



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"

Leonardo Medeiros Prado

Apoio automatizado para a gestão de conhecimento ágil

São José do Rio Preto, 2023

Leonardo Medeiros Prado

Apoio automatizado para a gestão de conhecimento ágil

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Orientadora: Profa. Dra. Rogéria Cristiane Gratão de Souza.

São José do Rio Preto, 2023

P896a

Prado, Leonardo Medeiros

Apoio automatizado para a gestão de conhecimento ágil / Leonardo Medeiros Prado. -- São José do Rio Preto, 2023

62 p.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientadora: Rogeria Cristiane Gratão de Souza

1. Gestão de conhecimento. 2. Gestão ágil de projetos. 3. Framework Scrum. I. Título.

Leonardo Medeiros Prado

Apoio automatizado para a gestão de conhecimento ágil

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título de Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Comissão Examinadora

Profa. Dra. Rogéria Cristiane Gratão de Souza
UNESP - São José do Rio Preto
Orientadora

Prof. Dr. Leandro Alves Neves
UNESP - São José do Rio Preto

Prof. Dr. Lucas Correia Ribas
UNESP - São José do Rio Preto

São José do Rio Preto, 2023

AGRADECIMENTOS

Agradeço a toda minha família e meus amigos por me apoiarem durante estes anos de graduação e por sempre me acolherem e aconselharem em momentos difíceis.

Agradeço minha orientadora Profa. Rogéria Cristiane Gratão de Souza por aceitar o convite para me orientar e por sempre estar disposta com muita dedicação a sanar dúvidas e aconselhar.

Por fim, agradeço aos professores do Ibilce por toda a orientação e o ensino.

RESUMO

A indústria de software se destaca pela criação de produtos de alto valor. Com a popularização das metodologias ágeis, deu-se início ao desenvolvimento ágil de software, garantindo mais flexibilidade aos requisitos dos produtos. Porém, é importante considerar que o conhecimento das pessoas representa um ativo importante, logo, deve ser incorporado em um processo de desenvolvimento. Para tanto, gerenciar eficientemente os recursos de conhecimento em ambientes de desenvolvimento de software é essencial para preservar o conhecimento organizacional e estimular a inovação. Assim, tem-se a área de gestão de conhecimento, a qual é definida como um conjunto de processos cujo objetivo é apoiar a criação, a transferência e a aplicação do conhecimento. Neste cenário, o objetivo deste trabalho é contribuir para a gestão ágil de conhecimento, por meio da criação de um protótipo denominado KamTool. Para avaliar sua adequação, 31 profissionais de tecnologia atuantes no mercado usaram o protótipo por um período de 7 dias e responderam a um questionário online. Da análise das respostas, verifica-se que a área de gestão do conhecimento continua em crescimento e se popularizando. Especificamente sobre o KamTool, observa-se que tais usuários consideraram o protótipo intuitivo e útil, sendo adequado para o promover o compartilhamento de informações entre os membros de uma equipe de desenvolvimento de software de uma forma simples e rápida.

Palavras-chave: Gestão de conhecimento, Gestão ágil de projetos, Framework Scrum.

ABSTRACT

The software industry stands out for creating high-value products. With the popularization of agile methodologies, agile software development began, ensuring more flexibility in product requirements. However, it is relevant to consider that people's knowledge represents an important asset, to incorporate into a development process. To this end, efficiently managing knowledge resources in software development environments is essential to preserve organizational knowledge and stimulate innovation. Thus, we have the area of knowledge management, which is a set of processes whose objective is to support the creation, transfer, and application of knowledge. In this scenario, the goal of this work is to contribute to agile knowledge management through the creation of a prototype called KamTool. To assess its suitability, 31 technology professionals working in the market used the prototype for seven days and responded to an online questionnaire. From the analysis of the responses, it can be seen that the area of knowledge management continues to grow. Specifically regarding KamTool, the users considered the prototype intuitive and useful, being suitable for promoting the sharing of information between members of a software development team in a simple and quick way.

Keywords: Knowledge management, Agile project management, Scrum Framework.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas do ciclo de gestão de conhecimento.	20
Figura 2 – Visão geral das etapas previstas no ciclo Scrum.	24
Figura 3 – Arquitetura de microsserviços definida para a aplicação KamTool.	36
Figura 4 – Diagrama de classe para a aplicação KamTool.	37
Figura 5 – Ilustração da tela de cadastro de usuário.	40
Figura 6 – Ilustração da tela de autenticação.	41
Figura 7 – Ilustração da tela para troca de senha.	41
Figura 8 – Ilustração da tela para alteração de nome do usuário.	42
Figura 9 – Ilustração da tela para gerenciar os projetos vinculados a um usuário.	42
Figura 10 – Ilustração da tela para a seleção do papel do usuário junto ao projeto.	43
Figura 11 – Ilustração da tela de criação de um projeto.	44
Figura 12 – Ilustração da tela para edição dos dados de um projeto pelo seu criador.	44
Figura 13 – Ilustração da tela de backlog do produto.	45
Figura 14 – Ilustração da tela criação de um sprint.	45
Figura 15 – Ilustração da tela de edição de um sprint.	46
Figura 16 – Ilustração da tela de criação do board.	46
Figura 17 – Ilustração da tela de edição do board.	47
Figura 18 – Ilustração da tela de painel de um sprint.	47
Figura 19 – Ilustração da tela da base de conhecimento.	48
Figura 20 – Ilustração da tela de criação de um documento.	49
Figura 21 – Ilustração da tela de seleção ou criação de categorias.	49
Figura 22 – Ilustração da tela de seleção de usuários que podem editar o documento.	50
Figura 23 – Ilustração da tela de edição de documento.	50
Figura 24 – Gráfico com resultado da avaliação sobre a pergunta: “Qual sua função atual no mercado de TI?”.	53
Figura 25 – Gráfico com resultado da avaliação sobre a pergunta: “Qual seu tempo de experiência no mercado de TI na função atual?”.	53
Figura 26 – Gráfico com resultado da avaliação sobre a afirmação: “O protótipo tem um uso fácil e intuitivo”.	54
Figura 27 – Gráfico com resultado da avaliação sobre a afirmação: “O design e os componentes na interface são agradáveis para o uso”.	55
Figura 28 – Gráfico com resultado da avaliação sobre a afirmação: “O conceito de gestão de conhecimento já era conhecido”.	55
Figura 29 – Gráfico com resultado da avaliação sobre a afirmação: “O protótipo foi útil para o auxílio na gestão ágil de projetos”.	56

Figura 30 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo foi útil para o auxílio na gestão de conhecimento".	56
Figura 31 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo facilita a captura e o compartilhamento de conhecimento juntamente com a gestão de projeto".	57
Figura 32 – Gráfico com resultado da avaliação sobre a afirmação: "A quantidade de funcionalidades disponibilizadas no protótipo foram satisfatórias".	57
Figura 33 – Gráfico com resultado da avaliação sobre a afirmação: "A organização e estrutura estabelecidas para a documentação da base de conhecimento e dos boards do backlog do produto são adequadas".	58

LISTA DE TABELAS

Tabela 1 – Resultado obtido a partir da análise de softwares voltados para gestão de conhecimento e de projetos.	32
Tabela 2 – Resultado de avaliação do KamTool sobre as características de qualidade almejadas para ferramentas de gestão ágil de projetos. .	59

LISTA DE ABREVIATURAS E SIGLAS

TCC	Trabalho de Conclusão do Curso
XP	Extreme Programming
CMS	Content Management System

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Contexto do trabalho	14
1.2 Justificativa e motivação	15
1.3 Objetivos	16
1.4 Metodologia	16
1.5 Organização da monografia	17
2 REVISÃO BIBLIOGRÁFICA	18
2.1 Gestão de conhecimento	18
2.2 Modelo de gestão de conhecimento	19
2.3 Metodologias ágeis	22
2.4 Framework Scrum	23
2.5 Software de apoio	26
2.5.1 Apoio automatizado à gestão do conhecimento	27
2.5.2 Apoio automatizado à gestão de projetos	29
2.5.3 Análise de software	30
2.6 Considerações finais	33
3 DETALHAMENTO DO KAMTOOL	34
3.1 Descrição Geral	34
3.2 Definição da Arquitetura	35
3.3 Definição das funcionalidades	39
3.4 Considerações finais	51
4 RESULTADOS DO PROCESSO DE AVALIAÇÃO	52
4.1 Perfil dos participantes	52
4.2 Avaliação do KamTool	54
4.3 Considerações finais	59

5 CONCLUSÃO	60
5.1 Contribuições do trabalho	60
5.2 Trabalhos futuros	61
REFERÊNCIAS	62
APÊNDICE A	68

Capítulo 1 - Introdução

O desenvolvimento de software é uma atividade intrinsecamente criativa que depende significativamente das habilidades e experiências das pessoas envolvidas. A qualidade dos resultados obtidos está profundamente ligada ao conhecimento técnico e à experiência prática que os membros da equipe possuem. A colaboração eficaz desses profissionais, combinada com uma gestão inteligente de seus conhecimentos, é crucial para o sucesso do projeto. A gestão ágil desses conhecimentos visa otimizar a produtividade sem comprometer os prazos estabelecidos. A agilidade na gestão do conhecimento não apenas fortalece a colaboração, mas também promove a inovação, permitindo que os desenvolvedores respondam de maneira eficiente às demandas do mercado em constante evolução.

Neste capítulo são expostos detalhes sobre o trabalho, com intuito de evidenciar sua importância. Para tanto, na seção 1.1 é descrito o contexto do trabalho, apresentando as hipóteses adotadas para sua realização. Já na seção 1.2, tem-se a justificativa e motivação, apresentando a relevância do estudo e também o impulso acadêmico que levou à seleção desse tema específico. Na seção 1.3 é descrito o objetivo, apresentando o objetivo geral e específico do trabalho. Na seção 1.4, tem-se a metodologia utilizada para o desenvolvimento do trabalho. Por fim, na seção 1.5, tem-se a estrutura da monografia.

1.1 Contexto do trabalho

A indústria de software apresenta características peculiares em relação a outros segmentos do mercado. Ela se diferencia por desenvolver atividades intensivas em conhecimento e gerar produtos de alto valor agregado (BJØRNSON; DINGSØYR, 2008; CHEN et al., 2014; MACHADO SILVANA SECINARO; LANZALONGA, 2022). A globalização dos mercados e o uso intensivo do conhecimento no desenvolvimento de novos produtos contribuíram para que a indústria de software se tornasse um setor de alta tecnologia (OLUWAFUNMILAYO et al., 2015), tornando-o complexo (NAWINNA, 2011). Assim, é necessário e fundamental que o conhecimento das pessoas seja incorporado aos processos e atividades da organização. Portanto, gerenciar com eficiência os recursos de conhecimento em um ambiente de desenvolvimento de software é fundamental para manter o conhecimento organizacional e promover a inovação (AURUM et al., 2008).

A gestão de conhecimento é definida como um conjunto de processos cujo

objetivo é apoiar a criação, a transferência e a aplicação do conhecimento (MEIHAMI; MEIHAMI, 2014; AURUM et al., 2008). Nesse sentido, para o conhecimento ser útil à organização, ele deve ser exteriorizado e socializado entre os indivíduos que a compõem (WIIG, 1997). A abordagem tradicional de desenvolvimento de software é considerada pouco flexível para lidar com mudanças e com a necessidade de compartilhamento de informações em um ambiente tão dinâmico como o de desenvolvimento de software. Com o Manifesto Ágil as metodologias ágeis se popularizaram ainda mais, dando início ao desenvolvimento ágil de software (NAGAI; SBRAGIA, 2023). Metodologias ágeis também podem ser chamadas de Agile, que é um método de desenvolvimento de software flexível, baseado no trabalho em equipe, alta inovação, criatividade, flexibilidade, mudanças, comunicações e interações (NAGAI; SBRAGIA, 2023).

No entanto, os métodos ágeis utilizam principalmente mecanismos informais de comunicação e coordenação por meio da exploração do conhecimento tácito, dificultando a captura do conhecimento explícito criado no desenvolvimento de software, fazendo com que muito deste conhecimento seja perdido nesse processo. Diante deste cenário é possível definir duas hipóteses para a condução do presente projeto:

- Hipótese 1: A gestão de conhecimento, aplicada em um contexto de desenvolvimento ágil, contribui para o compartilhamento de conhecimento entre membros de uma equipe, durante o desenvolvimento de um projeto de software;
- Hipótese 2: O apoio automatizado para a gestão de conhecimento contribui para manter a agilidade em projetos de software.

1.2 Justificativa e motivação

O conhecimento obtido no processo de desenvolvimento de software é um recurso fundamental para as organizações (ALMEIDA et al., 2019; Di Vaio et al., 2021). Para Machado e Elias (2020), a sociedade atual marcha em alta velocidade, tornando a gestão das informações uma prática importante e necessária. Mesmo que o processo de desenvolvimento de um novo software seja único e não possa ser replicável devido às especificidades de cada projeto, o conhecimento e a experiência do pessoal envolvido podem auxiliar os integrantes de projetos futuros na execução de suas atividades.

A gestão do conhecimento, de acordo com Becerra-Fernandez e Sabherwal (2015), tem impacto nas pessoas, aumentando sua capacidade de aprender, melhorando a adaptabilidade dos membros e aumentando a satisfação no trabalho. Tem

também impacto no processo realizado pela empresa ou organização, aumentando a eficácia do projeto, a eficiência e o nível de inovação no processo em andamento.

Em uma revisão sistemática da literatura sobre o uso de metodologias ágeis no desenvolvimento de software foi constatado que gestão de conhecimento é um dos temas mais populares entre os pesquisadores ágeis. Segundo os autores, isso ocorre porque as metodologias ágeis promovem um ambiente de aprendizado contínuo (DINGSØYR et al., 2012).

No que se diz respeito a ferramentas para gestão de conhecimento ágil, percebe-se que se torna mais complexa a busca por tal ferramenta. Por outro lado, é mais comum encontrar ferramentas que tratem especificamente sobre a gestão de conhecimento, como, por exemplo, KnowledgeOwl, Bloomfire, Document360 (OLIVA; KOTABE, 2019) e ferramentas para auxiliar no uso de metodologias ágeis, como, por exemplo, Jira, Trello, Asana (ÖZKAN; MISHRA, 2019). Dessa forma, tem-se como motivação para o desenvolvimento do presente projeto a criação de um apoio computacional que consiga unir a gestão de conhecimento a um ciclo de desenvolvimento ágil.

1.3 Objetivos

O objetivo geral deste trabalho é contribuir para a gestão ágil de conhecimento com intuito de promover o compartilhamento de informações entre os membros de uma equipe de desenvolvimento de software de uma forma simples e rápida.

Para isto, tem-se como o objetivo específico o desenvolvimento de um protótipo de software como apoio automatizado, contemplando conceitos e técnicas de gestão de conhecimento para identificar, criar, apresentar e distribuir o conhecimento obtido no contexto de um processo ágil de desenvolvimento de software, visando contribuir para uma gestão efetiva sem comprometer a agilidade almejada.

1.4 Metodologia

O desenvolvimento deste trabalho foi dividido nas seguintes etapas:

- Revisão bibliográfica - estudo e pesquisa envolvendo os temas relacionados ao trabalho, como gestão de conhecimento, metodologias ágeis e critérios almeçados para uma ferramenta de gestão de conhecimento ágil, a partir da identificação e

análise de ferramentas relacionadas a gestão de conhecimento ágil eventualmente existentes, destacando os seus pontos fortes e fracos;

- Estabelecimento dos requisitos do protótipo proposto - extração e estudo dos requisitos desejados para a criação do apoio automatizado para gestão de conhecimento ágil;
- Estabelecimento da arquitetura de software - documentação dos componentes estabelecidos e definição da arquitetura que irá nortear o desenvolvimento do protótipo, a partir dos requisitos especificados;
- Prototipação - implementação do protótipo usando recursos computacionais gratuitos;
- Avaliação - avaliação do protótipo por profissionais atuantes na área de desenvolvimento de software, para verificar e avaliar a usabilidade e adequação da aplicação, considerando as hipóteses estabelecidas.

1.5 Organização da monografia

O presente trabalho é composto por quatro capítulos, além deste introdutório, conforme segue:

Capítulo 2 - Revisão Bibliográfica: neste capítulo são apresentados fundamentos teóricos pertinentes ao trabalho, incluindo Gestão de conhecimento, Metodologias Ágeis, Software para a gestão de conhecimento e para ágil de projetos e análise de trabalhos correlatos;

Capítulo 3 - Detalhamento do KamTool: neste capítulo é exposta a arquitetura e organização do protótipo KamTool, bem como a descrição de suas funcionalidades e a ilustração das telas correspondentes;

Capítulo 4 - Resultados do processo de avaliação: neste capítulo são mostrados os resultados obtidos com a avaliação do protótipo desenvolvido, realizada por meio de um questionário online disponibilizado juntamente com o KamTool a funcionários de diferentes empresas que aceitaram o convite para participar;

Capítulo 5 - Conclusão: neste capítulo são mostradas as conclusões que podem ser vistas a partir dos resultados obtidos, bem como os possíveis trabalhos futuros.

Capítulo 2 - Revisão Bibliográfica

No intuito de assegurar a correta compreensão do leitor, este capítulo destaca-se como um alicerce fundamental para a estrutura do estudo. Nele, busca-se não apenas evitar ambiguidades, mas também auxiliar na assimilação dos argumentos e conclusões apresentados ao longo do trabalho.

Neste capítulo é exposta toda base teórica para o desenvolvimento do protótipo em questão. Para tanto, na seção 2.1, tem-se a gestão de conhecimento, apresentando toda a teoria envolvida na gestão de conhecimento. Na seção 2.2 é descrito o ciclo da gestão de conhecimento, apresentando todo o processo que corresponde a gestão de conhecimento. Na seção 2.3, tem-se as metodologias ágeis, apresentando toda a base teórica envolvida na gestão ágil de projetos, bem como a descrição de algumas das metodologias existentes. Já na seção 2.4 é descrito o framework Scrum, apresentando de uma forma mais detalhada uma metodologia ágil, no caso o Scrum, que foi escolhido para o trabalho. Seguindo para a seção 2.5 é descrito o software de apoio, apresentando alguns softwares voltados para a gestão de conhecimento e a gestão ágil de projetos, além de uma análise destes softwares mencionados. Por fim, tem-se a seção 2.6 com as considerações finais.

2.1 Gestão de conhecimento

No mundo dos negócios, há uma transição da era dos recursos naturais para a era do conhecimento, baseada em pesquisa e desenvolvimento (FRIEDMAN, 2005; GULBRANSON; AUDRETSCH, 2008). O recurso econômico básico não é mais o capital, nem os recursos naturais, nem o trabalho, mas sim o conhecimento (JELENIC, 2011; KHAN, 2014). O conhecimento tem sido considerado um dos ativos e mercadorias mais importantes e altamente valorizadas (HEGAZY; GHORAB, 2014).

Anand e Walsh (2016) afirmam que o conhecimento contém informações, habilidades e expertise. Essas informações podem ser usadas na tomada de decisões e ações (CHANG; LIN, 2015). O principal objetivo de compartilhar o conhecimento é torná-lo visível, mostrar o seu papel nas organizações e incentivar os funcionários a promover comportamentos, como o compartilhamento do conhecimento, e construir a infraestrutura do conhecimento (MERLO, 2016). Todavia, o conhecimento sem uma gestão adequada pode ser obsoleto e inútil (ANSARI et al., 2012; KARIMI; JAVANMARD, 2014). Assim, as organizações precisam implementar e aplicar uma série de processos para gerenciarem seu conhecimento (OUYANG, 2014).

O conhecimento pode ser dividido e agrupado em duas formas: tácito e explícito. Há uma interação entre conhecimento tácito e explícito para criar novas ideias. Joia e Lemos (2010) afirmam que a dimensão desses dois tipos de conhecimento é uma das questões mais discutidas no campo da gestão do conhecimento. Segundo Mooradian (2006) e Grant (2007), o conceito de conhecimento tácito está no centro da gestão do conhecimento. Este tipo de conhecimento pode ser definido como algo pessoal, uma habilidade para fazer algo, que é parcialmente baseado na própria experiência das pessoas. Ele é menos familiar, geralmente de forma não convencional e menos consciente (ALWIS; HARTMANN, 2008). O conhecimento tácito difere do conhecimento explícito, ao ser geralmente proferido e compartilhado por meio de desenhos ou escrita. Já o conhecimento explícito geralmente vem na forma de livros, periódicos, artigos, documentos, bancos de dados, etc. (HERSCHEL; JONES, 2005; NONAKA; KROGH, 2009). Do ponto de vista organizacional, o conhecimento tácito está embutido na cultura organizacional e no entendimento coletivo, enquanto o conhecimento explícito pode ser conceituado, armazenado em bancos de dados e transferível formalmente.

A gestão do conhecimento é a capacidade de gerenciar o conhecimento, como reunir conhecimento interno ou externo das organizações, convertê-los em uma nova ideia ou estratégia e aplicá-los e protegê-los (LYTRAS et al., 2002). Gold et al. (2001) definiram a gestão do conhecimento como uma aplicação sistemática e explícita do conhecimento que auxiliará as organizações a maximizar a eficácia relacionada ao seu conhecimento e aos retornos dos ativos dele. Também cria novos recursos, incentiva a inovação e o desempenho. Diante disso, a gestão do conhecimento pode ser dividida em quatro processos: de captura, de armazenamento, de compartilhamento e de uso do conhecimento (LEIDNER et al., 2006; CHANG; LIN, 2015). Além disso, também pode ser definido como um processo sistemático para reunir, organizar e comunicar conhecimento tácito e explícito dos funcionários, que pode ser usado em seus trabalhos (SCHULTZE; LEIDNER, 2002; ALAVI et al., 2005; MASSEY; MONTOYA-WEISS, 2006). Assim, o principal objetivo da gestão do conhecimento é permitir que as organizações estejam cientes de seu conhecimento e moldem-no, para poderem usá-lo de maneira eficaz e eficiente (NEWELL et al., 2004; ALAVI et al., 2005).

2.2 Modelo de gestão de conhecimento

Segundo o Comitê Europeu de Padronização (CEN, 2004), existem mais de 150 modelos de ciclos existentes de gestão de conhecimento. Cada um desses modelos compreende um conjunto de práticas que buscam atender aos quatro processos mencionados anteriormente. Cada modelo de gestão do conhecimento possui sua própria abordagem e enfoca diferentes aspectos do processo de gestão do conheci-

mento. A escolha do modelo mais adequado dependerá das necessidades, objetivos e contexto específico de uma organização. Além disso, muitas vezes os modelos podem ser combinados ou adaptados para melhor atender às necessidades específicas de uma organização.

Neste trabalho, o modelo adotado é o definido em Tenório et al. (2020), o qual representa uma adaptação de Tenório et al. (2017), Evans et al. (2014) e Goldoni e Oliveira (2010), conforme esquematizado na Figura 1.

Figura 1 – Etapas do ciclo de gestão de conhecimento.



Fonte: Adaptado de Tenório et al. (2020).

Segundo Turner et al. (2012), existem seis processos principais na gestão do conhecimento que são: captura, criação, armazenamento, disseminação, compartilhamento e uso. Conforme observado na Figura 1, todos eles estão presentes no modelo escolhido.

O processo de captura é responsável pela coleta e identificação de conhecimento relevante dentro da organização (CHEN; EDGINGTON, 2005; CARRION et al., 2012). Isso pode ser feito por meio de várias fontes, como documentos, relatórios, entrevistas, observação de processos e interações com especialistas (ABOU-ZEID, 2002). A captura de conhecimento pode ocorrer formalmente, por meio de proces-

sostruturados, ou informalmente, por meio de conversas informais e experiências cotidianas.

O processo de criação estabelece que o conhecimento novo e original é gerado por meio da interação entre pessoas e ideias. Pode ocorrer através de atividades como brainstorming, experimentação, pesquisa e desenvolvimento de novos produtos ou processos, promovendo a interação entre o conhecimento tácito e explícito dos indivíduos (LYLES; SCHWENK, 2007). Este processo tem sido considerado um papel importante, pois gera novos conhecimentos dentro da organização e isso pode ser transformado em fator-chave de sucesso e inovação contínua (BHATT, 2000; MALHOTRA, 2000; EVANS et al., 2014).

O processo de armazenamento ocorre após a captura e criação do conhecimento, é necessário armazená-lo de maneira acessível para uso futuro. Isso envolve a organização e a estruturação do conhecimento de forma clara e estruturada para facilitar sua busca e recuperação posterior (EVANS et al., 2014).

O processo de disseminação tem como foco a distribuição ativa do conhecimento para as pessoas ou áreas relevantes dentro da organização. A disseminação envolve tornar o conhecimento acessível, disponibilizando-o de maneira estruturada e organizada. Isso pode incluir a criação de bancos de dados, repositórios de documentos, sistemas de gestão do conhecimento ou outras formas de armazenamento e acesso ao conhecimento. O objetivo principal é garantir que o conhecimento esteja disponível para aqueles que possam se beneficiar dele. A disseminação é geralmente realizada por meio de canais formais, como treinamentos, apresentações, manuais, relatórios ou boletins informativos. Tenório et al. (2017) destaca que o foco da disseminação é garantir que as informações e conhecimentos sejam amplamente divulgados para alcançar um público mais amplo.

O processo de compartilhamento, por outro lado, é um processo mais interativo e colaborativo (IPE, 2003; EVANS et al., 2014). Envolve a troca ativa de informações e conhecimentos entre os membros da organização, visando criar um ambiente propício à colaboração, aprendizagem mútua e sinergia. O compartilhamento ocorre por meio de interações sociais, como discussões em grupo, reuniões, comunicação informal, comunidades de prática, plataformas colaborativas ou redes sociais internas. O foco do compartilhamento é estimular a participação ativa dos indivíduos, facilitar a transferência de conhecimento entre eles e promover uma cultura de colaboração e inovação (ASRAR-UL-HAQ; ANWAR, 2016).

O processo de uso corresponde à etapa final da gestão do conhecimento,

sendo a aplicação do conhecimento nas atividades e processos da organização. Isso implica utilizar o conhecimento disponível para tomar decisões informadas, resolver problemas, inovar, melhorar a eficiência e alcançar os objetivos organizacionais. Por mais explícito que o conhecimento esteja disponível, nesta etapa, para utilizá-lo, o conhecimento tácito será aplicado pelos indivíduos. Existem muitas formas de usar o conhecimento, como a criação de comunidades de prática (EVANS et al., 2014). Tenório et al. (2017) constatou que o processo de uso ocorre conforme o conhecimento será aplicado. A aplicação do conhecimento ocorre em todas as áreas funcionais da organização e é fundamental para gerar valor a partir do conhecimento adquirido.

A implementação de projetos de Gestão do Conhecimento tornou-se mais fácil com a ajuda de ferramentas tecnológicas. Todas as diferentes etapas do processo de gestão do conhecimento podem ser atendidas com a ajuda de modernas ferramentas (MITTAL; KUMAR, 2019). A gestão do conhecimento não é uma disciplina que envolve somente tecnologia, também trata-se de administrar pessoas, cultura, práticas e estruturas organizacionais e utilizá-las para os objetivos organizacionais, o que torna mais sensível o desenvolvimento de um apoio computacional. Para isso, é importante que todos os processos do ciclo sejam abstraídos e respeitados na ferramenta.

2.3 Metodologias ágeis

Uma das maiores inovações no desenvolvimento de software foi a introdução de métodos ágeis que promoveram a adaptação de processos em um ambiente em constante mudança (VLAANDEREN et al., 2011; HIGHSMITH; CONSORTIUM, 2002). As abordagens tradicionais de gerenciamento de projetos e desenvolvimento de software, como o modelo em cascata, mostraram-se inadequadas para lidar com as rápidas mudanças nas demandas dos clientes e as incertezas inerentes ao processo de desenvolvimento.

Foi nesse contexto que surgiram as metodologias ágeis. As metodologias ágeis oferecem uma abordagem alternativa e flexível para o desenvolvimento de projetos, que se concentra na entrega incremental de valor e na colaboração estreita entre as partes interessadas. Essas metodologias têm se tornado cada vez mais populares, não apenas no desenvolvimento de software, mas também em outras áreas que buscam eficiência, inovação e resultados rápidos.

No centro das metodologias ágeis estão os valores expressos no Manifesto Ágil, um documento criado por um grupo de profissionais de software em 2001 (COHN, 2010). Os quatro valores fundamentais - indivíduos e interações ao invés de processos

e ferramentas; executar software em vez de documentação extensa; colaboração do cliente em vez de negociação de contratos; responder a mudanças em vez de seguir um plano - destacam a importância de colocar as pessoas em primeiro lugar, entregar software funcional, envolver os clientes no processo e ser capaz de se adaptar às mudanças de requisitos.

As metodologias ágeis introduziram uma nova visão sobre como desenvolver software (DINGSØYR et al., 2012). Isso as diferenciou dos métodos tradicionais por ter um conjunto de práticas com foco maior nas pessoas do que nos processos (CRISTAL et al., 2008). Além disso, os métodos ágeis podem potencializar o crescimento da produtividade nas equipes de desenvolvimento de software (MELO et al., 2013), causando um impacto positivo também na qualidade do produto final (AHMED et al., 2010).

A entrega incremental e iterativa é uma das principais características dessas metodologias, permitindo que o trabalho seja dividido em partes menores e entregue em intervalos curtos de tempo, conhecidos como iterações ou sprints. Isso possibilita que as equipes obtenham feedback rápido dos clientes e ajustem o produto em desenvolvimento de acordo com suas necessidades.

Outro aspecto crucial das metodologias ágeis é a colaboração intensa entre as partes interessadas, os stakeholders. As equipes ágeis valorizam a comunicação face a face e a interação constante entre os membros da equipe, bem como a colaboração próxima com os clientes e usuários finais (SCHWABER; SUTHERLAND, 2017). Isso promove uma compreensão mais profunda das necessidades e requisitos, permitindo que as equipes atendam às expectativas dos clientes de forma mais eficaz.

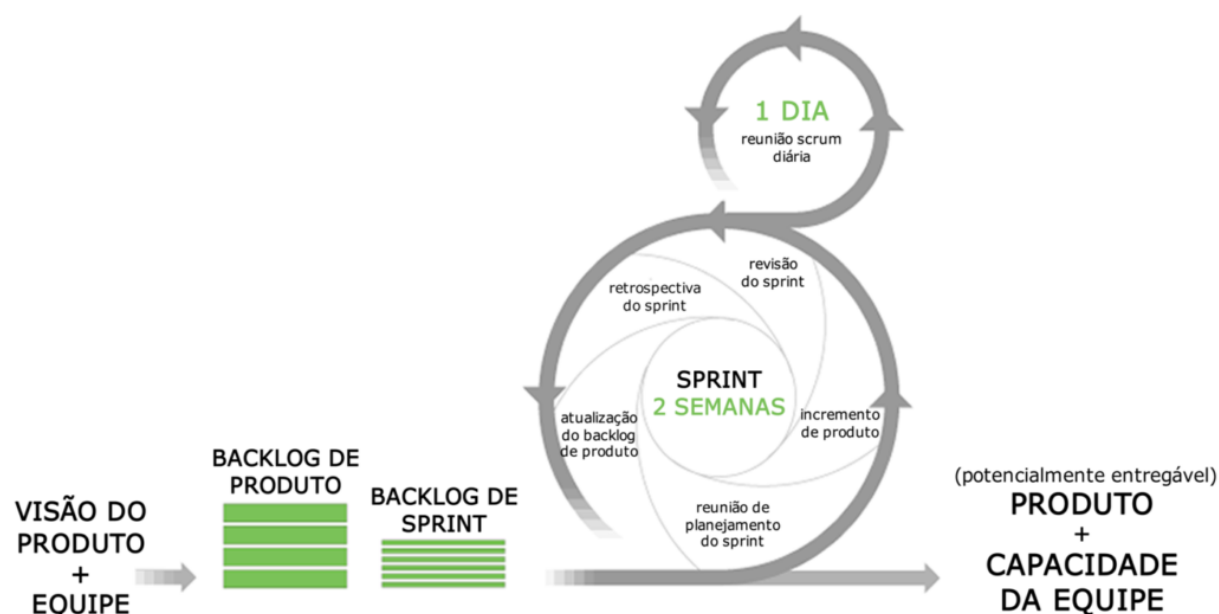
A qualidade é uma preocupação central nas metodologias ágeis. Em vez de deixar os testes e verificações para o final do projeto, as equipes ágeis incorporam práticas de qualidade em cada etapa do processo. Além disso, as metodologias ágeis incentivam a autogestão das equipes (ANDERSON, 2010). Ao invés de hierarquias rígidas e controle centralizado, as equipes ágeis têm mais autonomia para tomar decisões e se organizar internamente. Isso aumenta a responsabilidade individual e coletiva, promovendo uma maior motivação e engajamento dos membros da equipe.

2.4 Framework Scrum

Existem várias metodologias ágeis, como a Extreme Programming (XP) (JEFFRIES et al., 2000; AKHTAR et al., 2022), Lean (POPPENDIECK; POPPENDIECK, 2003) e Kanban (ANDERSON, 2010). Segundo Schwaber e Sutherland (2017) uma das metodologias ágeis mais populares e amplamente adotadas é o Scrum e, portanto,

foi selecionado para uso no presente trabalho. Ele fornece uma estrutura flexível para o gerenciamento de projetos, com ênfase na entrega contínua de valor e na colaboração entre as equipes (CERVONE, 2011; MORANDINI et al., 2021). Ele permite que as pessoas resolvam problemas complexos e adaptativos, entregando produtos de qualidade de forma produtiva e criativa (SCHWABER; SUTHERLAND, 2017). No Scrum, sua estrutura gira em torno de práticas de gerenciamento e desenvolvimento de produtos, dividindo o processo de desenvolvimento de software em fases iguais (chamadas de Sprints) que incluem atividades de monitoramento e feedback. Essas atividades são realizadas em reuniões diárias e rápidas com toda a equipe, que visa identificar e corrigir deficiências e/ou impedimentos durante a execução dos processos (DÖNMEZ et al., 2016). O Scrum estabelece práticas fundamentais em relação aos papéis, eventos e artefatos necessários para realizar o que é chamado de “Ciclo Scrum”, apresentado na Figura 2.

Figura 2 – Visão geral das etapas previstas no ciclo Scrum.



Fonte: <https://www.semeru.com.br/blog/o-ciclo-de-vida-do-framework-scrum/> Acesso em: 30 de maio de 2023. Último acesso em 11 de julho de 2023

Os papéis presentes no Scrum são: Product Owner, Equipe Scrum e Scrum Master. O Product Owner representa o interesse da equipe. A função dele é expressar com transparência o que deve ser realizado, determinando requisitos, funcionalidades, objetivos e plano de entrega (MARÇAL et al., 2007). A Equipe Scrum é composta por desenvolvedores, designers, testadores e arquitetos. Tem autoridade e responsabilidade por muitos aspectos do desenvolvimento de software, incluindo planejamento, programação, atribuição de tarefas aos stakeholders (partes interessadas) e tomada

de decisão para atingir os objetivos propostos (MOE et al., 2010). O Scrum Master desempenha o papel de líder facilitador do processo. Esclarece o andamento do projeto e propaga os valores do Scrum para toda a equipe, garantindo a prática de reuniões diárias e mantendo o desempenho da equipe (LEI et al., 2017).

Os eventos presentes no Scrum são divididos em: Planejamento da Sprint, Daily Scrum, Revisão da Sprint, Retrospectiva da Sprint. O Planejamento da Sprint é um evento realizado no início de cada sprint, no qual o Product Owner e a equipe de desenvolvimento definem o objetivo da sprint e selecionam as tarefas a serem realizadas. A equipe estima o esforço necessário e cria o plano para a sprint. Já a Daily Scrum é uma reunião diária de curta duração na qual a equipe de desenvolvimento compartilha o progresso do trabalho, sincroniza as atividades e identifica possíveis obstáculos. A Revisão da Sprint é uma reunião realizada ao final de cada sprint, na qual a equipe de desenvolvimento demonstra o trabalho concluído. O Product Owner e outros stakeholders dão feedback sobre o produto entregue e discutem possíveis ajustes e melhorias. Por fim, o evento de Retrospectiva da Sprint é uma reunião realizada ao final de cada sprint, na qual a equipe de desenvolvimento revisa o processo de trabalho e identifica oportunidades de melhoria. São discutidos os aspectos positivos, os problemas enfrentados e as ações a serem tomadas para aprimorar o próximo sprint (LEI et al., 2017).

São três os artefatos existentes no Scrum: Backlog do produto, Backlog da sprint, incremento. O Backlog do produto é uma lista que contém todos os requisitos, funcionalidades e melhorias desejadas para o produto. O Product Owner é responsável por manter e priorizar esse artefato. Já o Backlog da sprint é uma lista de tarefas selecionadas pelo time de desenvolvimento a partir do Backlog do produto para serem realizadas durante a Sprint atual. Essa lista é monitorada e atualizada ao longo da sprint. Por fim, tem-se o incremento, o qual é o resultado tangível do trabalho realizado durante uma sprint. É uma versão funcional do produto que pode ser potencialmente lançada ou entregue aos stakeholders.

O ciclo do Scrum tem início com o planejamento da Sprint. O Product Owner trabalha em colaboração com as partes interessadas para definir e priorizar os itens do backlog do produto (SCHWABER; SUTHERLAND, 2017). A equipe Scrum realiza uma reunião de planejamento do sprint para selecionar as tarefas que serão concluídas durante o próximo sprint. Durante essa reunião, a equipe estima a quantidade de trabalho necessário para cada tarefa e define os objetivos específicos da sprint.

Uma vez estruturada, a Sprint é iniciada. O período dela possui um tempo fixo, geralmente de 2 a 4 semanas, durante o qual a equipe Scrum trabalha para

concluir as tarefas selecionadas. A equipe realiza as reuniões Daily Scrum. Cada membro da equipe compartilha o progresso feito desde a última reunião, discute os obstáculos encontrados e planeja as próximas etapas. Durante a Sprint, a equipe colabora para desenvolver as funcionalidades selecionadas. As tarefas são divididas em itens menores e atribuídas aos membros da equipe com base em suas habilidades e expertise.

Ao final da sprint, a equipe realiza a reunião de revisão do sprint. Durante essa reunião, a equipe demonstra o trabalho concluído ao Product Owner e as outras partes interessadas. Esse trabalho concluído no final da sprint é chamado de incremento, que nada mais é que a soma de todas as histórias de usuário e funcionalidades desenvolvidas durante o sprint. O feedback é recebido e discutido, e podem ocorrer ajustes no backlog do produto com base nas necessidades do cliente e nas mudanças do mercado.

Após a revisão do sprint, a equipe realiza uma retrospectiva para refletir sobre o sprint anterior. Durante a retrospectiva, a equipe analisa o que funcionou bem, quais obstáculos foram enfrentados e identifica ações de melhoria para o próximo sprint. É uma oportunidade para a equipe aprender e aprimorar continuamente suas práticas e processos.

Com base no feedback recebido e nas prioridades atualizadas do backlog do produto, a equipe Scrum realiza uma reunião de planejamento do próximo sprint. Durante essa reunião, eles selecionam as tarefas a serem realizadas no próximo sprint e definem os objetivos específicos. Esse ciclo de planejamento, execução, revisão e retrospectiva continua até que o produto seja considerado concluído ou até que o projeto seja encerrado.

O ciclo do Scrum é projetado para ser iterativo e incremental, permitindo a entrega contínua de valor ao cliente. Por meio das iterações regulares, a equipe Scrum pode responder rapidamente a mudanças e ajustar suas abordagens para atender às necessidades em constante evolução.

2.5 Software de apoio

A gestão de conhecimento e as metodologias ágeis são duas áreas distintas, mas complementares, quando se trata de desenvolvimento de software. Enquanto a gestão de conhecimento é responsável pela captura, organização e compartilhamento do conhecimento em uma organização, a metodologia ágil Scrum é um framework para gerenciar projetos de desenvolvimento de software. Isso torna desafiador encontrar um

software capaz de atender a ambas.

Em relação à gestão de conhecimento, um software eficaz deve fornecer recursos avançados para gerenciar documentos, criar bases de conhecimento, estabelecer comunidades de prática e facilitar a colaboração e o aprendizado entre os membros da equipe. Além disso, um software de gestão de conhecimento geralmente oferece recursos de pesquisa avançados para facilitar a recuperação rápida de informações.

Por outro lado, o Scrum requer um conjunto diferente de recursos para auxiliar no gerenciamento de projetos ágeis. O Scrum envolve a definição de backlog de produtos, sprints, tarefas, atribuição de responsabilidades, monitoramento do progresso e colaboração da equipe. Um software voltado para o Scrum deve oferecer recursos como quadros Kanban ou de tarefas com o intuito de facilitar a visualização e o acompanhamento do progresso das atividades da equipe. Esses quadros permitem que as tarefas sejam visualizadas em colunas, representando diferentes etapas do fluxo de trabalho, como “A fazer”, “Em andamento” e “Concluído” (GARCIA et al., 2020). Gráficos de Burndown também são considerados um recurso que um software de gestão de projetos deve contemplar. Esses gráficos fornecem uma visão clara e objetiva do progresso do projeto, permitindo que a equipe acompanhe o trabalho restante e tome medidas corretivas, se necessário. Além disso, os gráficos de Burndown facilitam a comunicação entre a equipe de desenvolvimento e os stakeholders, ao oferecerem uma representação visual do progresso do projeto, ajudando a alinhar as expectativas e identificar possíveis problemas (ARAFEEN; BOSE, 2009).

A dificuldade de encontrar um software que atenda tanto a gestão de conhecimento quanto a aplicação do Scrum surge porque a maioria das ferramentas disponíveis no mercado tende a se especializar em uma área específica. Alguns softwares focam exclusivamente na gestão de conhecimento, enquanto outros são direcionados ao gerenciamento de projetos ágeis, com recursos específicos para o Scrum.

Dependendo das necessidades específicas de uma organização, por exemplo, pode ser necessário utilizar diferentes ferramentas especializadas em cada área. Nesses casos, é essencial garantir uma integração eficiente entre os diferentes softwares para permitir uma transição suave de informações e evitar lacunas de conhecimento.

2.5.1 Apoio automatizado à gestão do conhecimento

Para facilitar a gestão de conhecimento, algumas ferramentas e tecnologias estão disponíveis, porém entre essas ferramentas existem as que só oferecem a versão paga e as que oferecem versão paga e também a gratuita, sendo que a versão gratuita

normalmente apresenta algumas limitações que restringem o uso em relação à versão paga. Essas ferramentas oferecem recursos que auxiliam na criação, armazenamento, organização e disseminação de informações e conhecimentos relevantes para os membros da organização. Ao utilizar essas ferramentas, as empresas podem centralizar o conhecimento, promover a colaboração e permitir o acesso rápido e fácil às informações necessárias.

Um exemplo de software nessa categoria é o KnowledgeOwl¹. Ele permite criar uma base de conhecimento centralizada, onde os usuários podem acessar e contribuir com informações relevantes. O KnowledgeOwl oferece recursos de pesquisa avançados, categorização de conteúdo, controle de versões e comentários, facilitando a organização e a manutenção de um repositório de conhecimento atualizado.

Outra opção é o Bloomfire², um software que permite a criação de uma plataforma de compartilhamento de conhecimento interno. Ele permite criar comunidades de prática, onde os usuários podem compartilhar informações, fazer perguntas, colaborar em projetos e aprender mutuamente. O Bloomfire oferece recursos como busca inteligente, tags, comentários e métricas de engajamento para facilitar a descoberta e o acesso ao conhecimento relevante.

O Document360³ é mais um exemplo de sistema de gestão do conhecimento. Ele permite criar uma base de conhecimento bem estruturada, com categorias, tags e uma interface amigável para navegação. O Document360 também oferece recursos avançados de colaboração, permitindo que várias equipes contribuam e atualizem o conteúdo colaborativamente. Além disso, ele oferece recursos de análise e métricas para monitorar o desempenho e a utilidade do conhecimento compartilhado.

Esses sistemas de gestão do conhecimento desempenham um papel crucial na promoção da aprendizagem contínua, no compartilhamento eficiente de informações e no desenvolvimento de um ambiente de trabalho colaborativo. Eles auxiliam as organizações a preservar o conhecimento institucional, evitar a redundância de esforços e promover a inovação. Ao permitir o acesso rápido e fácil ao conhecimento relevante, esses softwares contribuem para a agilidade e a produtividade das equipes, facilitando a resolução de problemas, a tomada de decisões informadas e a melhoria contínua dos processos.

¹ <<https://www.knowledgeowl.com>> Último acesso em 24 de junho de 2023

² <<https://bloomfire.com>> Último acesso em 24 de junho de 2023

³ <<https://document360.com>> Último acesso em 24 de junho de 2023

2.5.2 Apoio automatizado à gestão de projetos

As ferramentas de gerenciamento de projetos, por sua vez, desempenham um papel essencial na gestão de projeto ágil, permitindo a organização, o acompanhamento e a coordenação de projetos eficientemente. Esses softwares fornecem recursos que facilitam a atribuição de tarefas, o monitoramento de prazos, o planejamento de recursos e a colaboração entre os membros da equipe.

Um exemplo popular de software nessa categoria é o Jira⁴. Grande parte do poder do Jira está em seus fluxos de trabalho personalizáveis, que oferecem transições, validações e notificações personalizáveis para ajustar seu processo de desenvolvimento nos mínimos detalhes. O Jira é adequado para projetos de desenvolvimento de software ágil por vários motivos, por exemplo: ele permite a criação de quadros Kanban para visualizar o fluxo de trabalho do projeto.

No Jira também há a possibilidade da criação de um backlog do produto, sendo que a equipe pode gerenciar e classificar o backlog do produto, atribuindo pontos de história, estimativas de esforço e prioridades a cada item. O Jira permite a criação de sprints e ajuda a planejar e acompanhar os sprints, incluindo a definição de datas de início e término, seleção de itens do backlog para inclusão no sprint e rastreamento do progresso durante o sprint.

O Jira facilita, ainda, a realização das reuniões do Scrum, como a reunião de planejamento do sprint, a reunião diária e a revisão do sprint. Os membros da equipe podem usar o Jira para revisar o backlog, atualizar o status das tarefas durante a reunião diária e dar feedback durante a revisão do sprint. O Jira também oferece vários relatórios específicos do Scrum, como o relatório de velocidade, que mostra a quantidade de trabalho concluída em cada sprint, e o relatório de tendência de sprint, que ajuda a acompanhar o desempenho ao longo do tempo. Esses relatórios fornecem informações valiosas para a equipe e facilitam a melhoria contínua do processo.

Embora o Jira seja mais conhecido por suas funcionalidades de gerenciamento de projetos e suporte ao desenvolvimento ágil de software, ele também oferece recursos que podem ser utilizados para a gestão de conhecimento. Ele permite que os usuários criem e anexem documentos, arquivos e imagens diretamente nos problemas, tarefas ou histórias. Isso possibilita a documentação de informações importantes, como especificações técnicas, manuais de usuário, exemplos de código e outros materiais relevantes para o conhecimento da equipe. Nele, também é possível a criação de categorias personalizadas e etiquetas para organizar e classificar os problemas ou tarefas.

⁴ <https://www.atlassian.com/software/jira> Último acesso em 24 de junho de 2023

Essa funcionalidade pode ser usada para categorizar o conhecimento por temas, áreas de especialização ou tipos de problema. As etiquetas podem ser adicionadas aos itens para facilitar a recuperação de informações relacionadas.

O Jira oferece recursos de comentários e discussões em tempo real. Os membros da equipe podem adicionar comentários aos problemas ou tarefas, permitindo a troca de informações, feedback e discussões sobre o trabalho realizado. Essa funcionalidade facilita a captura de conhecimento contextual e a colaboração entre os membros da equipe. A ferramenta conta também com recursos de pesquisa avançada que permitem aos usuários encontrar informações específicas em projetos, problemas ou tarefas. Isso facilita a localização de conhecimento anteriormente registrado, como soluções para problemas semelhantes, respostas a perguntas frequentes ou informações relevantes sobre um determinado assunto.

Outra opção é o Asana⁵, uma ferramenta de gerenciamento de projetos que oferece recursos para planejamento, rastreamento de tarefas e colaboração. O Asana permite criar projetos, atribuir tarefas, definir prazos e acompanhar o progresso em um formato de lista ou quadro. Ele também oferece recursos de comunicação integrados, como comentários e notificações, para facilitar a colaboração e a troca de conhecimento entre os membros da equipe.

O Trello⁶ também é uma ferramenta muito utilizada em metodologias ágeis. Ele é bem parecido com o Jira, porém mais simplificado. O Trello é conhecido por sua interface simples e intuitiva. É fácil de usar e ideal para equipes menores ou projetos simples. Por outro lado, o Jira é mais complexo e abrangente, projetado para lidar com projetos maiores e equipes mais técnicas. Ele oferece recursos mais avançados, mas também pode ter uma curva de aprendizado mais íngreme.

2.5.3 Análise de software

A partir das ferramentas citadas nas subseções 2.5.1 e 2.5.2, observa-se que as primeiras — KnowledgeOwl, Bloomfire, Document360 — são ferramentas criadas com foco em gestão de conhecimento, o que faz com elas tenham mais funcionalidades específicas, tornando mais restrita a utilização delas em um contexto ágil. Da mesma forma que as ferramentas citadas por fim — Jira, Asana, Trello — são ferramentas criadas com o foco na gestão ágil de projetos, o que as tornam mais restritas a um ambiente de gestão de conhecimento.

Diante desse cenário, tais ferramentas foram avaliadas, e o resultado obtido é

⁵ <https://asana.com/pt> Último acesso em 24 de junho de 2023


⁶ <https://trello.com> Último acesso em 24 de junho de 2023

ilustrado na Tabela 1. Para a avaliação, foram escolhidas as principais características almeçadas para softwares com foco na gestão de conhecimento e softwares com foco em metodologias ágeis, a saber:


- Armazenamento e organização: Deve permitir o armazenamento centralizado de informações, como documentos, artigos, tutoriais, FAQs, entre outros. Além disso, deve oferecer recursos robustos de organização, como categorias, tags e estruturas hierárquicas, para facilitar a navegação e a busca de informações;
- Pesquisa: Uma função de pesquisa é essencial para permitir que os usuários encontrem rapidamente as informações de que precisam no sistema de gestão de conhecimento. Isso pode incluir recursos como pesquisa por palavra-chave, filtros, pesquisa em texto completo e sugestões de pesquisa;
- Backlog do produto: Deve permitir a criação e o gerenciamento do backlog do produto, onde as histórias do usuário são capturadas, priorizadas e estimadas;
- Planejamento e rastreamento de sprints: Deve fornecer recursos para o planejamento e o acompanhamento dos sprints, incluindo a atribuição de tarefas, estimativas de tempo, rastreamento do progresso e a visualização das metas e dos prazos;
- Quadro Kanban: Um quadro Kanban visual, com colunas que representam o fluxo de trabalho, ajuda a equipe a acompanhar o status das tarefas e a priorizar o trabalho em andamento. Essa visualização facilita a colaboração e a comunicação na equipe;
- Colaboração em equipe: Deve permitir a colaboração e a comunicação efetiva entre os membros da equipe. Recursos como comentários, notificações, menções e compartilhamento de arquivos podem melhorar a colaboração e a produtividade da equipe;
- Métricas e relatórios: É importante que o software ofereça recursos para o acompanhamento de métricas e geração de relatórios, como velocidade da equipe e gráfico burndown. Essas informações auxiliam na tomada de decisões e no acompanhamento do desempenho do projeto.

Tabela 1 – Resultado obtido a partir da análise de softwares voltados para gestão de conhecimento e de projetos.


		Características							
Softwares		Armazenamento e Organização	Pesquisa Avançada	Backlog do Produto	Planejamento e rastreamento de sprints	Quadro Kanban	Colaboração em equipe	Métricas e relatórios	Versão gratuita
Gestão de conhecimento	KnowledgeOwl	Atende	Atende	Não atende	Não atende	Não atende	Atende	Atende em partes	Atende em partes
	Bloomfire	Atende	Atende	Não atende	Não atende	Não atende	Atende	Atende em partes	Não atende
	Document360	Atende	Atende	Não atende	Não atende	Não atende	Atende	Atende em partes	Atende
Gestão de projetos	Jira	Atende em partes	Atende em partes	Atende	Atende	Atende	Atende	Atende	Atende em partes
	Asana	Atende em partes	Atende em partes	Atende	Atende	Atende	Atende	Atende em partes	Atende
	Trello	Atende em partes	Atende em partes	Atende em partes	Atende em partes	Atende	Atende	Atende em partes	Atende



Atende



Atende em partes



Não atende

Fonte: Elaborado pelo autor.

Para a análise dos softwares, foram considerados três resultados possíveis. O primeiro deles indica que o software segue os critérios definidos pela característica, logo, ele “atende” os requisitos estabelecidos; se o software não atende completamente a característica definida, ele “atende em partes” o requisito; já se o software não segue nada do que foi definido pela característica, ele “não atende” ao requisito.

Pode-se depreender a partir dos resultados da avaliação que é difícil concentrar todas as características em um único software. Logo, a proposta de uma solução capaz de apoiar a gestão de conhecimento de forma integrada a uma abordagem ágil, como o Scrum, representa uma importante contribuição do presente trabalho.

2.6 Considerações finais

Neste capítulo, foram expostos os conceitos teóricos fundamentais para alicerçar o trabalho. Além disso, foram expostas as características necessárias para se ter um software para a gestão de conhecimento e o gerenciamento ágil de projetos. A partir delas, foi possível identificar carências nos softwares atuais desse segmento. Neste contexto, as maiores ausências estão relacionadas à união dos pilares da gestão de conhecimento — armazenamento, organização, pesquisa avançada — juntamente aos pilares do gerenciamento ágil de projetos — backlog do produto, planejamento e rastreamento de sprints, quadro Kanban, colaboração em equipe.

No próximo capítulo é apresentado o detalhamento sobre a construção do protótipo e a descrição de suas funcionalidades acompanhadas das ilustrações das telas correspondentes.

Capítulo 3 - Detalhamento do KamTool

Atualmente, muitas soluções no mercado abordam separadamente a gestão de conhecimento e a gestão de projetos, levando a lacunas e deficiências na colaboração e na execução de tarefas. A integração dessas duas áreas permite que equipes aproveitem o conhecimento acumulado de maneira ágil e eficaz durante todas as fases do projeto. Isso não apenas acelera a resolução de desafios complexos, mas também promove um ambiente de aprendizado contínuo, melhorando a capacidade de adaptação da equipe e impulsionando a inovação, demonstrando a importância de desenvolver uma ferramenta desse tipo.

Neste capítulo são expostos detalhes sobre o desenvolvimento do protótipo. Para tanto, na seção 3.1, tem-se a descrição geral, apresentando de uma forma mais resumida toda a arquitetura do protótipo, bem como os softwares envolvidos no processo de desenvolvimento. Já na seção 3.2, tem-se a definição da arquitetura, apresentando mais detalhadamente todo o aspecto técnico relativo à arquitetura do protótipo. Na seção 3.3, tem-se a definição das funcionalidades, apresentando todas as funcionalidades pertencentes ao protótipo, bem como ilustração das telas do mesmo. Por fim, na seção 3.4, tem-se as considerações finais sobre o capítulo.

3.1 Descrição Geral

Com base no levantamento bibliográfico e na análise de softwares, foram definidos os requisitos e funcionalidades almejadas para o desenvolvimento do protótipo. A partir destes requisitos, foi definida uma arquitetura de software com foco em deixar o software dinâmico e responsivo para a utilização no contexto de gestão ágil de conhecimento. Com isso, espera-se contribuir para o processo de compartilhamento de conhecimento entre membros de uma equipe, durante o desenvolvimento de um projeto de software.

A escolha do acrônimo KamTool para uma ferramenta de gestão ágil de conhecimento (Knowledge Agile Management) foi estratégica. A combinação das letras KAM reflete diretamente a natureza da ferramenta, indicando sua função principal de gestão de conhecimento de forma ágil. A inclusão do termo Tool (ferramenta em inglês) no final reforça claramente sua aplicação prática.

Para o desenvolvimento da camada de front end está sendo utilizado o React⁷,

⁷ <https://react.dev> Último acesso em 17 de agosto de 2023

biblioteca front end JavaScript. A camada de back end utiliza o Strapi⁸, um Content Management System - CMS, de código aberto baseado no Node JS. A camada de banco de dados é gerenciada pelo PostgreSQL⁹, um sistema gerenciador de banco de dados objeto-relacional, desenvolvido como projeto de código aberto. Os softwares utilizados são: Visual Studio Code para a construção da aplicação; PostgreSQL responsável pelo banco de dados; Photopea para a edição das imagens.

3.2 Definição da Arquitetura

O padrão de arquitetura estabelecido para o KamTool é de microsserviços. O gerenciamento de cada microsserviço é feito pelo Docker¹⁰, que é software de código aberto usado para implantar aplicativos dentro de containers virtuais. Um container¹¹ pode ser definido como um ambiente isolado, disposto em um servidor, que divide um único host de controle. Os containers funcionam de uma forma um pouco parecida das máquinas virtuais, mas de uma maneira muito mais específica e granular. Em uma máquina virtual, é possível utilizar diversos recursos e ferramentas, porém tudo roda em um mesmo sistema operacional. No caso dos containers, a ideia é que cada container faça apenas um serviço e assuma uma só responsabilidade. Desta forma, é possível isolar os processos de cada ferramenta, garantindo que nenhuma atrapalhe o funcionamento da outra. Cada camada importante da aplicação está colocada em um container, conforme representado na Figura 3, fazendo com que cada camada se torne independente e utilize a quantidade ideal de recursos do servidor.

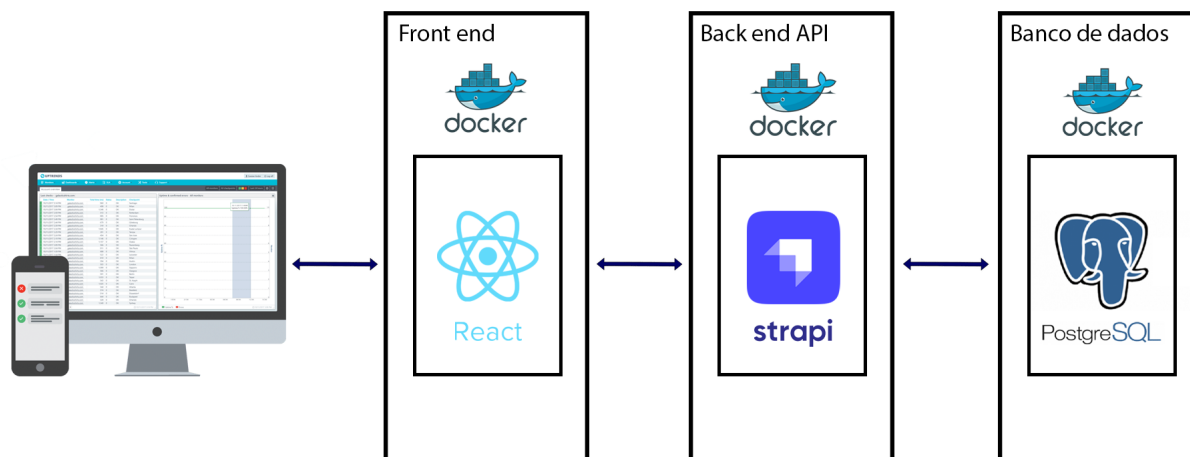
⁸ <https://strapi.io> Último acesso em 17 de agosto de 2023

⁹ <https://www.postgresql.org> Último acesso em 17 de agosto de 2023

¹⁰ <https://www.docker.com> Último acesso em 17 de agosto de 2023

¹¹ <https://esr.rnp.br/administracao-de-sistemas/containers-docker-como-utilizar/> Último acesso em 17 de agosto de 2023

Figura 3 – Arquitetura de microsserviços definida para a aplicação KamTool.



Fonte: Elaborado pelo autor.

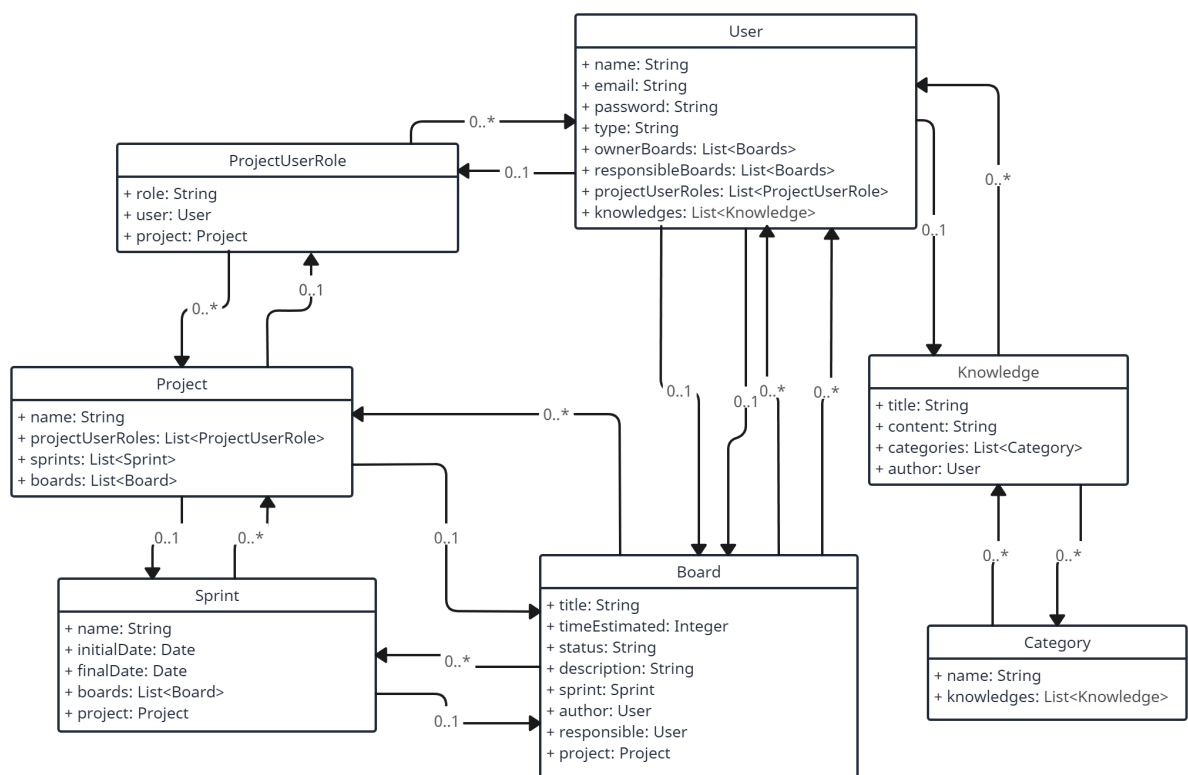
A partir da Figura 3 é possível observar que cada retângulo simboliza um container Docker, o primeiro é responsável por conter o Front end com React. O segundo container contém o Back end com Strapi. Por fim, tem-se o terceiro container com o banco de dados PostgreSQL.

Para um melhor detalhamento da estrutura estabelecida para a aplicação, têm-se as classes representadas na Figura 4. A classe User contempla, além dos dados pessoais de um usuário, como nome, email e senha, os seguintes atributos:

- type: atributo do tipo texto responsável por armazenar o tipo do usuário (default ou admin);
- ownerBoards: relação da classe User com a classe Board, no caso, relação de posse, ou seja, quando um usuário criar um board, ele será o dono do mesmo;
- responsibleBoards: relação da classe User com a classe Board, no caso, relação de responsabilidade, ou seja, o usuário pode não criar um board, mesmo assim ele pode ser o responsável por ele;

- projectUserRoles: relação da classe User com a classe ProjectUserRole, no caso, a classe ProjectUserRole é responsável por relacionar uma classe User a uma classe Project e atribuir um papel (Scrum Master, Product Owner ou Membro) para esse usuário nesse projeto. Caso o usuário seja o criador do projeto, é obrigatório que ele tenha o papel de Scrum Master ou de Product Owner;
- knowledges: relação da classe User com a classe Knowledge, no caso, relação de posse, ou seja, quando um usuário criar um documento de conhecimento, ele será o dono do mesmo.

Figura 4 – Diagrama de classe para a aplicação KamTool.



Fonte: Elaborado pelo autor.

Já a classe Project representa um projeto e possui os seguintes atributos:

- name: atributo do tipo texto responsável por armazenar o nome do projeto;
- projectUserRoles: relação da classe Project com a classe ProjectUserRole. Um projeto pode possuir vários usuários;
- sprints: relação da classe Project com a classe Sprint. Um projeto pode possuir vários sprints;

- boards: relação da classe Project com a classe Board. Um projeto pode possuir vários boards, no caso, todos que ele possui são os itens do Backlog do produto.

A classe Sprint representa um sprint de um projeto, essa classe possui:

- name: atributo do tipo texto responsável por armazenar o nome do sprint;
- initialDate: atributo do tipo data responsável por armazenar a data inicial do sprint;
- finalDate: atributo do tipo data responsável por armazenar a data final do sprint;
- boards: relação da classe Sprint com a classe Board. Um sprint pode possuir vários boards, no caso, todos são itens que existem no Backlog do produto, e então foram atribuídos a um sprint;
- project: relação da classe Sprint com a classe Project. Um sprint pode pertencer a somente um projeto.

A classe Board representa um item de um projeto, ou seja, um item pertencente ao backlog do produto desse projeto, e conseqüentemente, a uma sprint desse projeto. Essa classe possui:

- title: atributo do tipo texto responsável por armazenar o Título do board (item);
- timeEstimated: atributo do tipo inteiro responsável por armazenar o tempo estimado (em horas) que o responsável pelo item tem para concretizar a tarefa imposta no mesmo;
- status: atributo do tipo texto responsável por armazenar o status (Não iniciado, Em progresso ou Concluído) do board (item);
- description: atributo do tipo texto responsável por armazenar a descrição da tarefa do board (item);
- sprints: relação da classe Sprint com a classe Board. Um board pode pertencer a somente um sprint;
- author: relação da classe User com a classe Board. Um board pode pertencer a somente um usuário como dono;
- responsible: relação da classe User com a classe Board. Um board pode pertencer a somente um usuário como responsável;

- project: relação da classe Project com a classe Board. Um board pode pertencer a somente um projeto.

A classe Knowledge representa um documento de conhecimento pertencente à base de conhecimento do protótipo. Essa classe possui:

- title: atributo do tipo texto responsável por armazenar o Título do documento de conhecimento;
- content: atributo do tipo texto responsável por armazenar todo o conteúdo desse documento de conhecimento;
- categories: relação da classe Category com a classe Knowledge. Um documento de conhecimento pode possuir varias categorias;
- author: relação da classe User com a classe Knowledge. Um documento de conhecimento pode pertencer a somente um usuário como dono.

Por fim, classe Category representa cada categoria que um documento de conhecimento pode ter. Essa classe possui:

- name: atributo do tipo texto responsável por armazenar o nome da categoria;
- knowledges: relação da classe Knowledge com a classe Category. Uma categoria pode pertencer a vários documentos de conhecimento.

3.3 Definição das funcionalidades

As funcionalidades apresentadas pelo protótipo são descritas a seguir, tendo como base a ilustração das telas correspondentes. Porém, primeiramente é importante estabelecer os papéis dos potenciais usuários previstos para o protótipo. Existem 2 tipos de usuários:

- Usuário padrão (default): tem permissões específicas a depender do papel que ele, como membro de uma equipe vinculada a um projeto, está exercendo, bem como acesso à base de conhecimento;
- Usuário Administrador do sistema (admin): tem todas as permissões possíveis do protótipo, podendo fazer qualquer eventual alteração que seja necessária em qualquer segmento, seja qualquer projeto ou na base de conhecimento. Portanto,

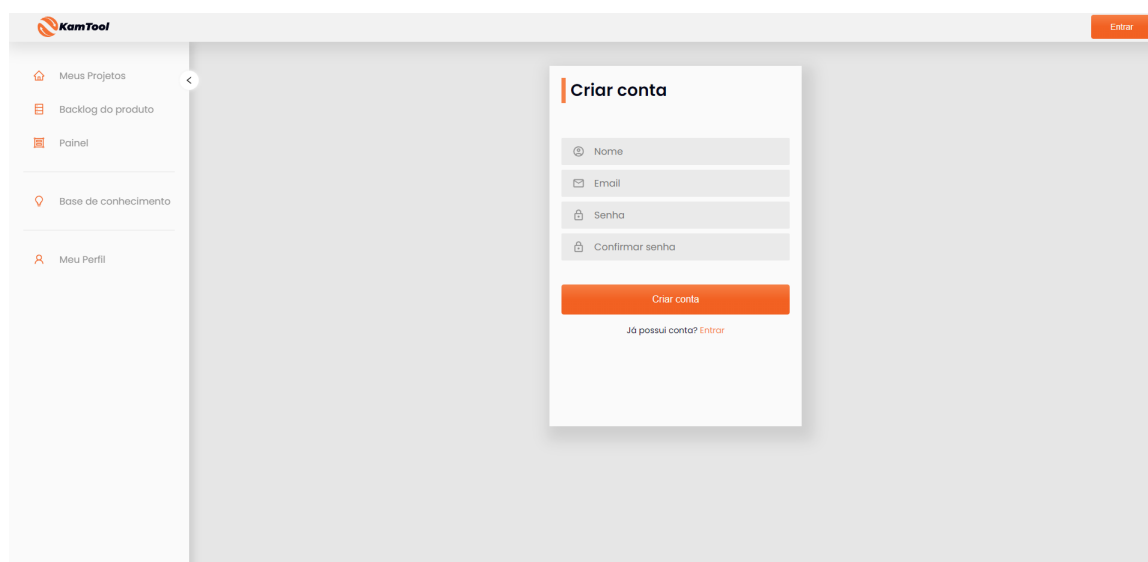
este usuário é responsável por manter a ferramenta funcional, uma vez que pode fazer as intervenções necessárias diretamente no código ou na base de dados para garantir o correto funcionamento.

Só existe um usuário Administrador, sendo definido como padrão no protótipo. Esse usuário é utilizado em casos onde se precisa fazer alguma alteração urgente de algum dado atrelado a um usuário default do qual não se tem acesso no momento ou alguma outra emergência. Já o usuário do tipo default representa todos os funcionários de uma organização que possuem acesso ao protótipo para exercer suas atividades junto aos diferentes projetos desenvolvidos. Logo, cada funcionário deve efetuar seu cadastro junto ao protótipo.

F1. Cadastro de usuários

O sistema aceita somente cadastros de usuários do tipo default, para funcionários. Cada funcionário que for utilizar o protótipo precisa efetuar o cadastro com seus dados. Os campos necessários no cadastro serão: nome, e-mail e senha, que deve possuir no mínimo 5 caracteres (Figura 5). O e-mail deve ser único para identificação do usuário dentro da aplicação e para recuperação de senha, se necessário.

Figura 5 – Ilustração da tela de cadastro de usuário.

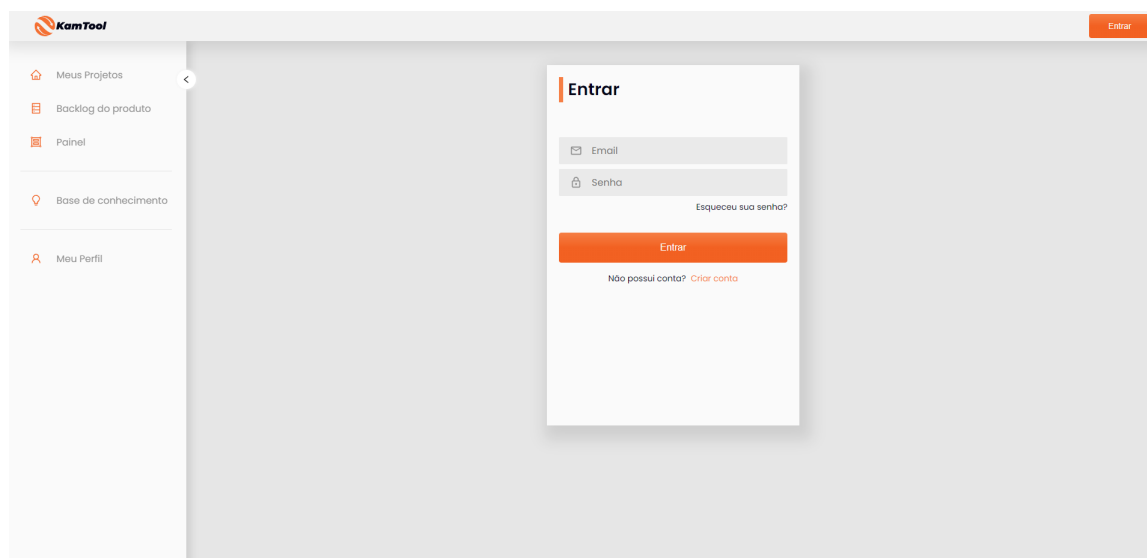
A imagem mostra a interface de usuário do KamTool. No topo, há o logotipo 'KamTool' e um botão 'Entrar' no canto superior direito. À esquerda, há um menu lateral com opções: 'Meus Projetos', 'Backlog do produto', 'Painel', 'Base de conhecimento' e 'Meu Perfil'. O conteúdo principal da tela é um formulário branco centralizado com o título 'Criar conta'. O formulário possui quatro campos de entrada: 'Nome', 'Email', 'Senha' e 'Confirmar senha'. Abaixo dos campos, há um botão laranja 'Criar conta' e um link 'Entrar' em vermelho para quem já possui uma conta.

Fonte: Elaborado pelo autor.

F2. Autenticação do usuário

Para se autenticar, o usuário deve colocar o e-mail e senha previamente indicados no momento do cadastro para acessar o protótipo (Figura 6).

Figura 6 – Ilustração da tela de autenticação.

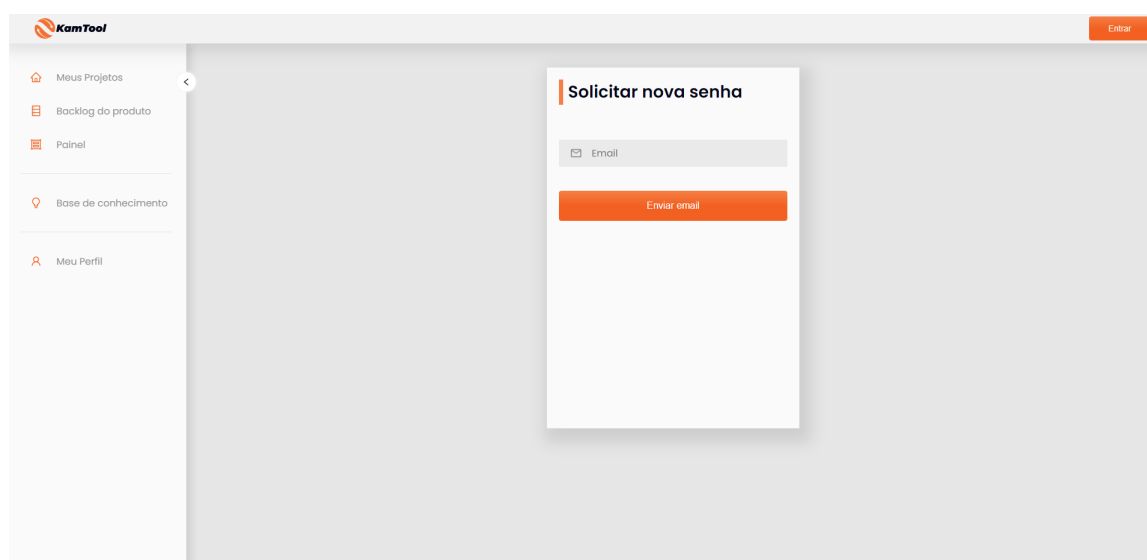


Fonte: Elaborado pelo autor.

F3. Alterações cadastrais

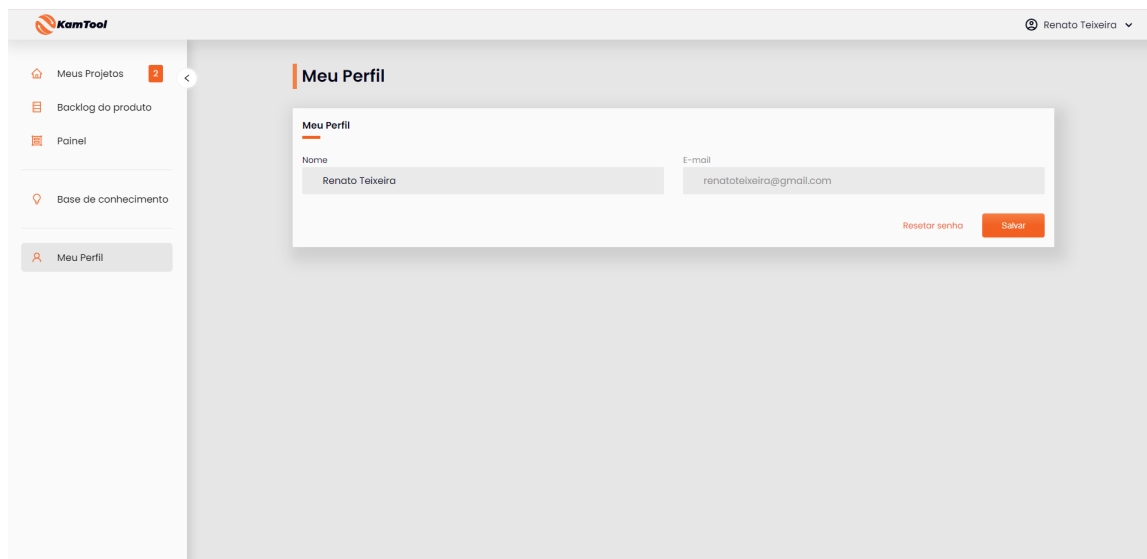
O usuário pode alterar os dados cadastrais sempre que desejar. Uma possibilidade é a troca de senha. Para tanto, o usuário deve fazer a solicitação (Figura 7) e, com isso, receber um e-mail com um link que o habilita a trocar a senha. Além disso, também é possível alterar o nome, conforme a Figura 8.

Figura 7 – Ilustração da tela para troca de senha.



Fonte: Elaborado pelo autor.

Figura 8 – Ilustração da tela para alteração de nome do usuário.

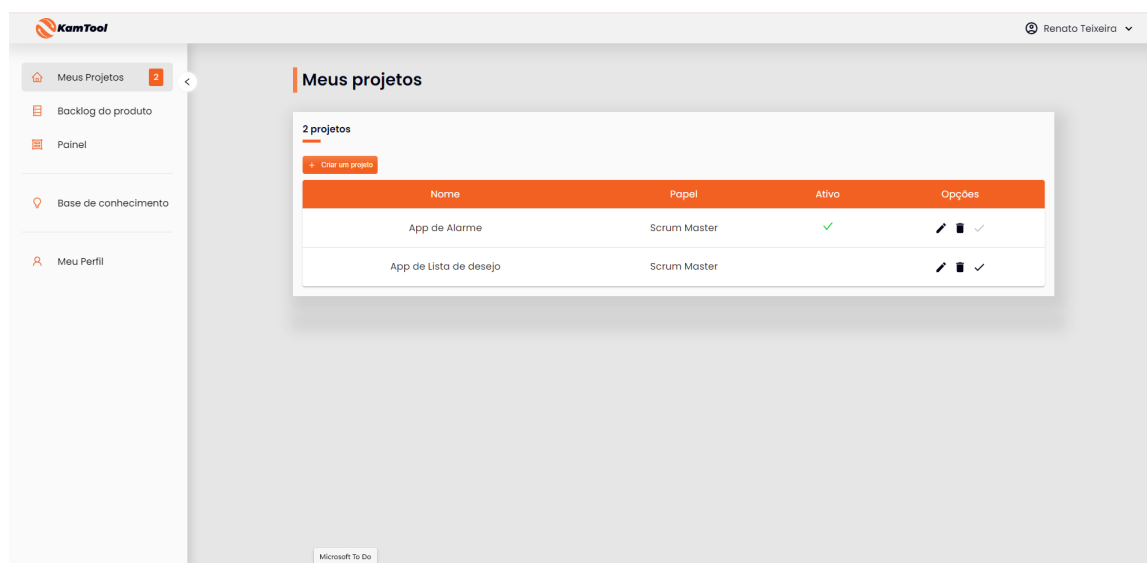


Fonte: Elaborado pelo autor.

F4. Gerenciar os projetos

O usuário pode gerenciar os seus projetos, tanto os que ele criou quanto os que ele faz parte. Ele pode excluir os projetos em que ele seja o criador. Sendo ele o criador ou integrante do projeto, ele pode ativar o projeto e então acessar as telas de backlog do produto e painel, que são referentes ao projeto ativado. (Figura 9).

Figura 9 – Ilustração da tela para gerenciar os projetos vinculados a um usuário.



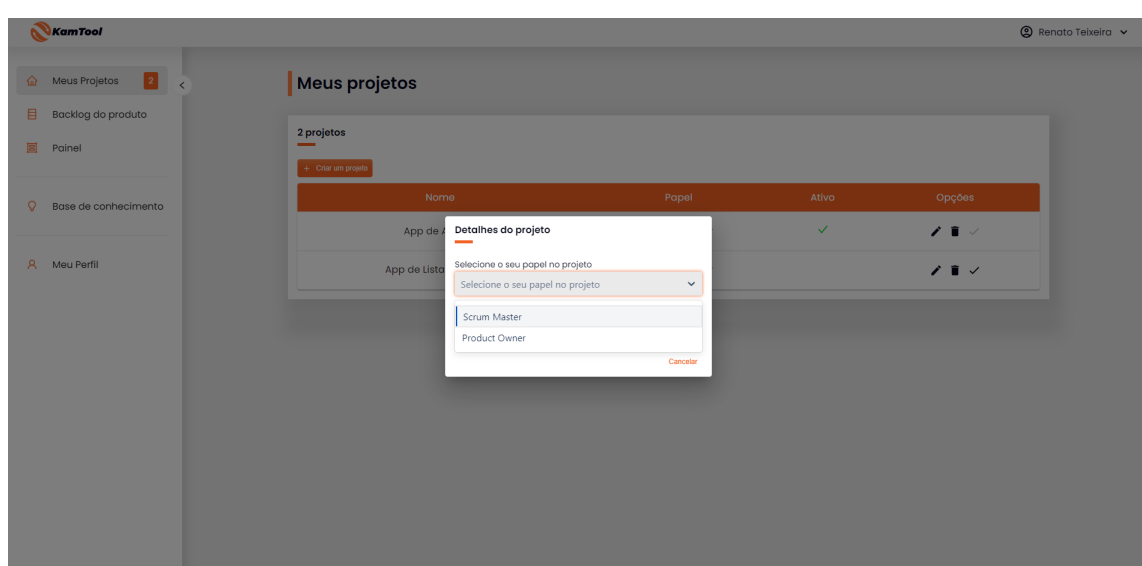
Fonte: Elaborado pelo autor.

F5. Criar e editar projeto

O usuário pode criar um projeto. Um projeto, ao qual devem ser vinculados os seguintes dados: nome; integrantes da equipe; backlog do produto, contendo todos os itens (boards) pertencentes ao projeto; sprints, sendo que um sprint terá boards já criados no backlog do produto.

O usuário que cria um projeto, primeiramente tem que escolher seu papel no projeto, ou de Scrum Master, ou de Product Owner (Figura 10).

Figura 10 – Ilustração da tela para a seleção do papel do usuário junto ao projeto.

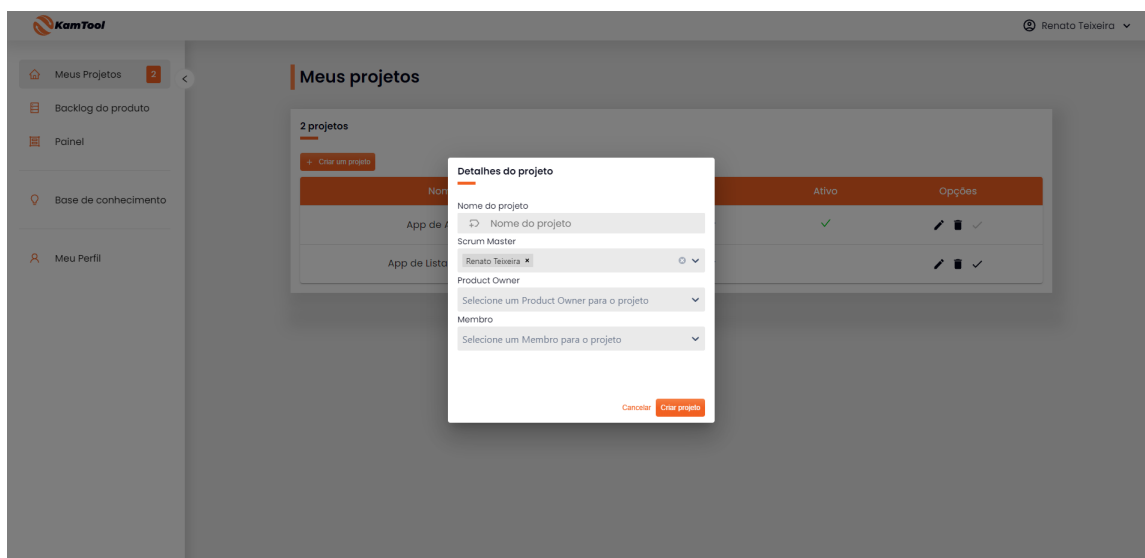


Fonte: Elaborado pelo autor.

Todo usuário que cria um projeto pode adicionar os integrantes da equipe Scrum ao projeto no momento da criação do mesmo, ou depois que o projeto estiver criado, editando-o. Quando um integrante da equipe for adicionado, um papel no projeto deve ser escolhido para ele (Figura 11). As opções disponíveis serão:

- Scrum Master ou Product Owner: estes papéis permitem ao usuário todas as permissões possíveis no projeto em questão, ou seja, este usuário tem controle pleno sobre o mesmo, podendo editar o nome do projeto, remover e adicionar usuários a ele.
- Membro: permite ao usuário um acesso mais restrito no projeto. Correlacionando com o Scrum, os funcionários que fizerem parte do papel ligado à Equipe Scrum, ou seja, desenvolvedores, designers, testadores e arquitetos, deverão ter esse papel de Membro no projeto cadastrado.

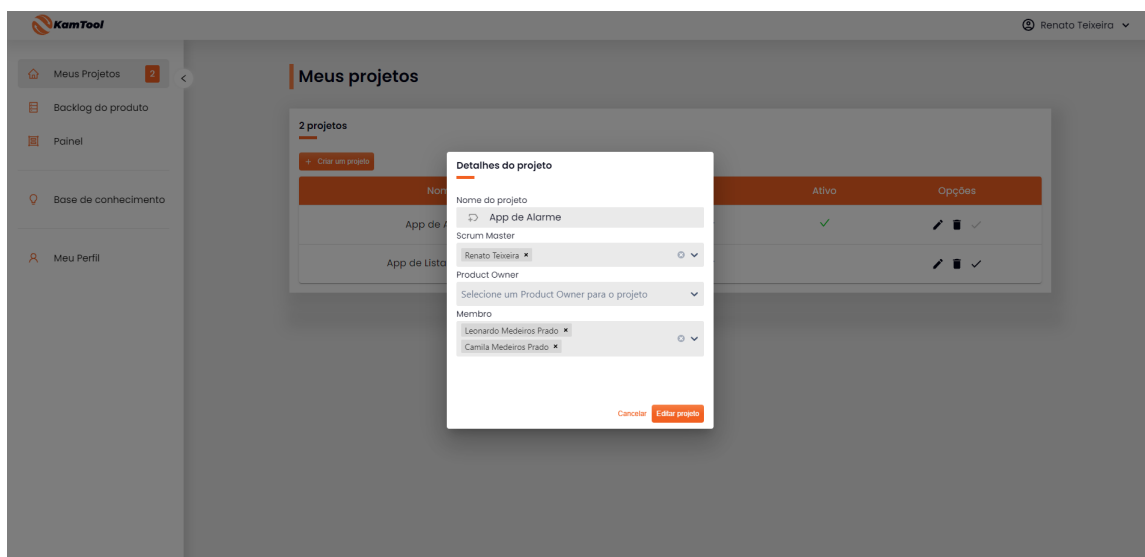
Figura 11 – Ilustração da tela de criação de um projeto.



Fonte: Elaborado pelo autor.

Após a criação de um projeto, o Scrum Master ou Product Owner pode mudar o nome do projeto que ele criou e também adicionar mais usuários ou remover usuários pertencentes ao projeto (Figura 12).

Figura 12 – Ilustração da tela para edição dos dados de um projeto pelo seu criador.



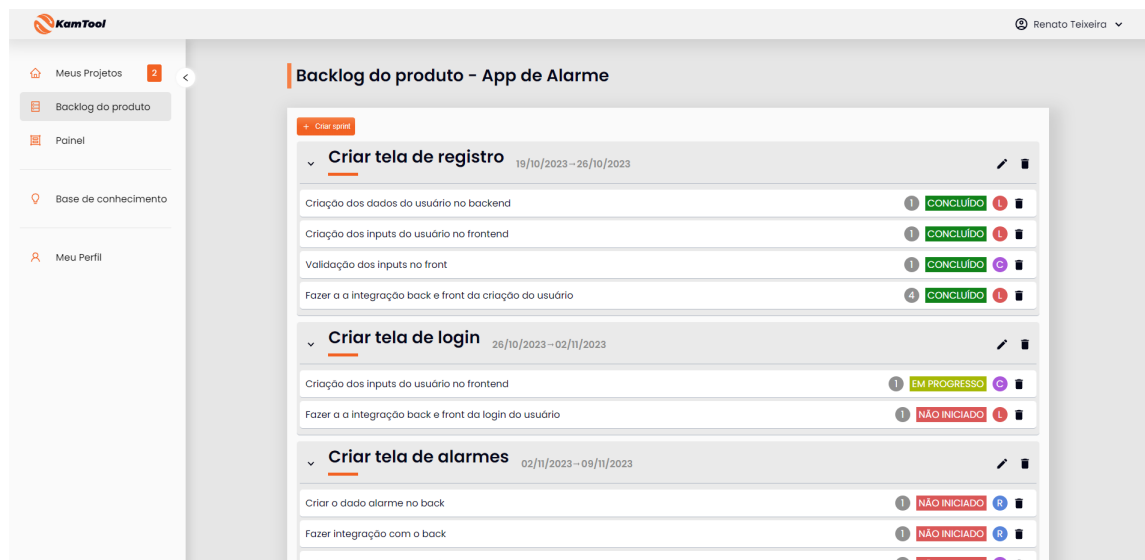
Fonte: Elaborado pelo autor.

F6. Gerenciar o backlog do produto do projeto

Independente de seu papel, o usuário pode visualizar o backlog do produto do projeto (Figura 13). Caso ele seja Scrum Master ou Product Owner, ele pode excluir ou

editar os boards e os sprints da tela de backlog do produto. Além disso, ele pode mudar os boards de sprints, arrastando-os ou editando o destino na tela de edição de board.

Figura 13 – Ilustração da tela de backlog do produto.

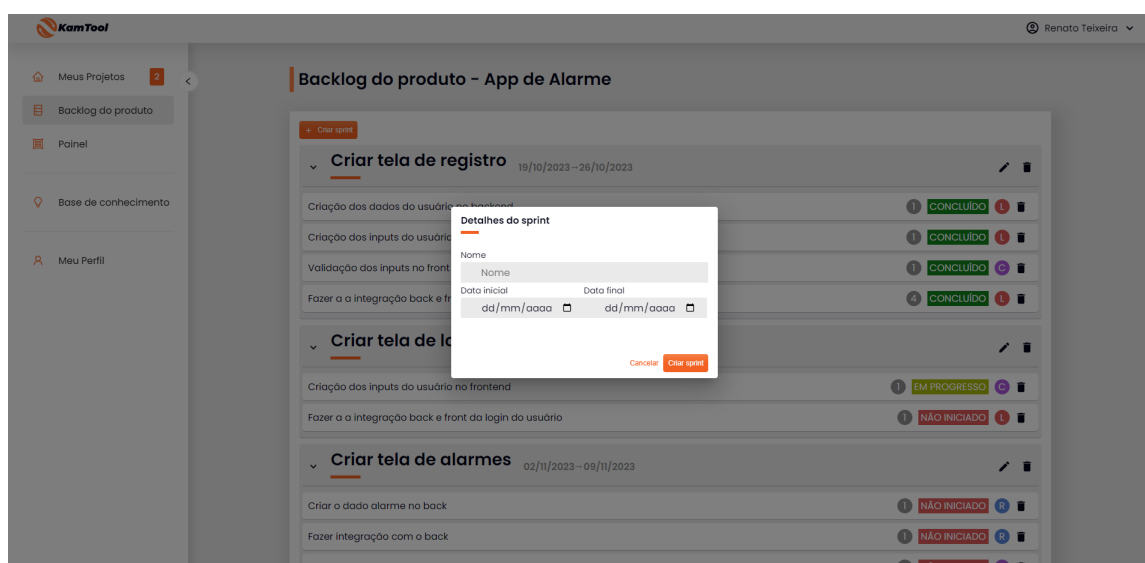


Fonte: Elaborado pelo autor.

F7. Criar e editar um sprint no projeto

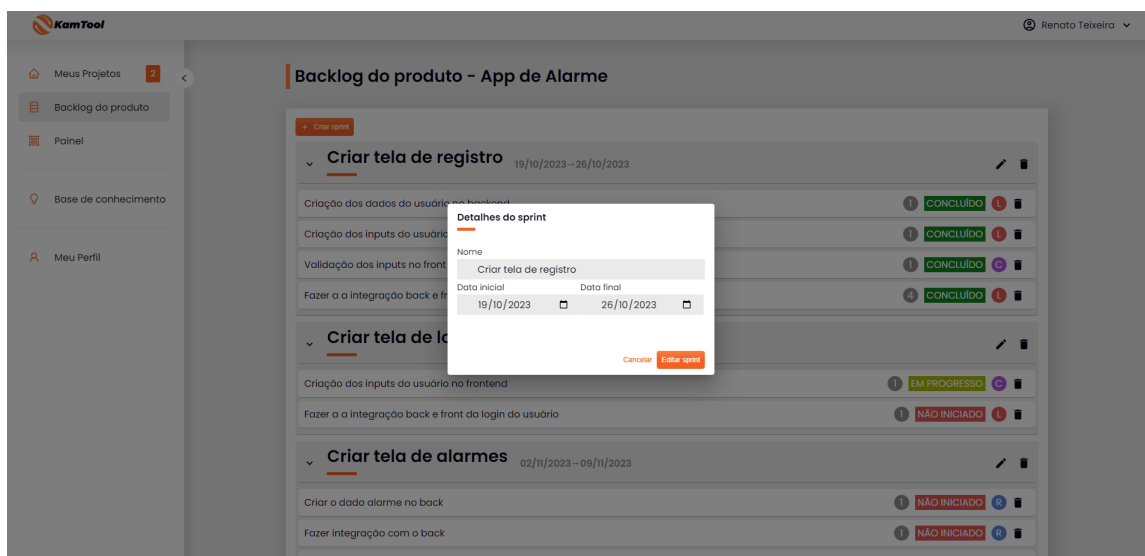
O Scrum Master ou Product Owner pode criar (Figura 14) ou editar (Figura 15) um sprint no projeto. Todo sprint terá: nome; data inicial; data final; boards.

Figura 14 – Ilustração da tela criação de um sprint.



Fonte: Elaborado pelo autor.

Figura 15 – Ilustração da tela de edição de um sprint.

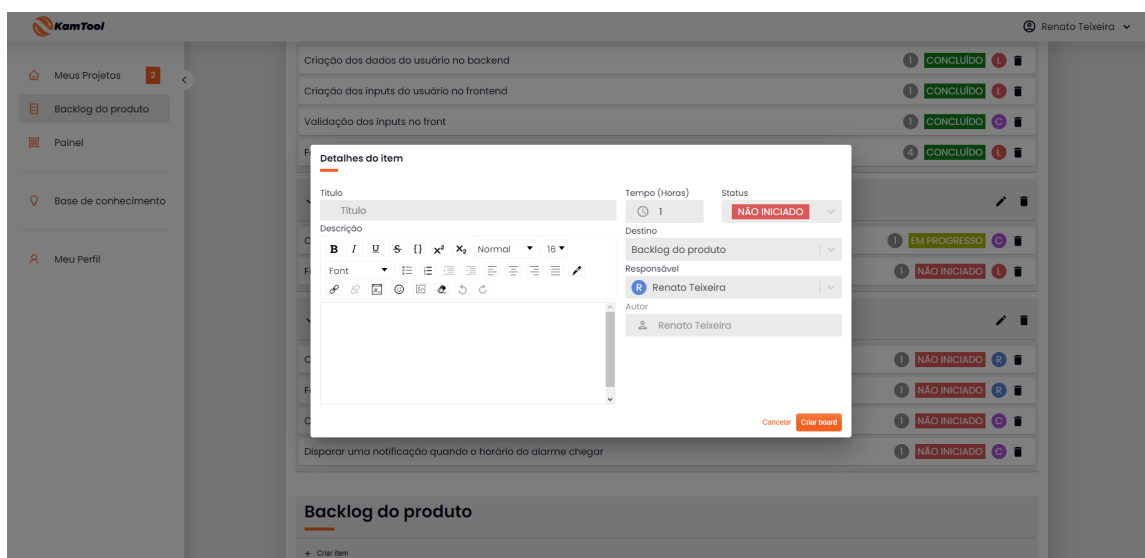


Fonte: Elaborado pelo autor.

F8. Criar e editar um item no backlog do produto no projeto

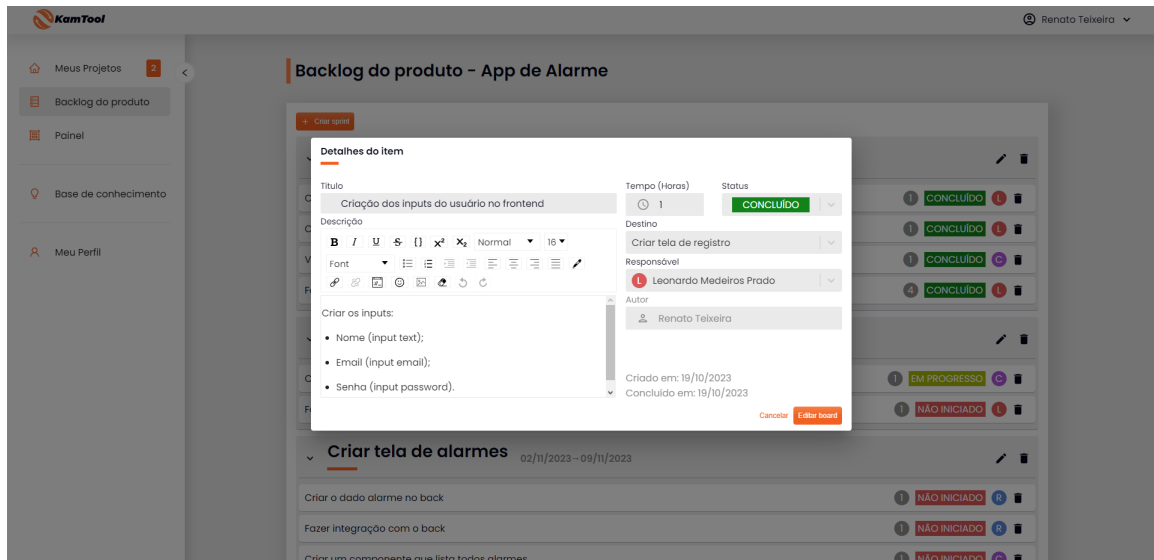
Todo projeto tem um backlog do produto criado e vazio assim que o projeto for criado. O Scrum Master ou Product Owner pode criar (Figura 16) ou editar (Figura 17) um item no backlog do produto do projeto. Um item (board) sempre tem: título; tempo estimado (em horas) para concluir o board; status, podendo ser Não iniciado, Em progresso ou Concluído; descrição sobre o que precisa ser feito para concluir o board; autor - Este campo não é editável, assim que o board é criado, ele é relacionado com o usuário que criou o board; responsável por concluir o board.

Figura 16 – Ilustração da tela de criação do board.



Fonte: Elaborado pelo autor.

Figura 17 – Ilustração da tela de edição do board.

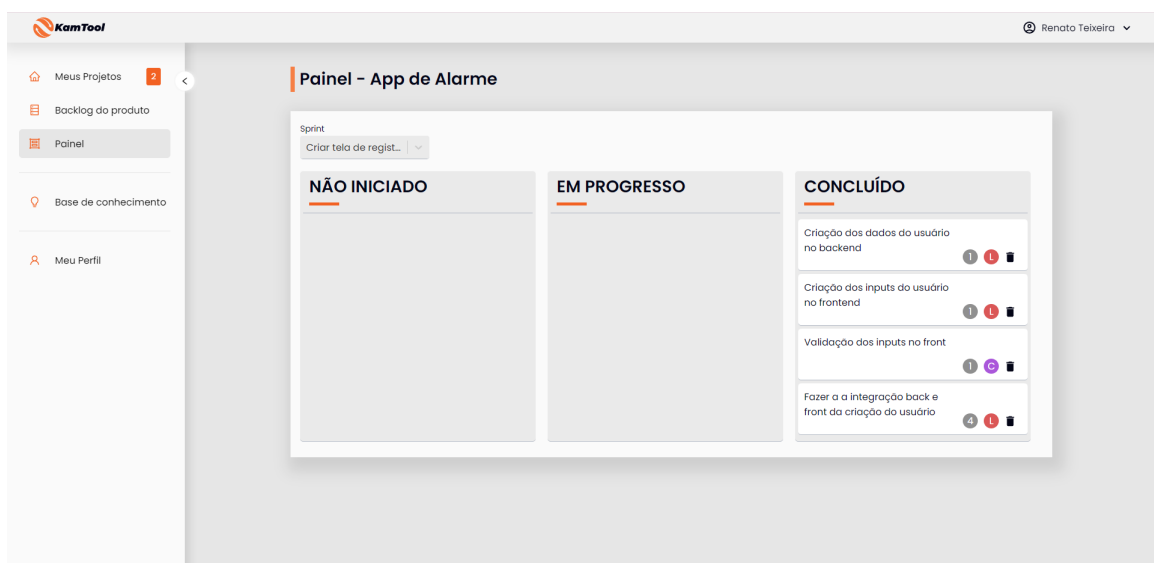


Fonte: Elaborado pelo autor.

F9. Mudar status dos boards de um sprint

O Scrum Master ou Product Owner pode editar o status de um board de um sprint apenas arrastando para a coluna do status desejado (Figura 18).

Figura 18 – Ilustração da tela de painel de um sprint.

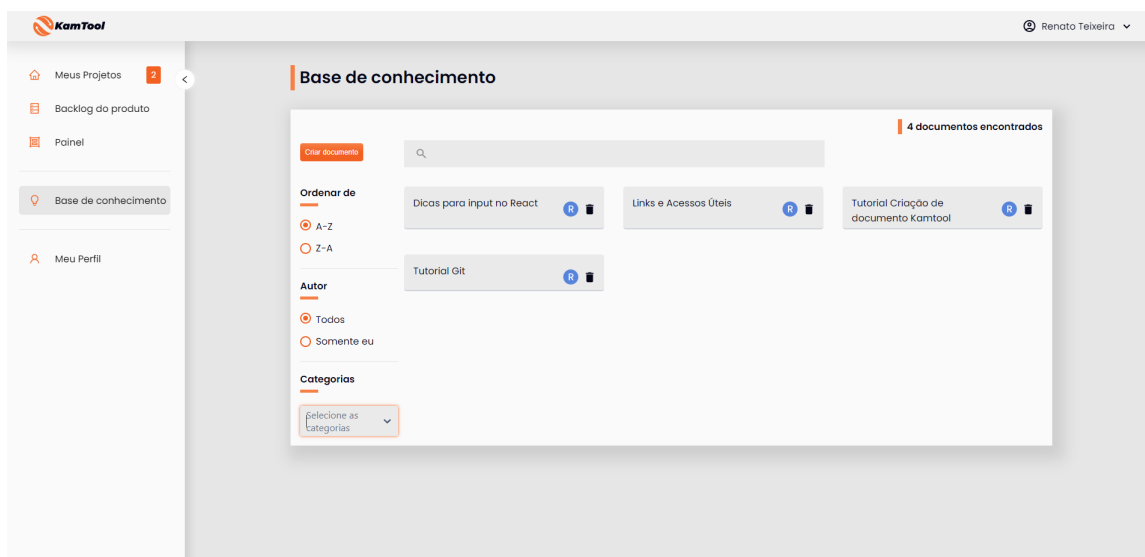


Fonte: Elaborado pelo autor.

F10. Visualizar a base de conhecimento do protótipo

Os usuários do protótipo podem visualizar os documentos da base de conhecimento do protótipo, podendo fazer uma filtragem por meio de categorias ou até por título dos documentos (Figura 19).

Figura 19 – Ilustração da tela da base de conhecimento.



Fonte: Elaborado pelo autor.

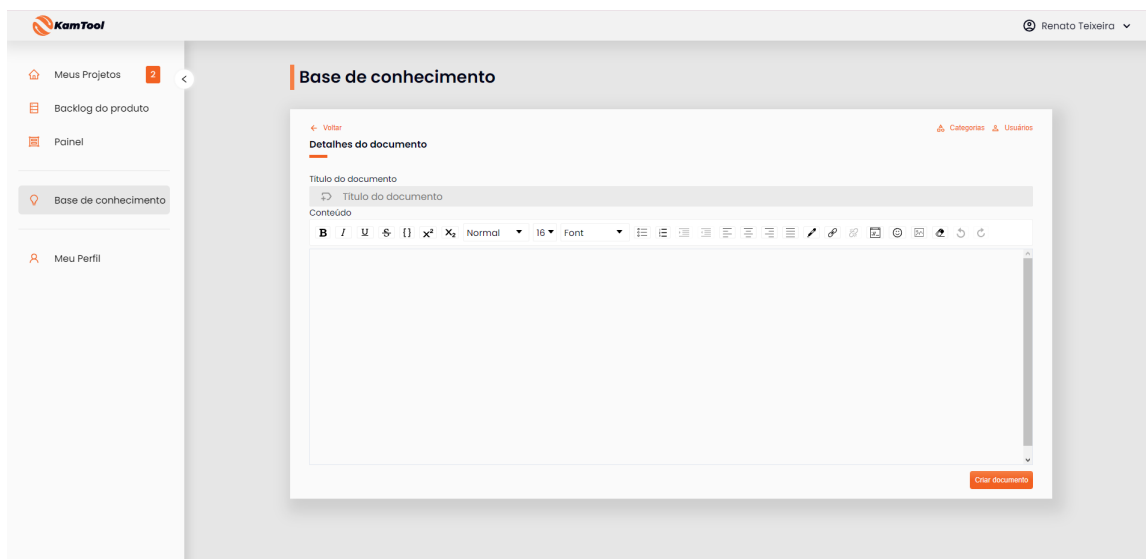
F11. Criar um documento na base de conhecimento do protótipo

Os usuários do protótipo podem criar um documento na base de conhecimento do protótipo (Figura 20). Todo documento de conhecimento tem:

- título do documento;
- categorias referentes ao documento de conhecimento, elas podem ser, por exemplo, linguagens de programação, tecnologias, ou qualquer outra categoria referente ao assunto abordado no documento, sendo que essas categorias podem ser criadas, se não existirem, ou selecionadas se já existirem (Figura 21);
- conteúdo contendo todas as informações relevantes para fomentar o conhecimento relacionado às categorias do mesmo, este campo conterá todo texto do documento; autor daquele documento de conhecimento;
- usuários que podem editar, ou seja, o criador do documento pode escolher usuários do sistema que podem editar o documento (Figura 22).

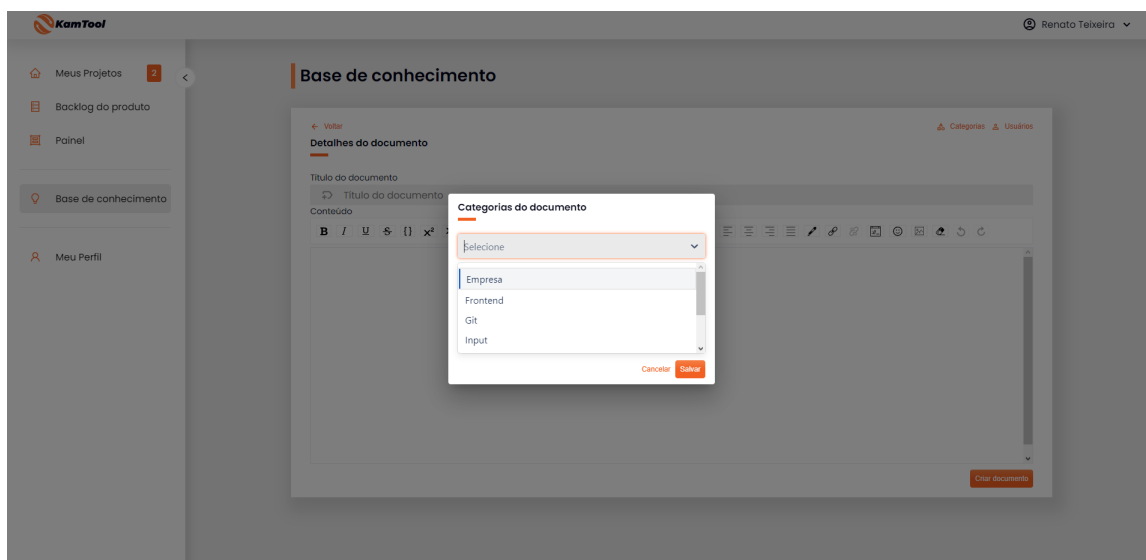
Um exemplo de documento de conhecimento seria a criação, por um usuário, de uma documentação para registrar como foi criada a tela de login de um projeto, tal documento terá como possíveis categorias: front end, React e o conteúdo dele seria um texto com todo passo a passo de tudo feito para se fazer a eventual tela.

Figura 20 – Ilustração da tela de criação de um documento.



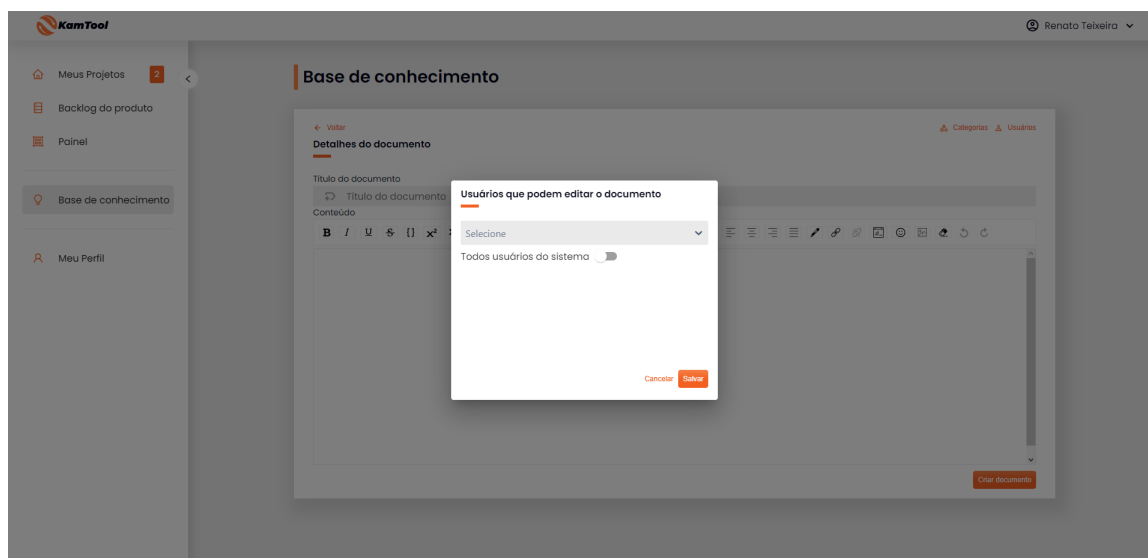
Fonte: Elaborado pelo autor.

Figura 21 – Ilustração da tela de seleção ou criação de categorias.



Fonte: Elaborado pelo autor.

Figura 22 – Ilustração da tela de seleção de usuários que podem editar o documento.

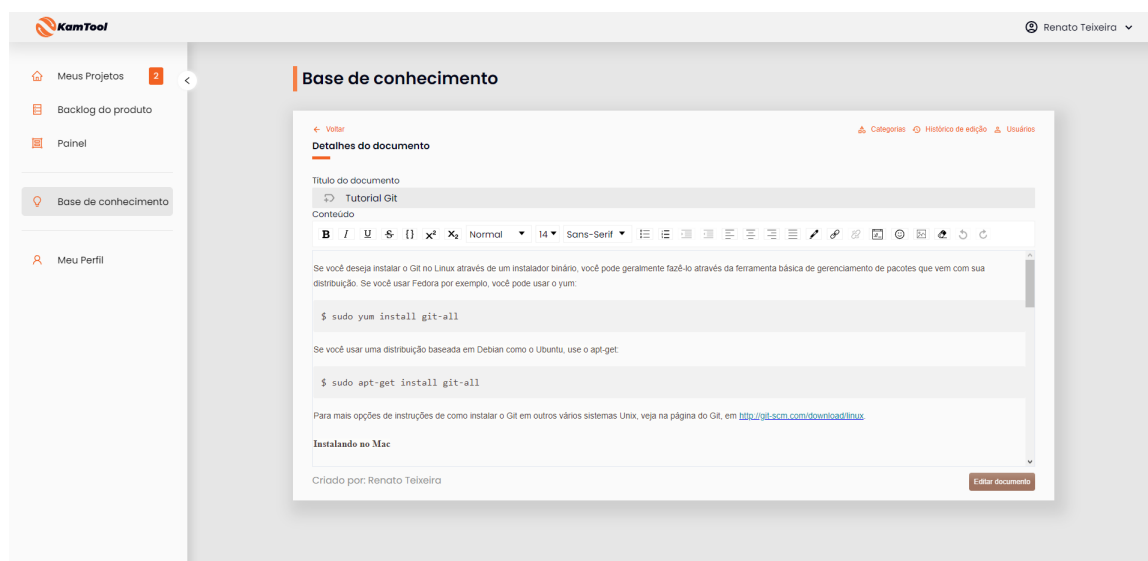


Fonte: Elaborado pelo autor.

F12. Editar um documento de conhecimento da base de conhecimento do protótipo

Os usuários do protótipo podem editar os seus documentos de conhecimento pertencentes à base de conhecimento do protótipo. Podem remover ou adicionar categorias ao documento, alterarem seu conteúdo ou título e também podem adicionar ou remover usuários que possuem permissão de editar o mesmo (Figura 23).

Figura 23 – Ilustração da tela de edição de documento.



Fonte: Elaborado pelo autor.

3.4 Considerações finais

Neste capítulo foram expostos detalhes sobre a arquitetura e implementação do protótipo KamTool, bem como sobre o seu funcionamento, por meio da apresentação das telas que correspondem a estas respectivas funcionalidades.

No próximo capítulo, são apresentados os resultados da validação do protótipo, ilustrados por meio de gráficos, e as conclusões obtidas

Capítulo 4 - Resultados do processo de avaliação

A avaliação do protótipo desempenha um papel fundamental na garantia da qualidade, não só verificando a conformidade com requisitos técnicos e funcionais, mas também avaliando a usabilidade e desempenho do mesmo, além do fato de auxiliar na identificação de carências para novas atualizações.

Neste capítulo são expostos os resultados do processo de avaliação. Para tanto, na seção 4.1, tem-se o perfil dos participantes, apresentando algumas características coletadas dos participantes no processo de avaliação. Já na seção 4.2, tem-se a avaliação do KamTool, apresentando os resultados da avaliação do KamTool. Por fim, na seção 4.3, tem-se as considerações finais, apresentando algumas considerações sobre o capítulo.

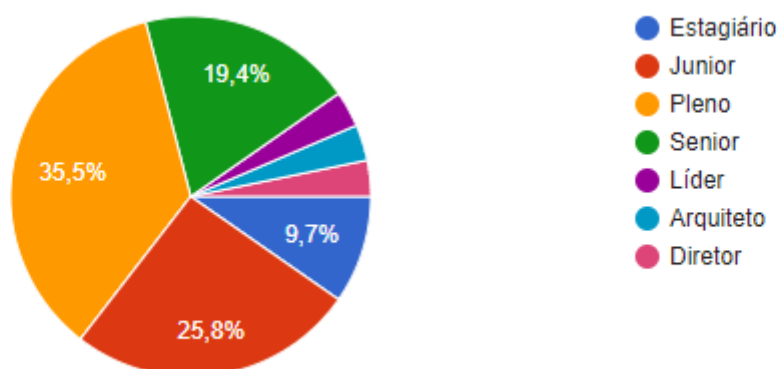
4.1 Perfil dos participantes

O protótipo ficou disponível online para avaliação, em conjunto com um formulário, previamente definido (Apêndice A), entre os dias 20 e 27 de outubro de 2023, via link disponibilizado em grupos de redes sociais. O público-alvo foram usuários da área de tecnologia em atividade no mercado de trabalho. O formulário teve 31 respostas, sendo ele composto por 12 questões.

A partir das respostas coletadas sobre gênero, tem-se que 25,8% dos participantes da pesquisa são do gênero feminino, sendo os 74,2% restantes do gênero masculino.

Analisando a função atual no mercado de cada participante, tem-se uma maior concentração nas funções: "Pleno"(35,5%), "Junior"(25,8%), "Sênior"(19,4%) e "Estagiário"(9,7%), ilustrado na Figura 24.

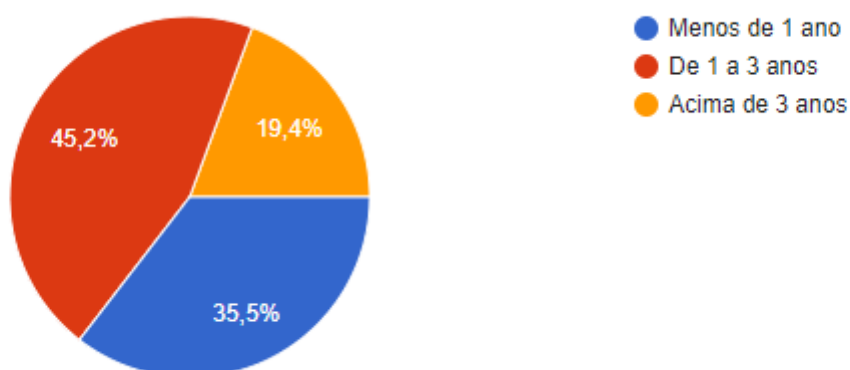
Figura 24 – Gráfico com resultado da avaliação sobre a pergunta: “Qual sua função atual no mercado de TI?”.



Fonte: Elaborado pelo autor.

Na questão de tempo de experiência, conforme a Figura 25, nota-se que 45,2% dos profissionais que responderam o formulário estão com menos de 1 ano exercendo a atual função informada, enquanto 35,5% estão de 1 a 3 anos exercendo a atual função, e 19,4% exercem a função atual por mais de 3 anos.

Figura 25 – Gráfico com resultado da avaliação sobre a pergunta: “Qual seu tempo de experiência no mercado de TI na função atual?”.



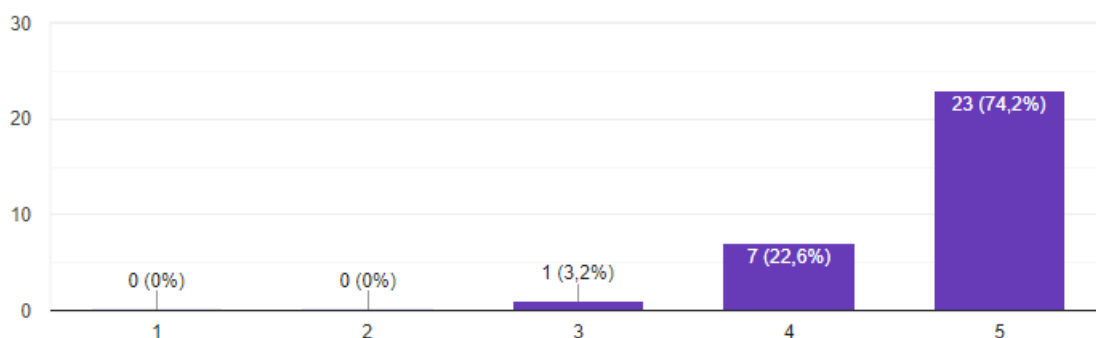
Fonte: Elaborado pelo autor.

4.2 Avaliação do KamTool

Para a avaliação do protótipo, foram estabelecidas algumas afirmações para serem respondidas por meio de uma escala Likert de concordância, com valores entre 1 e 5, sendo 1 indicando que discorda totalmente da afirmação e 5 indicando que concorda totalmente com a afirmação.

O resultado da avaliação, ilustrado na Figura 26, diz respeito à facilidade de utilização do protótipo, na qual vinte e três participantes atribuíram valor 5, sete atribuíram valor 4, e um atribuiu valor 3, demonstrando que o protótipo atende bem neste quesito.

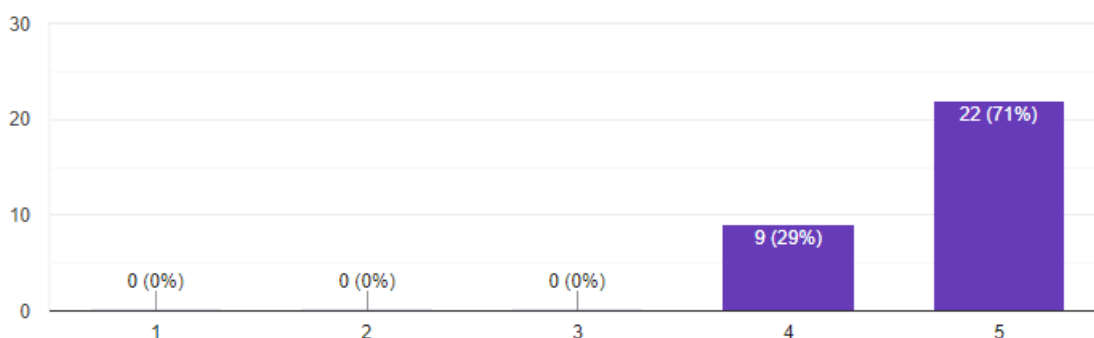
Figura 26 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo tem um uso fácil e intuitivo".



Fonte: Elaborado pelo autor.

As respostas coletadas e representadas na Figura 27, diz respeito ao design dos componentes na interface, se é agradável ao uso. Vinte e dois participantes atribuíram valor 5 e nove atribuíram valor 4. Com isso, é possível afirmar que o protótipo satisfaz os usuários na questão estética.

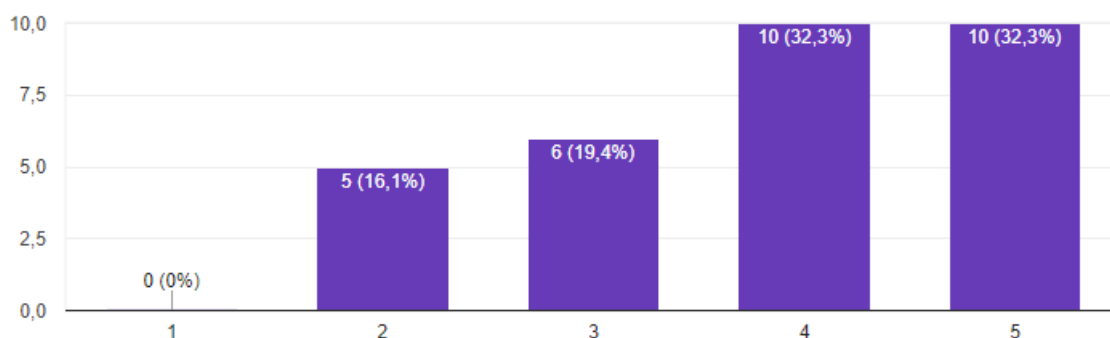
Figura 27 – Gráfico com resultado da avaliação sobre a afirmação: "O design e os componentes na interface são agradáveis para o uso".



Fonte: Elaborado pelo autor.

Na Figura 28, tem-se o resultado da afirmação sobre conhecer previamente o conceito de gestão de conhecimento. Dez participantes atribuíram valor 5, dez atribuíram valor 4, seis atribuíram valor 3 e cinco atribuíram valor 2. Com isso, pode-se perceber que nem todos os participantes possuem uma vasta experiência no assunto.

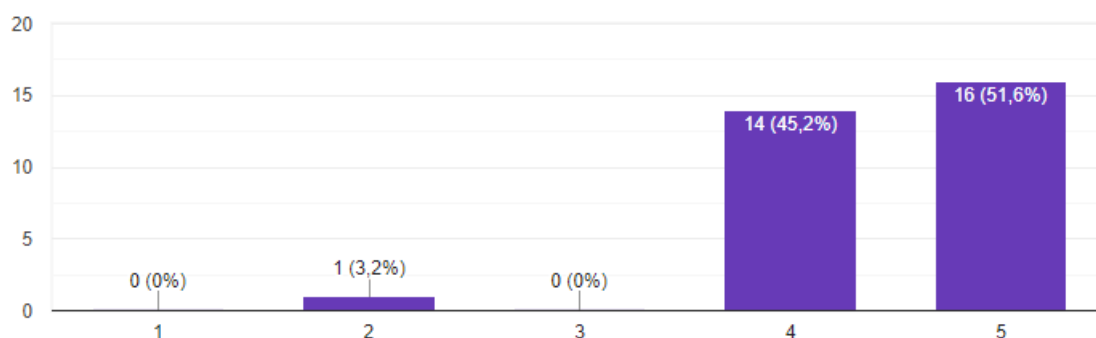
Figura 28 – Gráfico com resultado da avaliação sobre a afirmação: "O conceito de gestão de conhecimento já era conhecido".



Fonte: Elaborado pelo autor.

Já o resultado da avaliação, representado na Figura 29, diz respeito à utilidade do protótipo para a gestão ágil de projetos. Dezesseis participantes atribuíram valor 5 e Catorze atribuíram valor 4 e somente um atribuiu o valor 2. Com isso, é possível afirmar que o protótipo é competente na gestão ágil de projetos, contribuindo para manter a agilidade em projetos de software.

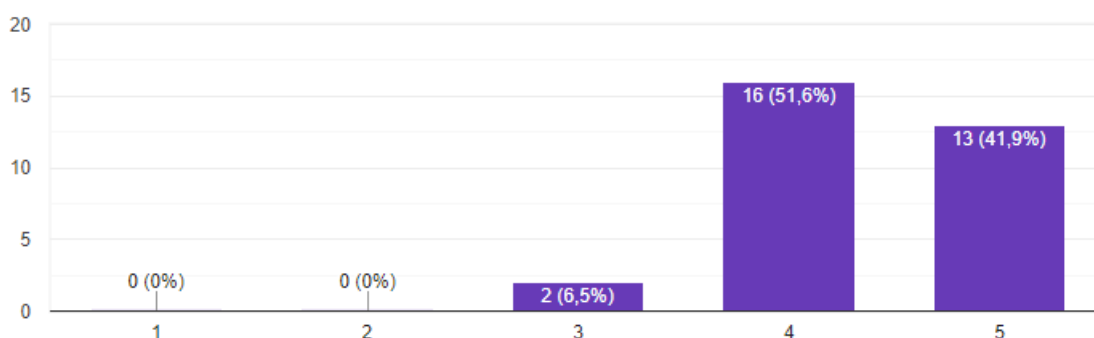
Figura 29 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo foi útil para o auxílio na gestão ágil de projetos".



Fonte: Elaborado pelo autor.

Na Figura 30, tem-se o resultado da avaliação sobre a utilidade do protótipo para a gestão de conhecimento. Treze participantes atribuíram valor 5, dezesseis atribuíram valor 4 e dois atribuíram valor 3. Com isso, é possível afirmar que o protótipo é competente na gestão de conhecimento.

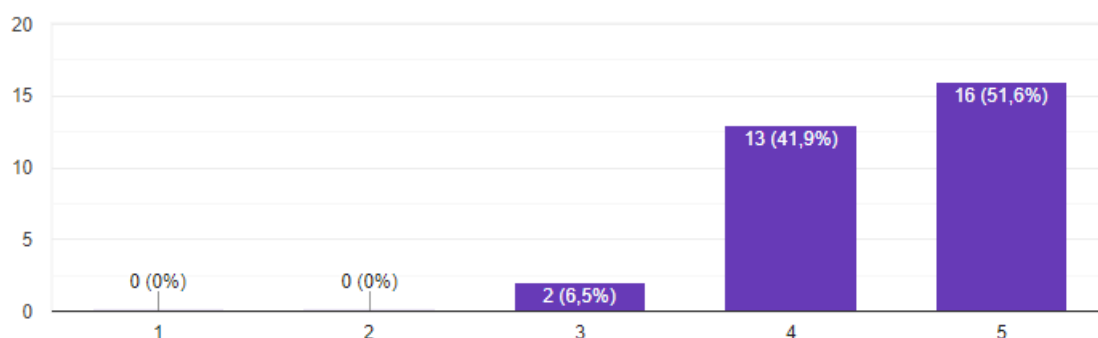
Figura 30 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo foi útil para o auxílio na gestão de conhecimento".



Fonte: Elaborado pelo autor.

Na Figura 31, por sua vez, tem-se o resultado da avaliação sobre a facilidade do protótipo no auxílio da captura e do compartilhamento de conhecimento juntamente com a gestão de projeto. Dezesseis participantes atribuíram valor 5, treze atribuíram valor 4 e dois atribuíram valor 3. Com isso, é possível afirmar que o protótipo favorece e facilita a gestão do conhecimento, aplicada em um contexto de desenvolvimento ágil.

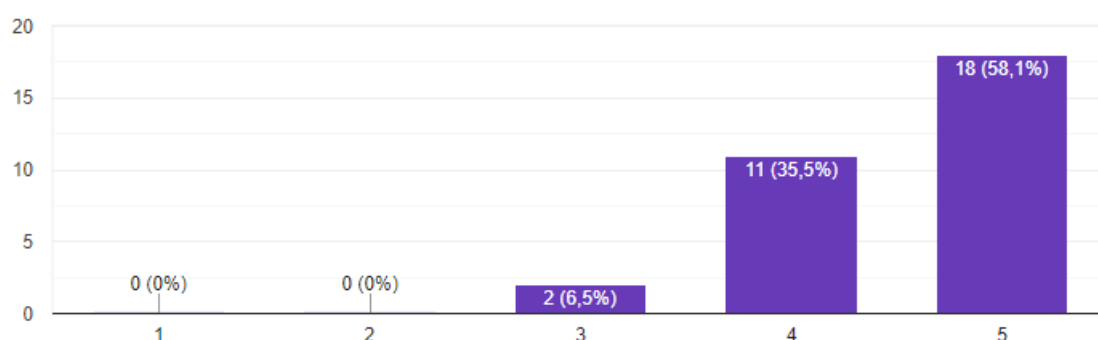
Figura 31 – Gráfico com resultado da avaliação sobre a afirmação: "O protótipo facilita a captura e o compartilhamento de conhecimento juntamente com a gestão de projeto".



Fonte: Elaborado pelo autor.

Conforme ilustrado na Figura 32, tem-se o resultado coletado a partir da avaliação sobre a quantidade de funcionalidades disponibilizadas no protótipo, se foram ou não satisfatórias. Dezoito participantes atribuíram valor 5, onze atribuíram valor 4 e dois atribuíram valor 3. Logo, de maneira geral, observa-se que a quantidade de funcionalidades disponibilizadas agradou à maioria dos usuários.

Figura 32 – Gráfico com resultado da avaliação sobre a afirmação: "A quantidade de funcionalidades disponibilizadas no protótipo foram satisfatórias".

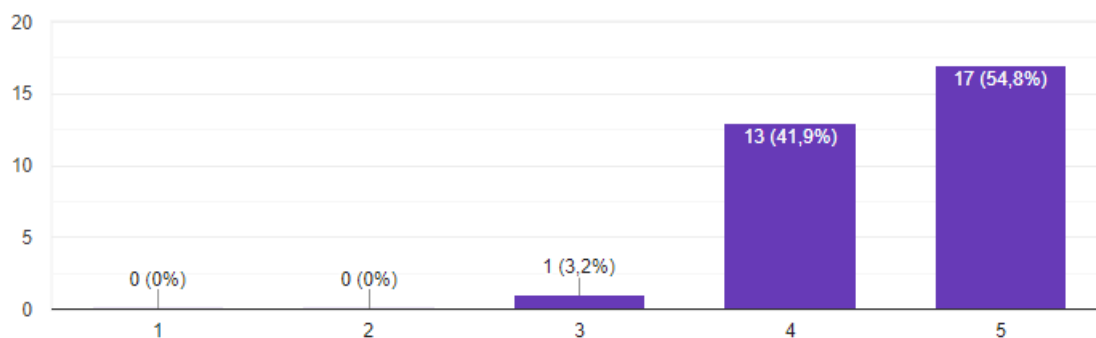


Fonte: Elaborado pelo autor.

Por último, ilustrado na Figura 33, teve-se a avaliação sobre a adequação da organização e estrutura estabelecidas para a documentação da base de conhecimento e dos boards do backlog do produto são adequadas. Dezesete participantes atribuíram valor 5, treze atribuíram valor 4 e um atribuiu valor 3. Logo, entende-se que o protótipo

conseguiu ser competente em suprir a documentação tanto da base de conhecimento, quanto a do backlog do produto.

Figura 33 – Gráfico com resultado da avaliação sobre a afirmação: "A organização e estrutura estabelecidas para a documentação da base de conhecimento e dos boards do backlog do produto são adequadas".



Fonte: Elaborado pelo autor.

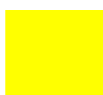
Na Tabela 2 é apresentado o resultado de uma análise pessoal realizada sobre o KamTool, considerando as mesmas subcaracterísticas de qualidade previamente utilizadas no Capítulo 2 para avaliação empírica de aplicativos existentes (Tabela 1). O objetivo é evidenciar as contribuições obtidas neste trabalho. Diante do resultado apresentado na Tabela 2, tem-se o atendimento pleno da ampla maioria das subcaracterísticas de qualidade consideradas. Cabe observar, porém, que o foco maior do KamTool foi sanar deficiências observadas nos principais softwares de gestão ágil de projetos, considerando os recursos voltados para gestão de conhecimento, ou seja, armazenamento e organização e pesquisa avançada. É possível justificar o resultado da avaliação pelo fato de que as subcaracterísticas estão bem atendidas, uma vez que o protótipo possibilita uma boa filtragem e organização dos documentos, juntamente com uma pesquisa. Em relação à parte de gestão ágil de projetos, ou seja, backlog do produto, planejamento e rastreamento de sprints e quadro Kanban, o protótipo atendeu bem e conseguiu ser bem efetivo, segundo os avaliadores. Na subcaracterística de métricas e relatórios pode-se apontar um possível ponto a ser explorado em trabalhos futuros, uma vez que tiveram sugestões sobre funcionalidades que contribuem para tal, como, por exemplo, permitir a criação de gráficos de Burndown e também permitir a criação de relatórios. Por fim, um ponto forte do protótipo é o mesmo ser gratuito e com código livre

Tabela 2 – Resultado de avaliação do KamTool sobre as características de qualidade almejadas para ferramentas de gestão ágil de projetos.

	Características							
Softwares	Armazenamento e Organização	Pesquisa Avançada	Backlog do Produto	Planejamento e rastreamento de sprints	Quadro Kanban	Colaboração em equipe	Métricas e relatórios	Versão gratuita
Kamtool								



Atende



Atende em partes



Não atende

Fonte: Elaborado pelo autor.

4.3 Considerações finais

Neste capítulo, foram apresentados os resultados obtidos por meio da avaliação do protótipo por profissionais da área de tecnologia, usando questionário online. A partir das respostas coletadas, observa-se que o KamTool foi classificado pela maioria dos usuários como sendo intuitivo, com uma boa interface e facilitador na gestão ágil de conhecimento.

No próximo capítulo, é apresentada uma conclusão baseada nos resultados mostrados, além dos possíveis trabalhos futuros.

Capítulo 5 - Conclusão

A conclusão de um trabalho desempenha um papel importante, refletindo sobre os resultados das avaliações coletadas e então verificando se as hipóteses anteriormente preestabelecidas são ratificadas.

Neste capítulo são expostas as conclusões que se tem a partir dos resultados obtidos. Para tanto, na seção 5.1, tem-se as contribuições do KamTool para disseminação da área de gestão de conhecimento e, por fim, na seção 5.2, tem-se os trabalhos futuros, apresentando todas as sugestões para melhoria do protótipo.

5.1 Contribuições do trabalho

A Gestão Ágil de Conhecimento vem se mostrando uma abordagem interessante, porém em crescimento. Considerando as dificuldades a respeito de se encontrar um software que reúna todas as características pertinentes, tanto para a gestão de conhecimento quanto para a gestão ágil de projetos, verificou-se então a pertinência do presente trabalho.

Porém, é importante observar que, em razão da amostra de conveniência, usada no processo de avaliação qualitativo realizado, abranger apenas profissionais de algumas poucas empresas, não é possível generalizar os resultados obtidos. Logo, ainda é preciso avaliar tais resultados por meio de uma pesquisa complementar quantitativa e com uma população de usuários mais diversificada com o intuito de obter resultados mais expressivos.

O KamTool foi avaliado por meio de um questionário online disponibilizado a profissionais de tecnologia atuantes no mercado. Com isso, foi possível notar que existe um certo desconhecimento de alguns ainda em relação à área de gestão de conhecimento, logo, isso mostra que a área está em pleno crescimento e precisa cada vez mais de exposição e incentivo para sua evolução.

Neste cenário, observa-se a contribuição do KamTool para disseminação desta área, uma vez que os conceitos abordados como ciclo de gestão do conhecimento e ciclo do Scrum são devidamente contemplados no protótipo. Segundo os participantes do processo de avaliação do KamTool, pode-se concluir que o presente trabalho contribui para a área e seu desenvolvimento.

Finalmente, destaca-se que é possível também verificar, por meio dos resultados obtidos com a avaliação do KamTool, que o apoio automatizado para a gestão de conhecimento contribui para manter a agilidade em projetos de software, além de que a gestão de conhecimento, aplicada em um contexto de desenvolvimento ágil, contribui para o compartilhamento de conhecimento entre membros de uma equipe, durante o desenvolvimento de um projeto de software, ratificando as hipóteses consideradas.

5.2 Trabalhos futuros

Embora o feedback dos usuários fora em sua maioria positivo, também foram feitas algumas sugestões de melhoria voltadas, principalmente, para a inserção de novas funções, como: colocar uma lista de escolhas na tela de Backlog e Painel para mudar de projeto sem precisar ficar voltando na primeira tela; incluir um modo noturno/noite, a fim de agradar ainda mais outros possíveis nichos de usuários; permitir a criação de relatórios; adicionar mais filtros de pesquisa; e permitir que as colunas do painel possam ser ordenáveis; gráficos de Burndown. Deste modo, fica claro que os trabalhos futuros sugeridos visam a melhora da praticidade do uso do protótipo, fazendo que ele fique ainda mais prático de ser usado. Por fim, perante o exposto, é possível afirmar que o objetivo do trabalho foi atingido.

REFERÊNCIAS

- ABOU-ZEID, E. A knowledge management reference model. **Journal of Knowledge Management**, v. 6(5), p. 486–499, 2002.
- AHMED, A.; AHMAD, S.; EHSAN, N.; MIRZA, E.; SARWAR, S. Agile software development: Impact on productivity and quality. In: IEEE. **2010 IEEE International Conference on Management of Innovation & Technology**. [S.l.], 2010. p. 287–291.
- AKHTAR, A.; BAKHTAWAR, B.; AKHTAR, S. Extreme programming vs scrum: A comparison of agile models. **International Journal of Technology, Innovation and Management (IJTIM)**, v. 2, n. 2, p. 80–96, 2022.
- ALAVI, M.; KAYWORTH, T.; LEIDNER, D. An empirical examination of the influence of organizational culture on knowledge management practices. **Journal of Management , 22(3)Information Systems**, v. 22(3), p. 191–224, 2005.
- ALMEIDA, F.; MIRANDA, E.; FALCÃO, J. Challenges and facilitators practices for knowledge management in large-scale scrum teams. **Journal of Information Technology Case and Application Research**, v. 21(2), p. 90–102, 2019.
- ALWIS, R. S.; HARTMANN, E. The use of tacit knowledge within innovative companies: Knowledge management in innovative enterprises. **Journal of Knowledge Management**, v. 12(1), p. 133–147, 2008.
- ANAND, A.; WALSH, I. Should knowledge be shared generously? tracing insights from past to present and describing a model. **Journal of Knowledge Management**, v. 20(4), p. 713–730, 2016.
- ANDERSON, D. J. **Kanban: Successful Evolutionary Change for Your Technology Business**. Blue Book ed. Edition, 2010. Disponível em: <<https://www.amazon.com/Kanban-Successful-Evolutionary-Technology-Business/dp/0984521402>>.
- ANSARI, M.; YOUSHANLOUEI, H. R.; MOOD, M. M. A conceptual model for success in implementing knowledge management: A case study in tehran municipality. **Journal of Service Science and Management**, v. 5(1), p. 212–222, 2012.
- ARAFEEN, J.; BOSE, S. Improving software development using scrum model by analyzing up and down movements on the sprint burn down chart: Proposition for better alternatives. **International Journal of Digital Content Technology and its Applications**, v. 3, n. 3, p. 109–115, 2009.
- ASRAR-UL-HAQ, M.; ANWAR, S. A systematic review of knowledge management and knowledge sharing: trends, issues, and challenges. **Cogent Business Management**, v. 3, 2016.
- AURUM, A.; DANESHGAR, F.; WARD, J. Investigating knowledge management practices in software development organizations - an australian experience. **Information and Software Technology**, v. 50(6), p. 511–533, 2008.

BECERRA-FERNANDEZ, I.; SABHERWAL, R. **Knowledge Management Systems and Processes**. [S.l.]: Routledge, 2015. v. 2.

BHATT, G. Organizing knowledge in the knowledge management development cycle. **Journal of Knowledge Management**, v. 4(1), p. 15–26, 2000.

BJØRNSON, F. O.; DINGSØYR, T. Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. **Information and Software Technology**, p. 1055–1068, 2008.

CARRION, G. C.; NAVARRO, J. G. C.; JIMENEZ, D. J. The effect of absorptive capacity on innovativeness: Context and information systems capability as catalysts. **British Journal of Management**, v. 23(1), p. 110–129, 2012.

CEN. European guide to good practice in knowledge management. **Management Science**, v. 50(3), p. 352–364, 2004.

CERVONE, H. F. Understanding agile project management methods using scrum. **International Digital Library Perspectives**, v. 27(1), p. 18–22, 2011.

CHANG, C. L.; LIN, T. The role of organizational culture in the knowledge management process. **Journal of Knowledge Management**, v. 19(3), p. 433–455, 2015.

CHEN, A.; EDGINGTON, T. Assessing value in organizational knowledge creation: Considerations for knowledge workers. **MIS Quarterly**, v. 29(2), p. 279–309, 2005.

CHEN, H.; RAGSDELL, G.; O'BRIEN, A. Identification of tacit knowledge associated with experience: a chinese software industry study. In **Proceedings of 15th European Conference on Knowledge Management**, Portugal, p. 1147–1155, 2014.

COHN, M. **Succeeding with Agile: Software Development Using Scrum**. Addison-Wesley Professional, 2010. Disponível em: <https://books.google.com.br/books/about/Succeeding_with_Agile_Software_Developme.html?id=fIMQrvWFmQ4C&redir_esc=y>.

CRISTAL, M.; WILDT, D.; PRIKLADNICKI, R. Usage of scrum practices within a global company. **IEEE International Conference on Global Software Engineering**, p. 222–226, 2008.

Di Vaio, A.; PALLADINO, R.; PEZZI, A.; KALISZ, D. E. The role of digital innovation in knowledge management systems: A systematic literature review. **Journal of Business Research**, v. 123, p. 220–231, 2021. ISSN 0148-2963. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0148296320306238>>.

DINGSØYR, T.; NERUR, S.; BALIJEPALLY, V.; MOE, N. B. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85(6), p. 1213–1221, 2012.

DÖNMEZ, D.; GROTE, G.; BRUSONI, S. Routine interdependencies as a source of stability and flexibility. a study of agile software development teams. **Information and Organization**, v. 26, p. 63–83, 2016.

EVANS, M.; DALKIR, K.; BIDIAN, C. A. Holistic view of the knowledge life cycle: The knowledge management cycle (kmc) model. **Electronic Journal of Knowledge Management**, v. 12(2), p. 85–97, 2014.

FRIEDMAN, T. L. **The World Is Flat: A Brief History of the Twenty-first Century**. [S.l.]: London: Farrar, Straus and Giroux, 2005. v. 1.

GARCIA, A. L.; MIGUEL, I. da R.; EUGÊNIO, J. B.; VILELA, M. da S.; MARCONDES, G. A. B. Scrum-based application for agile project management. **J. Softw.**, v. 15, n. 4, p. 106–113, 2020.

GOLD, A. H.; MALHOTRA, A.; SEGARS, A. H. Knowledge management: An organizational capabilities perspective. **Journal of Management Information Systems**, v. 18(1), p. 185–214, 2001.

GOLDONI, V.; OLIVEIRA, M. Knowledge management metrics in software development companies in brazil. **Journal of Knowledge Management**, v. 14(2), p. 301–313, 2010.

GRANT, K. Tacit knowledge revisited - we can still learn from polanyi. **The Electronic Journal of Knowledge Management**, v. 5(2), p. 173–180, 2007.

GULBRANSON, C. A.; AUDRETSCH, D. B. Proof of concept centers: Accelerating the commercialization of university innovation. **SSRN Electronic Journal**, v. 33(3), p. 249–258, 2008.

HEGAZY, F.; GHORAB, K. The influence of knowledge management on organizational business processes' and employees' benefits. **International Journal of Business and Social Science**, v. 5(1), p. 148–172, 2014.

HERSCHEL, R.; JONES, N. Knowledge management and business intelligence: The importance of integration. **Journal of Knowledge Management** 9(4), v. 9(4), p. 45–55, 2005.

HIGHSMITH, J.; CONSORTIUM, C. What is agile software development? **The Journal of Defense Software Engineering**, v. 15, p. 4–9., 2002.

IPE, M. Knowledge sharing in organizations: A conceptual framework. **Human Resource Development Review**, v. 2(4), p. 337–359, 2003.

JEFFRIES, R.; ANDERSON, A.; HENDRICKSON, C. **Extreme Programming Installed**. Addison-Wesley Professional, 2000. Disponível em: <<https://www.amazon.com/Extreme-Programming-Installed-Ron-Jeffries/dp/0201708426>>.

JELENIC, D. The importance of knowledgemanagement in organizations – with emphasis on the balanced scorecard learning and growth perspective. **Knowledge as Business Opportunity: Proceedings of the Management, Knowledge and Learning International Conference**, International School for Social and Business Studies, Celje, Slovenia, 2011.

JOIA, L. A.; LEMOS, B. Relevant factors for tacit knowledge transfer within organizations. **Journal of Knowledge Management**, v. 14(3), p. 410–427, 2010.

KARIMI, F.; JAVANMARD, M. Surveying the infrastructure and capabilities for knowledge management implementation in supply chain. **JIM Quest**, v. 10(1), p. 75–82, 2014.

KHAN, R. A. Sustainable competitive advantage through knowledge management. **International Journal of Advanced Research in Computer Engineering Technology**, v. 3(4), p. 1079–1082, 2014.

LEI, H.; GANJEIZADEH, F.; JAYACHANDRAN, P. K.; OZCAN, P. A statistical analysis of the effects of scrum and kanban on software development projects. **Robotics and Computer-Integrated Manufacturing**, v. 43, p. 59–67, 2017.

LEIDNER, D.; ALAVI, M.; KAYWORTH, T. The role of culture in knowledge management: A case study of two global firms. **International Journal of e-Collaboration**, v. 2(1), p. 17–40, 2006.

LYLES, M.; SCHWENK, C. R. Top management, strategy and organizational knowledge structures. **Journal of Management Studies**, v. 29(2), p. 155–174, 2007.

LYTRAS, M.; POULOUDI, A.; POULYMENAKOU, A. Knowledge management convergence - expanding learning frontiers. **Journal of Knowledge Management**, v. 6(1), p. 40–51, 2002.

MACHADO, H. P. V.; ELIAS, M. L. G. G. R. Gestão do conhecimento: constituição do campo, temas e perspectivas de pesquisas. **TransInformação**, v. 32, p. 1–12, 2020.

MACHADO SILVANA SECINARO, D. C. Andreia de B.; LANZALONGA, F. Knowledge management and digital transformation for industry 4.0: a structured literature review. **Knowledge Management Research & Practice**, Taylor Francis, v. 20, n. 2, p. 320–338, 2022.

MALHOTRA, Y. Knowledge management for e-business performance: Advancing information strategy to "internet time". **Information Strategy, The Executive Journal**, v. 16(4), p. 5–16, 2000.

MARÇAL, A. S. C.; SOARES, F. S. F.; BELCHIOR, A. D. Mapping cmmi project management process areas to scrum practices. **IEEE Software Engineering Workshop**, v. 31, p. 13–22, 2007.

MASSEY, A.; MONTOYA-WEISS, M. Unraveling the temporal fabric of knowledge conversion: A model of media selection and use. **MIS Quarterly**, v. 30(1), p. 99–114, 2006.

MEIHAMI, B.; MEIHAMI, H. Knowledge management a way to gain a competitive advantage in firms (evidence of manufacturing companies). **International Letters of Social and Humanistic Sciences**, v. 14, p. 80–91, 2014.

MELO, C.; CRUZES, D. S.; KON, F.; CONRADI, R. Interpretative case studies on agile team productivity and management. **Information and Software Technology**, v. 55(2), p. 412–427, 2013.

MERLO, T. R. Factors influencing knowledge management use in technology enterprises in southern united states. Vienna, Austria, Elsevier, 2016.

MITTAL, S.; KUMAR, V. Study of knowledge management models and their relevance in organisations. **International Journal of Knowledge Management Studies**, v. 10(3), p. 322–335, 2019.

MOE, N. B.; DINGSØYR, T.; DYBÁ, T. A. A teamwork model for understanding an agile team: a case study of a scrum project. **Information and Software Technology**, v. 52, p. 480–491, 2010.

MOORADIAN, N. Tacit knowledge: Philosophic roots and role in km. **Journal of Knowledge Management**, v. 9(6), p. 104–113, 2006.

MORANDINI, M.; COLETI, T. A.; OLIVEIRA, E.; CORRÊA, P. L. P. Considerations about the efficiency and sufficiency of the utilization of the scrum methodology: A survey for analyzing results for development teams. **Computer Science Review**, v. 39, p. 100314, 2021. ISSN 1574-0137. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1574013720304147>.

NAGAI, R. A.; SBRAGIA, R. As origens da metodologia ágil: de onde saímos e onde estamos? uma revisão sistemática da literatura. **Revista de Gestão e Projetos**, v. 14, n. 1, p. 11–41, 2023.

NAWINNA, D. A model of knowledge management: delivering competitive advantage to small & medium scale software et al industry. **In Proceedings of 6th International Conference on Industrial and Information Systems**, Sri Lanka, p. 414–419, 2011.

NEWELL, S.; TANSLEY, C.; HUANG, J. Social capital and knowledge integration in an erp project team: The importance of bridging and bonding. **British Journal of Management**, v. 15(1), p. 43–57, 2004.

NONAKA, I.; KROGH, G. V. Tacit knowledge and knowledge conversion: Controversy and advancement in organizational knowledge creation theory. **Organization Science**, v. 20(3), p. 635–652, 2009.

OLIVA, F. L.; KOTABE, M. Barriers, practices, methods and knowledge management tools in startups. **Journal of Knowledge Management**, v. 23(9), p. 1838–1856, 2019.

OLUWAFUNMILAYO, B. G.; OYEBISI, T. O.; ADEBOLA, O.; AFOLABI, S. Evaluation of the factors influencing the indigenous software products development. **International Journal on Advances in ICT for Emerging Regions (ICTer)**, Nigeria, v. 7(3), p. 1–8, 2015.

OUYANG, Y. A cyclic model for knowledge management capability - a review study. **Arabian Journal of Business and Management Review**, v. 4(4), p. 1–9, 2014.

ÖZKAN, D.; MISHRA, A. Agile project management tools: A brief comparative view. **Cybernetics and Information Technologies**, v. 23(9), p. 17–25, 2019.

POPPENDIECK, M.; POPPENDIECK, T. **Lean Software Development: An Agile Toolkit**. Addison-Wesley Professional, 2003. Disponível em: <https://www.amazon.com/Lean-Software-Development-Agile-Toolkit/dp/0321150783>.

SCHULTZE, U.; LEIDNER, D. Studying knowledge management in information systems research: Discourses and theoretical assumptions. **MIS Quarterly**, v. 26(3), p. 213–242, 2002.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide - The definitive guide to Scrum: the rules of the game**. Scrum Guides, 2017. Disponível em: <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>.

TENÓRIO, N.; PINTO, D.; SILVA, M. J.; ALMEIDA, I. C. de; BORTOLOZZI, F. Knowledge management in the software industry: how scrum activities support a knowledge management cycle. **Navus: Revista de Gestão e Tecnologia**, v. 10, p. 1–13, 2020.

TENÓRIO, N.; PINTO, D.; VIDOTTI, A. F.; SANTOS, M. O.; URBANO, G. C.; BORTOLOZZI, F. Tool based on knowledge management process: An interview protocol to gather functional requirements from software industry experts. **Matter: International Journal of Science and Technology**, v. 3(1), p. 45–54, 2017.

TURNER, J. R.; ZIMMERMAN, T.; ALLEN, J. Teams as a sub-process for knowledge management. **Journal of Knowledge Management**, v. 16(6), p. 963–977, 2012.

VLAANDEREN, K.; JANSEN, S.; BRINKKEMPER, S.; JASPERS, E. The agile requirements refinery: applying scrum principles to software product management. **Information and Software Technology**, v. 53, 2011.

WIIG, K. M. Knowledge management: an introduction and perspective. **Journal of Knowledge Management**, p. 6–14, 1997.

APÊNDICE A – formulário disponibilizado aos profissionais de tecnologia atuantes no mercado

Este formulário visa coletar percepções de usuários da área de tecnologia, com o intuito de avaliar os resultados obtidos com o desenvolvimento do Trabalho de Conclusão de Curso intitulado "Apoio automatizado para a gestão de conhecimento ágil". Portanto, este formulário deve ser preenchido após utilizar todas as funções disponibilizadas no protótipo denominado "KamTool". Observo que as informações coletadas serão tratadas de forma confidencial e serão usadas apenas para fins científicos e acadêmicos.

1) Seu gênero.

Feminino

Masculino

Outros

2) Função atual no mercado de TI.

Estagiário

Junior

Pleno

Sênior

Líder

Outros

3) Tempo de experiência no mercado de TI na função atual.

Menos de 1 ano

De 1 a 3 anos

Acima de 3 anos

Em uma escala de 1 (Discordo totalmente) a 5 (Concordo totalmente), avalie as afirmações a seguir:

4) O protótipo tem um uso fácil e intuitivo.

- 5) O design e os componentes na interface são agradáveis para o uso.
- 6) O conceito de gestão de conhecimento já era conhecido.
- 7) O protótipo foi útil para o auxílio na gestão ágil de projetos.
- 8) O protótipo foi útil para o auxílio na gestão de conhecimento.
- 9) O protótipo facilita a captura e o compartilhamento de conhecimento juntamente com a gestão de projeto.
- 10) A quantidade de funcionalidades disponibilizadas no protótipo foram satisfatórias.
- 11) A organização e estrutura estabelecidas para a documentação da base de conhecimento e dos boards do backlog do produto são adequadas.
- 12) Qual(is) funcionalidade(s) poderia(m) ser adicionadas ao protótipo?

Sua resposta: