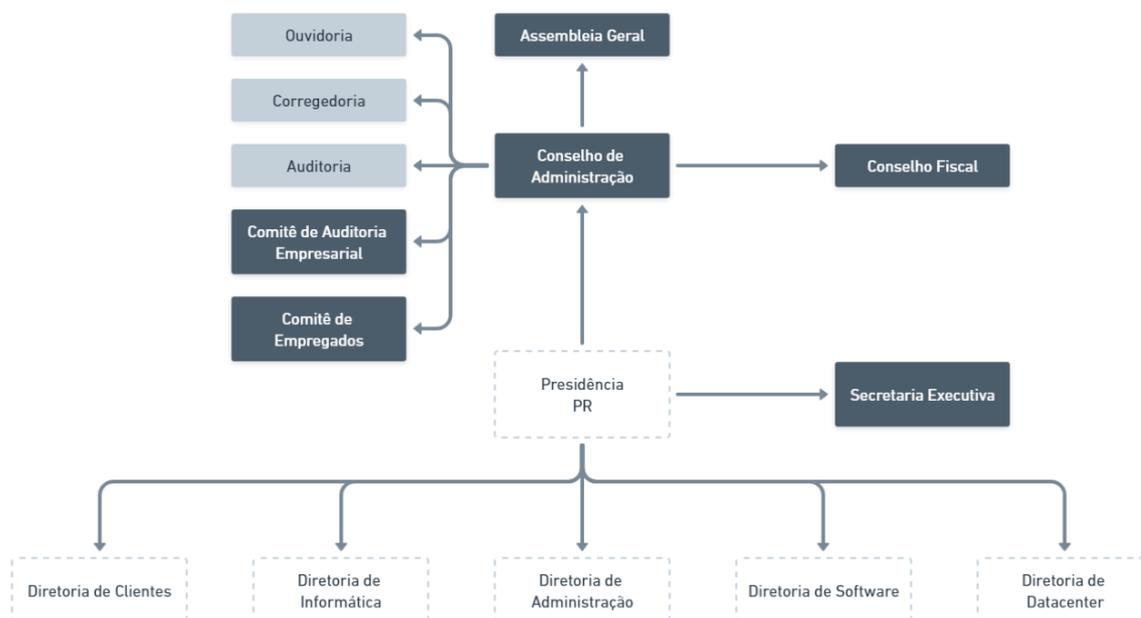
Em complemento, considerando que as entrevistas foram realizadas a partir de meios digitais, fez-se necessário dispor de um método de registro, coleta e sintetização de dados relevantes, assim optou-se pela utilização de fichas analíticas. Essas fichas (Ramalho, 2015) permitiram a identificação dos atributos a partir das entrevistas realizada. Segundo Rocha (2022) essa abordagem utilizando fichas analíticas possibilita uma compreensão mais profunda dos resultados obtidos, partindo das palavras contidas nas transcrições das entrevistas e das menções literais presentes nos textos dos documentos analisados. O modelo das fichas analíticas utilizadas está disponível no Apêndice C deste trabalho.

3.3 Caracterização da organização em estudo

A Empresa BETA é uma empresa pública que provê tecnologia de qualidade ao cidadão. Organizada em Diretorias, com responsabilidades distintas que abrangem desde o desenvolvimento de software e a manutenção de infraestrutura de datacenters até a gestão de relações empresariais e o bem-estar dos colaboradores.

As atividades essenciais de desenvolvimento de software estão a cargo da Diretoria de Software, o suporte a infraestruturas de datacenter está a cargo da Diretoria de Datacenter. A Diretoria de Clientes foca na busca por novos negócios, a Diretoria de Informática é responsável pelo ambiente corporativo de dados analíticos e, por fim, a Diretoria de Administração dedica-se à gestão de recursos humanos. A presidência da organização assume uma posição estratégica, visando a concretização dos objetivos corporativos e mantendo um relacionamento direto com órgãos internos e externos vitais, tais como secretarias executivas, auditoria interna, entre outros órgãos de governança e fiscalização. Um organograma, disponível em documentação interna da Empresa de Processamento BETA, ilustra a estrutura organizacional e as interconexões entre os diversos departamentos e funções.

Figura 9 - Organograma Empresa BETA.



Fonte: O autor adaptado a partir de dados da pesquisa.

3.4 Perfil dos participantes

Frente a análise de dados coletados da Empresa BETA, edição de janeiro de 2020, identificaram-se as áreas envolvidas no processo de desenvolvimento de

software, sustentação de datacenter e gestão contratual na organização. O objetivo foi mapear os perfis dos profissionais que, potencialmente, poderiam ser entrevistados ao longo da pesquisa.

Neste contexto, após identificar as Diretorias de Software, Datacenter e Administração como fontes potenciais de entrevistados, a seleção de participantes foi refinada para assegurar a obtenção de informações relevantes e detalhadas sobre o desenvolvimento do software em questão. Reconhecendo que nem todos os funcionários dessas diretorias estariam diretamente envolvidos no projeto de desenvolvimento dos softwares provedores do programa social Alfa devido à diversidade de sistemas em operação e as especificidades de cada função, a estratégia de seleção adotada buscou intencional.

A metodologia de seleção, inspirada nas recomendações de Patton (2015) para a seleção dos entrevistados, priorizou a identificação de indivíduos cujas contribuições ao projeto eram mais relevantes para os objetivos da pesquisa. Patton enfatiza a importância de selecionar participantes que possam fornecer conhecimentos profundos sobre o objeto de estudo, uma estratégia adotada aqui para garantir a coleta de dados primários pertinentes ao foco deste trabalho.

Além disso, a busca por uma variedade de perspectivas, incorporando desde gestores de projeto e desenvolvedores, arquitetos a especialistas em TI e profissionais da gestão contratual, foi alinhada à abordagem de Patton (2015) de garantir diversidade e profundidade nas entrevistas. Esta diversificação de pontos de vista, visa oferecer uma compreensão abrangente e detalhada dos processos envolvidos no desenvolvimento e gestão de projetos de software dentro da organização.

O Quadro 15 foi elaborado com o objetivo para apresentar o total de funcionários segmentados por diretoria que de alguma forma estiveram envolvidos no projeto de desenvolvimento dos softwares provedores do programa social Alfa, bem como o total de potenciais entrevistados por diretoria.

Quadro 15 – Total de funcionários por diretoria DIT e DPS.

Diretoria	Total de funcionários envolvidos com o processo de desenvolvimento de software do programa social ALFA	Total de potenciais entrevistados
Diretoria de Data Center	15	8
Diretoria de Desenvolvimento	28	8
Diretoria de Administração	5	2

Fonte: o autor, a partir de dados da pesquisa.

Adicionalmente, o Quadro 16 detalha as superintendências que estão sob a alçada das diretorias previamente identificadas como relevantes para o estudo, selecionadas com base em seu papel conforme definido no manual de atribuições.

O Quadro 16 introduz uma coluna denominada “atuação predominante”, especificamente desenhada para mapear a principal esfera de atuação de cada superintendência em relação aos elementos de classe apresentados no Quadro 11, conforme classificado pelo SEI. Tal especificação não restringiu a possibilidade de que, durante as entrevistas, fossem identificados atributos relacionados a outros elementos de classe vinculados à Taxonomia SEI.

A inclusão da coluna no Quadro 16, citada no parágrafo anterior e a subsequente categorização das diretorias têm como finalidade assegurar uma abordagem metódica e integrada à coleta de dados, evitando-se assim a consideração de elementos descontextualizados, em consonância com as práticas qualitativas recomendadas por Minayo (2017). Esta estratégia contribui para uma compreensão mais acurada dos riscos associados ao desenvolvimento de software, alinhando-se ao objetivo de produzir um estudo qualitativo robusto e coeso.

Quadro 16 – Visão de atuação predominante por classe

Classe	Diretoria	Área	Atuação predominante com base nos elementos de classe do quadro 11
Engenharia de Produto	Diretoria de Datacenter	Coordenação de Arquitetura	Especialidades de Engenharia
		Coordenação de Gestão de Data Center	Integração e Teste
		Coordenação de Gestão de Serviços de TIC	Design
	Diretoria de Desenvolvimento	Coordenação de Gestão Tecnológica	Requisitos
		Coordenação de Produtos	Design
Ambiente de Desenvolvimento	Diretoria de Datacenter	Coordenação de Gestão de Data Center	Métodos de Gestão
		Coordenação de Gestão de Serviços de TIC	Métodos de Gestão
	Diretoria de Desenvolvimento	Coordenação de Gestão Tecnológica	Sistema de Desenvolvimento
Restrições do Programa	Diretoria de Datacenter	Coordenação de Gestão de Serviços de TIC	Recursos
	Diretoria de Desenvolvimento	Coordenação de Produtos	Interface do Programa
	Diretoria de Administração	Coordenação de Compras	Contrato

Fonte: o autor, a partir de dados da pesquisa.

Após a escolha e organização dos potenciais entrevistados, o próximo passo consistiu em selecionar participantes para as entrevistas em um número que seja capaz de refletir toda a complexidade do objeto investigado. Essa estratégia, descrita por Minayo (1999), visa alcançar uma compreensão abrangente do tema em todas as suas dimensões e é definida por saturação teórica.

É importante destacar que na abordagem qualitativa, a seleção dos participantes não deve seguir critérios quantitativos pelo simples fato de que o objetivo não é quantificar opiniões ou fenômenos, mas explorar uma gama de depoimentos, realizar análise de conteúdo e se aprofundar nas relações e nas diferentes representações que um problema pode acarretar (Gaskell,2000).

Além de definir com clareza o perfil dos potenciais participantes da entrevista, é igualmente relevante estabelecer uma ordem para a coleta de entrevistas, uma vez que nem todos os indivíduos aptos para participar do estudo serão consultados. Essa abordagem permite concentrar os esforços nas fontes mais representativas para o aprofundamento da pesquisa.

Foi estabelecida uma sequência para as entrevistas com os potenciais entrevistados, intercalando a seleção deles entre diretorias, áreas e funções. Esse critério de intercalação foi estabelecido a fim de se garantir uma visão ampla sobre a gestão de risco empregada ao longo do processo a partir dos diversos pontos de vista.

Inicialmente foi definida uma ordem de participação das diretorias e superintendências, sendo que ao término de cada rodada completa, a lista foi reiniciada. Após definida a ordem de participação das áreas, os potenciais entrevistados foram numerados de E1 a E18, em seguida foi realizado um sorteio aleatório definido a ordem de realização da entrevista, conforme Quadro 17.

Quadro 17 - Ordem para realização das entrevistas.

Diretoria	Área	Entrevistado
Diretoria de Datacenter	Coordenação de Arquitetura	E01
Diretoria de Desenvolvimento	Coordenação de Gestão Tecnológica	E02
Diretoria de Administração	Coordenação de Contratos	E03
Diretoria de Datacenter	Coordenação de Gestão de Data Center	E04
Diretoria de Desenvolvimento	Coordenação de Produtos	E05
Diretoria de Administração	Coordenação de Contratos	E06
Diretoria de Datacenter	Coordenação de Gestão de Serviços de TIC	E07
Diretoria de Desenvolvimento	Coordenação de Gestão Tecnológica	E08
Diretoria de Administração	Coordenação de Contratos	E09
Diretoria de Datacenter	Coordenação de Gestão de Data Center	E10
Diretoria de Desenvolvimento	Coordenação de Produtos	E11
Diretoria de Administração	Coordenação de Contratos	E12
Diretoria de Datacenter	Coordenação de Gestão de Serviços de TIC	E13

Diretoria de Desenvolvimento	Coordenação de Gestão Tecnológica	E14
Diretoria de Administração	Coordenação de Contratos	E15
Diretoria de Datacenter	Coordenação de Gestão de Data Center	E16
Diretoria de Desenvolvimento	Coordenação de Produtos	E17
Diretoria de Administração	Coordenação de Contratos	E18

Fonte: o autor, a partir de dados da pesquisa.

3.5 Procedimentos de coleta de dados

Os procedimentos para a coleta de dados neste estudo de caso abrangeram pesquisas bibliográfica, documental e coleta de entrevistas. Inicialmente, foram obtidos os artigos científicos, normas, regulamentos, procedimentos e modelos de gestão de riscos que serviram como embasamento teórico para a presente pesquisa. Conforme destacado na Tabela 1, foram utilizadas diversas bases de dados para realizar investigações sobre o tema da gestão de riscos no desenvolvimento de software. Os filtros utilizados na busca incluíram os termos "Risk Management and Development Software" e foram aplicados critérios específicos, como área de assunto relacionada a "Computação" ou "Engenharia", e tipo de documento limitado a artigos de pesquisa.

As pesquisas às bases de dados citadas no parágrafo anterior foram realizadas nas bases SCOPUS (Elsevier), Scientific Electronic Library Online (SciELO), SAGE Journals Online, SpringerLink, Web of Science, Wiley Online Library e Biblioteca Digital Brasileira de Teses e Dissertações - BDTD. Esse embasamento teórico contribuiu para a elaboração da fundamentação teórica, incluindo a enumeração de fatores relacionados à implementação da gestão de riscos no desenvolvimento de software.

Para assegurar a coleta de documentos e das entrevistas, foi solicitado previamente o acesso a documentos não sigilosos relacionados às fases do processo de desenvolvimento de software da Empresa BETA. Adicionalmente, foi solicitada autorização para convidar empregados para participar de entrevistas. Essas solicitações para acesso a dados foram formalizadas nos termos descritos no Apêndice A deste estudo.

Com a autorização obtida, iniciou-se o processo de convite aos potenciais entrevistados, que foram selecionados com base em sua relevância e envolvimento no projeto em questão a partir dos critérios já apresentados. Para formalizar o convite e explicar os objetivos e a natureza da pesquisa, cada candidato recebeu, individualmente, um e-mail contendo uma apresentação detalhada do estudo e do

pesquisador, conforme documentado no Apêndice D. Neste Email, enfatizou-se a participação voluntária e foi oferecida a possibilidade de agendar a entrevista em uma data e horário que melhor conviessem ao entrevistado, respeitando sua disponibilidade. Esta abordagem, recomendada por McDERMID et al. (2014), visa facilitar a adesão e garantir que os participantes se sintam confortáveis e informados sobre o processo de pesquisa.

Em seguida, procedeu-se à coleta de documentos não sigilosos da Empresa BETA, utilizando-se de buscas nos repositórios internos, analisando a cadeia de valor, normas, regulamentos e as comunicações de serviço que detalham as etapas do processo de desenvolvimento de software adotadas pela empresa. Para identificar os documentos caracterizados como não sigilosos, foi efetuado, inclusive, exame da classificação de informação de documentos da referida empresa.

Além da pesquisa documental, foram realizadas doze entrevistas. O roteiro de entrevistas foi previamente citado na Seção 3.2 e consta do Apêndice B deste estudo. Durante as entrevistas, foram obtidas informações relevantes de profissionais que não necessariamente eram especialistas em gestão de risco, porém, participaram do projeto de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa. A seleção dos entrevistados foi circunscrita àqueles previamente identificados com base no Quadro 18, previamente exibido na Seção 3.4 deste estudo.

As entrevistas foram agendadas de acordo com a disponibilidade de cada entrevistado, com duração de até uma hora, de forma virtual na plataforma Teams da Microsoft. Antes de cada entrevista, foram reforçados os cuidados éticos, incluindo a preservação do anonimato do entrevistado e a análise conjunta e confidencial das respostas (Dermid et al., 2014).

As entrevistas foram gravadas com a autorização prévia dos entrevistados. Posteriormente, todas as entrevistas foram transcritas. Segundo Halcomb e Davidson (2006) transcrever consiste na reprodução das palavras faladas, como as que provêm de uma entrevista gravada, em texto escrito.

As entrevistas foram realizadas entre 25/09/2023 e 22/12/2023, sendo finalizada quando do atingimento da saturação teórica Falqueto et al. (2018). É importante destacar que quando o roteiro de entrevista é adequado, o ponto de saturação geralmente é atingido em 15 entrevistas (Thiry-Cherques, 2009).

Nesta dissertação o ponto de saturação ocorreu na entrevista de número nove, considerando todas as classes da Taxonomia SEI citadas no Quadro 10. Em outras palavras, nenhuma nova informação foi identificada e considerada relevante após a nona entrevista. Após essa constatação, três entrevistas foram realizadas para a

necessária confirmação, com base em Thiry-Cherques (2009) que recomenda que sejam feitas ao menos duas entrevistas adicionais depois de encontrado o ponto de saturação.

Os doze entrevistados selecionados para o estudo apresentam elevado grau de experiência em processos de desenvolvimento de software e diferentes níveis de responsabilidade durante o projeto. A maioria dos participantes, especificamente onze (91,66%) deles reportou possuir uma experiência substancial, de quatro anos ou mais, no desenvolvimento de software na época do projeto.

Quanto às responsabilidades ocupadas durante o processo de desenvolvimento, observa-se que a maioria, nove (75,00%) dos doze entrevistados, ocupava alguma função de confiança ou comissionada, o que sugere que esses participantes estavam envolvidos em papéis de liderança ou tomada de decisão crítica no projeto. Apenas três (25,00%) entrevistados relataram não ocupar tais funções.

Por fim, após a conclusão da pesquisa documental e realização das entrevistas, foi possível obter os dados necessários para a análise dos dados que permitiu o atingimento dos objetivos deste estudo. Os procedimentos de análise de dados constam da Seção 3.6, descrita a seguir.

3.6 Procedimentos de análise dos dados

Após a coleta dos dados, todas as informações foram compiladas, organizadas e consolidadas para fornecer suporte às análises dos dados. Os procedimentos de análise de dados desta pesquisa abrangeram a análise documental e a análise de conteúdo.

De acordo com Bardin (1977), a análise documental é uma etapa inicial na constituição de um serviço de documentação ou de um banco de dados, visando representar de forma concisa as informações contidas de maneira mais pertinente por meio de procedimentos transformadores. Essa análise documental foi particularmente importante para depurar o conjunto de documentos não sigilosos contendo informações sobre as etapas do processo de desenvolvimento de softwares adotadas pela Empresa BETA, selecionando-se informações sobre os softwares provedores do programa social Alfa. Essa análise foi particularmente útil para buscar o atingimento do primeiro objetivo específico deste estudo.

Adicionalmente, para as análises de dados coletados em entrevistas, adotou-se nesta pesquisa a Análise do Conteúdo. Segundo Bardin (1977), é um conjunto de

técnicas de análise das comunicações visando obter, por procedimentos sistemáticos e objetivos de descrição do conteúdo das mensagens, indicadores (quantitativos ou não) que permitam a inferência de conhecimentos relativos às condições de produção/recepção (variáveis inferidas) destas mensagens. A análise do conteúdo textual transcrito das entrevistas seguiu as seguintes fases: 1. Pré-análise; 2. exploração do material e 3. tratamento dos resultados, inferência e interpretação.

Embora a tecnologia ofereça opções que auxiliem processos de categorização¹⁴, foi decidido, para os propósitos desta pesquisa, realizar a categorização de forma manual. Essa abordagem foi adotada após a transcrição do conteúdo por meio da plataforma Microsoft Teams.

Para Identificar os principais fatores de riscos associados a cada etapa de desenvolvimento dos softwares provedores do Programa Social Brasileiro Alfa, foram considerados na análise de conteúdo os atributos da taxonomia de risco hierárquica do SEI citados no Quadro 10 deste estudo e apresentados por Carr et al. (1993), atendendo assim o segundo objetivo específico.

A identificação dos riscos a partir da análise das entrevistas de acordo com os atributos da taxonomia do SEI, foi realizada de acordo com uma descrição objetiva, sistemática do conteúdo extraído das comunicações e sua respectiva interpretação (Bardin, 1977).

Em adição, levando em conta as diversas fontes de evidência mencionadas por (Sampieri; Collado; Lúcio, 2013), procedeu-se à análise de documentos internos obtidos, com o propósito de verificar a correspondência dos atributos identificados nas entrevistas com as informações contidas nesses documentos, corroborando ou refutando os fatos narrados durante as entrevistas. Essa abordagem, empregando a triangulação de evidências obtidas das entrevistas e dos documentos, contribuiu para a confiabilidade dos resultados deste estudo.

Concluída a identificação dos riscos e o estabelecimento de sua correspondência aos atributos da Taxonomia SEI, foi realizada a classificação dos riscos identificados de acordo com elementos de classe da referida taxonomia. Com isso, buscou-se atingir o terceiro objetivo específico do estudo, efetuando-se, inclusive, considerações sobre os elementos de classe agrupados em cada classe da Taxonomia SEI.

Por fim, a partir dos atributos identificados em cada elemento de classe, foi realizada a comparação das ocorrências dos riscos identificados por classe com base

¹⁴ São exemplos de recursos tecnológicos para auxiliar a categorização os produtos 'Nvivo' e 'Iramuteq'.

na literatura, especialmente o estudo de Menezes et al. (2019). As três classes da Taxonomia SEI (Engenharia de Produto, Ambiente de Desenvolvimento e Restrições de Programa), representam áreas gerais que podem influenciar o projeto de software e que foram descritas na Figura 7 e Quadro 10 deste estudo, e são baseadas em (Carr et., 1993).

Em complemento, as etapas de análise de dados podem ser resumidamente observadas no Quadro 18, que apresenta a relação entre os quatro objetivos específicos da pesquisa e as técnicas de coleta e análise de dados propostas neste estudo.

Quadro 18 - Relação entre objetivos, coleta e análise dos dados

Objetivo Geral	Objetivo Específico	Coleta de Dados	Análise de Dados
Identificar os principais fatores de riscos associados às etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa.	Descrever as etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro ALFA	Documentos	Análise documental
	Verificar os principais fatores de riscos associados ao processo de desenvolvimento de software previamente descrito, considerando os atributos da taxonomia de risco hierárquica do SEI	Entrevistas Documentos	Análise de conteúdo
	Classificar os fatores de riscos verificados de acordo com elementos de classe da referida taxonomia	Entrevistas Documentos	Análise de conteúdo
	Comparar a ocorrência dos fatores de riscos verificados segundo as classes previamente referidas, com base na literatura	Entrevistas Documentos	Análise de conteúdo

Fonte: O autor, a partir de dados da pesquisa.

Por fim, para enriquecer a análise dos resultados foram empregados quadros e nuvem de palavras. A nuvem de palavras pode ter várias utilidades (Lunardi et., 2008) (Ramsden e Bate, 2008) e neste estudo foi usada para identificar palavras mais citadas nos textos transcritos das entrevistas. Para elaboração da nuvem de palavras foi utilizado o software Microsoft Power BI[®] para separar as organizar as palavras e em seguida o Word Cloud[®] para efetivamente gerar a nuvem de palavras.

4 DESCRIÇÃO, ANÁLISE E DISCUSSÃO DOS RESULTADOS

Neste capítulo são expostos os resultados alcançados por meio deste trabalho. As seções estão organizadas da seguinte maneira: a Seção 4.1 Etapas do processo de desenvolvimento (Objetivo 1); a Seção 4.2 Verificação dos riscos associados ao processo de desenvolvimento dos softwares, considerando os atributos da Taxonomia SEI (Objetivo 2); a Seção 4.3 Classificação dos riscos de acordo com elementos de classe correspondentes da Taxonomia SEI (Objetivo 3); e, por fim, a Seção 4.4 Comparação da ocorrência dos fatores riscos verificados segundo as classes (Objetivo 4).

4.1 Descrição das etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro ALFA (Associado objetivo específico 1)

O Processo de Desenvolvimento e Manutenção de Software da Empresa BETA, denominado PD-BETA, para os fins deste estudo, é constituído por 23 atividades. Essas atividades são estruturadas para direcionar, normatizar e registrar os procedimentos, visando, primordialmente, elevar a qualidade dos produtos e serviços fornecidos e, conseqüentemente, garantir a plena satisfação dos clientes.

Na filosofia subjacente ao PD-BETA, presume-se que cada software criado deve ser submetido a todas essas atividades estabelecidas. Esse percurso visa assegurar bons resultados, culminando na entrega de um software que atenda as expectativas do cliente em termos de qualidade, funcionalidade, desempenho e segurança.

A estrutura do PD-BETA é organizada em 23 atividades que se desdobram da maneira descrita no Apêndice H. De forma sintética o Quadro 19. apresenta as etapas do Processo de Desenvolvimento de Software associadas ao Programa Social identificando aquelas que foram efetivamente realizadas durante o desenvolvimento dos sistemas provedores do Programa Social Alfa. Este formato permite uma comparação direta entre as etapas teóricas do processo e as atividades efetivamente observadas na prática.

Quadro 19 – Atividades do processo de desenvolvimento de software

Descrição	Atividades identificadas na análise documental
Atividade 1 - Elaborar Visão da Solução	X
Atividade 2 - Avaliar Necessidade de Solução Arquitetural	X
Atividade 3 - Medir Tamanho Funcional	
Atividade 4 - Planejar Entregas	
Atividade 5 - Estruturar Iniciativa	
Atividade 6 - Consolidar Planejamento	
Atividade 7 - Planejar Release	
Atividade 8 - Planejar Sprint	
Atividade 9 - Especificar Item de Backlog	
Atividade 10 - Implementar Solução	X
Atividade 11 - Implementar Testes	
Atividade 12 - Preparar Testes	
Atividade 13 - Executar Testes Funcionais	X
Atividade 14 - Executar Testes Não-funcionais	
Atividade 15 - Avaliar Resultados dos Testes	X
Atividade 16 - Monitorar Progresso	
Atividade 17 - Gerenciar Fluxo de Trabalho	
Atividade 18 - Demonstrar Resultado da Sprint	
Atividade 19 - Realizar Retrospectiva	
Atividade 20 - Disponibilizar Release para Homologação	X
Atividade 21 - Medir Tamanho Funcional	
Atividade 22 - Implantar Release	X
Atividade 23 - Encerrar Execução	

Fonte: o autor, a partir de dados da pesquisa.

A partir das informações coletadas dos entrevistados e na análise da documentação não confidencial, foi verificado que sete dentre as vinte e três atividades previstas no PD-BETA, delineadas no Quadro 19, foram efetivamente

implementadas durante o desenvolvimento do software ligado ao Programa Social Brasileiro ALFA. As atividades específicas identificadas são: Elaborar Visão da Solução (Atividade 1), Avaliar Necessidade de Solução Arquitetural (Atividade 2), Implementar Solução (Atividade 10), Executar Testes Funcionais (Atividade 13), Avaliar Resultados dos Testes (Atividade 15), Disponibilizar Release para Homologação (Atividade 20), e Implantar Release (Atividade 22).

Esta identificação das atividades utilizadas reflete não apenas a abordagem metodológica seguida pelo time de desenvolvimento, mas também aponta para as áreas de maior relevância e atenção dentro do processo de desenvolvimento do software para o Programa Social Brasileiro ALFA.

É importante conceituar que o ciclo de vida do desenvolvimento de software - SDLC é um conceito fundamental na engenharia de software, utilizado principalmente para descrever um método estruturado para planejar, criar, codificar, testar e implementar uma especificação de requisitos do usuário. Esse processo envolve uma série de configurações de software e hardware e representa um procedimento detalhado para a criação de software de qualidade. O SDLC abrange diversas fases, que devem ser seguidas metodicamente, sendo essenciais para os engenheiros de software, incluindo análise, planejamento, codificação, design, teste e implementação. De maneira geral, cada processo de software é composto por pelo menos cinco fases principais: Requisitos, Arquitetura e Design, Implementação, Verificação e Lançamento e Manutenção(Abioye et al2020).

Ao comparar as atividades realizadas no desenvolvimento dos softwares provedores do programa social Alfa com as fases ideais do SDLC destacadas por Abioye et al. (2020), podemos analisar a cobertura e as lacunas existentes no processo.

Foram cobertas as seguintes fases:

- Elaborar visão da solução e avaliar necessidade de solução arquitetural: estas atividades alinham-se com as fases de Requisitos e Arquitetura e Design do SDLC.
- Implementar solução: Esta atividade cobre a fase de Implementação do SDLC.
- Executar testes funcionais e avaliar resultados dos testes: Essas atividades estão relacionadas à fase de Verificação do SDLC.
- Disponibilizar release para homologação e implantar release: Estas atividades correspondem à fase de Lançamento e Manutenção.

Contudo as etapas de medir tamanho funcional, planejar entregas, estruturar Iniciativa não foram cobertas. É importante destacar que a omissão dessas atividades pode comprometer etapas necessárias para a estimativa de esforço, planejamento detalhado e estruturação do projeto.

A falta de atividades dedicadas ao monitoramento e ajuste do escopo e ao gerenciamento de mudanças também pode resultar em dificuldades para se adaptar a requisitos alterados, afetando a relevância e a eficácia do software.

Autores corroboram com esse entendimento, por exemplo, Abdul Karim et al. (2016) destacam a prática de desenvolvimento seguro de software como uma integração essencial ao SDLC, sugerindo que a inclusão de atividades específicas de segurança poderia mitigar riscos de vulnerabilidades, um aspecto não explicitamente coberto pelas atividades identificadas.

A gestão de riscos por sua vez é enfatizada por Alharbi e Qureshi (2014), que argumentam ser crucial a sua inclusão no início do SDLC para identificar e mitigar riscos antes que eles se materializem. Este enfoque é complementar à abordagem do PD-BETA, indicando a necessidade de uma atividade dedicada à gestão de riscos para antecipar e planejar ações mitigadoras, contribuindo assim para a estabilidade e previsibilidade do projeto de software.

O papel da comunicação efetiva no sucesso do desenvolvimento de software é ressaltado por Andrade et al. (2019), que sugerem métodos para melhorar a comunicação entre as partes interessadas. Isso inclui a implementação de atividades que promovam o entendimento mútuo e o alinhamento entre os desenvolvedores, gestores e usuários finais do software, abordando uma das lacunas identificadas relacionada ao ajuste de escopo e gerenciamento de mudanças.

A importância da adaptabilidade e da resposta rápida às mudanças no desenvolvimento de software, princípios fundamentais das metodologias ágeis, é discutida por Beck et al. (2001). A integração de práticas ágeis poderia oferecer maior flexibilidade e capacidade de resposta, permitindo ajustes rápidos ao escopo e às necessidades dos usuários, melhorando assim a satisfação do cliente e a relevância do software desenvolvido.

Já Sbrocco e Macedo (2012) enfatizam a necessidade de uma estruturação detalhada do projeto, incluindo planejamento de entregas e estruturação da iniciativa, destacando a importância dessas atividades para a estimativa de esforço e planejamento detalhado. A inclusão dessas atividades poderia fornecer uma base mais sólida para o gerenciamento do projeto, garantindo que todos os aspectos sejam considerados e planejados desde o início.

A certificação e a conformidade com padrões internacionais, como ABNT NBR ISO 31000 podem melhorar a gestão de riscos e a qualidade do software (ABNT, 2018). A adoção desses padrões reforçaria a abordagem de gestão de riscos e garantiria que as práticas adotadas estejam alinhadas com as melhores práticas globais.

A perspectiva de Pressman e Maxim (2016) sobre a necessidade de adaptação contínua às mudanças tecnológicas e de mercado ressalta a importância de um desenvolvimento dinâmico, que possa se adaptar às novas tendências e exigências, garantindo que o software desenvolvido permaneça atualizado e relevante.

A análise de pontos de função, como medida de tamanho funcional, sugerida por diversos autores, como Jones (1994), proporciona uma base quantitativa para estimar o esforço de desenvolvimento. A inclusão de uma atividade dedicada a essa análise poderia melhorar a precisão das estimativas de esforço e custo, contribuindo para um planejamento mais eficaz.

A implementação de testes automatizados e a integração contínua, conforme discutido por Bass et al. (2015), são práticas recomendadas para melhorar a eficiência e a qualidade do desenvolvimento de software. A integração dessas práticas poderia reduzir erros e defeitos, melhorando a confiança na qualidade do software entregue.

Por fim, a consideração de práticas de desenvolvimento orientadas à usabilidade e à experiência do usuário, enfatizada por Ebert (2017), sugere a necessidade de uma abordagem ampla que considere não apenas os aspectos técnicos, mas a incorporação dessas práticas ao processo de desenvolvimento, resultando em um software mais alinhado às necessidades e expectativas dos usuários.

Portanto, ao integrar esses elementos, muitos já previstos no PD-BETA, o processo de desenvolvimento do software do Programa Social Brasileiro ALFA não apenas alinhar-se-ia mais estreitamente às melhores práticas da engenharia de software, mas também poderia aumentar suas chances de sucesso e aceitação pelos usuários finais.

4.2 Verificação dos principais fatores de riscos associados ao processo de desenvolvimento dos softwares, considerando os atributos da Taxonomia SEI (Associado objetivo específico 2)

Conforme já destacado na fundamentação teórica, a taxonomia do SEI é estruturada a partir de 03 classes, 13 elementos e 64 atributos. A cada entrevista realizada foram identificados, a partir da opinião do entrevistado transcrita, os atributos

que estiveram presentes durante o desenvolvimento dos softwares provedores do programa social brasileiro Alfa.

Foram realizadas ao total 12 entrevistas, conforme apresentado no Quadro 20 ocasiões na qual foi possível constatar a saturação teórica (Falqueto; Hoffmann; Farias, 2018), por conta da ausência de novas evidências oriundas das entrevistas, seguido o processo de análise de conteúdo definido por Bardin (2011) previamente citado na metodologia deste estudo.

No Quadro 20 exibida uma representação que segue a ordenação das entrevistas realizadas, demonstrando a evolução do processo de pesquisa e, mais especificamente, identificando o ponto em que a saturação teórica foi alcançado na entrevista E9, por inexistir elementos novos a serem inseridos em todas as classes. Foi adotada a convenção de atribuir "1", quando uma entrevista resultou na identificação de novos atributos para a classe correspondente, e "0" foi atribuído quando a entrevista não acrescentou novos atributos à análise (Falqueto; Hoffmann; Farias, 2018):

Quadro 20 - Ponto de saturação teórica

Classe	ENTREVISTADOS											
	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
Engenharia de produto	1	1	1	0	1	0	0	1	0	0	0	0
Ambiente de Desenvolvimento	1	1	1	0	0	1	0	1	0	0	0	0
Restrições do Programa	1	1	0	0	0	0	0	0	0	0	0	0

Fonte: O autor, a partir de dados da pesquisa, e baseado em Falqueto, Hoffman e Farias (2018).

Legenda:

Entrevistas (1) – Existência de nova informação (0) - Ausência de nova informação (0).

 Ponto de saturação

 Confirmação

É uma prática comum sugerir a condução de pelo menos duas entrevistas adicionais após a identificação do ponto de saturação, conforme recomendado por Falqueto, Hoffmann e Farias (2018) e Thiry-Cherques (2009). Neste estudo, foram realizadas três entrevistas adicionais para assegurar a clareza da saturação. Adicionalmente, a realização de um total de 12 entrevistas para alcançar a saturação teórica encontra respaldo na literatura, como evidenciado por Boddy (2016).

Após a conclusão das entrevistas, a investigação documental reforçou a importância dos atributos identificados, conferindo maior robustez aos achados da pesquisa, um processo que está em harmonia com o pensamento de Wieggers (2013) que enfatiza a importância da corroboração documental.

Durante a condução das entrevistas, foram identificados 39 atributos distintos dentre os 64 possíveis, que estão distribuídos nas 03 classes e nos 13 elementos, conforme demonstrado no Quadro 21.

O Quadro 21 apresenta a classificação dos atributos, incluindo as classes às quais pertencem, os elementos de cada classe, e um código sequencial de 1 a 63 para identificação de cada atributo. Em adição, o Quadro 21 mostra os entrevistados, de E1 a E12. O Quadro mostra, inclusive a atribuição "1" quando uma entrevista resultou na identificação do atributo. Caso contrário, mostra "0" correspondente a ausência daquele atributo na narrativa do entrevistado.

Quadro 21 - Atributos identificados durante a análise das entrevistas

Classe (03)	Elemento (13)	COD	Atributo (64)												
			E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	
Engenharia de Produto	Requisitos	1	a. Estabilidade	1	0	1	1	0	1	0	1	0	0	0	1
		2	b. Completude	1	0	1	1	0	0	1	1	0	1	0	1
		3	c. Clareza	1	0	0	1	0	0	0	0	0	0	0	0
		4	d. Validade	1	0	0	0	0	0	0	0	0	0	0	0
		5	e. Viabilidade	0	0	0	0	0	0	0	0	0	0	0	0
		6	f. Precedente	0	0	0	0	0	0	0	1	0	0	0	0
		7	g. Escala	1	0	0	0	0	0	0	1	0	0	0	0
	Design	8	a. Funcionalidade	0	0	0	0	0	0	0	0	0	0	0	0
		9	b. Dificuldade	0	0	0	0	0	0	0	0	0	0	0	0
		10	c. Interfaces	1	0	0	0	0	0	1	0	0	0	0	0
		11	d. Desempenho	0	1	0	0	0	1	0	0	0	0	0	0
		12	e. Testabilidade	1	0	0	1	1	0	1	0	1	0	0	0
		13	f. Restrições de Hardware	0	0	0	0	0	0	0	0	0	0	0	0
		14	g. Software Não-Desenvolvido	1	0	1	1	0	0	0	1	0	0	0	0
	Código e Teste Unitário	15	a) Viabilidade	0	0	0	0	0	0	0	0	0	0	0	0
		16	b) Teste de Unidade	0	0	0	0	1	0	0	0	0	0	0	0
		17	c) Codificação/Implementação	0	0	0	0	0	0	0	0	0	0	0	0
	Integração e Teste	18	a) Ambiente	1	0	0	0	0	0	0	0	0	0	0	0
		19	b) Produto	0	0	0	0	0	0	0	0	0	0	0	0
		20	c) Sistema	0	0	1	0	0	0	0	0	0	0	0	0
	Especialidades de Engenharia	21	a) Manutenibilidade	0	0	0	0	0	0	0	0	0	0	0	0
		22	b) Confiabilidade	0	1	0	0	0	0	0	0	0	0	0	0
		23	c) Segurança	0	1	0	1	0	0	0	0	0	0	0	0
		24	d) Segurança (Security)	0	0	0	0	0	0	0	0	0	0	0	0
		25	e) Fatores Humanos	0	0	0	0	0	0	0	1	0	0	0	1

	Contrato	55	a) Tipo de Contrato	1	0	0	0	0	0	1	0	0	0	1	0	
		56	b) Restrições	1	0	0	0	0	1	0	0	0	0	0	0	0
		57	c) Dependências	0	0	0	0	0	0	0	0	0	0	0	0	0
	Interfaces do Programa	58	a) Cliente	0	1	1	0	0	0	0	1	0	0	0	0	0
		59	b) Contratados Associados	0	0	1	1	0	0	0	0	0	0	0	0	0
		60	c) Subcontratados	0	0	0	0	0	0	0	0	0	0	0	0	0
		61	d) Contratante Principal	0	1	0	0	0	0	0	0	0	0	0	0	0
		62	e) Gestão Corporativa	0	0	0	0	0	0	0	0	0	0	0	0	0
		63	f) Fornecedores	0	1	0	0	0	0	0	0	0	0	0	0	0
		64	g) Política	0	0	0	0	0	0	0	0	1	0	0	0	0

Fonte: Elaborado pelo autor a partir das entrevistas. Legenda: '1' indica a presença do atributo, '0' indica a ausência do atributo.

Ademais, destaca-se que o detalhamento de cada atributo identificado no Quadro 21 é apresentado e discutido a seguir.

O primeiro atributo **1a.**, citado no Quadro 21, intitula-se 'Estabilidade' e refere-se à frequência de mudanças nos requisitos e ao impacto destas no produto, abrangendo a dificuldade em controlar essas alterações rápidas e que foi relatado por 50,00% dos entrevistados. Na sequência, exibe-se evidência do referido atributo.

(...) Aí teve uma dessas coisas, é aí teve aquela parte, né? (...) Então foram. Foram assim a gente foi mudando com um avião andando, né? Foi decolando, a gente foi ajeitando.

Inclusive porque, como os requisitos foram mudando com o tempo a as regras iniciais usadas no processamento e o volume de dados inicial, ele foi no final ele foi crescendo exponencialmente, né? (...) porque tem que cortar agora, corta agora, agrega +1 + 1 Parcela quando você achou que o benefício ia acabar, não bota mais uma parcela que agora mudou a regra faz de novo. Então esse processo que o benefício, que era algo de poucos meses, durou, foram 2 anos, se não me engano, né? (...)

De acordo com evidências de relatório de controladoria governamental, o projeto inicialmente empregou um conjunto reduzido de bases de dados, visando à qualificação dos dados e à identificação precisa dos beneficiários. No entanto, modificações substanciais foram registradas, conforme indicado pela União em 2020. Em junho, por exemplo, as regras pertinentes aos casos com indícios de óbito foram revisadas, visando otimizar o processo de cruzamento de dados e mitigar o risco de falsos-positivos. Tais mudanças, conforme destacado pelos entrevistados, validam a dinâmica de alteração frequente dos requisitos.

Paralelamente, a necessidade de inclusão e tratamento de dados adicionais foi explicitamente mencionada. Em resposta a União, o demandante do software, através de Ofício, esclareceu que, conforme Parecer Jurídico, a concessão do benefício deveria ser restrita. Foram utilizadas duas bases de dados, fornecidas por órgão governamental, contendo informações de indivíduos encarcerados. que permitiram verificar beneficiários que deveriam ser contemplados no programa social Alfa.

Adicionalmente, considerou-se a alteração legislativa referente ao auxílio reclusão, introduzida pela Lei nº 13.846, de 18 de junho de 2019. A mudança estipula que o benefício seja concedido exclusivamente a presos em regime fechado, o que, se não observado, pode resultar em exclusões errôneas, aumento de litígios judiciais

e retrabalho administrativo. O tribunal de contas, ao analisar o caso, reiterou a dificuldade de se obter informações precisas a partir das bases de dados, dada a falta de unificação delas. Esses elementos fortalecem as observações feitas durante as entrevistas, ressaltando a complexidade e os desafios enfrentados no processo de cruzamento e análise de dados governamentais.

Por fim, o Termo de abertura estimou uma contagem de Pontos de Função de 202,00 para o desenvolvimento completo do sistema. No entanto, o levantamento detalhado realizado e documentado no relatório não sigiloso de Contagem de Pontos de Função, sob o #ID da contagem 9857, revelou que somente para a funcionalidade de 'Consulta do cliente aos requerimentos/análise do benefício', houve a utilização de 28.000 pontos de função. Esse volume inesperado refletiu a extensa quantidade de alterações que foram necessárias entre a fase de especificação e a implementação efetiva da funcionalidade. Tal discrepância evidencia não apenas a complexidade do sistema, mas também o dinamismo e os desafios enfrentados na adaptação às necessidades em evolução durante o ciclo de vida do desenvolvimento.

Esse cenário espelha os desafios abordados por Menezes et al. (2019), enfatizando a necessidade de estratégias adaptativas e flexíveis em gerenciamento de riscos para acomodar a fluidez das demandas e as mudanças rápidas nos requisitos, que são características fundamentais em projetos de desenvolvimento de software em ambientes voláteis.

O segundo atributo identificado **2b. 'Completeness'** que se refere a requisitos faltantes ou mal especificados que podem levar a implementações baseadas em suposições desalinhadas com as expectativas do cliente, o atributo foi relatado por 58,33% dos entrevistados.

(...) O primeiro pagamento, ocorreu 7 dias depois da demanda Colocada, né? A demanda não é a especificação, tá! Então, assim, quando você olha, é para questão de software, não é? Então, assim, todas aquelas fases tradicionais, né? De uma especificação bem-feita, né? É de uma coisa que tem um dever, que é uma coisa que está bem consolidada definida. Isso não aconteceu para o ALFA, tá? Então, o ALFA. Tinha essa necessidade inicial, definição de público-alvo.

(...)Tinha também morava no exterior, teve várias questões que foram sendo trazidas à tona, né? Como restrição ao direito de receber, né? E que isso ia sendo verificado mês a mês, então esse controle, né? (...)

O atributo **2b.**, pode ser constatado por meio de dois documentos. O Decreto de Abril de 2020 que regulamenta também a Lei de 2020, apresenta requisitos de sistema. Cada seção ou artigo do decreto descreve diferentes aspectos relacionados

ao benefício e os critérios que os trabalhadores devem atender para serem elegíveis a receber o benefício. Estes requisitos abordam questões como idade, inscrição em sistemas do governo, ausência de renda formal, contribuição previdenciária, entre outros. Cada um desses critérios é um requisito de sistema que precisa ser cumprido para que os trabalhadores possam receber o benefício.

Adicionalmente, o documento não sigiloso define como escopo o fornecimento de informações ao cliente para auxiliar a concessão do direito ao benefício para os cidadãos. Dentre as funcionalidades solicitadas para esta versão do projeto, foram citadas: módulo de inclusão de decisão judicial e integração sistemas jurídicos, módulo de controle de lista de pagamento e ordem bancária, apresentar informações de pagamento das parcelas nas consultas do cidadão e do cliente, funcionalidade de contestação da análise (via parceiros, Alfa e manual), carga de novos lotes/remessas para apresentação dos resultados das análises, processamento de múltiplas competências de outros benefícios.

Adicionalmente foi analisada a documentação relacionada a demanda - Plataforma digital ALFA, onde foi possível verificar as diversas versões, módulos e evoluções na etapa de desenvolvimento do sistema.

Esse confronto da análise documental com a percepção dos entrevistados demonstra que os requisitos não estavam completos e foram definidos em tempo de projeto, acompanhando as mudanças e necessidades emergentes.

O cenário descrito, com a incorporação progressiva de novas necessidades e critérios regulamentares, exemplifica um desenvolvimento dinâmico e adaptativo, características que ressoam com a natureza fluida dos projetos de software e a necessidade de gerenciamento de riscos efetivo, conforme discutido por Menezes et al. (2019). Este cenário destaca a importância de uma abordagem ágil e responsiva no gerenciamento de projetos de software, especialmente em contextos de rápida evolução e incerteza.

O atributo **3c. 'Clareza'**, mencionado por 16,67% dos entrevistados, enfoca a precisão e a nitidez na formulação dos requisitos, uma vez que ambiguidades ou imprecisões podem levar a retrabalhos significativos para alinhar o produto final às expectativas do cliente. Esse aspecto foi não apenas evidenciado nas entrevistas, mas também corroborado por achados na análise documental. A seguir, apresenta-se um relato extraído de uma entrevista que ilustra a relevância prática desse atributo no contexto do projeto.

(...) Muito dinâmico, né? Porque efetivamente é o cliente, não tinha é noção, né? Ou ele não tinha toda não sei se é noção, mas noção ele

tinha, mas ele não tinha toda a visualização do que ele precisava ter de regra, né?

(...) Muito dinâmico, né? Porque efetivamente é o cliente, não tinha é noção, né? Ou ele não tinha toda não sei se é noção, mas noção ele tinha, mas ele não tinha toda a visualização do que ele precisava ter de regra, né? Algumas coisas efetivamente foram sendo descobertas ao longo do tempo e aí, efetivamente, isso foi uma característica, né? Então, assim, as bases mudavam, os as regras mudavam, né? E também a captação dos dados, né?

De acordo com o corpus documental, o escopo principal e os requisitos do software foram definidos por dois documentos: o Termo de Abertura e a legislação aplicável. Ambos estabelecem um escopo conciso e limitado para a complexidade da solução proposta, uma delimitação que parece ser uma resposta direta à incerteza gerada pela duração indeterminada dos impactos da pandemia. No entanto, essa limitação de escopo foi acentuada pela lacuna no entendimento do cliente em relação à definição precisa dos requisitos, o que se mostrou um fator crítico ao longo do projeto.

A exemplificar, o Termo de Abertura esboça, de forma restrita, apenas cinco componentes essenciais do software: o Serviço de Requerimento, que, ironicamente, foi desenvolvido por um banco parceiro; o Serviço de Concessão/Elegibilidade, focado na extração, atualização e batimento de dados, e que constitui o cerne das regras de negócio; o Serviço de Manutenção de Benefício, que engloba o Controle de Pagamento, Bloqueio e Desbloqueio de benefícios – este último, posteriormente transformado em Serviço de Controle de Pagamento; e, por fim, o Serviço de Elaboração e Análise de Relatórios.

Tais limitações apresentadas inicialmente refletiram em uma série substancial de modificações durante a execução do projeto. Isso é particularmente evidente na documentação 'Regras e Atualizações', que registrou a alteração e inclusão de 190 requisitos ao longo do desenvolvimento. Essa dinâmica realça não só a volatilidade do ambiente de projeto em resposta às incertezas, mas também a essencialidade de uma gestão de requisitos ágil e adaptativa para o sucesso do desenvolvimento de softwares.

A narrativa dos entrevistados ilustra um cenário típico de muitos projetos de desenvolvimento, onde os requisitos evoluem e se refinam à medida que a compreensão do problema e do domínio do projeto se aprofunda. A literatura em gerenciamento de projetos também destaca essa dinâmica, notadamente em métodos

ágeis, onde a evolução dos requisitos é uma expectativa e parte integrante do processo de desenvolvimento, mas que por vezes não estão completamente definidos no momento inicial do projeto por falta de clareza da complexidade do projeto (Beck et al., 2001; Schwaber e Sutherland, 2020). Portanto, o relato dos entrevistados corrobora com a literatura, enfatizando a necessidade de flexibilidade, comunicação efetiva e processos iterativos para o gerenciamento eficaz da clareza dos requisitos.

O atributo **4d. 'Validade'** mencionado por 8,33% dos entrevistados, concentra-se na adequação e precisão com que os requisitos capturados refletem as verdadeiras necessidades e expectativas do cliente. Este atributo é importante para garantir que o produto final esteja alinhado com os objetivos do projeto, evitando mal-entendidos e retrabalhos. A expectativa pelo alcance desse atributo é evidenciada pelo seguinte relato do entrevistado: *"(...) Era aquela incerteza. Estamos fazendo certo, não é? É aí a arquitetura ali, de uma forma rápida (...)"*

Como pôde ser observado na regulamentação em vigor bem como no termo de abertura do projeto, a definição completa dos requisitos era incerta devido às limitações das regras estabelecidas inicialmente. Ademais, o projeto experimentou várias alterações durante seu desenvolvimento, com inclusões e exclusões de regras de negócio críticas, refletindo a dinâmica e a adaptabilidade necessárias em resposta às circunstâncias emergentes. Como exemplo ilustrativo, a demanda DM.085734 documenta o processo evolutivo do sistema, que passou por atualizações significativas no período de março a dezembro de 2020, destacando mudanças de requisitos durante o desenvolvimento do projeto.

A validade dos requisitos é fundamental para a entrega bem-sucedida de qualquer projeto de software. Como afirma Sommerville (2019), requisitos válidos são aqueles que refletem as necessidades dos stakeholders e que podem ser claramente compreendidos por todas as partes envolvidas.

O atributo **6f. 'Precedente'** mencionado por 8,33% dos entrevistados aborda o desafio de lidar com requisitos ou capacidades tecnológicas que são novidades para a equipe de desenvolvimento. A falta de experiência anterior com tais requisitos eleva o risco e exige uma abordagem cuidadosa para garantir uma implementação bem-sucedida.

(...) A gente não tinha, como é que eu vou dizer? A gente não tinha parâmetro para poder definir adequadamente o ambiente, então eu acho que o maior risco quando a gente começou a discutir o ambiente era, vamos nos basear no quê? Não é o que é parecido que a gente

tem. Qual a quantidade de espaço armazenado, qual a quantidade de processadores, qual a máquina adequada que vai suportar isso, qual a tecnologia?

(...) Sabes que a falta de estudo de capacidade e tempo hábil para isso não é não, que é eu estar não quisesse, mas falta de tempo hábil para o para o estudo de capacidade, até a falta, né? De um estudo de caso, também que pudesse apoiar, né? Nisso, né? Ineditismo disso, né? Acho que cada equipe foi herói, cada uma no seu pedacinho, né? Pra tentar alcançar algumas vezes não se conseguia, né, e aí se dava-se a volta, se discutia-se muitas madrugadas se passaram e várias equipes, acho que cada um no seu momento fez.

Conforme destacado por McConnell (1996), o precedente, ou a falta dele, pode ser um fator determinante para o sucesso ou fracasso de um projeto. Corroborando com o relato do entrevistado, a designação de uma Emergência de Saúde Pública de Importância Internacional – ESPII pelo Regulamento Sanitário Internacional – RSI ressalta a gravidade de eventos extraordinários que representam riscos à saúde pública global, necessitando de uma coordenação e resposta internacionais ágeis para prevenir a disseminação transfronteiriça de doenças, como indicado por Marins et al. (2021). No caso deste estudo, o desenvolvimento do software ocorreu no período em que houve a decretação da pandemia decorrente da COVID-19.

Historicamente, a medida de decretação de emergência de saúde pública foi invocada em raros momentos, evidenciando a sua importância para casos de extrema urgência: a pandemia de H1N1 em 25 de abril de 2009, a propagação do poliovírus em 5 de maio de 2014, o surto de Ebola na África Ocidental em 8 de agosto de 2014, o surto do vírus Zika em 1 de fevereiro de 2016, e o surto de Ebola na República Democrática do Congo em 18 de maio de 2018. Cada um desses eventos trouxe desafios significativos à saúde pública internacional, mas nenhum se comparou ao impacto da COVID-19, que transcendeu as precedentes emergências pela sua vasta escala e consequências globais, afetando a saúde, economia e estruturas sociais no mundo (Marins et).

Mencionado por 16,67% o atributo **7g. 'Escala'**, destaca os desafios singulares enfrentados em sistemas de grande complexidade, abrangendo aspectos tanto técnicos quanto gerenciais. Tais desafios incluem a integração de múltiplos componentes, o gerenciamento de dependências e a coordenação eficiente de tarefas e equipes. Este atributo enfatiza a necessidade de estratégias bem definidas para navegar na complexidade e garantir a eficácia do projeto.

(...) Se eu diria que é em se falando de um projeto desafiador como aquele em termos de volume e concorrência e acessos, né? A gente está falando só, ou seja, em escala.

(...) É uma ideia que vai amadurecendo. Então, como que a gente vai trazer isso para o nosso mundo? (...) Não é assim com muita responsabilidade ali, não é? Quantas pessoas dependiam daquele trabalho?

Analisando os documentos do projeto, constata-se que o desenvolvimento do software enfrentou desafios relacionados ao desconhecimento dos requisitos por parte do cliente. Isso fica claro na delimitação de escopo tanto no Decreto quanto no Termo de Abertura do projeto. A situação foi agravada pelo contexto da pandemia, que impôs desafios adicionais.

Conforme evidenciado pelo corpus de documentos não sigilosos, a mudança para o regime de teletrabalho para todos os funcionários aumentou os desafios gerenciais. O projeto exigiu o desenvolvimento de um software complexo, que demandava o cruzamento de bases de dados, uma complexidade técnica destacada no documento explicativos do projeto. Além disso, havia a necessidade de apoiar os funcionários na transição para o trabalho remoto, ampliando a escala de gestão do projeto.

Esses desafios são citados na literatura de gerenciamento de projetos, que enfatiza a importância de uma abordagem sistemática e estratégica na coordenação de componentes multifacetados e na gestão de recursos e equipes PMI (2017). Especialmente em projetos de grande escala, onde a interdependência entre os componentes é significativa, a necessidade de uma comunicação eficiente, gerenciamento de riscos e métodos ágeis de desenvolvimento torna-se ainda mais premente (Larson e Gray (2011).

O atributo **10c. 'Interfaces'**, mencionado por 16,67% dos entrevistados concentra-se na gestão e definição de todas as interfaces de hardware e software que compõem o escopo do programa de desenvolvimento. Trata-se de um aspecto necessário para assegurar a integração harmoniosa e o funcionamento eficiente do sistema, envolvendo a aplicação de técnicas para gerenciar essas interfaces críticas.

(...) A primeira coisa que me veio à cabeça foi isso (...). Assim, a nossa grande, o nosso maior desafio, foi integrar bases, ou seja, diferentes pontes. E aí você fala de base que pode ser federal, base que pode ser estadual, base que pode ser, sei lá, municipal é fazer esse cruzamento de dados que não estão ali normalizados de certa forma, né? Em algum

momento vou ter um CPF em outro momento vou ter só o nome da pessoa e às vezes aquilo até que com o nível de confiabilidade.

(...) É, mas eu acho que para fazer todo o desenvolvimento, o processo como um todo, nossos dos maiores desafios aí é. Eu vejo muito nessa linha da integração que a gente basicamente é o no geral, é fazer um cruzamento de uma grande massa de dados, né? De Fontes distintas e fazer aquele processamento todo e sair com um resultado ali de pessoas que são elegíveis ou não né? E ali eu acho que estava o maior ponto de risco.

Na análise dos documentos disponibilizados, incluindo o "Brazilian ALFA Benefit", onde são apresentadas as bases de dados empregadas, identificou-se a necessidade de estabelecer uma interface para troca de arquivos com a instituição financeira. Contudo, conforme descrito pelos entrevistados, essa metodologia de troca de arquivos revelou-se ultrapassada diante das tecnologias contemporâneas disponíveis. A utilização de *Application Programming Interfaces* - APIs, por exemplo, ofereceria uma alternativa mais moderna e eficiente. APIs permitem a comunicação direta e em tempo real entre diferentes sistemas, eliminando a necessidade de geração, envio e processamento manual de arquivos. Esta abordagem não só acelera a troca de informações, como também reduz erros, aumenta a segurança dos dados e melhora a integração entre sistemas, tornando o modelo de troca de arquivos uma prática obsoleta em comparação.

A complexidade descrita pelos entrevistados em relação ao atributo 10c reflete a natureza da gestão de interfaces em projetos de TI, especialmente quando envolvem a integração de bases de dados diversas e heterogêneas. Esta complexidade é corroborada por Sommerville (2019), que destaca a importância crítica da gestão de interfaces para a eficácia e a eficiência de sistemas de software.

A integração de diferentes bases de dados, cada uma com seus formatos, padrões e níveis de confiabilidade, exige uma abordagem cuidadosa e métodos específicos para garantir a consistência e a precisão dos dados. A necessidade de normalizar e cruzar informações de múltiplas fontes, cada uma com seu próprio conjunto de desafios, ressalta a importância de um gerenciamento de interfaces robusto e eficiente, conforme discutido por Pressman (2014), enfatizando que a qualidade do processo de integração é fundamental para a confiabilidade e a integridade do sistema como um todo.

Mencionado por 16,67% dos entrevistados o atributo **11d 'desempenho'** aborda os requisitos críticos relacionados ao tempo de resposta e taxa de

transferência do sistema, ressaltando a importância de análises de desempenho e modelagem durante o ciclo de desenvolvimento.

(...) a gente precisava desenvolver algo que fosse possível de ser entregue num curto período de tempo e que também aguentasse essa condição de corrida que gerava concorrência.

(...) E um outro ponto é que, por exemplo, a gente pensava muito sobre é elasticidade né? É, a gente tem os nossos padrões de desenvolvimento e a gente tem os recursos que a gente já tem na casa disponibilizado como padrão e hoje a gente não tem recurso imediatamente disponível que me provê elasticidade automatizada. Então a gente teve que suprir isso com uma... o pessoal diz que solução recebe mais funcionalidades, funções e recursos do que realmente precisa, sendo desenvolvida de maneira mais complexa do que o necessário, né? Então assim, a gente provisionou muito mais do que o necessário.

Baseando-se nas informações coletadas do acervo documental, especificamente do Projeto ALFA - ambientes iniciais, e alinhando-se às observações dos entrevistados sobre o excesso de solução, observou-se a implementação estratégica de ambientes especializados, incluindo *proxies*, servidores *web*, *proxies* de saída, *backends*, caches, processamentos *batch*, bases de dados e *gateways* de API. Estes ambientes foram organizados em desenvolvimento, homologação e produção, distribuídos em redes externas, intermediárias e internas. Esta abordagem multifacetada não apenas evidencia a busca pelo desempenho otimizado diante da exigência de níveis de concorrência e demanda, mas também destaca a necessidade de prover elasticidade e resiliência ao sistema. A decisão por uma configuração super dimensionada reflete uma medida preventiva diante da incerteza quanto à capacidade necessária, assegurando que o sistema pudesse suportar picos de carga sem comprometer a experiência do usuário.

A ênfase dada pelos entrevistados ao citado atributo 11d reflete a relevância da eficiência e da escalabilidade em sistemas. Esta preocupação é encontrada na literatura, como apontado por Patterson e Hennessy (2013), que discutem a importância de projetar sistemas com capacidades de desempenho que possam atender a demandas intensas e variáveis. A necessidade de desenvolver uma solução em um curto período de tempo, capaz de suportar uma alta concorrência e garantir elasticidade, alinha-se aos princípios de desenho para alta disponibilidade e tolerância a falhas. O conceito de "overengineering" mencionado por um dos entrevistados,

embora possa implicar um excesso de recursos, é uma estratégia defensiva comum para garantir que o sistema possa lidar com picos de carga inesperados, ecoando a importância de um planejamento de capacidade proativo, conforme destacado por Menasce e Almeida (2002), para prevenir gargalos de desempenho e garantir uma experiência de usuário consistente e confiável.

O quarto atributo **12e 'testabilidade'**, que engloba a capacidade de se testar o design de forma eficaz, o desenvolvimento de funcionalidades pensadas para facilitar essa testagem e a integração de especialistas responsáveis por planejar e executar os testes do produto no processo de design. Esse atributo foi mencionado por 41,67% dos entrevistados, que destacaram a ausência ou as dificuldades encontradas durante a fase de testes. Tal situação decorre, em parte, pelo ineditismo da solução, que impacta na orientação do processo. Ademais, os desafios impostos pelos atributos discutidos anteriormente acentuaram essas dificuldades, resultando em uma abordagem onde a rapidez na entrega do produto foi priorizada em detrimento das etapas de teste e qualidade, aceitando-se o risco de problemas potenciais no desempenho em estágios futuros.

Seria incorreto afirmar que os testes foram completamente negligenciados; de fato, evidências como o 'Relatório de teste de carga e estresse' indicam que testes foram conduzidos até certo ponto. No entanto, é possível que esses testes tenham sido limitados em escopo ou não tenham explorado a solução em toda a sua complexidade. Esse entendimento é reforçado pelos relatos obtidos durante as entrevistas, em que transpareceu que, embora tenham sido realizados esforços para testar o sistema, talvez não tenham sido suficientes para garantir uma avaliação abrangente.

(...) Essas regras, elas mudavam, muito, então, assim não era aquela questão, né constante, com o dever bem definido, específica, programa testa implementa. Isso não existiu.

(...) É natural e acontece muito isso, né? É, quem perde é. São as etapas de teste.

(...) Quem perde são as etapas de qualidade, né? Em detrimento de você entregar um produto mais rápido, né?

E eu creio que não foi possível fazer todos os testes, por exemplo, de desempenho, né? E isso poderia ter dado algum problema mais na frente, né? Era um risco. Só que aí é devido ao tempo todo resolveu-se correr esse risco, né? Para ter o produto logo no ar.

A ênfase na celeridade de entrega em detrimento da realização de testes abrangentes em projetos de software, conforme destacado no contexto do projeto ALFA, ressoa com as considerações de Pressman e Maxim (2021) sobre o delicado equilíbrio entre rapidez e qualidade. Estes autores realçam que, apesar da importância crítica dos testes para assegurar a qualidade e a estabilidade de um produto de software, a pressão por prazos apertados pode levar a decisões onde a qualidade é comprometida para cumprir cronogramas exigentes. Ainda, segundo Wiradarma e Sasmita (2019), a integração eficaz de práticas de testabilidade no ciclo de vida do desenvolvimento de software é crucial para mitigar riscos e garantir que o produto final atenda aos requisitos de desempenho e segurança, enfatizando a necessidade de um planejamento cuidadoso e uma execução rigorosa dos testes, mesmo sob condições de prazos restritos.

O atributo **14g. 'Software Não Desenvolvido'** refere-se a soluções de software que não são originariamente concebidas para atender a requisitos específicos de um determinado sistema. Em vez disso, são selecionadas por serem consideradas a "melhor adequação", adaptando-se de forma mais genérica às necessidades do projeto. Este atributo engloba softwares que foram desenvolvidos para outros projetos, mas que, de alguma forma, podem ser reaproveitados ou integrados para contribuir com o desenvolvimento atual. A utilização desses softwares pré-existentes exige uma avaliação cuidadosa para assegurar que eles possam ser efetivamente adaptados e integrados ao novo contexto, cumprindo os objetivos e requisitos do projeto em questão.

Este atributo foi mencionado por 33,33% dos entrevistados, que enfatizaram os desafios inerentes à integração de sistemas heterogêneos. Os entrevistados apontaram as dificuldades encontradas ao trabalhar com múltiplos cadastros e sistemas legados, evidenciando a complexidade de estabelecer uma comunicação eficaz entre plataformas distintas. Além disso, foi destacada a questão de componentes essenciais, como interfaces de cadastro, serem desenvolvidos por terceiros, o que adiciona uma camada de complexidade e potencializa os riscos associados ao projeto.

(...) Tinha vários pedaços, né? Tinha um Cadastro que já existia e tinha o pessoal que não estava no Cadastro, que ia ser captado de alguma maneira e aí construiu-se, né um aplicativo?

A equipe chegou a preparar um protótipo na época para entregar correndo, e o banco (...) desenvolveu a carinha do aplicativo (...) que aí impôs uma quantidade de riscos porque a gente virou no sistema da

década de 80, arquivo para lá, arquivo para cá (...) é batimento de arquivo (...) o maior risco que a gente teve ali foi o fato de ter perdido a camada de aplicação no aplicativo ponta visível para o cliente, para o banco.

A apresentação disponível no repositório da Empresa BETA que foi conduzida pela autoridade máxima da Empresa BETA em evento aberto ao público, reforçou as observações feitas pelos entrevistados ao demonstrar o uso de bases de dados distintas, que culminaram na manipulação de unidades de informação. Durante o mesmo evento, enfatizou-se que solicitações foram registradas por meio do aplicativo e/ou portal do Banco, alinhando-se aos relatos anteriores. Adicionalmente, o corpus documental desta pesquisa permitiu identificar documento não sigiloso que detalha a arquitetura da solução implementada, incluindo o método de integração baseado na troca de arquivos de texto com a instituição bancária para concretizar os pagamentos após o processo de verificação cruzada. A existência do protótipo proposto pela Empresa BETA e mencionada durante a entrevista, foi confirmada pelas evidências documentais. É importante ressaltar que o aplicativo desenvolvido pelo banco desempenhou papel necessário durante a implementação do programa social Alfa, estando disponível para acesso público.

Corroborando com as informações anteriores, foi localizado no corpus documental deste estudo template que estabelece o formato padrão para troca de informações entre a instituição financeira e a Empresa BETA. Este template foi utilizado para padronizar a comunicação e garantir a precisão na identificação daqueles que deveriam receber o benefício social decorrente do referido programa social Alfa.

Esse contexto reflete as ideias de Abdul Karim et al. (2016), que discutem a prática do desenvolvimento seguro de software dentro do SDLC, enfatizando a importância da integração e validação de componentes de software não desenvolvidos originalmente para o projeto em questão. Além disso, Alenezi e Almuaihiri (2019a) ao dispor sobre riscos de segurança no ciclo de vida do desenvolvimento de software destaca que a integração de componentes de terceiros pode introduzir vulnerabilidades e complexidades, exigindo uma avaliação criteriosa e um planejamento robusto para mitigar os riscos associados. Essas perspectivas sublinham a necessidade de uma abordagem sistemática e proativa para gerenciar a integração de software pré-existente, assegurando que os componentes funcionem harmoniosamente e atendam aos objetivos do projeto.

O atributo **16b. 'Teste de Unidade'** é um aspecto a ser considerado no desenvolvimento de software, enfocando a verificação da funcionalidade de componentes individuais foi mencionado 8,33% dos entrevistados. Este atributo contribui para identificar e corrigir erros em estágios iniciais, buscando garantir a qualidade e a confiabilidade do sistema. A este respeito, vide relato de entrevistado: *“(...) extremamente eficaz por outro lado, a própria necessidade de urgência, né? De tempestividades faz com que as equipes (...) invistam menos, né? (...) Em qualidade de testes, né? Em automação (...)”*

O corpus documental desta pesquisa evidenciou que foram realizados testes. A partir deles foi possível identificar falhas que, uma vez corrigidas, asseguraram a funcionalidade do sistema em conformidade com os requisitos estabelecidos. As evidências documentais não permitiram identificar que estes testes tenham sido insuficientes.

Myers, Sandler e Badgett (2011) salientam a importância dos testes de unidade como uma prática essencial para assegurar a qualidade do software. A realização efetiva de testes de unidade não só ajuda a detectar e corrigir falhas precocemente, mas também contribui para a documentação do código, tornando-o mais compreensível e manutenível.

O atributo **18a. 'Ambiente'** mencionado por 8,33% refere-se à adequação do ambiente de desenvolvimento, teste e integração para refletir os cenários operacionais reais. Um ambiente bem estruturado contribui para garantir que o produto final seja robusto e esteja preparado para o ambiente operacional. A este respeito, vide depoimento dos entrevistados: *“(...) precisava fazer um teste de desempenho. Essa parte de integração em teste, tipo o ambiente deixá-lo redondo, o produto, o sistema. (...) Eles rodavam o teste, aí precisava calibrar a infraestrutura, voltava para a gente. (...)”*

Conforme evidências do corpus documental deste trabalho, foram criados os ambientes de homologação e desenvolvimento, sendo ambas réplicas arquiteturas do ambiente de produção e com distinção apenas em quantidade de equipamentos, de tal modo que a cada teste realizado e explicitado esses ambientes eram ajustados e posteriormente refletidos no ambiente produtivo, porém, em maior escala.

A adequação do ambiente de desenvolvimento e teste é um ponto enfatizado por Bass, Clements e Kazman (2012), que discutem a importância de um ambiente que simule com precisão as condições operacionais reais. Um ambiente de teste inadequado pode levar a suposições errôneas sobre o desempenho e o comportamento do sistema no mundo real, resultando em falhas não antecipadas.

O atributo **20c. 'Sistema'** mencionado por 8,33% dos entrevistados aborda os riscos associados à integração do produto dentro do sistema, incluindo a precisão na reprodução das condições de interface e a suficiência do tempo destinado aos testes. Segue transcrição do entrevistado a respeito desse atributo: *"(...) é essa interlocução entre ambientes diversos, né? O ambiente externo pra poder chegar no nosso ambiente, acho que foi um impacto também. Foi um, foi algo que impactou bastante. É a metodologia, foi a metodologia ágil."*

A integração do produto dentro do sistema, contemplando integração com ambiente externo, conforme destacado pelo entrevistado, ressalta desafios na interlocução entre ambientes diversos. Evidências documentais que delineiam os requisitos e escopo do benefício do programa social Alfa indicaram a complexidade dessa integração, enfatizando a necessidade de uma metodologia ágil para adaptar-se rapidamente às mudanças.

A integração de sistemas é um aspecto crítico discutido por Gomaa (2011), que ressalta a complexidade de garantir que os componentes de software interajam corretamente. A precisão nas condições de interface e um tempo adequado de teste contribuem para evitar problemas de integração que podem comprometer a funcionalidade do sistema como um todo.

O atributo **22b. 'Confiabilidade'** mencionado por 8,33% dos entrevistados enfatiza a importância de um sistema que opere de maneira consistente e confiável ao longo do tempo, atendendo às expectativas de desempenho e disponibilidade. A respeito dessa disponibilidade, um dos entrevistados informou o seguinte: *"(...) Um dos pontos mais críticos para nós ali diziam respeito à disponibilidade."* Essa disponibilidade foi destacada como um elemento crítico, uma ênfase compartilhada por evidências documentais, que prescrevem critérios para o funcionamento eficiente do sistema.

A prioridade na confiabilidade é reforçada pelas práticas de monitoramento, conforme documentado no corpus documental, com evidências sobre a disponibilidade do sistema. Adicionalmente, os relatórios fornecidos por empresas terceirizadas, que avaliam a integridade da rede usada pela Empresa BETA, estabelecem diretrizes para assegurar tanto a precisão quanto a acessibilidade contínua do sistema, evidenciando que houve disponibilidade de 99% ao longo do período de 15/03/2020 a 15/12/2020.

A confiabilidade é destacada por Laprie (1992), que a define como a capacidade de um sistema funcionar sem falhas por um período de tempo especificado em condições determinadas. Garantir a confiabilidade requer uma

atenção à qualidade do design, à robustez dos componentes e à eficácia dos processos de teste.

O atributo **23c. 'Segurança'**, mencionado por 16,67% dos entrevistados enfatiza o risco associado à correta execução dessas medidas e na subsequente dificuldade de comprovar sua efetividade, particularmente antes que o sistema esteja plenamente instalado e operacional. O relato dos entrevistados sobre o atributo 23c evidencia as complexidades envolvidas na verificação de elegibilidade de beneficiários em um sistema de grande escala. Ver a evidência desse relato, na sequência:

(...) É informava seu CPF, né? Fazia a composição da sua família, e entrava com requerimento via aplicativo. Aí esse requerimento, então ele chegava, né? E a gente analisava a pessoa ver se ela tinha renda e uma série de outros critérios definidos pelos clientes, né? Em que a gente elege-se aquele cara, tinha ou não tinha direito e qual era o valor que ele tinha direito? Então assim, uma pessoa podia entrar com seu CPF. Né? Ela podia entrar ali, seu CPF (...). Aí a gente tinha que na verdade, não é ver se você (...) era ou não era elegível, mas não fui eu que pedi, né? Então, assim, tinham pessoas politicamente expostas que acabaram, né concedendo benefício, né?

A preocupação verificada com base no relato do entrevistado, citado no parágrafo anterior, é refletida na literatura de segurança da informação. A situação descrita ressalta a importância de mecanismos de segurança para validar a identidade do usuário e verificar a sua elegibilidade ao benefício do programa social Alfa. Conforme apontado por Stallings (2017), a implementação de controles rigorosos de segurança é crucial para proteger contra ameaças internas e externas, garantindo que apenas usuários autorizados tenham acesso aos recursos do sistema e que todas as transações sejam devidamente auditadas e registradas para garantir a transparência e a responsabilização, aspectos fundamentais para a integridade de sistemas críticos como o descrito.

Mencionado por 16,67% dos entrevistados o atributo **25e. 'Fatores Humanos'** reconhece o risco de não atender adequadamente aos requisitos relacionados aos fatores humanos, devido à complexidade de compreender o ambiente operacional e chegar a um consenso com clientes e usuários. Destaca a necessidade de prototipagem e demonstração contínua para alinhar as expectativas e garantir uma experiência do usuário final coerente e eficaz. Na sequência exibe-se o relato de um entrevistado a respeito do atributo citado neste parágrafo.

(...) Só faltou uma comunicação mais clara com o cidadão. Assim, você o teu auxílio (...) foi negado porque você, anteriormente, no grupo familiar, você fez isso, entendeu? Então faltou informação (...). Eu acho que teve uma falta assim, eu não sei se assim leia aqui que o pessoal acaba não lendo assim.

O relato evidencia uma lacuna em relação ao atributo 25e, em especial no que tange à comunicação eficaz com os usuários finais. Esta lacuna ressalta as preocupações levantadas por Norman (2013) sobre a importância da usabilidade e da experiência do usuário na concepção de sistemas, destacando que o design centrado no usuário e a comunicação eficaz são importantes para atender às necessidades e às expectativas dos usuários, minimizando riscos e maximizando a satisfação do usuário final. A falta de clareza na comunicação dos motivos de inelegibilidade para o benefício do programa Social Alfa ressalta a necessidade de interfaces de usuário intuitivas e informativas, que não apenas facilitem a interação, mas também forneçam feedback claro e compreensível. Isso contribui para evitar mal-entendidos e contribuir para que os usuários estejam cientes dos processos e das decisões que afetam diretamente suas necessidades.

O atributo **26f. 'Especificações'** mencionado por 8,33% dos entrevistados aborda o risco associado à implementação das especificações do sistema e à qualidade de atributos como estabilidade, completude, clareza e verificabilidade. A este respeito, vide o depoimento de um entrevistado *"(...) Era aquela incerteza. Estamos fazendo certo, não é? É aí a arquitetura ali, de uma forma rápida (...)"*

As incertezas sobre estar "fazendo certo" mencionadas no trecho entrevista do parágrafo anterior refletem a dinâmica e a volatilidade dos requisitos, como evidenciado no corpus documental, que evidencia registro de alterações nos requisitos ao longo do desenvolvimento. Isso destaca a importância de especificações claras e verificáveis para alinhar o desenvolvimento às expectativas do projeto, conforme discutido em documentos como o Termo de Abertura.

A importância de Especificações claras e verificáveis é enfatizada por Wieggers e Beatty (2013), que discutem como a falta de especificações claras pode levar a mal-entendidos e erros de implementação. Uma definição precisa de requisitos e a verificação contínua durante o desenvolvimento contribuem para garantir que o produto final esteja alinhado com as expectativas do cliente.

O próximo atributo **27a. 'Formalidade'** que foi mencionado por 25,00% dos entrevistados e diz respeito ao risco de inconsistências e falhas devido à falta de

definição, documentação e comunicação formal do processo de desenvolvimento em todas as suas fases. A este respeito, ver o trecho do relato de entrevistado, na sequência.

(...) O processo natural de desenvolvimento da BETA que a vida é um pouco burocrática, né? Não foi seguido à risca, né? Ponto.

(...) não preencher a matriz de responsabilidade tem umas coisas muito burocráticas na parte do desenvolvimento (...)

A documentação ficou um pouco assim a desejar né?

A confirmação destas afirmações foi realizada na sua totalidade, uma vez que o processo de desenvolvimento da Empresa BETA é composto por 23 etapas e apenas 7 delas foram cumpridas, conforme descrito na sessão 4.1. Apesar do não cumprimento total das etapas, foram identificadas evidências de documentos que sugerem uma documentação das etapas que estavam sendo desenvolvidas.

No entanto, ao analisar o documento "Gestão de Mudança" vinculado à cadeia de valor da Empresa BETA, observou-se uma discrepância entre a prática relatada e a formalidade esperada, com registros de mudanças após a realização dessas mudanças, contrariando o protocolo estabelecido e ilustrando um desafio intrínseco à gestão de projetos em contextos dinâmicos e urgentes. A situação espelha a complexidade de equilibrar a formalidade procedural com a necessidade de resposta rápida, um tema recorrente em discussões sobre gerenciamento ágil de projetos e desenvolvimento de software Schwaber e Sutherland (2020).

O atributo **28b. 'Adequabilidade'**, foi mencionado por 16,67% dos entrevistados e aborda o risco associado à escolha de modelos, processos, métodos e ferramentas de desenvolvimento que podem não ser inteiramente adequados ao escopo e às demandas específicas do programa. Ressalta a importância de uma seleção criteriosa para assegurar que as abordagens de desenvolvimento estejam alinhadas às necessidades do projeto. A este respeito, segue relato obtido nas entrevistas

(...) A gente usou pra aquela velocidade uma tecnologia SAS, um processo de desenvolvimento muito fora da rota normal, né? Havia pessoas ali naquela tecnologia, SAS. O pessoal lá, por exemplo, num pedaço ali muito importante, que praticamente vestiu a camisa e botou, bateu no peito e eu vou resolver isso aqui. E aí a galera tentando dar suporte para que eles fizessem aquilo daquela forma fora de um processo normal, a partir de processamento eu quero dizer, né? O

benefício, se a gente separar em pedacinhos, houve vários pedacinhos. Eu acho que esse é o mais fora do ciclo produtivo normal, né? (...)

Analisando as informações extraídas do conjunto documental, fica evidente que a escolha da Plataforma de Analytics SAS como núcleo da solução teve um papel importante no cruzamento de dados, fundamentais para a tomada de decisão. Esta plataforma não apenas facilitou o processamento analítico, mas também contribuiu para a alimentação da base de dados relacional, fornecendo recursos essenciais para consultas e para a geração de relatórios detalhados. Tal escolha, embora desalinhada dos processos de desenvolvimento padrão, foi uma medida adotada diante das necessidades do projeto.

A escolha da tecnologia SAS e a adoção de processos de desenvolvimento não convencionais alinham-se ao estudo de Boehm e Turner (2004), que enfatiza a importância de adaptar as práticas de desenvolvimento ao contexto específico do projeto. O comprometimento e a adaptabilidade da equipe são importantes, mas a escolha de ferramentas e métodos apropriados são importantes, também, para garantir a eficácia e a eficiência do desenvolvimento. O estudo referido neste parágrafo, também, reforça que a seleção adequada de processos e ferramentas não apenas facilita a realização dos objetivos do projeto, mas também minimiza os riscos associados à incompatibilidade entre as metodologias de desenvolvimento e as necessidades específicas do projeto.

O atributo **29c. 'Controle de Processo'** mencionado por 8,33% dos entrevistados reflete a importância de manter um controle rigoroso e sistemático do processo de desenvolvimento, especialmente em projetos com vários locais de desenvolvimento e a necessidade de monitoramento da aderência ao processo definido. A necessidade de controle do processo de desenvolvimento, foi apontada por entrevistado, conforme trecho a seguir: *"(...) Tomar decisões em salas online e aplicar essas decisões com o nível de documentação mínimo"*.

A necessidade de controle evidenciada pelo entrevistado foi confirmada pela abordagem documentada no corpus documental, identificando-se evidências que estabeleceram um quadro para a aderência ao processo definido. Esse controle foi considerado para gerenciar a complexidade de desenvolver os softwares provedores do Programa Social Alfa a partir de múltiplos locais de desenvolvimento e monitoramento.

O controle efetivo de processos é destacado por Humphrey (1989), que discute a importância de práticas como a Engenharia de Software Baseada em Processos para garantir que os projetos sejam realizados de maneira previsível e controlada. O monitoramento rigoroso do processo e a melhoria contínua são fundamentais para alcançar eficiência e qualidade no desenvolvimento de software.

O atributo **30d. 'Familiaridade'**, dispõe sobre o risco associado à falta de conhecimento, experiência e conforto do pessoal do programa com o processo de desenvolvimento definido. Ele foi mencionado por 33,33% dos entrevistados, exibindo-se os seguintes trechos de relatos dos entrevistados, na sequência:

(...) As pessoas estavam aprendendo uma nova forma de trabalhar, né?

(...) O motor de reconhecimento ele foi desenvolvido ali na equipe de estatísticos, né? São pessoas fantásticas que têm um conhecimento. É muito bom, principalmente da ferramenta. Só que eles são estatísticos, né? Eles não são engenheiros de software, né? Eles não têm informação de engenharia de software, né? E por conta disso, (...) A solução (...) ela foi se construindo ali.

(...) Certo, acho que o primeiro fator foi a questão da tecnologia utilizada, né? A gente partiu para uma solução que não era tão comum na casa.

(...) É o conhecimento da equipe, porque a gente não pôde escolher as pessoas com as quais a gente iria trabalhar porque o resto dos caras que eram os medalhões da empresa estavam também no Alfa, mas em outras equipes. Então assim é o que determinou foi isso

O atributo 30.d diz respeito a familiaridade da equipe, composta por estatísticos, atuando na construção da versão inicial da solução em uma plataforma analítica. A diversidade de origens das bases de dados e a consequente variação na qualidade das informações apresentaram um obstáculo adicional, influenciando a qualidade dos resultados esperados. A solução foi conceber uma base analítica consolidada, reunindo informações de várias bases governamentais em um único repositório. Esse esforço de padronização visava assegurar a precisão dos resultados de acordo com as diretrizes do Cliente. Para validação, amostras estatísticas foram elaboradas, possibilitando ao Cliente a realização de uma homologação prévia. Após a homologação, os dados precisavam ser transmitidos para as instituições financeiras em um formato compatível com seus sistemas de processamento.

Naquele momento duas condições afetavam o projeto, a primeira diz respeito a instituição de novo modelo de trabalho em função da crise sanitária, o teletrabalho, regulamentado internamente na Empresa BETA e que afetou todos os empregados. A segunda condição tratou-se do processo de construção dos softwares provedores do programa social Alfa, verificando-se, segundo as evidências documentais, a necessidade de cruzamento de volume significativo de dados. Assim, a equipe inicialmente desenvolveu modelos de dados utilizando ferramentas analíticas, conforme descrito no parágrafo imediatamente anterior a este, o que levou ao envolvimento de estatísticos que auxiliaram esse desenvolvimento de modelos, mas não eram, necessariamente, familiarizados com o processo de desenvolvimento de software.

O atributo **32a. 'Capacidade'** foi mencionado por 25,00% dos entrevistados e diz respeito ao risco de limitações devido a poucas estações de trabalho, poder de processamento insuficiente ou armazenamento de banco de dados, ou outras inadequações do equipamento para suportar atividades paralelas de desenvolvimento, teste e suporte.

Os relatos das entrevistas evidenciaram, principalmente, a capacidade em suportar o volume de acessos e a quantidade de informações a serem processadas, conforme descrito a seguir.

(...) Como é que a gente vai dar conta de um número gigantesco de requisições que entrariam em simultâneo, né?

Escalabilidade, né? (...)

Então, assim que a gente soltava Lote do benefício e a gente tinha um pico de acesso que a gente precisava conseguir aguentar esse pico de acesso e esse pico de acesso gerava concorrência.

E aí o mais arriscado para os ambientes, porque não se sabia qual é o ambiente que vai, será que o ambiente vai suportar aquela capacidade de aquela quantidade de processamento?

Embora as preocupações expressadas pelos entrevistados se concentrassem no caráter inovador da solução e na urgência de sua implementação aliados a falta de informação sobre a capacidade necessária, o corpus de documentos revelou que a infraestrutura de TI foi preparada, com a configuração capaz de suportar uma demanda elevada. A análise de desempenho dos três meses após o início do processamento, indicou que a utilização máxima de CPU alcançou 38,7%, enquanto o consumo de memória RAM atingiu 99,8%, memória Swap 14,7%, e uso de disco 85%. Adicionalmente, o relatório de disponibilidade apontou que, nos três meses

subsequentes ao lançamento da solução, a plataforma manteve-se totalmente operacional, sem registros de indisponibilidade. A ausência de indisponibilidade nos três meses pós-lançamento depõe a favor das medidas de capacidade adotadas, desafiando a percepção de risco. Destaque-se a importância de uma infraestrutura de TI bem dimensionada e testada em projetos de grande envergadura (Gartner, 2021).

Com 41,67% das menções, o atributo **39a. 'Planejamento'** emergiu no discurso dos entrevistados. Este atributo destaca as dificuldades enfrentadas na elaboração de um plano adaptável a imprevistos e alinhado a metas de longo prazo. A menção deste atributo reflete desafios de formalizar e adaptar os planos de acordo com as mudanças necessárias, um aspecto crítico para a gestão eficiente do desenvolvimento.

(...) Acredito que a gente gastou muito mais energia do que a gente gastaria. Se a gente tivesse investido um pouco, é um pouco mais, né? Em análise em projeto da solução como um todo. Para mim, esse é o problema.

(...) Foi tudo muito de repente, né? Teve essa necessidade, então começaram a pensar em nível de governo, né? E depois, como fica?

A perspectiva do entrevistado encontra suporte no histórico normativo. Exemplificando, houve extensão dos benefícios do programa social. Estas modificações legislativas ilustram a complexidade de se estabelecer um planejamento conforme delineado pela taxonomia do SEI, confirmando as observações dos entrevistados sobre os obstáculos inerentes ao planejamento em um contexto de mudanças rápidas.

Além desse benefício, as evidências documentais indicaram a implementação governamental de programas de assistência social destinados a diferentes categorias profissionais. Essas iniciativas demandaram novos esquemas de planejamento, mas enfrentaram desafios decorrentes da limitação ou insuficiência de informações. Essa carência de dados claros e precisos complicou a implementação e a eficácia desses programas, refletindo a complexidade de adaptar as medidas de apoio às necessidades específicas de cada setor profissional em um contexto dinâmico.

Os desafios de planejamento no desenvolvimento de softwares em situações emergenciais, como ilustrado pelo projeto ALFA, ecoam nas considerações de Boehm (1991), que destaca a importância de um planejamento flexível e adaptativo em projetos de software para acomodar mudanças e incertezas. o estudo citado neste parágrafo salienta que, em ambientes onde as necessidades e os requisitos estão em

constante evolução, um planejamento rígido pode levar a esforços excessivos e a resultados a quem do esperado, uma vez que o projeto pode não responder adequadamente às mudanças necessárias.

O atributo, **42d. Interfaces do Programa**, com 33,33% de menção entre os entrevistados, trata do risco de problemas de comunicação e coordenação decorrentes da interação dos gestores em todos os níveis com o pessoal do programa e com pessoal externo, como clientes e gerência sênior.

(...) A gente tinha um pouco de interação aí falando de externa, externa mesmo, né? (...) com o Banco, né? e com o Cliente.

Então a gente teve dificuldades, eu diria mais com o Banco. Porém, mais com o banco porque, é na questão das negociações, porque como a gente tinha muito assim, todo o tráfego era de arquivos daqui para lá e de lá para cá, entre o banco e a Empresa BETA. E aí, é negociação de layout, negociação de prazos, é tudo muito difícil.

(...) E assim, o que eu, o que eu presenciei também de várias pessoas é que é a pessoa fazia, é uma, vamos lá (...) deixa eu organizar meu raciocínio aqui para você, eu acho que foi uma falta de comunicação assim, de entendimento entre Empresa BETA e o Banco, no sentido de solucionar o problema como um todo, entendeu? (...)

Ao abordar o atributo 42d, pode-se notar que a experiência dos entrevistados, indicou dificuldades de comunicação entre Empresa BETA e o Banco. Entretanto, não foi possível identificar evidências documentais conclusivas que comprovem ou refutem a existência dessas dificuldades. Esta constatação, entretanto, pode sinalizar a necessidade de uma estrutura mais transparente para a sociedade sobre a comunicação e a coordenação entre as partes, o que pode contribuir para o sucesso de iniciativas que visam atender aos direitos sociais em contextos complexos como o delineado por Castro (2012).

O atributo **44b. 'Gestão de Pessoal'** foi mencionado por 25,00% dos entrevistados e observa o risco relacionado à seleção e treinamento inadequados dos membros do programa, o que pode comprometer o desempenho de acordo com o planejamento e a interação com o cliente. A respeito desse atributo, na sequência destaca-se a mudança do regime de trabalho, afetando, inclusive, a equipe de tecnologia.

(...) As pessoas não estavam mais trabalhando na Alfa, né? Elas tinham sido obrigadas a ficar em casa, então tem toda uma adaptação

envolvida nisso, né? Uma adaptação da rotina de trabalho, uma adaptação é até do grupo de tecnologia, né? (...)

Constatou-se que 91,66%, dos entrevistados possuía uma experiência de mais de quatro anos de atuação em suas respectivas áreas. No entanto, as circunstâncias excepcionais impostas pela pandemia trouxeram consigo desafios, especialmente no que diz respeito à adaptação das equipes. O maior desafio, conforme relatado pelos entrevistados, foi a transição para o trabalho remoto.

Neste contexto, a Empresa BETA implementou estratégias, conforme evidenciado no corpus de documentos coletados neste estudo, em especial comunicações e normativos internos. Esses documentos indicaram o estabelecimento de estrutura para o gerenciamento de equipes em regime de teletrabalho, buscando assegurar que, apesar dos desafios associados ao trabalho remoto, a capacidade operacional da equipe envolvida no processo de desenvolvimento dos softwares provedores do programa social Alfa fosse preservada. Essas medidas podem servir de exemplo da adaptabilidade e do treinamento contínuo em gestão de pessoal (Gartner, 2021).

Mencionado por 25,00% dos entrevistados e o atributo **45c. 'Garantia de Qualidade'** destaca o risco de comprometimento da qualidade do programa devido à implementação inadequada de processos e padrões, ou devido à equipe de garantia de qualidade estar subdimensionada. Esse risco foi evidenciado nas entrevistas da seguinte forma:

(...) É, no mínimo de qualidade aceitável, né? Porque o risco nesse caso aí de você sacrificar a qualidade é enorme, é porque você tem pouco tempo para testar aí.

(...) O risco é o Ambiente de desenvolvimento é a questão de qualidade mesmo (...) puxada pela tempestividade, uma vez que a gente tem uma urgência, que a gente precisa entregar o mais rápido possível.

A Garantia de Qualidade neste projeto se estendeu a fatores como o gerenciamento do tempo, a execução de testes e a precisão dos dados. A qualidade, um conceito multifacetado, foi avaliada principalmente pela eficiência na operação e na geração de resultados confiáveis. Isso se reflete no sistema de performance que realizou cinco ciclos de checagem de dados, otimizando processos de 16 horas para 2 horas e processando milhões de solicitações de elegibilidade. As evidências

documentais indicam essas realizações, evidenciando esforços para manter a qualidade mínima aceitável, corroborando o relato dos entrevistados.

Apesar de todo o esforço, o corpus documental revelou, por meio de relatório internacional sobre a COVID-19, a heterogeneidade e qualidade variável das bases de dados utilizadas, introduzindo uma complexidade adicional ao processo. A dificuldade de integração dessas bases, pontuada, também, por evidências documentais de órgão de controle externo, ressalta as barreiras enfrentadas em relação à identificação inequívoca dos indivíduos e à qualidade dos dados, contribuindo para interferir na eficácia dos procedimentos.

O atributo **48b. 'Cooperação'**, mencionado por 8,33% dos entrevistados, destaca a necessidade de um trabalho em equipe e da remoção de barreiras para a realização eficaz do trabalho, tanto internamente quanto entre diferentes grupos de trabalho. Na sequência, expõe-se a percepção dos entrevistados que se referem ao atributo citado neste parágrafo:

(...) A geração espontânea, uma solução que não teve o olhar da arquitetura, não teve acompanhamento, orientação da arquitetura. Se a gente estava para atuar em uma fase de planejamento e orientação para aquele caso, a gente não teve essa oportunidade (...)

Sem prejuízo do relato da entrevista, o corpus documental evidenciou que a colaboração com as equipes de arquitetura técnica, padrões tecnológicos, engenharia de dados e segurança para a definição de soluções foi explicitada por meio de manual de atribuições de responsabilidades de área responsável pela provisão da infraestrutura tecnológica da Empresa BETA. Além disso, a documentação dos resultados da solução arquitetural evidenciou que a atuação dessa área alinhou-se com as atribuições previamente estabelecidas pelo referido manual.

A respeito do Atributo 48b, embora tenha sido verificada a percepção do entrevistado de que a arquitetura poderia ter sido mais envolvida no acompanhamento do desenvolvimento do projeto, os documentos internos sugeriram que a atuação da arquitetura se manteve nos limites de suas responsabilidades. Vale citar que a cooperação efetiva entre equipes é um fator chave para o sucesso de projetos de software, conforme discutido por Brooks (1975). A comunicação eficaz, o entendimento mútuo e a colaboração são essenciais para superar desafios, otimizar recursos e alcançar os objetivos do projeto de forma harmoniosa.

O atributo **49c. 'Comunicação'** foi mencionado por 25,00% dos entrevistados e destaca o risco decorrente da comunicação deficiente, que pode resultar em falta de conhecimento sobre a missão, requisitos, objetivos de design do sistema, ou sobre a importância dos objetivos do programa para a empresa ou projeto.

Constatam-se, a partir das entrevistas, relatos de experiências positivas de comunicação interna em função da utilização de plataformas online, contudo, por vezes foram relatadas dificuldades de comunicação em especial com o cliente para definição de requisitos, regras de negócio e obtenção de informação.

(...) É interface com o terceiro, a gente teve apesar do time lá do Cliente está bastante mobilizado na implantação disso, a gente se deparava com cenários de dificuldade de interação e, principalmente, de entendimento. É do que é que se esperava em determinado espaço.

(...) Só faltou uma comunicação mais clara com o cidadão. Assim, (...) o teu benefício foi negado porque você, anteriormente, no grupo familiar, você fez isso, entendeu? Então faltou informação. (...)

Apesar dos relatos dos entrevistados sobre as dificuldades de comunicação, especialmente em interações com o cliente e na definição de requisitos e regras de negócio, não foram encontrados registros que corroborassem essas dificuldades de forma sistemática. Além disso, conforme evidência de corpus documental, existiu um modelo formal de comunicação entre as partes envolvidas no projeto por meio de contrato, estipulando que a comunicação formal deve ser mantida por meio de sistemas de gestão de incidentes ou demandas previamente acordados e que todas as comunicações relevantes devem ser consideradas efetivas quando entregues ou enviadas através de ofício, carta protocolada, telegrama, fac-símile ou e-mail, com a devida confirmação.

Essa formalização da comunicação tem o propósito de assegurar que todas as partes estejam alinhadas e que as informações essenciais sejam compartilhadas de maneira eficiente e documentada, minimizando assim os riscos associados à comunicação deficiente. Portanto, apesar dos desafios comunicacionais relatados nas entrevistas, o contrato administrativo nº 12/2020 proporciona um quadro de formalização que visa garantir a clareza e a eficácia na troca de informações entre os envolvidos no projeto.

O relato dos entrevistados sobre as falhas na comunicação, especialmente em termos de interações cliente-desenvolvedor e na articulação clara de requisitos e regras de negócio, reflete uma preocupação que se estende a outros projetos de

tecnologia da informação, em que a comunicação eficaz contribui para o sucesso Kieling e Magnagnagno (2021).

Já o atributo **50d. 'Moral'** mencionado por 25,00% dos entrevistados trata do risco relacionado a baixa moral que pode levar a baixo entusiasmo, desempenho, produtividade ou criatividade; raiva que pode resultar em danos intencionais ao projeto ou produto; saída em massa do pessoal do projeto; e uma reputação negativa dentro da empresa que dificulta o recrutamento.

(...) eu lembro (...) que eu tinha um umas questões aí, intelectual, sim, que legal e desafiador ao mesmo tempo, (...) eu lembro que esse dia eu trabalhei (...) horas direto e quando eu parei eu não conseguia dormir.

(...) Cansadas no esforço da entrega e esse cansaço pode virar impacto no código ou até mesmo informação como houve da gente pelo cansaço, passar informações erradas para a frente para outras entidades.

(...) Eu soube de salas que equipes que tiveram essas discussões sérias assim a gente nunca teve nenhuma discussão. O clima sempre foi muito bom na sala. Mas a gente teve muita questão assim, de desabafo mesmo, das equipes, a galera entrar na sala e assim desabar, né? (...) dizer que não aguentava mais, que não tinha condição (...)

Conforme observado nos relatos das entrevistas, a participação no projeto motivou a equipe envolvida. Porém, em função do contexto pandêmico WHO (2020), aliado à necessidade de adoção do teletrabalho, evidenciando-se, também, pelo corpus documental, a participação formal de cinco funcionários no projeto, tudo isso ressaltou a severidade com que o moral da equipe pode ter sido impactado e refletido no atributo 50d. Essa combinação de elementos pode ter trazido uma tensão significativa, destacando a complexidade dos desafios enfrentados pela equipe.

Como evidenciado no

Quadro 19 foram identificados um total de 39 atributos ao longo das entrevistas. Dentre esses atributos, o atributo **51a. 'Cronograma'**, sendo mencionado por 100% dos entrevistados. Esse fenômeno está intimamente relacionado ao desafio representado pelo prazo reduzido entre a solicitação dos softwares provedores do programa social Alfa e sua entrega com resultados funcionais aos usuários, um período que se limitou a aproximadamente seis dias.

(...) Não tínhamos tempo hábil para isso, então o tempo que a gente tinha determinou a arquitetura que a gente ia utilizar baseado no conhecimento da equipe.

(...) Quem determinou a arquitetura foi o tempo, o tempo foi o crítico.

(...) Se você tinha que especificar de manhã, desenvolver de tarde já implantar produção de noite é, era o tempo que já terminaram, não tinha tempo. Você não tinha tempo para fazer o melhor. Você tinha tempo para desenvolver aquilo que era possível dentro daquela agenda aí.

(...) O tempo acabou determinando tudo, né? Determinada arquitetura, determinou como ele ia ser implementado a solução e determinou também o quanto de qualidade a gente ia conseguir implementar no curto tempo que a gente tinha.

O contrato cujo objetivo era a prestação de serviços especializados em TI para o desenvolvimento de sistemas, foi formalizado e os resultados gerados por meio desse sistema foram disponibilizados para os usuários a partir do terceiro dia da formalização do contrato. Apenas três dias antes da assinatura do contrato, a OMS, proclamou a pandemia de COVID-19 (WHO, 2023). Esses fatos evidenciam a limitação de tempo imposta para o desenvolvimento dos softwares.

A situação emergencial da pandemia, oficialmente reconhecida pela OMS, pressionou o processo de desenvolvimento, demandando a entrega rápida da solução. Nesse contexto, o tempo tornou-se determinante não apenas na definição da arquitetura e na implementação da solução, mas também na medida em que influenciou a qualidade do projeto. A impossibilidade de dedicar um período maior para o desenvolvimento resultou em desafios para a equipe, que se viu obrigada a priorizar a funcionalidade e a entrega rápida. A análise desses eventos ressalta a relevância do fator temporal como um elemento crítico que moldou a abordagem e os resultados do projeto.

A urgência do projeto Alfa, com apenas seis dias entre a solicitação e a entrega, destaca o fator tempo como um elemento crítico, moldando a arquitetura, a

implementação e a qualidade da solução desenvolvida. Esse cenário ressoa com a constatação de Menezes et al. (2019), permitindo identificar pressões considerando o tempo. A natureza urgente da entrega, aliada ao contexto pandêmico, sublinha a importância de um gerenciamento de risco.

O atributo **52b. 'Pessoal'** foi mencionado por 41,67% dos entrevistados, ilustra as complexidades associadas à gestão de recursos humanos em um ambiente de projeto exigente. Este atributo engloba não apenas a instabilidade numérica da equipe, mas também a adequação de suas competências, experiências e disponibilidade. As narrativas das entrevistas ressaltam a carga substancial suportada por um núcleo pequeno de funcionários.

(...) A gente pode contar 10-15 pessoas. E, é claro, a equipe envolvida nisso era maior, mas essas 10-15 pessoas pontuais, se elas não estivessem no quadro da empresa, talvez os outros não conseguissem fazer. Elas lideraram, abriram caminhos, inovaram.

(...) Nós tivemos uma dificuldade inicial para poder alocar as equipes, porque as equipes tiveram que trabalhar quase 24 por 7, então fazer a alocação de recursos (...).

(...) E isso potencializou o quê? Jornadas excessivas de trabalho, né? As noites viradas né.

(...) A quantidade de pessoas ali (...) que se acometeram de algumas situações de estresse (...) por conta dos prazos, por conta da tempestividade das entregas (...).

O corpus documental evidenciou a participação formal de cinco funcionários no projeto. No entanto, reconhecendo a possibilidade de um envolvimento mais amplo conforme destacado nas entrevistas, foram conduzidas análises adicionais. A análise revelou a existência de grupos especialmente designados para trabalhar diretamente na solução proposta. Essa evidência expandiu o escopo do envolvimento do pessoal, elevando o número total de funcionários engajados para aproximadamente 28, o que indica uma força de trabalho maior do que a inicialmente documentada.

A respeito do atributo 52b, Boehm (1988) argumenta que a qualidade e a produtividade de um projeto são afetadas pela competência e pelo empenho da equipe. Além disso, a investigação de Carvalho e Rabechini Junior (2015) enfatizam que a gestão eficaz de recursos humanos é um fator crítico para o sucesso do projeto, destacando a necessidade de alinhar as habilidades da equipe com as demandas do projeto e de manter uma comunicação eficiente entre os membros da equipe.

O atributo **54d. 'Instalações'** mencionado por 8,33% dos entrevistados, refere-se à adequação e disponibilidade das instalações físicas e infraestrutura necessárias para o desenvolvimento, integração e teste do produto. Segundo relato do entrevistado, o assunto deveria, inclusive, ser abordado '*(...) do ponto de vista de estrutura física*'. Uma vez que no período de realização do projeto os funcionários nele envolvidos estavam em regime de teletrabalho, as evidências documentais abrangendo as instalações físicas consideraram a oferta de infraestrutura tecnológica para os funcionários atuarem remotamente, em suas residências.

Ainda a respeito do atributo 54.d, a adequação das instalações é um fator crítico na realização de projetos de desenvolvimento de software, conforme enfatizado por Allen (2002). Instalações apropriadas não apenas facilitam a comunicação e a colaboração entre os membros da equipe, mas também garantem que os recursos necessários estejam disponíveis e acessíveis, contribuindo para a eficiência e a eficácia do processo de desenvolvimento.

Mencionado por 25,00% dos entrevistados e o atributo **55a. 'Tipo de Contrato'** engloba risco associado aos termos de pagamento e aos requisitos contratuais como o escopo de trabalho e a quantidade e condições do envolvimento do cliente.

(...) se não me engano, o contrato do Alfa, ele só veio a ser fechado em outubro ou novembro de 2020. Daí o benefício começou em março
 (...) então (...) a gente fez toda a primeira parte toda sem contrato.
 (...) Como foi uma demanda que a gente recebeu (...) vamos fazer, né? Essa questão da relação contratual. Ela até veio depois para a precificação (...) né?

O extrato de contrato constante do corpus documental contradiz a narrativa das entrevistas, indicando que o contrato foi formalizado em 14/05/2020, o que sugere uma divergência entre a percepção dos entrevistados e os registros oficiais. Este contrato, firmado no contexto da pandemia, estabeleceu entregas, prazos, multas, entre outros elementos essenciais para a execução do projeto.

A discrepância nas datas sobre a formalização contratual destaca a importância do conhecimento prévio dos termos e condições contratuais, fato que não só contribui para uma execução mais alinhada às expectativas do cliente, mas também minimiza riscos e garante que todas as partes, incluindo os funcionários envolvidos, estejam cientes de suas responsabilidades, prazos e possíveis penalidades, contribuindo para um desempenho mais eficaz e uma satisfação sobre os objetivos alcançados do projeto.

Em relação ao atributo **58a. 'Cliente'**, mencionado por 25,00% dos entrevistados, enfoca os riscos associados a desafios decorrentes da possível falta de familiaridade do cliente com aspectos técnicos ou de aplicação do programa. Além disso, abrange a complexidade das interações, destacando as dificuldades nas relações de trabalho e nos procedimentos para alcançar consenso e aprovações por parte do cliente.

Durante as entrevistas, os participantes enfatizaram especificamente esses aspectos, ressaltando sua relevância e impacto no decorrer do projeto.

(...) A gente teve um fator externo que foi relevante. Até certa altura do tempo, a gente entendia que a nossa Empresa BETA faria o sistema de requerimentos do benefício e investiu bastante tempo nisso. Do nada, a gente soube que quem ia fazer esse sistema era o banco.

(...) sem dúvida o maior risco eu diria que foi a falta de conhecimento do cliente para o assunto.

Conforme evidenciado no documento de mudança arquitetural bem como no protótipo desenvolvido, havia um trabalho em andamento para que todo o software ficasse sob responsabilidade da Empresa BETA incluindo o portal de requerimento, os softwares de processamento e envio do arquivo resultante para os bancos, contudo, houve uma mudança de estratégia ao longo do caminho que ensejou em mudanças, passando o desenvolvimento do portal de requerimento do benefício para responsabilidade de instituição financeira. Não foram identificadas formalizações sobre os motivos que levaram a tal fato, contudo, esse particionamento do software bem como a necessidade de integração, trouxeram riscos ao projeto. Tanto a instituição financeira quanto a Empresa BETA tinham como cliente o órgão responsável pela gestão do benefício, o qual detém conhecimento sobre políticas públicas, e que apesar de possuir em sua estrutura uma área de Tecnologia da Informação, ele não possuía conhecimento completo sobre as necessidades, conforme relatado pelos entrevistados.

Essa falta de conhecimento pode afetar o desenvolvimento do software, pois conforme destaca Andrade et al. (2019) os requisitos definem restrições globais do software, do processo de desenvolvimento e do processo de implantação, sendo considerados globais no sentido de que eles surgem de todas as partes, podendo afetar toda a arquitetura do sistema e não apenas componentes individuais do sistema e suas interações com outros sistemas ou o ambiente.

O atributo **59b. 'Contratados Associados'** foi mencionado por 16,67% dos entrevistados. Esse atributo destaca o risco de desafios relacionados a conflitos de agendas políticas, problemas nas interfaces com sistemas desenvolvidos por organizações externas e dificuldades na cooperação para a coordenação de cronogramas e mudanças de configuração. O atributo, inclusive, enfatiza a importância de uma colaboração efetiva e de estratégias de comunicação para mitigar esses riscos e garantir a harmonia entre todas as partes envolvidas. A respeito da atuação dessas partes envolvidas, vide o relato de entrevistado, a seguir:

(...) Porque a gente precisava receber, é as informações, fazer o batimento aqui dentro e depois poder devolver. Então, os entes parceiros assim foi algo que também impactou bastante.

Focando na narrativa apresentada pelo conjunto completo de documentos, vários foram os órgãos que colaboraram com o fornecimento de bases dados, contudo, além do recorte temporal limitado, distorções de atualização de cadastros negaram a concessão do benefício citada por Marins et al. (2021) ou concederam de forma indevida a quem não deveria.

Esta situação sublinha a complexidade de sincronizar dados entre sistemas distintos, uma tarefa que exige coordenação e comunicação eficazes. Os desafios do projeto transcenderam os aspectos técnicos, envolvendo a necessidade de alinhamento e interação entre diversas entidades atuantes como parceiras. Portanto, a colaboração eficaz e a comunicação clara entre todas as partes revelaram-se elementos que contribuíram para o êxito do projeto, evidenciando a importância de acordos colaborativos e canais de comunicação bem definidos.

O atributo **56b. 'Restrições'** mencionado por 16,67% dos entrevistados, reflete os riscos advindos de diretrizes contratuais que impõem o uso de métodos ou equipamentos específicos de desenvolvimento. Destaca as complicações que podem surgir dessas imposições, como a necessidade de adquirir direitos de dados para o uso de software não desenvolvido internamente.

(...) É, eu. Eu vou citar algumas, porque foram mais de uma tá, mas assim, a Mais relevante de todas. Para mim foi a questão de é, a gente é uma empresa pública que precisa fazer licitação na hora de fazer uma contratação, né? Então, por exemplo, quando a gente começou a fazer o levantamento dos requisitos, a gente percebeu que a cota de

uso do recaptha que é um sistema de proteção ali que a gente queria utilizar é necessário que a gente se utiliza justamente para a segurança do produto. É, ia passar do nível gratuito, né? E isso era uma restrição grave.

Considerando as informações obtidas a partir dos documentos analisados e a situação particular da época de desenvolvimento do software, observa-se que a Empresa BETA não tinha um contrato estabelecido para o uso intensivo do reCaptcha, por exemplo. De acordo com o documento "Regulamento de licitações e contratos da BETA", é exigido que os contratos com terceiros para a prestação de serviços, aquisição, locação e alienação de bens e ativos, execução de obras, ou implementação de ônus real sobre bens do patrimônio, sejam formalizados por meio de licitação, salvo exceções previstas no regulamento e na legislação aplicável. Esta exigência ressalta as limitações enfrentadas para a aquisição de tecnologias essenciais ao desenvolvimento de produtos e serviços. O entrevistado, embora não cite a impossibilidade de contratação, enfatiza o desafio representado pela demora entre a identificação da necessidade e a efetivação da contratação. Tal situação evidencia as barreiras burocráticas e temporais que podem afetar a agilidade e a eficiência no processo de desenvolvimento de soluções tecnológicas.

O uso de serviços como o reCAPTCHA, cuja cota de uso gratuito pode ser rapidamente excedida em projetos de grande escala, exemplifica a complexidade de equilibrar a necessidade de funcionalidades de segurança robustas com as limitações contratuais e financeiras. Esta situação ressoa com a perspectiva de Boehm e Turner (2004), que destacam a importância de uma gestão cuidadosa das restrições contratuais e a necessidade de flexibilidade e planejamento estratégico na seleção e uso de ferramentas e métodos de desenvolvimento. A navegação bem-sucedida por essas restrições contratuais é crucial para o sucesso do projeto, exigindo uma compreensão profunda das implicações dessas limitações e uma abordagem proativa para mitigar seus impactos potenciais.

O atributo **61d. 'Contratante Principal'** mencionado por 8,33% dos entrevistados, aborda os riscos associados a mal-entendidos e dependências quando o programa é parte de um contrato maior, com complexidades na definição de tarefas e na coordenação de informações técnicas e programáticas. A respeito desse atributo, e os desafios decorrentes do relacionamento com contrapartes, apresenta-se, a seguir, o seguinte relato de entrevistado:

(...) A gente jogou fora toda a primeira parte da entrada e o banco passou a fazer os requerimentos. Isso não só fez a gente perder tempo com aquela interface de requerimento, mas também a gente teve que criar um modelo para receber esses dados do banco.

Os desafios associados a mal-entendidos e dependências em contratos maiores são evidenciados pela decisão de transferir a interface de requerimento para o banco, como mencionado na entrevista. Esta decisão, pode ser confirmada em evidência integrante do corpus documental que destaca também a complexidade na coordenação entre diferentes partes do contrato, alinhando-se ao risco evidenciado pela menção do entrevistado.

A complexidade na gestão de subcontratos é um desafio em projetos de software, como discutido por Kerzner (2013). A clareza nas definições de tarefas, os arranjos de reporte bem estruturados e a gestão eficaz da transferência de informações contribuem para garantir que todos os envolvidos estejam alinhados e que o projeto avance de forma integrada.

O atributo **63f. 'Fornecedores'** mencionado por 8,33% dos entrevistados, destaca a dependência de componentes do sistema fornecidos por terceiros, enfatizando a importância de uma gestão eficaz das relações com fornecedores e da qualidade dos produtos fornecidos. A dependência de componentes críticos fornecidos por terceiros, como o sistema bancário para o processamento de requerimentos, é um desafio destacado pelo entrevistado, com base no seguinte trecho: *"(...) A própria integração com o Banco implicava desafios de comunicação"*.

A gestão de fornecedores foi, também, confirmada por evidências do corpus documental. A gestão eficaz de fornecedores é crucial para o sucesso de projetos que dependem de componentes externos, conforme argumentado por Gadde e Snehota (2000). Uma comunicação clara, acordos bem definidos e monitoramento contínuo da qualidade dos produtos fornecidos contribuem para minimizar riscos e garantir que os componentes externos atendam às necessidades e expectativas do projeto.

O atributo **64g. 'Política'** mencionado por 8,33% dos entrevistados, refere-se ao impacto das relações e agendas políticas no desenvolvimento do projeto, ressaltando a necessidade de navegar cuidadosamente nessas dinâmicas para evitar conflitos e garantir a continuidade técnica e programática. A este respeito, exhibe-se na sequência relato de entrevistado:

(...) acho que se for conversar com as pessoas, não sei quantos você já conversou. Deve ser uma unanimidade. O banco tivesse a restrição

política disso não ter ido para eles, ter ficado com a gente. A camada de aplicação mesmo, né? A gente teria um conjunto de riscos (...) infinitamente menor (...) um produto muito melhor. Essa necessidade de geração de arquivos, batimentos e tudo mais seria tudo aqui dentro.

A influência das agendas políticas, mencionada pelo entrevistado, destaca o desafio de evitar conflitos. O contexto do desenvolvimento do Alfa, como o relato de transferência de responsabilidades de partes do desenvolvimento para o banco, no tocante à camada de aplicação, sugerem a complexidade de alinhar interesses técnicos e políticos para o sucesso do projeto. A este respeito, acrescenta-se que o corpus documental não contemplou a formalização dessa transferência de responsabilidades.

A influência da política em projetos de software é um aspecto frequentemente subestimado, mas que pode ter um impacto significativo, como discutido por Pinto e Slevin (1987). A compreensão das agendas políticas, a comunicação eficaz e a negociação habilidosa são cruciais para mitigar conflitos, alinhar expectativas e garantir que decisões técnicas sejam tomadas com base em critérios objetivos e não em interesses políticos.

Concluída a análise das entrevistas, apresenta-se a **Erro! Fonte de referência não encontrada.**², que oferece uma representação visual dos dados coletados nas entrevistas através de uma nuvem de palavras. A criação da nuvem de palavras, com base nos termos frequentemente mencionados durante as entrevistas, oferece uma visão agregada e sintetizada das preocupações e focos principais dos entrevistados. Esses termos estão relacionados, também, aos atributos identificados que se vinculam às fases do processo de desenvolvimento dos softwares provedores do Programa Social Brasileiro Alfa.

Com o objetivo de promover maior clareza e precisão na identificação e categorização dos elementos, foram omitidos termos que introduziam ambiguidade na interpretação dos resultados, tais como: conjunções, advérbios, preposições e pronomes. Palavras específicas como “Alfa”, “BETA”, “gestão”, “riscos”, “atributo” e seus respectivos sinônimos também foram removidas.

Essa abordagem metodológica ajuda na categorização dos atributos, facilitando a identificação de riscos para aprimoramento e foco na gestão do desenvolvimento de software no contexto do programa social em questão. Este esforço de síntese não apenas reflete a complexidade do projeto, mas também

destaca a importância de uma análise criteriosa para o sucesso e eficácia do desenvolvimento de soluções tecnológicas em programas sociais.



Figura 10 - Nuvem de palavras dos termos mais apresentados nas entrevistas
Fonte: Elaborado pelo autor a partir de dados da pesquisa.

Na condução deste estudo, ao analisar na Figura 10 as palavras que mais se destacaram devido à sua frequência significativa de menções, o que as torna indicativas de áreas críticas de atenção ao longo do projeto, é possível perceber a predominância de palavras específicas.

Na Figura 10, a palavra "Processo" demonstra que os entrevistados deram importância à necessidade ao processo de desenvolvimento. Conforme as análises prévias dos atributos, verificou-se que tal processo apresentou metodologia flexível para enfrentar os desafios inerentes ao projeto, especialmente considerando a pressão do tempo e as incertezas decorrentes da pandemia. Como destacam Boehm e Turner (2004), a escolha adequada de processos não apenas facilita a realização dos objetivos do projeto, mas também minimiza os riscos associados.

A segunda palavra-chave a ser lembrada na Figura 10 é "Tempo", que reflete a constante pressão que houve para concluir com celeridade o desenvolvimento dos softwares devido à urgência da situação de saúde pública. Segundo Patterson e Hennessy (2013), a necessidade de uma resposta rápida e eficaz ao problema de saúde pública adicionou uma camada significativa de complexidade ao projeto, exigindo uma gestão cuidadosa dos prazos e recursos disponíveis.

A terceira palavra-chave que merece destaque na Figura 10 é "Equipe", destacada pelos entrevistados. Conforme o Project Management Institute (PMI, 2017), a gestão dos recursos humanos e a promoção de uma cultura de colaboração e apoio

contribuem para enfrentar os desafios impostos, por exemplo pela pandemia, e garantir o progresso do desenvolvimento dos softwares, a exemplo daqueles softwares provedores do Programa Social Brasileiro 'Alfa'.

A palavra "Comunicação" citada na Figura 10 emergiu, também, da narrativa dos entrevistados. As análises das entrevistas indicaram, porém, que embora o objetivo fosse identificar atributos vinculados a riscos nas etapas do processo, os entrevistados relataram que o uso de ferramentas virtuais e o teletrabalho facilitaram o processo de comunicação. Nesse contexto, pode-se destacar que, segundo Kieling e Magnagnagno (2021), uma comunicação eficaz interfere no sucesso de projetos de tecnologia da informação.

Outra palavra-chave lembrada na Figura 10 é "Cliente", destacando a importância de entender e atender às necessidades e expectativas do cliente ao longo do projeto, bem como sua importância na definição de requisitos e escopo (Carr et al., 1993) enfatizam que a participação do cliente é fundamental para a redução das taxas de falha do projeto, refletindo uma abordagem centrada no cliente.

A palavra "Conhecimento", presente na Figura 10, sublinha a importância do domínio do conhecimento e da expertise técnica para lidar com os desafios do projeto. Como visto nas análises de atributos efetuada nesta seção, a equipe precisou contar com um domínio das tecnologias envolvidas para tomar decisões informadas e eficazes. Stallings (2017) reforça que a implementação de controles rigorosos de segurança contribui para proteger contra ameaças internas e externas, demonstrando que o conhecimento e a expertise técnica são elementos positivos dentro do projeto.

Por fim, as palavras 'Problema' e 'Requisitos', citadas na Figura 10, também, destacaram-se. Elas indicam que a equipe teve que abordar problemas específicos ao longo do projeto e que a gestão adequada dos requisitos do projeto foi crítica para atender às necessidades do cliente ou partes interessadas. Nesse contexto, conforme destaca Sommerville (2019), requisitos válidos são aqueles que refletem as necessidades dos stakeholders e que podem ser compreendidos por todas as partes envolvidas.

As palavras-chave mais representativas na Figura 10, descritas nos parágrafos anteriores, formam a espinha dorsal que resume a narrativa do projeto e, ao analisá-las em conjunto, é possível perceber como os desafios enfrentados em um ambiente de pandemia impactaram o desenvolvimento do software provedor do programa social brasileiro 'Alfa'. Esta análise das palavras-chave contribui para uma compreensão mais abrangente do projeto e dos fatores que influenciaram seu curso.

A partir da **Erro! Fonte de referência não encontrada.**10 é possível perceber que as palavras que se destacaram no corpus de pesquisa estão relacionadas aos atributos apresentados no Quadro 10 e à frequência com que esses atributos foram mencionados nas entrevistas. Esse padrão reforça a utilidade desses atributos para auxiliar a identificação de riscos no referido processo de desenvolvimento de softwares provedores do programa social brasileiro Alfa.

4.3 Classificação dos fatores riscos de acordo com elementos de classe correspondentes da Taxonomia SEI (Associado ao objetivo específico 3)

Após análise das entrevistas, e da identificação dos 39 atributos relacionados aos riscos em cada etapa do processo de desenvolvimento de software exibida na Seção 4.2, foi possível classificar os fatores riscos de acordo com os elementos de classe correspondentes da Taxonomia SEI, buscando atingir o 3º objetivo específico do estudo. Conforme descrito no Quadro 10, citado na metodologia deste estudo, foi possível agrupar os atributos identificados no Quadro 22 em cada um dos elementos de classe da referida taxonomia, os quais, por sua vez, estão agrupados em três classes.

Inicialmente, o Quadro 22 mostra os cinco elementos da classe "Engenharia de Produto". O destaque foi para o elemento de classe 'Requisitos', com 85,7% de menções de atributos a ele associados. A definição precisa de requisitos contribui para o sucesso do projeto, estabelecendo a fundação sobre a qual todo o desenvolvimento de software é construído (Wiegers e Beatty, 2013).

Quadro 22 – Elementos da classe Engenharia de produto

Elemento de classe	Total de Atributos	Total de Atributos Identificados	%
Requisitos	7	6	85,71
Design	7	4	57,14
Código e Teste Unitário	3	1	33,33
Integração e Teste	3	2	66,66
Especialidades de Engenharia	6	4	66,66

Fonte: o autor, a partir de dados da pesquisa, considerando resultados do Quadro 22.

Legenda: '%' indica o percentual de atributos identificados do elemento de classe.

Observação: em cada linha da coluna '%', 100% referem-se ao total de elementos de classe descrito na linha

Os elementos 'Integração e Teste', bem como 'Especialidades de Engenharia', foram descritos no Quadro 22 com 66,66% menções de atributos a eles associados. A integração eficaz dos componentes de software mostra-se importante para o desenvolvimento de um sistema coeso e funcional (Laprie, 1992).

O elemento de classe 'Design', citado no Quadro 22, obteve 57,14% dos atributos a eles associados presentes nas análises. Um design bem elaborado contribui significativamente para a manutenção, extensibilidade e qualidade do produto de software (Gomaa, 2011).

O elemento de classe 'Código e Teste Unitário', descrito no Quadro 22, exibiu 33,33% de menções de atributos a eles associados. Myers, Sandler e Badgett (2011) entendem que a codificação e os testes unitários representam etapas a serem consideradas no processo de desenvolvimento, assegurando a funcionalidade e a confiabilidade do software.

Dando sequência às análises, o Quadro 23 apresenta as evidências obtidas sobre os elementos da classe 'engenharia de produto' e realça a interdependência e a importância de múltiplos aspectos do ambiente de desenvolvimento. Iniciando com o elemento de classe "Processo de Desenvolvimento", houve a menção a 80,00% dos atributos a ele associados. Humphrey (1989) argumenta que a adoção de processos de desenvolvimento bem definidos e sistemáticos contribui para o sucesso do projeto de software.

Quadro 23 - Elementos da classe Ambiente de Desenvolvimento

Elemento de classe	Total de Atributos	Total de Atributos Identificados	%
Processo de Desenvolvimento	5	4	80,00
Sistema de Desenvolvimento	7	1	14,28
Processo de Gestão	4	2	50,00
Métodos de Gestão	4	2	50,00
Ambiente de Trabalho	4	3	75,00

Fonte: o autor, a partir de dados da pesquisa, considerando resultados do Quadro 20.

Legenda: '%' indica o percentual de atributos identificados do elemento de classe.

O elemento de classe "Ambiente de Trabalho", descrito no Quadro 23, obteve 75,00% de seus atributos citados pelos entrevistados. Brooks (1975), em seu estudo, havia enfatizado como um ambiente de trabalho adequado pode ser tão importante quanto as ferramentas de desenvolvimento na promoção da criatividade, da motivação e da eficácia das equipes.

No que diz respeito ao "Processo de Gestão" e aos "Métodos de Gestão" citados no Quadro 23, ambos obtiveram uma menção de 50,00% dos atributos a eles associados. Conforme Kerzner (2013), a gestão de projetos não se limita ao controle de tarefas, mas abrange a liderança, a comunicação e a resolução de conflitos, aspectos que são considerados para a navegação bem-sucedida das complexidades do desenvolvimento de software.

Em complemento, o elemento de classe "Sistema de Desenvolvimento", descrito no Quadro 23, apresentou 14,28% dos atributos a ele vinculados como presentes. Esse achado, obtido a partir da percepção dos entrevistados, não é consistente com o que o estudo de Allen (2002) discute sobre a necessidade de instalações e sistemas que não apenas suportem, mas também potencializem a produtividade e a colaboração entre as equipes.

Ao analisar o Quadro 24 sobre os elementos da classe "Restrições do Programa", observamos que esta classe abrange aspectos fundamentais que influenciam o desenvolvimento e a entrega de um projeto. Primeiramente, o elemento de classe "Recursos" exibiu 75,00% dos atributos a ele associados. Como ressaltado por PMI (2017) no "PMBOK® Guide", a gestão de recursos é um pilar fundamental na gestão de projetos, enfatizando a importância de otimizar o uso de recursos para alcançar os melhores resultados.

Quadro 24 - Elementos da classe Restrições do Programa

Elemento de classe	Total de Atributos	Total de Atributos Identificados	%
Recursos	4	3	75,00
Contrato	3	2	66,66
Interfaces do Programa	7	5	71,42

Fonte: o autor, a partir de dados da pesquisa, considerando resultados do Quadro 20.

Legenda: '%' indica o percentual de atributos identificados do elemento de classe.

O elemento de classe "Contrato", citado no Quadro 24, obteve 66,66% de seus atributos mencionados pelos entrevistados. Esse elemento de classe ressalta a importância das obrigações contratuais e das especificações que devem ser atendidas. Kerzner (2013) destaca que a gestão contratual eficaz é essencial para minimizar riscos e assegurar que ambas as partes cumpram suas responsabilidades.

Por fim, as "Interfaces do Programa" mostrado no Quadro 24, com 71,42% de menções, sublinham a importância da integração e da interoperabilidade entre diferentes sistemas, componentes ou departamentos envolvidos no projeto. Segundo Sommerville (2019), a engenharia de software moderna dá grande ênfase à gestão

de interfaces, reconhecendo que a complexidade das interações entre os componentes do sistema é um desafio significativo no desenvolvimento de software.

Por fim, as análises dos elementos de classe permitiram afirmar que as menções a atributos a eles vinculados não foi uniforme em cada classe, considerando os dados previamente coletados. Para os elementos de classe vinculados à classe 'Restrições do Programa', contudo, houve certo predomínio das menções aos atributos a eles vinculados, considerando os percentuais acima de 66,66 % citados no Quadro 24. A comparação entre as ocorrências dos riscos considerando essas classes é detalhada na Seção 4.4, a seguir.

4.4 Comparação da ocorrência dos fatores de risco verificados segundo as classes (Associado ao objetivo específico 4)

Esta seção dos resultados apresenta a comparação entre as ocorrências dos riscos verificados segundo as classes da Taxonomia SEI, com base na literatura. Para desenvolver a argumentação, inicialmente foram apurados os atributos presentes em todos os elementos de classe de cada classe definida na Taxonomia SEI e descrita no Quadro 10 desse estudo: 1. 'Engenharia de Produto'; 2. 'Ambiente de Desenvolvimento' e 3. 'Restrições do Programa'. Em seguida, foram exibidas as comparações entre as classes, considerando o total de atributos identificados e o total de atributos identificados por classe segundo Taxonomia SEI. Essa comparação das classes entre si considerou a literatura descrita no Capítulo 2 deste trabalho, especialmente o estudo de Menezes et al. (2019) e a citada Taxonomia SEI.

O Quadro 25 apresenta os resultados das análises a partir das classes presentes na Taxonomia SEI. Em termos do total de 39 atributos distribuídos entre as três classes, a classe "Engenharia de Produto" apresentou com 43,58% dos atributos identificados, seguida pelo "Ambiente de Desenvolvimento" com 30,76% dos atributos identificados e as "Restrições do Programa" com 25,64%. A análise revela uma conscientização dos entrevistados em relação aos atributos associados à "Engenharia de Produto" e "Ambiente de Desenvolvimento", demonstrando a importância dessas áreas no sucesso do desenvolvimento de software. Esses dados destacam a necessidade de uma gestão de risco que considere todas as classes de riscos para minimizar impactos negativos no desenvolvimento de softwares provedores do programa social Alfa.

Quadro 25 – Análise a partir das classes presentes na Taxonomia SEI

Classe	Total de Atributos da classe	Total de Atributos Identificados	% da Classe¹	% Total²
Engenharia de produto	26	17	65,38	43,58
Ambiente de Desenvolvimento	24	12	50,00	30,76
Restrições do Programa	14	10	71,42	25,64

Fonte: o autor a partir dos dados da pesquisa, considerando resultados dos Quadros 23 a 25.

Legendas: 1. '% da Classe' – indica o percentual de atributos identificados da classe.

2. '% Total' – indica o percentual de atributos identificados considerando o total de 39 atributos identificados em todas as classes.

Na classe "Engenharia de Produto", citada no Quadro 25, a análise qualitativa dos fatores de risco aponta para desafios comuns e distintos entre os dados trazidos por Menezes et al. (2019) e das entrevistas. O estudo de Menezes et al. (2019) identifica a ambiguidade de requisitos como o atributo mais prevalente, seguido por mudanças de requisitos e a introdução de novas tecnologias. Estes aspectos destacam a dinâmica incerta e em constante mudança do desenvolvimento de software, onde as necessidades do cliente e as soluções tecnológicas evoluem rapidamente.

A análise ainda revela que tanto os resultados exibidos na Seção 4.2 deste estudo quanto aqueles exibidos por Menezes et al. (2019) tratam a centralidade dos requisitos e dos testes no desenvolvimento de software, mas diferem em como esses e outros fatores de risco são ponderados e relatados. Isso pode refletir diferenças contextuais nos projetos examinados.

Em seguida foram analisados os dados relacionados a classe 'Ambiente de Desenvolvimento', citada no Quadro 25. No trabalho de Menezes et al. (2019), os riscos mais citados incluem a falta de planejamento ou planejamento inadequado, baixo comprometimento da equipe, disciplina e padronização insuficiente, objetivos do projeto não claros, e comunicações ineficazes entre os membros da equipe. Estes fatores destacam uma preocupação com a estrutura organizacional e a clareza de comunicação dentro da equipe de desenvolvimento.

Por fim, foram analisados os resultados relativos à classe "Restrições do Programa" mencionada no Quadro 25, que encapsula fatores externos que influenciam o desenvolvimento do software. Estes fatores abrangem as limitações impostas por recursos, os termos contratuais do projeto, e a necessidade de interações com clientes, outros contratantes, a gestão corporativa e fornecedores

(Jones, 1994). Eles refletem as condições contratuais, organizacionais e operacionais sob as quais o software é desenvolvido, representando os elementos externos ao projeto que podem afetar seu sucesso.

A classe 'Restrições do Programa' ressaltou fatores externos como cronograma, pessoal e contratos, com 100% das menções ao "Cronograma" destacando a pressão externa para desenvolver rapidamente os softwares provedores do programa social 'Alfa' em resposta à emergência sanitária. Conforme discutido pelo PMI (2017) e Kerzner (2013). As interações com clientes e contratados, assim como as restrições de instalações e fatores políticos, também apresentaram obstáculos, ilustrando a complexa rede de influências externas que afetaram o projeto.

O desenvolvimento dos softwares provedores do programa social 'Alfa' durante a pandemia trouxe à tona desafios únicos, refletidos nas classes "Engenharia de Produto", "Ambiente de Desenvolvimento", e "Restrições do Programa", conforme detalhado nas entrevistas. A classe "Engenharia de Produto" ressaltou a dificuldade na definição de requisitos em um cenário volátil, alinhando-se com as observações de Wiegers e Beatty (2013) sobre a essencialidade de requisitos claros e adaptáveis. A classe "Ambiente de Desenvolvimento" indicou os desafios de planejamento e adaptação a novas tecnologias e metodologias, conforme discutido por Highsmith (2009) e Brooks (1995). Questões como capacidade, gestão de pessoal, garantia de qualidade e comunicação também foram enfatizadas, ressaltando a necessidade de manter padrões de qualidade e cooperação em meio ao trabalho remoto. Esses desafios, contextualizados pelas circunstâncias excepcionais da pandemia, sublinham a importância da adaptabilidade e da gestão eficaz de riscos em projetos de desenvolvimento de software, reforçando a necessidade de uma abordagem flexível e dinâmica para enfrentar condições imprevisíveis e em rápida mudança.

Além do que foi exposto nesta seção previamente, foi possível, a partir do Quadro 25, efetuar uma análise comparativa das classes considerando os atributos presentes na Taxonomia SEI constantes da literatura e referentes a cada classe, citados no Quadro 11 deste estudo. Verifica-se que a classe "Restrições do Programa" é composta por 14 atributos associados a seus respectivos elementos de classe, com 10 atributos identificados durante as entrevistas, o que equivale a 71,42% do total de atributos possíveis de serem identificados. Em adição, a classe "Engenharia de Produto" conta com um total de 26 atributos a ela associados por meio de seus elementos de classe, dos quais 17 foram identificados nas entrevistas, representando 65,38% do total possível de ser identificado para esta classe. Por fim, a classe "Ambiente de Desenvolvimento" apresenta o total de 24 atributos a ela associados por

meio de seus elementos de classe, com 12 atributos identificados nas entrevistas, correspondendo a 50,00% do total possível de ser identificado.

A análise dos atributos identificados nas diferentes classes, conforme apresentado no Quadro 25, relata questões importantes sobre a aderência à taxonomia do SEI. A classe "Restrições do Programa", apesar de compor apenas 25,64% do total de atributos identificados, exibe uma identificação de 71,42% de atributos específicos vinculados a essa classe. Esta proporção contrasta com outras classes como "Engenharia de Produto" e "Ambiente de Desenvolvimento", sugerindo uma aderência particularmente robusta às restrições definidas pela Taxonomia SEI. Tal concentração de atributos nas "Restrições do Programa" pode ser interpretada como a influência externa de fatores, fora da gestão da Empresa BETA (Sommerville, 2011).

A comparação entre as classes "Engenharia de Produto", "Ambiente de Desenvolvimento", e "Restrições do Programa", inspirada na Taxonomia SEI e reforçada pelas constatações do estudo de Menezes et al. (2019), destaca a importância de uma gestão ampla de projetos de software. Menezes et al. (2019), enfatizam a necessidade de um entendimento aprofundado dos riscos associados a cada aspecto do desenvolvimento de software, desde a engenharia de produto até as restrições impostas pelo programa. Este entendimento permite uma abordagem mais eficaz na mitigação de riscos e na adaptação às mudanças, essenciais em um contexto de crise como o enfrentado durante a pandemia.

A análise das evidências comparando as classes revelou que, enquanto "Engenharia de Produto" e "Ambiente de Desenvolvimento" atrelaram-se a atributos que sinalizaram desafios internos relacionados à definição de requisitos, adaptação a novas tecnologias e garantia de qualidade, a classe "Restrições do Programa" destacou a influência de fatores externos. Estes fatores externos incluem não apenas restrições de cronograma e recursos, mas também a complexidade das interações com stakeholders externos e o impacto de fatores políticos e sociais.

A identificação de atributos na classe "Restrições do Programa" aderentes à Taxonomia SEI, conforme previamente ilustrado no Quadro 25 e previamente citado nesta seção, sublinha a importância de uma abordagem de desenvolvimento de software que deve considerar o reconhecimento de que o sucesso do desenvolvimento de software não se baseia exclusivamente nas habilidades técnicas internas e na gestão de projetos, mas também na habilidade de gerenciar efetivamente as restrições impostas pelo ambiente externo. Reconhece-se que, enquanto os fatores internos do projeto podem ser gerenciados até certo grau, os

fatores externos exigem uma abordagem proativa para antecipação, adaptação e mitigação de riscos, assegurando a continuidade e sucesso do desenvolvimento do software (Menezes et al., 2019).

Assim, a gestão de projetos de desenvolvimento de software de programas sociais requer uma atenção a fatores internos e externos, apontando para a necessidade de estratégias de gestão de riscos abrangentes que considerem a totalidade do ambiente de desenvolvimento. Portanto, integrar as lições aprendidas com a experiência dos softwares provedores do programa social ALFA, as recomendações de Menezes et al. (2019) no entendimento da gestão de projetos de software e as evidências obtidas segundo a aderência a atributos da Taxonomia SEI não apenas enriquece a abordagem à identificação de riscos, mas também prepara melhor as equipes de desenvolvimento para navegar na incerteza e alcançar sucesso em projetos futuros.

5 CONSIDERAÇÕES FINAIS

Este estudo teve como objetivo geral identificar os principais fatores de riscos associados às etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa. A pesquisa, de natureza descritiva, seguiu uma abordagem qualitativa fundamentada na coleta de entrevistas e documentos e análise dos dados por meio de análise de conteúdo, considerando a taxonomia de risco do SEI, pontuada por elementos adicionais da revisão de literatura. O alcance do objetivo geral e de quatro objetivos específicos a ele vinculados estão descritos na sequência.

O primeiro objetivo específico foi descrever as etapas do processo de desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa. Efetuou-se análise documental das 23 atividades do processo de desenvolvimento e Manutenção de Software da Empresa BETA, denominado PD-BETA, conforme citado na Seção 4.1 deste estudo. O resultado demonstrou que das 23 atividades descritas, 7 delas foram identificadas. Essas atividades identificadas delimitaram o foco das análises que contribuíram para o atingimento dos demais objetivos específicos.

O segundo objetivo enfatizou a verificação dos principais fatores de riscos associados ao processo de desenvolvimento de softwares previamente descrito, considerando os atributos da Taxonomia SEI. Os resultados do atingimento deste objetivo específico estão detalhados na Seção 4.2 deste trabalho. Foram identificados 39 atributos a partir de um total de 64 possíveis citados na referida taxonomia. Cada um dos atributos foi verificado a partir das entrevistas e confrontados com a análise do corpus de documentos não sigilosos coletados para os fins deste estudo, complementado pelo exame da literatura. Verificou-se que entre os atributos presentes, o atributo 'Cronograma' foi identificado no relato de 100,00% dos entrevistados.

Em seguida, o terceiro objetivo buscou classificar os fatores riscos previamente verificados de acordo com elementos de classe da referida Taxonomia SEI. As análises dos elementos de classe, descritas na Seção 4.3 deste estudo, permitiram afirmar que as menções a atributos a eles vinculados não foi uniforme em cada classe, considerando os dados analisados. Para os elementos de classe vinculados à classe 'Restrições do Programa', contudo, houve certo predomínio das menções aos atributos a ele vinculados com percentuais acima de 65,00%.

O quarto e último objetivo comparou a ocorrência dos fatores riscos identificados segundo as classes da Taxonomia SEI, com base na literatura. Os

resultados descritos na Seção 4.4 deste estudo mostraram que do total de 39 atributos distribuídos entre as três classes, a classe "Engenharia de Produto" apresentou 17 dos atributos identificados (43,58% do total de atributos), seguida pelo "Ambiente de Desenvolvimento" com 13 dos atributos identificados (30,76% do total de atributos) e a classe "Restrições do Programa" com 10 dos atributos identificados (25,64% do total de atributos).

Adicionalmente, com relação ao 4º objetivo específico, verificou-se que a classe "Restrições do Programa" apresentou 71,42% do total de atributos possíveis de serem identificados com base na Taxonomia SEI. A análise revela a importância de uma abordagem de desenvolvimento de software que deve considerar o/ sucesso do desenvolvimento de software não baseado exclusivamente nas habilidades técnicas internas e na gestão de projetos, mas também na habilidade de gerenciar efetivamente as restrições impostas pelo ambiente externo, inclusive o relacionamento com partes interessadas.

O alcance dos quatro objetivos específicos possibilitou atingir o objetivo geral e responder o problema de pesquisa. Após descrever as 7 atividades referentes às etapas de desenvolvimento de softwares provedores do referido programa social, foi possível, com base na Taxonomia SEI, identificar 39 atributos, sendo 17 vinculados à classe 'Engenharia de Produto' 12 vinculados à classe 'Ambiente de Desenvolvimento' e 10 vinculados à classe 'Restrições do Programa'. O exame dos atributos identificados enfatizou a classe 'Restrições do Programa', mostrando a influência de fatores externos no processo de desenvolvimento de software, em consonância com os achados de estudos anteriores e com a referida Taxonomia SEI. Adicionalmente, as análises permitiram detalhar os atributos segundo elementos de classe, destacando-se a identificação do atributo 'Cronograma', vinculado ao elemento de classe 'Recursos'.

A distribuição das classes de riscos, citada no parágrafo anterior, sublinha a importância de abordagens específicas de mitigação, particularmente no que tange às restrições impostas por fatores externos, ressaltando a necessidade de estratégias de gestão de riscos adaptativas e bem-informadas. Os achados deste estudo, também, permitem que sejam apresentadas contribuições para o aperfeiçoamento do processo de desenvolvimento de software na Empresa BETA com foco na gestão de risco:

- Implementação de Práticas Estruturadas de Gerenciamento de Riscos: A adoção de um framework estruturado para o gerenciamento de riscos, baseado na taxonomia do SEI, pode permitir à empresa identificar proativamente, avaliar e mitigar os riscos associados ao desenvolvimento

de software. Isso inclui a implementação de ferramentas de monitoramento e análise de riscos, bem como a capacitação da equipe em práticas de gerenciamento de riscos.

- Aprimoramento do Processo de Elicitação de Requisitos: A melhoria contínua dos métodos de elicitação de requisitos, enfatizando a importância da comunicação efetiva com os stakeholders e a utilização de técnicas de documentação claras e precisas, pode ajudar a garantir que os requisitos coletados reflitam as necessidades reais dos usuários finais e reduzam a possibilidade de ambiguidades ou mal-entendidos.
- Fortalecimento das Estratégias de Teste e Validação: Desenvolver e implementar uma estratégia abrangente de teste e validação que aborde desde testes unitários até integração e testes de aceitação do usuário. Isso inclui a adoção de automação de testes onde for aplicável, para aumentar a eficiência dos testes e garantir que o software atenda aos padrões de qualidade exigidos antes do lançamento.
- Fomento de um Ambiente de Desenvolvimento Ágil: Encorajar a adoção de metodologias ágeis de desenvolvimento de software para promover a flexibilidade, a colaboração entre as equipes e a capacidade de resposta rápida a mudanças nos requisitos. Isso pode envolver a reestruturação das equipes de desenvolvimento, a implementação de ciclos de feedback regulares com os stakeholders e a capacitação da equipe em práticas ágeis.
- Considerar as influências de fatores internos, como condições dadas para adaptação da equipe a regimes de trabalho presencial e remoto, bem como o exame de fatores externos, especialmente a possibilidade de articulação e acordos que promovam uma relação contratual transparente entre parceiros e demais stakeholders envolvidos nas etapas do processo de desenvolvimento de software.

As conclusões deste estudo oferecem contribuições para a prática e a teoria da gestão de riscos em desenvolvimento de softwares, especialmente ligados ao atendimento de programas sociais. A identificação e classificação dos riscos conforme a taxonomia do SEI fornecem oportunidades para profissionais e pesquisadores, sugerindo a necessidade de abordagens de gestão de riscos que sejam tanto abrangentes quanto específicas para cada fase do desenvolvimento desses softwares.

É importante destacar que, embora o desenvolvimento de softwares provedores do Programa Social Brasileiro Alfa tenha seguido quatro etapas essenciais do SDLC, a ausência de atividades sugeridas por Abioye et al. (2020), como avaliações mais rigorosas de qualidade, gestão de mudanças aprimorada e estratégias focadas na satisfação do cliente, pode ter introduzido riscos adicionais relacionados à qualidade do software, gerenciamento de mudanças e satisfação do cliente. Esse fato sublinha a importância de uma abordagem de desenvolvimento abrangente que considere todos os aspectos críticos para minimizar riscos e aumentar a eficácia dos softwares desenvolvidos, contribuindo, assim, para reflexões que busquem a melhoria da qualidade e sucesso destes projetos em contextos similares.

No que tange à confiabilidade da pesquisa, deve-se destacar que roteiro de entrevista de pesquisa foi elaborado e validado por meio teste piloto. Ademais, os apêndices A, D e E formalizam o distanciamento do pesquisador do contexto investigado. Os convites para participação foram enviados de forma transparente e os potenciais entrevistados tiveram a liberdade de recusar o convite, o que contribui para garantir a ética e a integridade da pesquisa. Os procedimentos adotados seguiram passos que podem ser replicados por outros pesquisadores.

Entre as limitações desta pesquisa, existiram aspectos fora do controle do autor deste trabalho, como o tempo decorrido entre o pedido e análise de autorização para realização da coleta de dados, o tempo para autorização e realização das entrevistas, o período de greve ao qual atravessou a Empresa BETA, lócus da pesquisa. Adicionalmente, considerando o teor do retorno da autorização do pedido de coleta de dados e a consulta às regras do programa de pós-graduação, optou-se pela anonimização do nome da empresa, dos softwares provedores e do programa social. No caso do pedido de autorização à Empresa Beta (Apêndice A), de coleta de dados por meio de entrevista (Apêndice D) e do teste piloto (Apêndice E) optou-se por tarjar o nome da empresa e o cargo ocupado pelo pesquisador. As limitações descritas neste parágrafo não restringiram o atingimento do objetivo geral proposto neste estudo, mas impactaram o andamento do cronograma da pesquisa, acarretando ajustes pontuais que se fizeram necessários.

É adequado citar que a análise e a profundidade da investigação do caso em questão consideraram a triangulação dos dados coletados, fato que ajuda promover a confiabilidade e a mitigar eventuais limitações apontadas em pesquisas qualitativas. Em complemento, é adequado lembrar que as conclusões deste estudo estão circunscritas ao caso estudado, considerando os dados de entrevistas e documentos

que foram coletados. Entretanto, os achados do estudo, também, podem fornecer subsídios para auxiliar a formulação de argumentos propostos em outras pesquisas.

Nesse sentido, sugere-se, para estudos futuros, a expansão desta pesquisa para incluir uma gama mais ampla de projetos de software abrangendo outros programas sociais, assim como uma análise mais detalhada sobre o tratamento dos riscos identificados. Além disso, seria proveitoso explorar a eficácia de diferentes estratégias de mitigação de riscos, adaptadas às especificidades de cada fase do processo de desenvolvimento de software.

Adicionalmente, pode-se considerar, como sugestão de estudo futuro, a ampliação deste estudo em outras empresas ou outros órgãos governamentais envolvidos em projetos de desenvolvimento de software específicos. Isso permitiria uma comparação entre organizações dos riscos identificados, avaliando como diferentes organizações públicas atuam, considerando inclusive a influência de fatores internos e externos que seriam capazes de afetar o gerenciamento dos riscos no desenvolvimento de softwares. Tal abordagem poderia, inclusive, revelar padrões ou práticas de mitigação de riscos identificados que são eficazes, oferecendo alternativas para aprimorar as estratégias de gerenciamento de riscos em projetos de software de uma forma mais generalizada.

Avaliar a transparência da relação contratual estabelecida entre colaboradores internos e entre partes interessadas, também, é mais uma oportunidade de pesquisa. Assim, análises sobre articulações e acordos entre parceiros num processo de desenvolvimento de software, bem como o grau de transparência estabelecido a tais acordos, podem servir como oportunidades de pesquisa para aprimorar as estratégias de mitigação de riscos e proporcionar a implementação de processos de desenvolvimento de softwares mais eficientes.

Este estudo, ao abordar a gestão de riscos no desenvolvimento de softwares do Programa Social Brasileiro Alfa, não apenas atingiu seus objetivos propostos, mas também abriu caminhos para aprofundar o entendimento e a eficácia da gestão de riscos em projetos de desenvolvimento de software em programas sociais, contribuindo para reflexões que busquem a melhoria da qualidade e sucesso destes projetos em contextos similares, tanto no País quanto no Exterior.

REFERÊNCIAS

ABDUL KARIM, N. S.; ALBUOLAYAN, A.; SABA, T.; REHMAN, A. **The Practice of Secure Software Development in SDLC: An Investigation Through Existing Model and a Case Study.** *Security and Communication Networks*, v. 9, n. 18, p. 5333–5345, 2016.

ABIOYE, T. E.; AROGUNDADE, O. T.; MISRA, S.; AKINWALE, A. T.; ADENIRAN, O. J. **Toward ontology-based risk management framework for software projects: An empirical study.** *Journal of Software: Evolution and Process*, v. 32, n. 12, p. 1–24, 2020.

ABNT. **ISO 31000 Gestão de Riscos.** ABNT Associação Brasileira de Normas Técnicas, p. 17, 2018.

ABRAH, J. **Política Social no Brasil: marco conceitual e análise da ampliação do escopo, escala e.**, p. 66–95, 2012.

ABUNADI, I.; ALENEZI, M. **An Empirical Investigation of Security Vulnerabilities within Web Applications.** *Journal of Universal Computer Science*, v. 22, n. 4, p. 537–551, 2016.

ALBADARNEH, A.; ALBADARNEH, I.; QUSEF, A. **Risk management in agile software development: A comparative study.** *Proc. IEEE Jordan Conf. Applied Electrical Engineering and Computing Technologies. Anais...*, 2015.

ALBERTO, J. D. B. F. **Atlântica – Instituto Universitário Criação de um Modelo de Análise de riscos a implementar em Projetos de 1–133,** 2021.

ALENEZI, M.; ALMUAIRFI, S. **Security Risks in the Software Development Lifecycle,** n. 3, p. 7048–7055, 2019a.

ALENEZI, M.; ALMUAIRFI, S. **Security risks in the software development lifecycle.** *International Journal of Recent Technology and Engineering*, v. 8, n. 3, p. 7048–7055, 2019b.

ALHARBI, E. T.; QURESHI, M. R. J. **Implementation of risk management with SCRUM to achieve CMMI requirements.** *Int. J. Comput. Netw. Inf. Secur.*, v. 6, n. 11, p. 20–25, 2014.

AMARAL, D. C.; CONFORTO, E. C.; BENASSI, J. L. C.; ARAÚJO, C. **Gerenciamento ágil de projetos - aplicação em produtos inovadores.** São Paulo: Saraiva, 2011.

ANDRADE, A. C. DA S.; BRAGA, J. L.; LEAL, A. L. DE C.; ZAIDAN, F. H. **Gestão de riscos em projetos de software: uma abordagem baseada em requisitos não funcionais.** *Sistemas & Gestão*, v. 14, n. 2, p. 188–196, 2019.

ARBACHE, J. S. **Pobreza e mercados no Brasil: uma análise de iniciativas de políticas públicas.** Brasília: Cepal, DFID, 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, A. **ABNT NBR 31010:2012 - Gestão de riscos - Técnicas de avaliação de riscos.** Rio de J, 2012.

- BARDIN, L. **Análise de conteúdo**. Lisboa: Edições 70, 1977.
- BASS, L.; WEBER, I.; ZHU, L. DevOps: **A Software Architect's Perspective**. Westford, Massachusetts: Addison-Wesley, 2015.
- BECK, K.; BEEDLE, M.; BENNEKUM, A. VAN; et al. **Manifesto for Agile Software Development**, 2001.
- BELEI, R. A.; GIMENEZ-PASCHOAL, S. R.; NASCIMENTO, E. N.; MATSUMOTO, P. H. V. R. O uso de entrevista, observação e videogravação em pesquisa qualitativa. **Cadernos de Educação**, n. 30, p. 187–199, 2008.
- BERNSTEIN, P. **Desafio aos deuses: a fascinante história do risco**. Rio de Janeiro: Campus, 1997.
- BISHT, P.; HEIM, M.; IFLAND, M.; SCOVETTA, M.; SKINNER, T. **Managing security risks inherent in the use of third-party components**, 2017.
- BOEHM, B. W. Software risk management: Principles and practices. **IEEE Software**, v. 8, n. 1, p. 32–41, 1991.
- BOEHM, B. W. **A spiral model of software development and enhancement**. **Computer**, 1988.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. UML: **Guia do Usuário**. São Paulo: Elsevier, 2006.
- BRASIL. **Conselho Monetário Nacional**. RESOLUÇÃO N° 3380., p. 4–7, 2006.
- BRASIL. **Lei Complementar nº 7, de 7 de setembro de 1970**. Brasília, DF, set. 1970.
- BRASIL. **Lei nº 6.125, de 4 de novembro de 1974**. Brasília, DF, nov. 1974.
- BRASIL. **Lei nº 8.742, de 7 de dezembro de 1993**. Brasília, DF, dez. 1993.
- BRASIL. **Lei nº 10.420, de 10 de abril de 2002**. Brasília, DF, abr. 2002.
- BRASIL. **Lei nº 10.741, de 1 de outubro de 2003**. Brasília, DF, out. 2003a.
- BRASIL. **Lei nº 10.779, de 25 de novembro de 2003**. Brasília, DF, nov. 2003b.
- BRASIL. **Decreto nº 4.962, de 22 de janeiro de 2004**. Brasília, DF, jan. 2004.
- BRASIL. **Decreto nº 9.203, de 22 de novembro de 2017**. Brasília, DF, p. 1-7, 2018.
- BRASIL. **Decreto nº 10.316, de 7 de abril de 2020**. Brasília, DF, 2020.
- BRASIL. **Lei nº 14.237, de 19 de novembro de 2021**. Brasília, DF, nov. 2021a.
- BRASIL. **Lei nº 14.284, de 29 de dezembro de 2021**. Brasília, DF, dez. 2021b.
- BRASIL. **MPV 1162/2023 Inteiro teor**. Brasília, DF, 2023a.

BRASIL. **Medida Provisória nº 1.164, de 2 de março de 2023**. Brasília, DF, mar. 2023b.

BROWN, Y. F., CANAL, K., CULLEN R., ELLIOTT, M., FAULKNER, J., LACKNER, M., OWENS, C., **PEERCY, D., REISZ, G., TEMPEL, P.; TRELLUE, P. Software risk management: A practical guide**. Department of energy quality managers software quality assurance subcommittee reference document SQAS21.01.00. Software risk management: A practical guide. Department of energy quality managers software quality assurance subcommittee reference document SQAS21.01.00., 2000.

CANHOTA, C. **Qual a importância do estudo piloto?** In: E. E. Silva (Org.); *Investigação passo a passo: perguntas e respostas para investigação clínica*. p.69–72, 2008. Lisboa: APMCG.

CARR, C. N. **Estrutura de Governança de T.I:** Aprimorando a eficiência e eficácia com o modelo COBIT como framework., p. 1–12, 2023.

Carr M. J., Konda S.L, Monarch I., Ulrich F.C., Walker C.F. **"Taxonomy-Based Risk Identification" Technical Report CMU/SEI-93-TR-6, Software Engineering Institute**, Carnegie Mellon University, EUA, July, 1993.

DE CARVALHO, M. M.; RABECHINI JUNIOR, R. Impact of risk management on project performance: The importance of soft skills. **International Journal of Production Research**, v. 53, n. 2, p. 321–340, 2015. Taylor e Francis. Disponível em: <<http://dx.doi.org/10.1080/00207543.2014.919423>>.

CASTRO, J. A. DE. Política social e desenvolvimento no Brasil., p. 1011–1042, 2012. CAVALCANTE, P. L. C.; LOTTA, G. S.; YAMADA, E. M. K. Exploring mid-level bureaucracy: a tentative typology. **Revista Brasileira de Ciência Política**, n. 26, p. 187–222, 2018.

CEC, A.; ZABEU, P.; JOMORI, S. M.; LUIZ, R.; VOLPE, D. Desenvolvimento de software., p. 66–68, 2006.

CERVO, A. L.; BERVIAN, P. A. **Metodologia científica**. 5o ed. São Paulo: Prentice Hall, 2002.

CEZERINO, A.; PAES NASCIMENTO, F. Utilização da Técnica de Desenvolvimento Orientado Por Comportamento (Bdd) No Levantamento De Requisitos. **Revista Interdisciplinar Científica Aplicada**, v. 10, n. 3, p. 40–51, 2016.

CHAPMAN, R. J. **The rules of project risk management: implementation guidelines for major projects**. London: Routledge, 2016.

CHITTISTER, C.; HAIMES, Y. Y. **Risk Associated with Software Development: A Holistic Framework for Assessment and Management**., v. 23, n. 3, p. 710–723, 1993.

CHITTISTER, C.; KIRKPATRICK, R.; VANSKOY, R. **Gerenciamento de riscos na prática. Amer. Programador**, v. 5, n. 7, 1992.

CMMI INSTITUTE. **Capability Maturity Model Integration**. 2018. Pittsburgh, PA. Disponível em: <<https://cmmiinstitute.com/cmmi>>. Acesso em: 30/9/2021.

CORVALÁN, J. G. Administración Pública digital e inteligente: transformaciones en la era de la inteligencia artificial. **Revista de Direito Econômico e Socioambiental**, p. 26–66, maio 2017. Curitiba.

COSTA, E. M. V. DA. **Processo de gestão de riscos na análise de demanda de novos softwares**, 2022.

DA, E. M. T.; CUBAS, M. R.; ALVAREZ, A. M. **Enfermagem gerontológica no cuidado ao idoso em tempos da 3**. 2022.

DAVENPORT, T. H. Process Innovation: Reengineering Work Through Information Technology. **1st ed. Boston**, MA, USA: Harvard Business Press, 1993.

DAVIS, J.; DANIELS, R. Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale. **Sebastopol**, California: O'Reilly Media, 2016.

DEVANBU, P. T.; STUBBLEBINE, S. Software Engineering for Security: A Roadmap. Proceedings of the Conference on the Future of Software Engineering. **Anais....** p.227–239, 2000.

DICIO. **Dicionário Escolar da Língua Portuguesa**. Disponível em: <<https://www.dicio.com.br/risco/>>. Acesso em: 3/7/2023.

DIRGAHAYU, T.; PRAYUDI, Y.; FAJARYANTO, A. Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server. **Jurnal Ilmiah NERO**, v. 1, n. 3, p. 190–197, 2015. Disponível em: <<http://nero.trunojoyo.ac.id/index.php/nero/article/download/29/27>>.

DU S., KEIL M., MATHIASSEN I., SHEN Y., & T. A. Attention-shaping tools, expertise, and perceived control in IT project risk assessment. **Decision Support Systems**, v. 43, n. 1, p. 269–283, 2010.

EBERT, C. Cyber Security Requirements Engineering. In: M. Ramachandran; Z. Mahmood (Orgs.); Requirements Engineering for Service and Cloud Computing. p.209–228, 2017. **Cham: Springer International Publishing**. Disponível em: <https://doi.org/10.1007/978-3-319-51310-2_9>.

EDUARDO, F. **Revista Boletim do Gerenciamento de Riscos: Uma Comparação entre o Guia PMBOK 6a Edição e a ISO 31000: 2018.**, v. 4, p. 22–32, 2019.

FALQUETO, J. M. Z.; HOFFMANN, V. E.; FARIAS, J. S. Saturação teórica em pesquisas qualitativas: relato de uma experiência de aplicação em estudo na área de administração. **Revista de Ciências da Administração**, v. 20, n. 52, p. 40–53, 2018.

FAN, C. F.; YU, Y. C. BBN-based software project risk management. **Journal of Systems and Software**, v. 73, n. 2, p. 193–203, 2004.

FERNANDES LIMA, A. C. **Ferramenta de gestão de riscos aplicada a ambientes de desenvolvimento de software com foco na garantia da qualidade do produto**, 2019.

FERREIRA, F. H.; ET AL. Ascensão e queda na desigualdade de renda no Brasil. **Econômica**, v. 8, n. 1, 2006.

FERRO, D. DOS S. **Gestão De Riscos Corporativos**: Um Estudo Multicaso Sobre Seus Métodos e Técnicas, n. 1, p. 1–119, 2015.

FILIPPETTO, A. S.; LIMA, R.; LUIS, J. **A risk prediction model for software project management based on similarity analysis of context histories**. v. 131, n. November 2020, 2021.

FLEURY, S. A segurança social e os dilemas da inclusão social. **RAP**, v. 39, n. 3, p. 449–469, 2005.

FONG, E.; OKUN, V. Web application scanners: Definitions and functions. Proceedings of the 40th Hawaii International Conference on System Sciences. **Anais...**, 2007.

FURLAN, L. D. M.; PACHECO, A. S. V. Gestão de risco: estudo de caso sobre os desafios na implantação e implementação. **Revista Ibero-Americana de Estratégia**, v. 20, n. 1, p. e18818, 2021.

FURTADO, C. **Formação econômica do Brasil**. 23a ed. ed. São Paulo: Nacional, 1989.

GARCIA, F. W. S., OLIVEIRA, S. R. B., SALVIANO, C. F. Uma abordagem metodológica para a implementação multimodelos de qualidade de software adotando a CERTICS e o CMMI-DEV. **Revista Eletrônica de Sistemas de Informação**, v. 15, n. 2, p. 1–28, 2016.

GENTILINI, U.; ALMENFI, M.; DALE, P.; et al. Social Protection and Jobs Responses to COVID-19: A Real-Time Review of Country Measures. “Living paper” version 10. Washington, D.C.: **World Bank**. Social Protection and Jobs Global Practice, p. 1–907, 2020.

GHOZALI, B.; KUSRINI, K.; SUDARMAWAN, S. Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating. **Creative Information Technology Journal**, v. 4, 2019.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4aed. ed. São Paulo: Editora Atlas S.A, 2002.

GJERDRUM, B. D.; PETER, M. **Risk Management The New International Standard on the Practice of Risk**, n. 21, 2011.

GUPTA, S. **A Proactive Approach to Information Security**. 2003.

HADABRUS, B. **Taxonomy of Risks in Software Development Projects**. 2022 63rd International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS), p. 1–7, 2022. IEEE.

HALCOMB, E. J.; DAVIDSON, P. M. Is verbatim transcription of interview data always necessary? **Applied Nursing Research**, v. 19, n. 1, p. 38–42, 2006.

HALL, E. M. **Managing risk: methods for software systems development**. Pearson Education. 1998.

AL HAMED, T.; ALENEZI, M. Business Continuity Management & Disaster Recovery Capabilities in Saudi Arabia ICT Businesses. **International Journal of Hybrid Information Technology**, v. 9, n. 11, p. 99–126, 2016.

HANIF, H.; MD NASIR, M. H. N.; AB RAZAK, M. F.; FIRDAUS, A.; ANUAR, N. B. The rise of software vulnerability: Taxonomy of software vulnerabilities detection and machine learning approaches. **Journal of Network and Computer Applications**, v. 179, n. February, p. 103009, 2021. Elsevier Ltd. Disponível em: <<https://doi.org/10.1016/j.jnca.2021.103009>>.

HERNÁNDEZ SAMPIERI, R.; FERNÁNDEZ COLLADO, C.; BAPTISTA LUCIO, P. **Metodologia de pesquisa**. 5th ed. Porto Alegre: Penso, 2013.

HEYMANN, D. L.; SHINDO, N. **COVID-19: what is next for public health?** The Lancet, v. 395, n. 10224, p. 542–545, 2020.

HILLSON, D. **Risk management isn't hard**. Reino Unido, 2009.

HM TREASURY. **The Orange Book**. Nature Reviews Drug Discovery, v. 4, n. 8, p. 48, 2020.

HOFMAN, M.; GRELA, G. Project portfolio risk identification--application of Delphi method. **Journal of Business and Economics**, v. 6, n. 11, p. 1857–1867, 2015.

HUBBARD, D. **Failure of Risk Management: Why It's Broken and How to Fix It**. New Jersey (EUA): John Wiley & Sons, Inc, 2009.

HÜTTERMANN, M. **DevOps for Developers**. New York, New York: Apress, 2012.

IBGC, I. B. DE G. C. **Guia de orientação para gerenciamento de Riscos Corporativos**. 2007.

IEEE. **Guide to the Software Engineering Body of Knowledge Version 3.0 (SWEBOK Guide V3.0)**. 2014.

ISO/IEC. Tecnologia da informação — Técnicas de segurança — **Gestão de riscos de segurança da informação** Sumário., v. 27005, p. 12–24, 2011.

ISO/IEC 16085. **International Standard Iso / Iec / Ieee Iteh Standard Preview Iteh Standard Preview**, v. 2011, 2021.

ISO. NBR ISO / IEC 12207 **Tecnologia de informação - Processos de ciclo de vida de software.**, 2017.

ISO, S. I. O. FOR. ISO 9001:2015 **Quality management systems - Requirements**, 2015.

JOHN R. S. FRASER, ROB QUAIL, B. S. **Enterprise Risk Management: Today's Leading Research and Best Practices for Tomorrow's Executives**. 2o ed. 2021.

JOIA, L. A. A teoria da representação social e a definição de constructos na área de administração da informação. EnADI--Encontro Nacional de Administração da Informação--ANPAD. **Anais...** 2017. Curitiba, PR.

JONES, C. **Assessment and Control of Software Risks**. 1994.

JORION, P. **Value at Risk**. São Paulo: BMF, 1998.

JOSE, P.; ALVAREZ, M.; JR, R. R. **Gestão de riscos na abordagem ágil e o sucesso de projetos**. Risk management in the agile approach and project success gestión de riesgos en el abordaje ágil y el éxito de los proyectos., v. 23, p. 138–162, 2023.

JÚNIOR, W. M. P.; FERREIRA JÚNIOR, J. L. A.; SILVA, L. P. DA. Um estudo dos processos de ciclo de vida de software a partir da norma ISO 12207. **Intercursos Revista Científica**, v. 8, n. 2, p. 21–23, 2009.

KIELING, D. L.; MAGNAGNAGO, O. **FAG Journal of Health The importance of telemedicine in the COVID-19 pandemic context**, 2021.

KONDA, S. L.; ULRICH, F. C.; WALKER, C. F. **Taxonomy-Based Risk Identification**. January, 2014.

KUTSCH E. **The influence of intervening conditions on the over and underestimation of risk**. **International Journal of Managing Projects in Business**, v. 1, n. 4, p. 602–210, 2008.

LAVILLE, C.; DIONNE, J. **A construção do saber: manual de metodologia da pesquisa em ciências humanas**. Belo Horizonte: UFMG, 1999.

LEOPOLDINO, C. B. **Avaliação De Risco Em Desenvolvimento De Software**. Mestrado, 2019.

LÍCIO, E. C.; FILHO, A. C. C.; BARTHOLO, L.; PASSOS, L.; MESQUITA, A. C. S. **Programas Estaduais de Transferências de Renda com Condicionais no Âmbito do Plano Brasil Sem Miséria**, 2018.

LUNARDI, M. S.; CASTRO, J.; MONAT, A. Visualização dos resultados do Yahoo em nuvens de texto: uma aplicação construída a partir de web services. **InfoDesign. Revista Brasileira de Design da Informação**, v. 5, n. 1, p. 21–35, 2008.

MACKEY, A.; GASS, S. **Common data collection measures. Second language research: methodology and design**. p.43–99, 2005. Mahwah: Lawrence Erlbaum.

MANIASI, S.; BRITOS, P.; GARCÍA-MARTÍNEZ, R. **A Taxonomy-Based Model for Identifying Risks**, n. January, 2006.

MARCONI, M. DE A.; LAKATOS, E. M. **Técnicas de pesquisa: planejamento e execução de pesquisas, amostragens e técnicas de pesquisas, elaboração e interpretação de dados**. 3o ed. São Paulo: Atlas, 1996.

MARCONI, M. DE A.; LAKATOS, E. M. **Fundamentos de Metodologia Científica**. 8o ed. São Paulo: Atlas, 2017.

MARINS, M. T.; RODRIGUES, M. N.; DA SILVA, J. M. L.; DA SILVA, K. C. M.; CARVALHO, P. L. Emergency income in times of pandemic. **Sociedade e Estado**, v. 36, n. 2, p. 669–692, 2021.

MARSHALL, C. **Medindo e gerenciando riscos operacionais em instituições financeiras**. Rio de Janeiro: Qualitymark, 2002.

MARTINS, J. C. C. **Gerenciando Projetos de Desenvolvimento de Software com PMI, RUP e UML**. 2010.

MARY BETH CHRISSIS, MIKE KONRAD, S. S. **CMMI for Development: Guidelines for Process Integration and Product Improvement** (SEI Series in Software Engineering). 3o ed. Addison-Wesley Professional, 2011.

MATSUURA, S. Populares com a quarentena, aplicativos para reuniões virtuais viram alvo de hackers. **O Globo**, 2020. Disponível em: <<https://oglobo.globo.com/economia/tecnologia/populares-com-quarentena-aplicativos-para-reunioes-virtuaisviram-alvo-de-hackers-24357276>>.

MCDERMID, F.; PETERS, K.; JACKSON, D.; DALY, J. Conducting qualitative research in the context of pre-existing peer and collegial relationships. **Nurse Researcher**, v. 21, n. 5, p. 28–33, 2014.

MCGRAW, G. Four Software Security Findings. **Computer**, v. 49, n. 1, p. 84–87, 2016.

MEDEIROS, B. P. **O uso do ciberespaço pela administração pública na pandemia da COVID-19: diagnósticos e vulnerabilidades**, v. 54, n. 4, p. 650–662, 2020.

MENEZES, J.; GUSMÃO, C.; MOURA, H. Risk factors in software development projects: a systematic literature review. *Software Quality Journal*, v. 27, n. 3, p. 1149–1174, 2019.

MINAYO, M. C. S. **O Desafio do Conhecimento: pesquisa qualitativa em saúde**. 2o ed. São Paulo: Hucitec/Abrasco, 1999.

MINAYO, M. C. S. Amostragem e Saturação em Pesquisa Qualitativa: consensos e controvérsias. **Revista Qualitativa**, v. 5, n. 7, p. 1–12, 2017.

MIRANDA, R. F. **Implementando a gestão de riscos no setor público**, 2017.

MOHADDES DEYLAMI, H.; ARDEKANI, I.; MUNIYANDI, R. C.; SARRAFZADEH, H. Effects of Software Security on Software Development Life Cycle and Related Security Issues. **International Journal of Computational Intelligence and Information Security**, v. Volume, n. Number, p. Pages, 2015.

MPS, M.; MPS, G. G.; SW, S. M.-; DIREITOS, S.; SOFTEX, S. MPS. BR - Melhoria de Processo do Software Brasileiro. **Guia Geral MPS de Software Sumário**, 2021.

NETO, J. S.; PFEIFER, L. **Cartilha Cobit 2019**. October, 2021.

NEVES, S. M., SILVA, C. E. SALOMON, V. A. P., SILVA, A. F.; SOTOMONTE, B. E. P. Risk management in software projects through Knowledge Management

techniques: Cases in Brazilian Incubated Technology-Based Firms. **International Journal of Project Management**2, v. 32, p. 125–138, 2014.

NEVES, S. M.; SILVA, C. E. S. DA. Risk management applied to software development projects in incubated technology-based companies: literature review, classification, and analysis. **Gestão & Produção**, v. 23, n. 4, p. 798–814, 2016.

ODZALY, E. E.; GREER, D. **Agents. Journal of Ambient Intelligence and Humanized Computing**, v. 0, n. 0, p. 0, 2017. Springer Berlin Heidelberg.
OGC, O. OF G. C. Managing Successful Projects with PRINCE2®, v. 1, 2009.

OLIVEIRA, T. **Dependência no escalonamento dos métodos ágeis: identificação dos desafios enfrentados e das práticas utilizadas no gerenciamento**, 2019.

OLIVEIRA, T.; SANTOS, P. L. B.; MEDEIROS JÚNIOR, J. V.; GURGEL, A. M.; SILVA, B. J. P. Proposta de framework para o processo de gestão de Riscos no setor Público (PROGERIS). **Revista Gestão Universitária na América Latina - GUAL**, p. 256–277, 2020.

OWASP. OWASP Top 10. Disponível em: <https://owasp.org/Top10/pt_BR/>.

PAWAR, R. P. A comparative study of agile software development methodology and traditional waterfall model. **IOSR Journal of Computer Engineering (IOSR-JCE)**, v. 2, n. 2, p. 1–8, 2015.

PEZZOTTA, G.; CAVALIERI, S.; GAIARDELLI, P. A spiral process model to engineer a product service system: an explorative analysis through case studies. **CIRP Journal of Manufacturing Science and Technology**, v. 5, n. 3, p. 214–225, 2012.

PMBOK, P. M. I. **Project Management Institute**. Um Guia do Conhecimento em Gerenciamento de Projetos. 6o ed. Newtown Square, PA, EUA, 2017.

PONTES, R. E. DA S.; NETO, J. S. C. Contratação do desenvolvimento ágil de software na administração pública federal: riscos e ações mitigadoras. 66o ed. **Revista do Serviço Público**, 2015.

PONTES, T. B.; ARTHAUD, D. D. B. Metodologias ágeis para o desenvolvimento de softwares. Ciência e Sustentabilidade, **Periódicos UFCA**, v. 4, n. 2, 2018. Disponível em: <<https://periodicos.ufca.edu.br/ojs/index.php/cienciasustentabilidade/article/view/314/308>>. Acesso em: 9/7/2023.

PRATAMA, E.; WIRADARMA, A. Open Source Intelligence Testing Using the OWASP Version 4 Framework at the Information Gathering Stage (Case Study: X Company). **International Journal of Computer Networks and Information Security**, v. 11, n. 7, p. 8–12, 2019. Disponível em: <<https://doi.org/10.5815/ijcnis.2019.07.02>>.

PRESIDÊNCIA, DA R. **Brasília-julho/2020 Presidência Da República Secretaria De Governo**, p. 1–34, 2020.

PRESSMAN, R.; MAXIM, B. R. **Engenharia de Software: uma abordagem profissional**, 9o ed. Porto Alegre: Mc Graw Hill Education, 2021.

PRESSMAN, R. S.; MAXIM, B. R. E. **Engenharia de software**: uma abordagem profissional. São Paulo, 2016.

PRESSMAN, J.; WILDAVSKY, A. **Implementation**. Berkeley, CA: University of California Press, 1973.

PRESSMAN, R. S. **Software Engineering**: A Practitioner's Approach. McGraw-Hill, 1997.

PRESSMAN, R. S. **Engenharia de Software**: Uma abordagem profissional. 7o ed. Porto Alegre: AMGH, 2011.

RAMALHO, W. D. **Portal de Periódicos da Capes**: uma análise do conteúdo mediante a aplicação da ferramenta SWOT, 2015. Porto Alegre, BRRS: Universidade Federal do Rio Grande do Sul, Instituto de Ciências Básicas da Saúde, Programa de Pós-Graduação em Educação em Ciências: Química da Vida e Saúde. Disponível em: <<https://www.lume.ufrgs.br/handle/10183/119182>>. Acesso em: 22/7/2023.

RAMSDEN, A.; BATE, A. **Using word clouds in teaching and learning**. University of Bath, 2008.

RASHID, A.; CHIVERS, H.; DANEZIS, G.; LUPU, E.; MARTIN, A. The Cyber Security Body of Knowledge (CyBoK) 1.0. CyBOKVersion 1.0 **The National Cyber Security Centre 2019**, p. 808, 2019. Disponível em: <<https://www.cybok.org/media/downloads/CyBOK-version-1.0.pdf>>.

RASHID, A.; DANEZIS, G.; CHIVERS, H. Scoping the Cyber Security Body of Knowledge. **IEEE Security and Privacy**, v. 16, n. 3, p. 96–102, 2018.

RAWLS, J. **Justiça como equidade**. São Paulo: Martins Fontes, 2003.

RENN, O. The role of risk perception for risk management. **Reliability Engineering and System Safety**, v. 59, p. 49–62, 1998a.

RENN, O. Three decades of risk research: accomplishments and new challenges. **Journal of risk research**, v. 1, n. 1, p. 49–71, 1998b.

ROCHA, B. C. G. DA. **Fatores Inibidores e Facilitadores à Implementação da Gestão de Riscos no Departamento Penitenciário Nacional**. 2022.

ROCHA, S. **Pobreza no Brasil: afinal, de que se trata?** Rio de Janeiro: Fundação Getúlio Vargas, 2003.

ROGER S. PRESSMAN, B. M. **Software Engineering**: A Practitioner's Approach. 8o ed. 2014.

ROYCE, W. Managing the Development of Large Software Systems. Proceedings of the 9th International Conference on Software Engineering. **Anais...** 1987. Disponível em: <<http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>>. Acesso em: 9/7/2023.

S. BAYONA-ORÉ, J. A. CALVO-MANZANO, G. C. AND T. S.-F. Critical success factors taxonomy for software process deployment. **Software Quality Journal**2, v. 22, n. 1, p. 21–48, 2014.

SÁDABA, S. M.; EZCURDIA, A. P.; LZCANO, A. M. E.; VILLANUEVA, P. Project risk management methodology for small firms. *Int. J. Project Manage.*, v. 2, p. 327–340, 2014.

SAMPAIO, R. C. e-Orçamentos Participativos como iniciativas de e-solicitação: uma prospecção dos principais casos e reflexões sobre a e-Participação. **Revista de Administração Pública**, v. 50, n. 6, p. 937–958, 2016.

SANCHES DE OLIVEIRA, W. **Universidade de Brasília Gestão de riscos de projetos aplicada ao gerenciamento de cronogramas: um estudo de caso do Sicoob Confederação.**, p. 132, 2020.

SBROCCO, J.; MACEDO, P. Metodologias Ágeis: Engenharia de Software sob Medida. São Paulo: Érica, 2012.

SECCHI, L. **Políticas públicas: conceitos, esquemas de análise**, casos concretos. São Paulo: Cengage Learning, 2010.

SEDEK, K. A.; OSMAN, N.; OSMAN, M. N.; JUSOFF, H. K. Developing a Secure Web Application Using OWASP Guidelines. **Computer and Information Science**, v. 2, n. 4, p. 137–143, 2009.

SELLTIZ, CLAIRE, MARIE JAHODA, M. D. E S. W. C. Métodos de pesquisa nas relações sociais., p. 690–690, 2011.

SEN, A. **A ideia de justiça. São Paulo:** Companhia das Letras, 2011.

SHEHABUDDEEN, N.; PROBERT, D.; PHAAL, R.; PLATTS, K. Representing and approaching complex management issues: Part 1 - Role and definition. USA, 1999.
SIDDIQUE, L.; HUSSEIN, B. A. Practical insight about risk management process in agile software projects in Norway. *Proc. IEEE Int. Technology Management Conf. Anais...* 2014.

SILVA, E. L. DA E E. M. M. **Metodologia da pesquisa e elaboração de dissertação.** 2005.

SILVA, F. L. A. D. **Análise do Impacto do Gerenciamento de Riscos no Sucesso de Projetos:** Um Estudo de Caso em uma Organização de Desenvolvimento de Software. Recife, PE, 2017.

SILVA, R. C. DA. A falsa dicotomia qualitativo-quantitativo: paradigmas que informam nossas práticas de pesquisas. *Diálogos metodológicos sobre prática de pesquisa. Anais...* p.159–174, 1998.

SILVA, R. O.; MARTINS, B. R.; DINIZ, W. G. A Complexibilidade da UML e seus Diagramas. **Tecnologias em Projeção**, v. 8, n. 1, 2017. Disponível em: <<http://revista.faculdadeprojecao.edu.br/index.php/Projecao4/article/view/825>>. Acesso em: 9/7/2023.

SIMKINS, B.; RAMIREZ, S. Enterprise-Wide Risk Management and Corporate Governance. **Loyola University Chicago Law Journal**, v. 39, p. 571–594, 2008. Chicago.

SINGH, N. **Framework of Goal-driven Risk Management in Software Development Projects Using the Socio-technical Systems Approach**, p. 1–15, 2021.

SOMMERVILLE, I. **Engenharia de Software**. 10th ed. São Paulo: Pearson Prentice Hall, 2019.

SOUSA NUNES, N. T.; MOTA, S. C.; CABRAL, A. C. DE A.; DOS SANTOS, S. M. Produção Científica Brasileira Sobre Gestão De Riscos No Setor Público: Uma Análise Bibliométrica. **Revista do Serviço Público**, v. 71, n. 4, p. 887–920, 2020.

SPOSATI, A. Modelo brasileiro de proteção social não contributiva: concepções fundantes. Concepção e gestão da proteção social não contributiva no Brasil. **Anais...** 2009. Brasília: Unesco/MDS.

TABASSI, E.; BURNS, K. J.; HADJIMICHAEL, M.; MOLINA-MARKHAM, A. D.; SEXTON, J. T. **A taxonomy and terminology of adversarial machine learning**, 2019.

TAIPALUS, T.; SEPPÄNEN, V.; PIRHONEN, M. The Journal of Systems & Software Uncertainty in information system development: Causes, effects, and coping mechanisms. *The Journal of Systems & Software*, v. 168, p. 110655, 2020. Elsevier Inc. Disponível em: <<https://doi.org/10.1016/j.jss.2020.110655>>.

TAVARES, B. G.; EDUARDO, C. **Practices to Improve Risk Management in Agile Projects**. v. 29, n. 3, p. 381–399, 2019.

TAVARES, B. G.; KEIL, M.; EDUARDO, C.; et al. A Risk Management Tool for Agile Software Development A Risk Management Tool for Agile Software Development. **Journal of Computer Information Systems**, v. 61, n. 6, p. 561–570, 2021. Taylor & Francis. Disponível em: <<https://doi.org/10.1080/08874417.2020.1839813>>.

TAVARES, B. G.; SILVA, C. E. S.; SOUZA, A. D. **Risk management analysis in scrum software projects**. *Int. Trans. Oper. Res.*, 2017.

TEIXEIRA, A. O conceito de seguridade social e a Constituição de 1998. **Previdência em Dados**, v. 7, 1992. Rio de Janeiro.

TERAQUEST. **Risk Management Process for Projects**. Argentina, 1998.

THE INSTITUTE OF INTERNAL AUDITORS - IIA. **Modelo das Três Linhas do IIA 2020. Uma atualização das três linhas de defesa**. IIA Brasil, p. 13, 2020. Disponível em: <<https://iiabrasil.org.br/korbillload/upl/editorHTML/uploadDireto/20200758glob-th-editorHTML-00000013-20072020131817.pdf>>.

THIRY-CHERQUES, H. R. Saturação em pesquisa qualitativa: estimativa empírica de dimensionamento. **Revista Brasileira de Pesquisas em Marketing (PMKT)**, v. 3, 2009. Disponível em: <http://www.revistapmkt.com.br/Portals/9/Edicoes/Revista_PMKT_003_02.pdf>. Acesso em: 18/4/2017.

THIRY-CHERQUES, R. H. Saturação em pesquisa qualitativa: estimativa empírica de dimensionamento. **Af-Rev PMKT**, v. 4, n. 08, p. 20–27, 2009. Disponível em: <http://www.revistapmkt.com.br/Portals/9/Edicoes/Revista_PMKT_003_02.pdf>. USP. Os instrumentos de pesquisa mais frequentes na pesquisa de intervenção. Usp, p. 1–6, 2012. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/4463430/mod_resource/content/1/Conteudos_de_20112012/Modulo_3_2011_2012/Abertura/Os_instrumentos_de_pesquisa_mais_frequentes.pdf>.

VACARI, I.; PRIKLADNICKI, R. **Desenvolvimento de software na administração pública**: Uma revisão sistemática da literatura. 2014.

VAZ, I. F.; PELICIONI, B. B. A importância do uso das TICS no processo de ensino-aprendizagem frente à Pandemia do novo Coronavírus (COVID-19) The importance of the use of TICS in the teaching-learning process facing the new Coronavirus Pandemic (COVID-19). , p. 10294–10300, 2021.

VETORAZZO, A. S. **Engenharia de software**. 2018.

VIANNA, M. L. **A americanização (perversa) da seguridade social no Brasil**: estratégias de bem-estar e políticas públicas. 2o ed. Rio de Janeiro: Revan/Ucam/IUPERJ, 1998.

WAZLAWICK, R. **Engenharia de Software**: Conceitos e Práticas. São Paulo: Elsevier, 2019.

WHO, O. W. H. **World Health Organization - WHO**. WHO Director-General's opening remarks at the média briefing on COVID-19. Disponível em: <<https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>>. Acesso em: 5/6/2023.

WHO, W. H. O. **Statement on the fifteenth meeting of the IHR (2005) Emergency Committee on the COVID-19 pandemic**. Disponível em: <[https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-\(2005\)-emergency-committee-regarding-the-coronavirus-disease-\(covid-19\)-pandemic](https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic)>. Acesso em: 16/6/2023.

WIRADARMA, A. A. B. A.; SASMITA, G. M. A. IT Risk Management Based on ISO 31000 and OWASP Framework using OSINT at the Information Gathering Stage (Case Study: X Company). **International Journal of Computer Network and Information Security**, v. 11, n. 12, p. 17–29, 2019.

YIN, R. K. **Estudo de caso: planejamento e métodos**. 3o ed. Porto Alegre: Bookman, 2005.

YIN, R. K. **Estudo de caso: planejamento e métodos**. Bookman, 2015.

ZUCULE DE BARROS, M. J. **Engenharia de Software**. Informática Aplicada: CSI 4305. Universidade Virtual Africana, 2017.

APÊNDICE A - PEDIDO DE AUTORIZAÇÃO PARA REALIZAÇÃO DE PESQUISA DE CAMPO

Assunto: Pesquisa Acadêmica – [REDACTED] – Mestrado UnB
Prezado Diretor, Nome

Meu nome é Rodrigo Franco de Souza, [REDACTED], apresentando-me neste documento na condição de aluno vinculado ao Programa de Pós-Graduação Profissional em Engenharia Elétrica da Universidade de Brasília - UnB (PPEE/UnB), sob a orientação do Prof. Dr. Carlos André de Melo Alves.

Venho por meio deste e-mail informar que realizo pesquisa de mestrado profissional, intitulada "Identificação de fatores de riscos associados ao processo de desenvolvimento dos softwares provedores dos programas sociais do Governo: o caso do Programa Social Brasileiro Alfa".

Para efetuar esse estudo de forma completa, solicito o acesso a documentos não sigilosos relacionados às etapas do processo de desenvolvimento de software da [REDACTED], em especial ligados ao Programa Social Brasileiro Alfa. Além disso, solicito sua autorização para convidar empregados que atuam na diretoria que coordena para a realização de entrevistas.

Por fim, é importante destacar que a coleta de dados seguirá os critérios estabelecidos no Ofício Circular nº 2/2021/CONEP/SECNS/MS, de 24 de fevereiro de 2021, o qual fornece orientações para procedimentos em pesquisas com qualquer etapa em ambiente virtual. Além disso, é importante frisar que o requerente irá seguir os princípios éticos, assumindo plena responsabilidade pelas informações coletadas. Os dados serão utilizados exclusivamente para fins acadêmicos, garantindo-se a confidencialidade das informações e a proteção do anonimato dos empregados que concordarem em participar das entrevistas.

Desde já, agradeço a atenção dispensada e pela disponibilidade em colaborar com esta pesquisa, que visa contribuir para o avanço do conhecimento na área de gestão de riscos aplicada ao desenvolvimento de softwares em nosso contexto de programas sociais do Governo.

Atenciosamente,

Rodrigo Franco

Mestrando em Engenharia Elétrica – PPEE/UnB - 07/08/2023

APÊNDICE B - ESTRUTURA DO ROTEIRO DE ENTREVISTAS

1 - Em qual fase do processo de desenvolvimento dos softwares 'provedores do Programa Social Brasileiro ALFA' você atuou predominantemente?

2 - Durante o processo de desenvolvimento dos softwares 'provedores do Programa Social Brasileiro ALFA', em sua percepção quais foram os principais atributos que interferiram na engenharia do produto?

Observação: ao pensar na 'engenharia de produto', pense sobre questões que possam ter interferido na definição de requisitos inclusive de engenharia, design e testes.

3 - Durante o processo de desenvolvimento dos softwares 'provedores do Programa Social Brasileiro ALFA', em sua percepção quais foram os principais atributos relacionados ao ambiente de desenvolvimento?

Observação: ao citar 'ambiente de desenvolvimento', fale um pouco sobre eventuais obstáculos ao processo de desenvolvimento, os métodos, metodologias e ambiente de trabalho com outras equipes.

4 - Durante o processo de desenvolvimento dos softwares 'provedores do Programa Social Brasileiro ALFA', em sua percepção, quais foram os principais atributos relacionados às restrições enfrentadas para o desenvolvimento do referido software?

Observação: ao citar 'restrições', procure relatar algo sobre recursos empregados, relações contratuais e eventuais interfaces do programa com outras partes interessadas.

5 - Existe algum comentário que deseja acrescentar em relação ao que foi exposto nas questões anteriores?

6 - Quanto tempo de experiência em processos de desenvolvimento do software você possuía a época do projeto?

- Menos de 1 ano
- De 1 até menos de 4 anos
- A partir de 4 anos

7 - Você ocupava alguma função (confiança/comissionada) durante processos de desenvolvimento do software:

- Sim
- Não

APÊNDICE C – MODELO DE FICHA ANALÍTICA

Ficha de análise para identificação de atributos	
Código do Entrevistado Exemplo: Ex	
Unidade de contexto: (análise temática) Parágrafo/Contexto que contém o atributo	
Análise de atributo	
Código do atributo: Fx.	Unidades de registro: (análise temática) Atributo
Classificação	
Classe / Elemento	Artefato Documental

Fonte: Adaptado de Ramalho (2015).

APÊNDICE D – E-MAIL CONVITE PARA ENTREVISTA

Assunto: Pesquisa Acadêmica – [REDACTED] – Mestrado UnB

Prezado(a) NOME DO ENTREVISTADO(A),

Meu nome é Rodrigo Franco de Souza, [REDACTED], apresentando-me neste documento na condição de aluno vinculado ao Programa de Pós-Graduação Profissional em Engenharia Elétrica da Universidade de Brasília - UnB (PPEE/UnB), sob a orientação do Prof. Dr. Carlos André de Melo Alves.

Venho por meio deste e-mail informar que realizo pesquisa de mestrado profissional, intitulada " Identificação de fatores de riscos associados ao processo de desenvolvimento dos softwares provedores dos programas sociais do Governo: o caso do Programa Social Brasileiro ALFA".

Para conduzir esse estudo de maneira abrangente, envio este convite para entrevista a ser realizada na plataforma Microsoft Teams no próximo dia XX/XX/XXXX às 00h. Nesse estágio, busco coletar as percepções dos entrevistados, frente a sua participação no processo de desenvolvimento dos softwares provedores dos programas sociais do Governo.

É importante destacar que a coleta de dados seguirá os critérios estabelecidos no Ofício Circular nº 2/2021/CONEP/SECNS/MS, de 24 de fevereiro de 2021, o qual fornece orientações para procedimentos em pesquisas com qualquer etapa em ambiente virtual. Além disso, é importante frisar que o requerente irá seguir os princípios éticos, assumindo plena responsabilidade pelas informações coletadas. Os dados serão utilizados exclusivamente para fins acadêmicos, garantindo-se a confidencialidade das informações e a proteção do anonimato dos empregados que participarem das entrevistas. Por fim, informo que a entrevista e coleta de dados foi apreciada e autorizada pela Diretoria em XX/XX/2023.

Desde já, agradeço a atenção dispensada e pela disponibilidade em colaborar com esta pesquisa, que visa contribuir para o avanço do conhecimento na área de gestão de riscos aplicada ao desenvolvimento de softwares em nosso contexto de programas sociais do Governo Federal.

Atenciosamente,

Rodrigo Franco de Souza

Mestrando em Engenharia Elétrica – PPEE/UnB

APÊNDICE E – E-MAIL CONVITE PARA VALIDAÇÃO DO PRÉ-TESTE

Assunto: Solicitação para Colaborar em Teste Piloto - Pesquisa Acadêmica - [REDACTED]
[REDACTED] – Mestrado UnB

Prezado(a) Nome,

Meu nome é Rodrigo Franco de Souza, [REDACTED], apresentando-me neste documento na condição de aluno vinculado ao Programa de Pós-Graduação Profissional em Engenharia Elétrica da Universidade de Brasília - UnB (PPEE/UnB), sob a orientação do Prof. Dr. Carlos André de Melo Alves.

Realizo pesquisa de mestrado profissional, intitulada "Identificação de fatores de riscos associados ao processo de desenvolvimento dos softwares provedores dos programas sociais do Governo: o caso do Programa Social Brasileiro Alfa". Para conduzir esse estudo é necessário realizar um teste piloto sobre as perguntas a serem apresentadas aos entrevistados. Nesse estágio, busco contar com a sua colaboração para colher sua opinião sobre o entendimento e a clareza dos enunciados das questões, bem como sobre a pertinência das questões em relação à temática proposta por esta pesquisa.

Esclareço que as opiniões colhidas serão utilizadas exclusivamente para fins acadêmicos, garantindo-se a confidencialidade das informações e a proteção do anonimato dos que concordarem em participar das entrevistas. Informo, também, que antes de efetuar esse contato com você recebemos a anuência da diretoria em que atua para consulta de servidores com esse objetivo.

Desde já, agradeço pela atenção dispensada e pela disponibilidade em colaborar com esta pesquisa. Agradeço o seu retorno indicando se está de acordo em participar do teste piloto e indicando melhor momento para contato.

Atenciosamente,

Rodrigo Franco de Souza

Mestrando em Engenharia Elétrica – PPEE/UnB

XX/XX/XXXX

APÊNDICE F – MEDIDAS DE PROTEÇÃO SOCIAL

Quadro 26 - Visão geral das medidas de proteção social por componente e país

Países	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO			
	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido
Afganistão	✓		✓	✓			✓			✓		
Albânia	✓		✓	✓	✓		✓	✓	✓		✓	✓
Argélia	✓		✓	✓	✓	✓	✓	✓			✓	
Samoa Americana	✓		✓								✓	✓
Andorra	✓			✓	✓			✓	✓			
Angola	✓		✓	✓							✓	
Anguila (Reino Unido)	✓		✓	✓	✓			✓		✓		
Antígua e Barbuda	✓		✓	✓		✓			✓	✓	✓	✓
Argentina	✓		✓	✓	✓	✓	✓	✓	✓		✓	
Armênia	✓	✓	✓	✓			✓		✓		✓	✓
Aruba			✓	✓				✓	✓		✓	✓
Austrália	✓			✓	✓			✓	✓	✓	✓	✓
Áustria	✓	✓		✓	✓			✓	✓	✓	✓	✓
Azerbaijão	✓	✓	✓	✓	✓		✓	✓	✓		✓	
Bahamas	✓		✓	✓	✓	✓	✓					
Bahrein	✓			✓		✓			✓			
Bangladesh	✓		✓	✓		✓			✓		✓	
Barbados	✓		✓	✓	✓			✓	✓	✓	✓	
Bielorrússia	✓		✓	✓	✓				✓			✓
Países	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO			
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido

							invalid ez						
Bélgica	✓			✓	✓		✓	✓	✓	✓	✓	✓	
Belize	✓		✓	✓	✓		✓	✓	✓		✓	✓	
Benim	✓			✓					✓		✓	✓	
Bermudas	✓				✓		✓						
Butão	✓	✓	✓	✓					✓				
Bolívia	✓		✓	✓	✓	✓		✓			✓	✓	
Bósnia e Herzegovina	✓		✓	✓	✓	✓		✓	✓	✓			
Botsuana	✓	✓	✓	✓	✓	✓			✓				
Brasil	✓		✓	✓	✓		✓	✓			✓	✓	
Ilhas Virgens Britânicas			✓	✓	✓								
Brunei Darussalam	✓			✓			✓	✓	✓	✓			
Bulgária	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	
Burkina Faso	✓		✓	✓				✓			✓		
Burundi	✓		✓	✓					✓				
Camboja	✓	✓	✓					✓	✓	✓			
Camarões	✓		✓	✓			✓	✓					
Canadá	✓		✓	✓	✓		✓		✓	✓	✓		
cabo Verde	✓		✓	✓	✓			✓	✓				
Caribe Holanda			✓	✓					✓				
Ilhas Cayman	✓		✓	✓		✓	✓	✓	✓	✓	✓		
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO				
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido	
República Centro-Africana	✓		✓	✓								✓	
Chade			✓	✓									
Chile	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	
China	✓		✓	✓		✓	✓	✓		✓		✓	
Colômbia	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	
Comores	✓			✓					✓				

Congo, Dem. Rep.	✓	✓	✓	✓		✓					✓	✓
Congo, Rep.	✓	✓	✓	✓		✓					✓	✓
Ilha Cook	✓		✓	✓	✓				✓	✓		
Costa Rica	✓		✓	✓			✓	✓		✓	✓	✓
Costa do Marfim	✓		✓	✓		✓				✓	✓	
Croácia	✓		✓	✓			✓	✓	✓		✓	✓
Cuba	✓							✓	✓		✓	✓
Curaçao	✓		✓	✓				✓	✓	✓		
Chipre	✓		✓	✓	✓			✓	✓	✓	✓	✓
República Checa	✓				✓	✓	✓	✓	✓	✓		
Dinamarca				✓	✓				✓		✓	✓
Djibuti	✓		✓	✓	✓			✓	✓		✓	✓
Dominica	✓		✓	✓						✓		
República Dominicana	✓		✓	✓	✓				✓		✓	✓
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO			
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido
Equador	✓		✓	✓	✓	✓		✓			✓	✓
Egito, Rep. Árabe.	✓		✓		✓	✓	✓				✓	
El Salvador	✓		✓	✓	✓	✓				✓	✓	✓
Guiné Equatorial			✓	✓				✓				
Eritreia	✓		✓	✓								
Estônia	✓		✓	✓	✓			✓	✓	✓		
Eswatini	✓		✓	✓	✓	✓		✓		✓		
Etiópia	✓	✓	✓	✓	✓					✓	✓	✓
Ilhas Faroe									✓			
Fiji	✓		✓	✓	✓		✓	✓				
Finlândia	✓		✓	✓	✓	✓		✓			✓	✓
França	✓		✓	✓	✓			✓	✓	✓	✓	✓
Polinésia Francesa	✓		✓									
Gabão	✓		✓	✓					✓		✓	

Gâmbia, A	✓		✓	✓								✓	
Geórgia	✓			✓	✓		✓		✓		✓	✓	
Alemanha	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	
Gana	✓	✓	✓	✓		✓	✓				✓		
Gibraltar					✓			✓	✓				
Grécia	✓			✓	✓			✓	✓	✓	✓	✓	
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO				
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de desemprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido	
Groenlândia									✓				
Granada	✓	✓	✓	✓	✓			✓	✓		✓		
Guam	✓	✓	✓	✓	✓								
Guatemala	✓		✓	✓			✓	✓	✓		✓	✓	
Guernsey	✓				✓				✓	✓			
Guiné	✓	✓	✓	✓				✓		✓			
Guiné-Bissau	✓	✓	✓	✓									
Guiana	✓		✓	✓			✓	✓		✓			
Haiti	✓		✓						✓				
Honduras	✓		✓	✓		✓		✓	✓	✓	✓	✓	
Hong Kong	✓		✓	✓	✓			✓	✓			✓	
Hungria				✓	✓		✓	✓	✓	✓		✓	
Islândia	✓				✓		✓	✓	✓		✓		
Índia	✓	✓	✓	✓		✓	✓	✓			✓		
Indonésia	✓	✓	✓	✓		✓			✓	✓	✓		
Irã, República Islâmica.	✓		✓	✓	✓								
Iraque	✓		✓	✓				✓					
Irlanda	✓		✓	✓	✓				✓		✓		
Ilha de Man					✓				✓				
Israel	✓		✓	✓	✓		✓	✓	✓				
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO				
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de desemprego	Trabalho Público	Apoio ao seguro de saúde	Contribuições para a segurança social	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de	

maurício	✓	✓		✓					✓	✓	✓	✓	
México	✓	✓		✓			✓	✓					
Micronésia, Fed. Sts.	✓			✓	✓								
Moldávia	✓				✓								
Mônaco	✓				✓			✓	✓	✓	✓	✓	
Mongólia	✓		✓	✓	✓			✓	✓			✓	
Montenegro	✓		✓	✓			✓	✓	✓		✓		
Montserrat (Reino Unido)	✓		✓	✓					✓			✓	
Marrocos	✓		✓	✓	✓	✓	✓	✓	✓		✓		
Moçambique	✓	✓		✓	✓			✓				✓	
Mianmar	✓	✓	✓	✓	✓	✓		✓					
Namíbia	✓		✓	✓	✓				✓		✓	✓	
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO				
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido	
Nepal	✓	✓	✓	✓		✓		✓		✓			
Holanda	✓			✓	✓			✓	✓	✓	✓		
Nova Caledônia	✓				✓			✓	✓				
Nova Zelândia	✓		✓	✓	✓				✓	✓			
Nicarágua			✓		✓						✓	✓	
Níger	✓	✓	✓	✓									
Nigéria	✓	✓	✓	✓					✓		✓		
Macedônia do Norte	✓		✓	✓	✓	✓		✓	✓	✓	✓		
Ilhas Marianas do Norte	✓		✓	✓	✓								
Noruega	✓			✓	✓			✓	✓	✓	✓		
Omã			✓	✓	✓	✓		✓			✓		
Paquistão	✓	✓	✓	✓		✓	✓			✓	✓	✓	
Palau	✓	✓	✓	✓	✓								
Panamá	✓		✓	✓	✓					✓	✓	✓	
Papua Nova Guiné		✓	✓					✓			✓	✓	

	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO			
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por invalidez	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido
Eslovênia	✓			✓	✓		✓	✓	✓		✓	✓
Ilhas Salomão			✓	✓				✓		✓		
Somália	✓		✓	✓			✓					✓
África do Sul	✓	✓	✓		✓		✓		✓		✓	
Sudão do Sul	✓	✓	✓								✓	✓
Espanha	✓		✓	✓	✓		✓	✓			✓	✓
Sri Lanka	✓		✓	✓			✓				✓	
São Cristóvão e Nevis	✓		✓	✓	✓					✓		✓
Santa Lúcia	✓	✓	✓	✓	✓					✓		
São Vicente e Granadinas	✓	✓	✓	✓	✓		✓				✓	✓
Sudão	✓		✓								✓	✓
Suriname	✓		✓	✓	✓	✓				✓	✓	
Suécia	✓				✓	✓		✓	✓	✓		
Suíça	✓							✓	✓			
República Árabe da Síria	✓		✓	✓						✓	✓	
Taiwan, China	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
Tadjiquistão	✓		✓	✓	✓		✓	✓		✓		
Tanzânia	✓			✓	✓							
tailândia	✓			✓	✓	✓		✓		✓		
Timor-Leste	✓		✓	✓				✓	✓			
	ASSISTÊNCIA SOCIAL				SEGURO SOCIAL				MERCADO DE TRABALHO			
Países	Transferências em dinheiro	Trabalho Público	Alimentação escolar	Utilidade e apoio financeiro	Licença remunerada/de emprego	Trabalho Público	Apoio ao seguro de saúde Pensões e benefícios por	Contribuições para a segurança social (isenção/subsídio)	Subsídio salarial	Ativação (treinamento)	Ajuste da regulamentação trabalhista	Subsídio por tempo de trabalho reduzido

							invalid ez						
Tonga	✓			✓				✓	✓				
Trindade e Tobago	✓		✓	✓	✓								✓
Tunísia	✓			✓	✓	✓	✓	✓	✓				
Peru	✓		✓	✓			✓	✓	✓			✓	✓
Turquemenistão	✓							✓				✓	
Ilhas Turcas e Caicos	✓			✓									
Tuvalu	✓			✓				✓					
Uganda	✓	✓	✓	✓			✓	✓					
Ucrânia	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	
Emirados Árabes Unidos			✓	✓	✓	✓		✓		✓	✓	✓	✓
Reino Unido	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	
Estados Unidos	✓		✓	✓	✓			✓				✓	
Uruguai	✓		✓	✓	✓	✓	✓	✓	✓			✓	✓
Uzbequistão	✓	✓	✓	✓	✓	✓	✓				✓	✓	✓
Vanuatu	✓			✓			✓			✓			
Venezuela, RB	✓		✓	✓						✓		✓	✓
Vietnã	✓			✓	✓	✓	✓	✓	✓	✓			
Ilhas Virgens (EUA)	✓		✓	✓	✓						✓		
Cisjordânia e Gaza	✓	✓	✓	✓				✓	✓				
Iémen	✓		✓	✓									
Zâmbia	✓	✓	✓	✓			✓					✓	✓
Zimbábue	✓	✓	✓	✓		✓					✓	✓	

Fonte: Do autor adaptado de Gentilini et al. (2020)

APÊNDICE G – GLOSSÁRIO ROTEIRO DE ENTREVISTAS

Taxonomia – A taxonomia oferece uma estrutura para organizar e estudar a amplitude das questões de desenvolvimento de software e, portanto, fornece uma estrutura para identificar e organizar os riscos do desenvolvimento de software

Processo – O processo compreende a definição, planejamento, documentação, adequação, aplicação e comunicação dos métodos e procedimentos utilizados para desenvolver o produto.

Engenharia de produto – A classe engenharia de produto refere-se às atividades de engenharia de sistemas e engenharia de software envolvidas na criação de um sistema que atende aos requisitos especificados e às expectativas do cliente. Essas atividades incluem análise e especificação de requisitos de sistemas e software, design e implementação de software, integração de componentes de hardware e software, e teste de software e sistema.

Ambiente de Desenvolvimento – A classe ambiente de desenvolvimento diz respeito tanto ao contexto no qual o projeto está inserido quanto ao método aplicado para desenvolver um produto de software. Este contexto envolve o fluxo e o sistema de desenvolvimento em si, as técnicas de gestão adotadas e o próprio ambiente de trabalho. Cada um desses aspectos do ambiente é descrito detalhadamente por meio de seus atributos específicos.

Restrições do programa – A classe Restrições do programa referem-se aos fatores "externos" do projeto. Estes são fatores que podem estar fora do controle do projeto, mas ainda podem ter grandes efeitos em seu sucesso ou constituir fontes de risco substancial.

Classe - A classe define o campo de atividade da engenharia de software com o qual determinado risco pode estar associado. No contexto da identificação de riscos baseada em taxonomia, uma classe refere-se a uma categoria de alto nível que aborda um conjunto amplo de riscos relacionados a um aspecto particular do desenvolvimento de software.

Elemento – O elemento de classe indica uma área específica de risco no campo relevante de atividade. Dentro de uma classe, existem elementos mais específicos que detalham certas áreas do risco geral abordado pela classe.

Atributo – O atributo do elemento define um fator de risco em uma determinada área de risco, que pode estar associado a um evento indesejável, ação ou fato que é uma fonte de risco. Atributos são detalhes ainda mais específicos dentro de um elemento. Eles descrevem características particulares ou aspectos do elemento em questão

APÊNDICE H – ATIVIDADES DO PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

<p>Atividade 1 Elaborar Visão da Solução</p>
<p>Descrição: Na elaboração do Documento de Visão para produtos existentes sem mudanças arquiteturais, o processo é simplificado, focando em necessidades e funcionalidades já documentadas. Para novos produtos ou com alterações arquiteturais, a estratégia envolve a elicitação de requisitos de negócio, identificação de necessidades e envolvidos, e a definição de arquitetura da solução. Este último passo exige a colaboração de um Especialista Técnico para moldar a visão e estruturar a arquitetura preliminar da solução.</p>
<p>Atividade 2 Avaliar Necessidade de Solução Arquitetural</p>
<p>Descrição: No início do desenvolvimento de produtos novos ou com mudanças complexas, o Líder do Produto deve colaborar com o Arquiteto de Solução e outros especialistas para delinear a arquitetura da solução. Dependendo da necessidade de infraestrutura, pode-se optar por um Blueprint de Plataforma padrão para acelerar o desenvolvimento, ou, se necessário, aguardar aprovação para detalhar a solução e infraestrutura. Este planejamento prévio facilita a autonomia da equipe e agiliza o processo de desenvolvimento.</p>
<p>Atividade 3 Medir Tamanho Funcional</p>
<p>Descrição: O processo inicia pela coleta de documentação necessária para a contagem de pontos de função, variando conforme o tipo: Inicial (Nesma ou Detalhada, exigindo apenas o Documento de Visão) ou final (Detalhada, requerendo Documento de Visão, Especificações Principal e Complementar, Modelo de Dados e Telas do Sistema). Após a análise de pontos de função e escolha do método de contagem, a contagem é submetida para validação, podendo requerer ajustes. A validação final atualiza a Baseline, assegurando a precisão dos dados.</p>
<p>Atividade 4 Planejar Entregas</p>

Descrição:

O Líder do Produto deve elaborar o planejamento das Entregas, com base no Documento de Visão Contagem de Pontos de Função Inicial e as necessidades do Cliente identificadas na PA do Produto ou na PA Portal de Necessidades. As informações das entregas são fundamentais para elaboração da Proposta de Atendimento e devem ser materializadas de acordo com o artefato de saída Plano de Entregas.

Atividade 5

Estruturar Iniciativa

Descrição:

O processo envolve definir responsabilidades, formatos de atendimento e estratégias de execução, adaptando-se à duração e natureza da demanda. O mapeamento de riscos, capacidade e planejamento detalhado das entregas são cruciais. A necessidade de testes de desempenho é avaliada e incorporada ao cronograma. A abordagem varia entre fluxo Kanban para tarefas de curto prazo e Time-boxed para projetos mais longos, considerando normativas específicas para projetos gratificados e complexos.

Atividade 6

Consolidar Planejamento

Descrição:

No planejamento, define-se as datas de entrega e ajusta-se o cronograma. Para iniciativas de curta duração, uma visão geral é suficiente, enquanto projetos mais longos exigem detalhamento e alocação de recursos. Após o planejamento, a ferramenta é usada para marcar a transição para a fase de execução, estabelecendo uma linha de base para monitoramento. É crucial registrar o dimensionamento inicial e garantir que o planejamento esteja alinhado com as estratégias e diretrizes organizacionais.

Atividade 7

Planejar Release

Descrição:

O processo envolve a priorização e ajuste do Backlog do Produto, focando em valor de negócio, necessidades do usuário e viabilidade técnica. A evolução do produto considera débitos técnicos e riscos, negociando soluções de qualidade com o cliente. O planejamento e estimativa do trabalho da Release são cruciais, definindo escopo, sprints e estratégias colaborativas. Mudanças podem ser introduzidas conforme acordos com o cliente, respeitando procedimentos formais ou mais flexíveis,

conforme a natureza da demanda.

Atividade 8

Planejar Sprint

Descrição:

O planejamento da Sprint envolve o Líder do Time e stakeholders para definir tarefas e metas. Após alinhar o cronograma, objetivos e selecionar itens de backlog, o Time detalha e estima as tarefas, considerando recursos e prazos. A aceitação do cliente é crucial para confirmar o alinhamento. O planejamento é detalhado em tarefas específicas, visando a entrega de valor na Sprint, e a responsabilidade é distribuída entre os membros do Time, com registro detalhado no Plano da Sprint e no Plano de Testes.

Atividade 9

Especificar Item de Backlog

Descrição:

A especificação de itens de backlog envolve selecionar formatos de documentação e detalhar requisitos, regras, casos de uso, e UX para alinhar com as necessidades do cliente e do projeto. Inclui a definição de testes e a garantia de rastreabilidade para integração e compreensão total do projeto. Também abrange a especificação técnica detalhada para a coleta de dados, respeitando as restrições e garantindo a qualidade dos dados obtidos.

Atividade 10

Implementar Solução

Descrição:

O processo de construção de soluções de TIC envolve modelagem de dados, codificação de itens de backlog, implementação de testes unitários e profiling de transações. A modelagem é essencial para estruturar a informação e a codificação segue padrões de desenvolvimento para garantir qualidade e manutenibilidade. Testes unitários asseguram que o código atende às funcionalidades especificadas, enquanto o profiling visa otimizar recursos e desempenho, identificando e corrigindo problemas de maneira eficiente.

Atividade 11

Implementar Testes

Descrição:

O desenvolvimento de códigos de testes automatizados, tanto funcionais quanto não-funcionais, deve estar alinhado ao planejamento e às ferramentas de automação. Inclui compreender, automatizar e

validar roteiros de testes, além de parametrizar as entradas com Dados de Testes. Esses testes visam replicar o comportamento do usuário, respeitando as regras de negócio e critérios de aceitação, para garantir a integridade e performance do sistema.

Atividade 12

Preparar Testes

Descrição:

A seleção do escopo de teste envolve a escolha criteriosa de casos de teste funcionais e de desempenho, visando cobrir novas funcionalidades, áreas impactadas por mudanças, falhas anteriores e funcionalidades críticas. A preparação para a execução dos testes é essencial, abrangendo a definição de procedimentos e a configuração de ambientes, com atenção especial para sistemas móveis, garantindo que todas as validações sejam planejadas e executadas.

Atividade 13

Executar Testes Funcionais

Descrição:

A execução dos testes funcionais, tanto manuais quanto automatizados, é essencial para verificar a conformidade das funcionalidades do sistema. A equipe deve executar testes exploratórios, avaliar os resultados e registrar defeitos. Em sistemas móveis, os testes e registros de defeitos seguem procedimentos específicos. Para soluções envolvendo Chatbots, IA ou Ciência de Dados, os testes incluem a validação do modelo e a avaliação prática pelo cliente, com registro de todos os resultados e defeitos detectados.

Atividade 14

Executar Testes Não-funcionais

Descrição:

Os Testes Não Funcionais incluem desempenho, segurança e outros requisitos como acessibilidade. Os de Desempenho são realizados conforme agendamento, e os de Segurança atendem requisitos específicos com análises dinâmicas. Requisitos adicionais como usabilidade podem demandar testes especiais. Todos os defeitos encontrados devem ser registrados adequadamente para garantir a qualidade e segurança do sistema.

Atividade 15

Avaliar Resultados dos Testes

Descrição:

O Time de Desenvolvimento e o Líder do Time avaliam os resultados dos testes para definir ações

corretivas ou melhorias. Itens com defeitos podem ser reagendados para sprints futuras se não forem resolvidos na sprint atual. A análise das falhas dos testes visa identificar e tratar as causas raízes, e vulnerabilidades identificadas nos testes de segurança são registradas e priorizadas para tratamento subsequente.

Atividade 16

Monitorar Progresso

Descrição:

O acompanhamento do progresso das tarefas, o gerenciamento de mudanças e a satisfação do cliente são fundamentais. Isso inclui atualizações contínuas, avaliação de mudanças propostas, monitoramento constante da satisfação do cliente, motivação da equipe, gerenciamento de riscos e impedimentos, e uma atenção especial à integração contínua. Cada passo é crucial para garantir que as entregas estejam alinhadas com as necessidades do cliente e os objetivos da Sprint sejam alcançados com eficácia.

Atividade 17

Gerenciar Fluxo de Trabalho

Descrição:

O monitoramento eficiente do fluxo de trabalho, integração contínua e comunicação são essenciais. Inclui ajustes em cadências, tratamento de impedimentos, uso de indicadores de desempenho, repriorização do backlog e replanejamento de testes. Essas práticas visam manter o alinhamento da equipe, otimizar o processo de desenvolvimento e garantir entregas de acordo com as necessidades do negócio e expectativas do cliente.

Atividade 18

Demonstrar Resultado da Sprint

Descrição:

A preparação do ambiente para demonstração e a reunião de apresentação dos resultados da Sprint são cruciais. O ambiente deve ser preparado, preferencialmente de homologação, com uma baseline clara dos itens de configuração. A reunião deve ser bem planejada, com duração controlada e participação ativa do cliente para validação dos resultados. Após a reunião, é essencial atualizar os itens de backlog e tarefas com base no feedback e nos resultados apresentados.

Atividade 19

Realizar Retrospectiva

Descrição:

A reunião de retrospectiva é um momento para o Time de Desenvolvimento e o Líder do Time refletirem sobre a Sprint que passou. Discutem-se as práticas bem-sucedidas e as falhas, focando em melhorias para o futuro. É um espaço aberto para feedback construtivo, estruturado em torno de três perguntas principais sobre o que deu certo, o que deu errado e o que pode ser melhorado. As ações de melhoria identificadas são implementadas nas Sprints futuras, contribuindo para a evolução contínua do processo.

Atividade 20

Disponibilizar Release para Homologação

Descrição:

O processo de preparação para a homologação é fundamental e envolve a criação da versão implantável e atualização da linha do tempo. O ambiente de homologação é preparado, com testes de regressão para assegurar a qualidade. A seleção cuidadosa dos casos de testes e a interação com o cliente são vitais para validar o alinhamento das funcionalidades com os objetivos de negócio. A baseline registrada reflete o produto homologado, e qualquer problema ou melhoria identificada pelo cliente é devidamente registrada e tratada.

Atividade 21

Medir Tamanho Funcional

Descrição:

Para o processo de contagem de pontos de função, a documentação relevante deve ser coletada e avaliada utilizando a Análise de Ponto por Função (APF). Esse procedimento inclui a escolha do método de contagem apropriado e a análise detalhada dos componentes do software, como arquivos lógicos internos, entradas e saídas externas, entre outros. A contagem é então submetida para validação, e ajustes podem ser necessários com base no feedback recebido. Este processo ajuda a estabelecer uma baseline para o produto ou módulo em desenvolvimento.

Atividade 22

Implantar Release

Descrição:

Para a preparação da passagem à produção, o código deve estar na branch correta, com publicação final em lojas para sistemas mobiles e procedimentos específicos para desenvolvimento de API, incluindo a solicitação de RDM. É fundamental verificar a correta implementação e funcionalidade do serviço pós-implantação, seguindo as etapas e aprovações necessárias para garantir a segurança e conformidade do processo.

Atividade 23

Encerrar Execução**Descrição:**

Para concluir uma iniciativa de desenvolvimento, é necessário adquirir o aceite formal da entrega pelo demandante. Isso inclui tornar a entrega disponível para homologação, solicitar a aprovação através de ferramentas apropriadas e registrar métricas finais de desempenho, como o tamanho funcional. Finalmente, o encerramento oficial do cronograma ou da ideia deve ser realizado, seguindo procedimentos específicos para projetos formalmente estabelecidos e acompanhados pelo departamento responsável, garantindo a conformidade e o fechamento adequado do projeto.