



INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE PERNAMBUCO

Recife

Departamento acadêmico de sistema, processos e controles eletro-eletrônico

Análise e desenvolvimento de sistemas

GUILHERME HENRIQUE SANTOS OLIVEIRA

PLATAFORMA GENAGILE: relatório técnico

Recife

2024

GUILHERME HENRIQUE SANTOS OLIVEIRA

PLATAFORMA GENAGILE: relatório técnico

Trabalho de conclusão de curso apresentado à Coordenação de Pós-Graduação em Análise e desenvolvimento de sistemas do Instituto Federal de Ciência e Tecnologia de Pernambuco, como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Eduardo de Melo Vasconcelos

Recife

2024

Ficha catalográfica elaborada pela bibliotecária Danielle Castro da Silva CRB4/1457

O48p
2024

Oliveira, Guilherme Henrique Santos

PLATAFORMA GENAGILE: relatório técnico / Guilherme Henrique Santos Oliveira. --- Recife: O autor, 2024.

85f. il. Color.

Trabalho de Conclusão (Curso Superior Tecnológico em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Pernambuco, Recife, 2024.

Inclui Referências.

Orientador: Professor Eduardo de Melo Vasconcelos.

1. Gerenciamento de Software. 2. Gerenciamento de Projetos. 3. Metodologia Ágil. 4. Scrum. I. Título. II. Vasconcelos, Eduardo de Melo (orientador). III. Instituto Federal de Pernambuco.

CDD 658.404 (23 ed.)

Trabalho de Conclusão de Curso apresentado pelo estudante **Guilherme Henrique Santos Oliveira** à coordenação de Análise e Desenvolvimento de Sistemas, do Instituto Federal de Pernambuco, sob o título de “**PLATAFORMA GENAGILE: Relatório Técnico**”, orientado pelo **Prof. Dr. Eduardo de Melo Vasconcelos** e aprovado pela banca examinadora formada pelos professores:

Recife, 06 de novembro de 2024.

Prof Dr. Eduardo de Melo Vasconcelos
CTADS/DACS/IFPE

Prof Me. Anderson Luiz Souza Moreira
CTADS/DACS/IFPE

Prof. Dr. Fernando Ferreira de Carvalho
IFPE

Aluno: Guilherme Henrique Santos Oliveira

Dedico esse trabalho a toda minha família.

AGRADECIMENTOS

Gostaria de expressar meus sinceros agradecimentos a todas as pessoas que contribuíram para a conclusão deste trabalho. Este TCC representa o resultado de meses de dedicação, esforço e a concretização de um sonho, que não teria sido possível sem o apoio de muitas pessoas. Em especial ao meu pai, Carlos Henrique da Silva Oliveira, que sempre ficou cobrando para que eu estudasse e desde que nasci fez todo o possível para que pudesse chegar nesse momento da minha vida, a minha esposa, Naira Caroline Marinho de Andrade, que esteve comigo desde o início desse ciclo me apoiando e incentivando, a minha tia, Claudia Regina da Silva Oliveira, que me ajudou na minha trajetória desde o início e sempre acreditou em mim e a toda minha família, que ofereceram apoio emocional, compreensão e incentivo durante todo o processo, expresse minha profunda gratidão. Seu apoio foi essencial para enfrentar os desafios e superar os obstáculos.

Por fim, agradeço a todos os que de alguma forma contribuíram para a realização deste projeto acadêmico. Cada um de vocês desempenhou um papel crucial e é parte integrante deste trabalho.

RESUMO

Este relatório técnico utiliza como base a experiência de um Brasileiro em uma empresa de desenvolvimento de software situada no Porto Digital em Recife, Pernambuco. Após dois anos de experiência e imersão nas práticas utilizadas para gerenciamento de projetos, foi observado que para adoção de metodologias ágeis, diversas plataformas eram utilizadas, trazendo uma complexidade maior a equipe nesse processo. Com isso, foram levantados os principais problemas no ambiente de desenvolvimento e criado um sistema web com o objetivo de propor soluções aos itens identificados.

Palavras-chave: Ágil; scrum; metodologia.

ABSTRACT

This technical report uses as a basis the experience of a Brazilian in a software development company located in Porto Digital in Recife, Pernambuco. After two years of experience and immersion in the practices used for project management, it was observed that to adopt agile methodologies, several platforms were used, bringing greater complexity to the team in this process. With this, the main problems in the development environment were raised and a web system was created with the aim of proposing solutions to the identified items.

Keywords: Agile; scrum; methodology.

LISTA DE ILUSTRAÇÕES

Figura 1 - Processos no modelo em cascata.....	18
Figura 2 - Processos no modelo ágil.....	21
Figura 3 - Métodos ágeis mais usados.....	22
Figura 4 - Fluxo do Scrum.....	24
Figura 5 - Modelo de Sprint retrospective.....	25
Figura 6 - Diagrama de processos utilizados.....	27
Figura 7 - Arquitetura monolítica.....	38
Figura 8 - Arquitetura de microsserviços.....	39
Figura 9 - Arquitetura do projeto desenvolvido.....	40
Figura 10 - Login usuário linked.....	46
Figura 11 - Listagem de times.....	47
Figura 12 - Dashboard geral parte 1.....	48
Figura 13 - Dashboard geral parte 2.....	49
Figura 14 - Área de trabalho do usuário.....	50
Figura 15 - Visualização de anotação do usuário.....	51
Figura 16 - Central de integrações.....	52
Figura 17 - Configuração de integração com Gitlab.....	53
Figura 18 - Quadro de atividades.....	54
Figura 19 - Visualização simplificada de uma atividade.....	55
Figura 20 - Visualização detalhada de uma atividade.....	56
Figura 21 - Cronometrando atividade.....	56
Figura 22 - Ferramentas.....	57
Figura 23 - Configuração de planning poker.....	58
Figura 24 - Informações do planning poker.....	59
Figura 25 - Planning poker visão de usuário master.....	60
Figura 26 - Planning poker visão de usuário sem permissão.....	61
Figura 27 - Alteração de voto no Gitlab.....	62
Figura 28 - Configuração de ferramenta Sprint.....	63
Figura 29 - Menu de ferramenta Sprint.....	63
Figura 30 - Planejamento da Sprint.....	64
Figura 31 - Adição de atividades em Sprint planejada.....	65
Figura 32 - Configuração de período da Sprint.....	66
Figura 33 - Adição de atividade com Sprint bloqueada.....	67
Figura 34 - Configuração de scrum master da Sprint.....	68
Figura 35 - Quadro de atividades da Sprint.....	69
Figura 36 - Relatórios da Sprint atual por quantidade parte 1.....	70
Figura 37 - Relatórios da Sprint atual por quantidade parte 2.....	70
Figura 38 - Relatórios da Sprint atual por quantidade parte 3.....	71

Figura 39 - Relatórios da Sprint atual por voto parte 1.....	71
Figura 40 - Relatórios da Sprint atual por voto parte 2.....	72
Figura 41 - Relatórios da Sprint atual por voto parte 3.....	72
Figura 42 - Configuração da Sprint.....	73
Figura 43 - Finalização da Sprint.....	74
Figura 44 - Listagem de retrospectivas de Sprints.....	74
Figura 45 - Retrospectiva da Sprint.....	75
Figura 46 - Dashboard do time parte 1.....	76
Figura 47 - Dashboard do time parte 2.....	77
Figura 48 - Dashboard do time parte 3.....	77
Figura 49 - Listagem de automações.....	78
Figura 50 - Criação de automação parte 1.....	79
Figura 51 - Criação de automação parte 2.....	80
Figura 52 - Criação de automação parte 3.....	81

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Motivação.....	13
1.2	Justificativa.....	14
2	OBJETIVOS.....	16
2.1	Objetivo geral.....	16
2.2	Objetivos específicos.....	16
3	FUNDAMENTAÇÃO TEÓRICA.....	17
3.1	Metodologias ágeis.....	17
3.1.1	<i>Histórico.....</i>	17
3.1.2	<i>Manifesto ágil.....</i>	19
3.1.3	<i>Scrum.....</i>	21
3.1.3.1	Artefatos.....	23
3.1.3.2	Fluxo do processo.....	23
3.1.4	<i>Planning poker.....</i>	26
4	MÉTODO DE PESQUISA.....	27
4.1	Experiência que motivou o projeto.....	27
4.1.1	<i>Problemas identificados na experiência.....</i>	28
4.1.1.1	[P1] Múltiplos sistemas de gerenciamento de projeto.....	28
4.1.1.2	[P2] Acompanhamento de atividades em diferentes times.....	29
4.1.1.3	[P3] Planejamento de Sprint.....	29
4.1.1.4	[P4] Análise de Sprints anteriores.....	30
4.1.1.5	[P5] Obter duração de atividade.....	30
4.1.1.6	Planning poker.....	31
4.1.1.6.1	[P6] Possibilidade de alteração da votação.....	31
4.1.1.6.2	[P7] Inserção manual dos votos.....	31
4.1.1.6.3	[P8] Rastreamento de atividades votadas.....	31
4.1.1.6.4	[P9] Determinar tempo em votação.....	31
5	ESPECIFICAÇÃO DO SISTEMA DESENVOLVIDO.....	33
5.1	Sistema Genagile.....	33
5.2	Tecnologias utilizadas.....	33
5.2.1	<i>Angular.....</i>	34
5.2.2	<i>Java.....</i>	34
5.2.3	<i>Spring Boot.....</i>	34
5.2.4	<i>Banco de dados Mariadb.....</i>	35

5.2.5	Gateway	35
5.2.6	Netflix Eureka	36
5.2.7	RabbitMQ	36
5.3	Arquitetura	37
5.3.1	Arquitetura monolítica	37
5.3.2	Arquitetura de microsserviços	38
5.3.3	Arquitetura do sistema desenvolvido	39
5.4	Módulos desenvolvidos	40
5.4.1	Genagile	41
5.4.2	Genagile-auth	41
5.4.3	Genagile-activity	41
5.4.4	Genagile-methodology	41
5.4.5	Genagile-calendar	42
5.4.6	Genagile-automation-api	42
5.4.7	Genagile-automation	42
5.4.8	Genagile-integrations-api	43
5.4.9	Genagile-integrations-gitlab	43
5.4.10	Genagile-event-lib	43
5.4.11	Genagile-common-lib	44
6	RESULTADOS	45
6.1	Definições	45
6.1.1	Usuários	45
6.1.2	Times	46
6.2	Funcionalidades desenvolvidas	47
6.2.1	Funcionalidades externas ao time	48
6.2.1.1	Dashboard geral.....	48
6.2.1.2	Área de trabalho do usuário.....	49
6.2.2	Funcionalidades interna ao time	51
6.2.2.1	Integrações.....	51
6.2.2.2	Quadro de atividades.....	53
6.2.2.3	Ferramentas ágeis.....	57
6.2.2.3.1	Planning poker.....	57
6.2.2.3.1.1	Informações de votações.....	58
6.2.2.3.1.2	Votação com permissionamento e cronometragem.....	59
6.2.2.3.1.3	Integração com Gitlab.....	61
6.2.2.3.2	Sprint.....	62
6.2.2.3.2.1	Planejamento da Sprint.....	64
6.2.2.3.2.2	Sprint atual.....	68
6.2.2.3.2.3	Retrospectiva da Sprint.....	74
6.2.2.4	Dashboard do time.....	76
6.2.2.5	Automatização de ações.....	78

7.	CONCLUSÃO E TRABALHOS FUTUROS.....	82
	REFERÊNCIAS.....	84

1 INTRODUÇÃO

Em um mundo de negócios extremamente competitivo, encontrar novas formas de trabalhar para obter o máximo de eficiência, tornou-se uma necessidade para sair na frente da concorrência, e uma prática que está se tornando mais comum a cada ano é o uso de métodos ágeis. Essas formas de trabalho ágeis, como Scrum e Kanban, têm sido amplamente adotadas por empresas de diversos setores, não apenas na área de Tecnologia da Informação (TI), mas também na área de marketing, manufatura e recursos humanos.

Conforme um estudo realizado pela Digital.ai em 2022, 94% das empresas do setor de software relataram alguma experiência com o desenvolvimento ágil. Este dado é um indicativo da relevância crescente das metodologias ágeis, que, em 2021, registraram um aumento significativo em sua aplicação, passando de 37% para 86% entre as equipes da indústria de software. Além disso, o estudo destaca que setores não relacionados à Tecnologia da Informação (TI) também vêm experimentando uma rápida adoção dessas práticas, com a adoção dobrando ano a ano. Onde a integração eficaz de diferentes ferramentas e processos ágeis proporcionam às empresas a manterem-se competitivas e adaptáveis, criando um ambiente de trabalho mais colaborativo e eficiente.

1.1 Motivação

As metodologias ágeis levaram à adoção de diversas táticas para aumentar a produtividade e promover uma abordagem mais flexível e individualizada aos desafios de desenvolvimento de projetos. Segundo autores como Schwaber, essas metodologias, inicialmente concebidas como respostas eficazes às demandas do setor tecnológico, transcenderam suas origens e passaram a ser adotadas em diversos campos, impactando positivamente a maneira como as equipes abordam seus objetivos e entregas.

No entanto, a integração destas metodologias representa um desafio inerente à eficácia da implementação, uma vez que múltiplas ferramentas se tornam comuns a serem utilizadas durante as diferentes fases da metodologia adotada, fazendo com que as equipes utilizem diversas plataformas para gerenciar as diferentes fases de seus processos ágeis.

Ao fragmentar onde cada item da metodologia é realizado, o resultado pode ser a diminuição da eficiência do processo como um todo, uma vez que a integração de múltiplas plataformas pode gerar complexidade extra, ocasionando em aumento de tempo e recursos necessários para coordenar as atividades de forma harmoniosa.

1.2 Justificativa

Com base na experiência prática do autor, que atuou por mais de dois anos em uma empresa de desenvolvimento de software onde o gerenciamento de atividades seguia a metodologia Scrum, foi possível observar que, para adotar as práticas dessa metodologia, diversas plataformas eram utilizadas com o objetivo de atingir os resultados esperados. No entanto, ao longo do tempo, a divisão dos processos entre essas diferentes ferramentas gerou uma série de desafios, impactando a fluidez e a eficiência do desenvolvimento. A complexidade de integrar e coordenar as informações entre as plataformas se tornou um obstáculo significativo, prejudicando a agilidade do processo e exigindo mais tempo e recursos para a gestão das atividades.

O projeto proposto visa abordar esses desafios por meio da criação de uma plataforma web dedicada à integração e centralização das ferramentas utilizadas no Scrum. De acordo com Schwaber & Sutherland (2020), o Scrum depende de transparência, inspeção e adaptação, e uma solução centralizada pode facilitar esses pilares ao consolidar informações e simplificar fluxos de trabalho. A proposta inclui funcionalidades que permitem a unificação de projetos em um único ambiente, proporcionando praticidade e eficiência na gestão de atividades.

A iniciativa visa oferecer aos usuários a capacidade de unificar projetos que adotam a metodologia Scrum, proporcionando praticidade e otimização no processo de gestão de atividades, centralizando as informações, permitindo que os usuários gerenciem projetos com as ferramentas da metodologia em um único ambiente.

Além da praticidade, a plataforma busca aprimorar a tomada de decisões ao fornecer relatórios informativos sobre o progresso dos projetos, permitindo a identificação de pontos de melhoria e acompanhamento do desempenho geral do projeto. Essa abordagem está alinhada à recomendação de Cockburn (2001), que destaca a importância de feedback constante para ajustar estratégias e melhorar resultados em equipes ágeis.

Assim, o projeto não apenas visa a integração de ferramentas, mas também almeja criar um ambiente colaborativo e eficiente, proporcionando aos usuários uma solução completa para a gestão eficaz de projetos e atividades.

2 OBJETIVOS

2.1 Objetivo geral

Desenvolver uma plataforma web voltada a resolver os problemas identificados no gerenciamento de projetos, fundamentando-se na experiência prática do autor, que atuou diretamente em um ambiente de desenvolvimento ágil. Com base em boas práticas e normas reconhecidas para o gerenciamento ágil de projetos, como as estabelecidas pelo Scrum Guide (Schwaber & Sutherland, 2020), identificando os pontos que causam perda de desempenho nos processos existentes e implementando soluções que visam otimizar a produtividades e garantir a qualidade das entregas.

2.2 Objetivos específicos

- Desenvolver integração de plataformas, permitindo que os usuários acessem e gerenciem suas atividades de forma centralizada através do ambiente de gerenciamento de projetos.
- Implementar recursos de rastreamento de atividades e progresso, incluindo a criação de gráficos e relatórios informativos para uma análise mais detalhada.
- Disponibilizar ferramentas de metodologias ágeis, incorporando práticas recomendadas do Scrum, para que os usuários possam gerenciar seus projetos com base nas melhores práticas.
- Desenvolver propostas de soluções aos problemas identificados na experiência do autor em um ambiente de desenvolvimento de software.

3 FUNDAMENTAÇÃO TEÓRICA

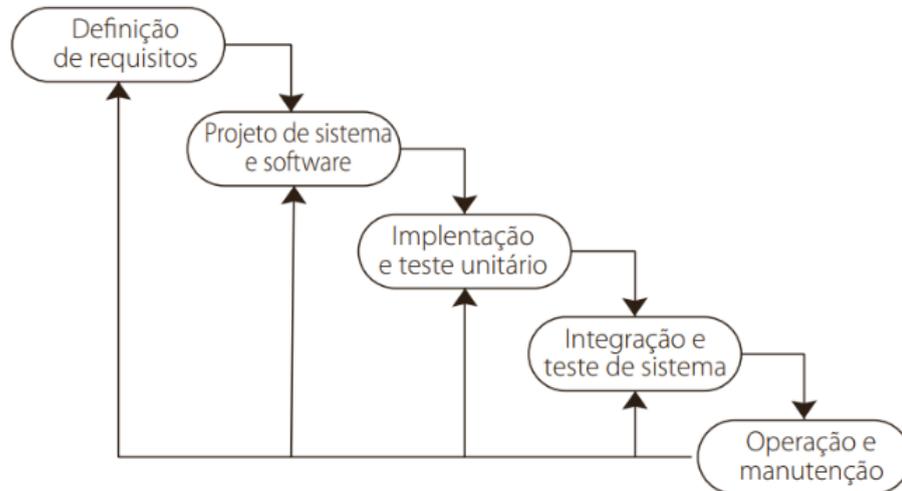
3.1 Metodologias ágeis

Segundo Camargo (2019), “As metodologias ágeis são abordagens para o desenvolvimento de produtos que estão alinhadas com os valores e princípios descritos no Manifesto Ágil para Desenvolvimento de Software”, e por conta disso tem como núcleo a busca pela flexibilidade e adaptabilidade, especialmente em cenários que requerem mudanças rápidas e constantes. Ao adotar o método ágil, as equipes se propõem a responder de forma eficaz e eficiente às transformações inerentes a projetos, promovendo uma maior capacidade de ajuste e uma abordagem mais ágil para atingir os objetivos estabelecidos, pois “a abordagem ágil prioriza as pessoas que fazem o trabalho e como elas trabalham juntas” (Agile Alliance, s.d.). Essa filosofia ágil não apenas redefine a forma como os projetos são conduzidos, mas também estimula uma cultura de colaboração, aprendizado contínuo e entrega de valor incremental de maneira consistente.

3.1.1 Histórico

Na década de 90, metodologias como a cascata eram amplamente empregadas no desenvolvimento de software, originando-se das disciplinas de engenharia, que colocam muita ênfase na fase de planejamento antes da construção.

Figura 1 - Processos no modelo em cascata



Fonte: Sommerville (2011, p. 20)

Entretanto, ao passar do tempo, foi identificado que essa abordagem não produzia os resultados desejados, tornando-se um problema crônico comum, caracterizado por atrasos e insatisfação na entrega de aplicações funcionais em muitas empresas de software. Conforme destacado por Martin Fowler (*"The new methodology"*, 2001), "Há um refrão que ouvi em todos os projetos problemáticos que encontrei. Os desenvolvedores vêm até mim e dizem: o problema desse projeto é que os requisitos estão sempre mudando.". Dessa forma, projetos que precisavam ter um replanejamento de forma constante, precisavam de uma outra forma de gerenciamento.

O surgimento das metodologias com o termo de "ágil" surgiu a partir de uma reunião em fevereiro de 2001 em Utah, EUA, quando 17 profissionais que trabalhavam com metodologias diferentes se reuniram e perceberam que mesmo usando abordagens e métodos distintos no desenvolvimento de software, existiam fundamentos compartilhados entre os mesmos (agilemanifesto, 2001). Ao fim da reunião, um documento foi criado, sendo uma declaração relatando os valores e princípios essenciais para metodologias ágeis, onde foram documentados no Manifesto Ágil para Desenvolvimento de Software.

3.1.2 Manifesto ágil

A criação do manifesto ágil representou uma resposta às deficiências das metodologias aplicadas naquele momento e trouxe consigo uma nova maneira de construir um software, entretanto, para que uma metodologia se encaixe no método ágil, precisa seguir os seguintes valores:

- Indivíduos e interações acima de processos e ferramentas.
- Software operante ao invés de documentações completas.
- Colaboração do cliente ao invés de negociações contratuais.
- Responder às mudanças ao invés de seguir um planejamento.

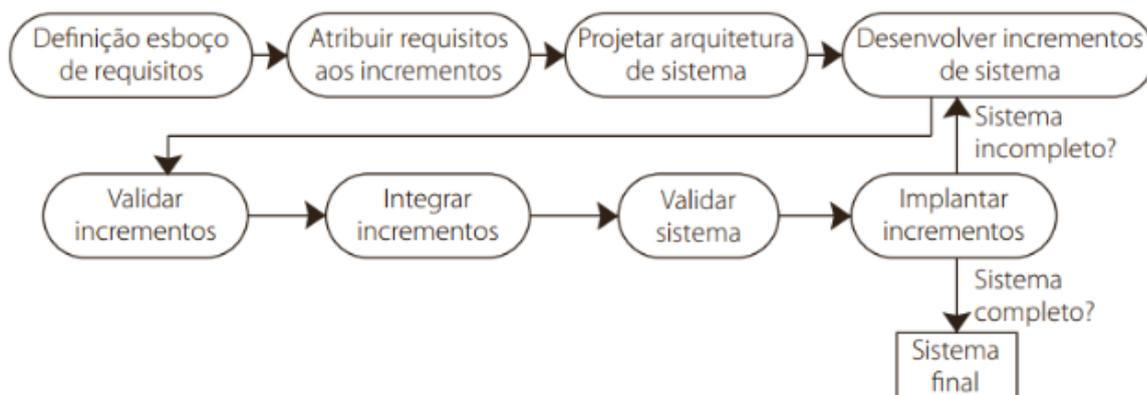
Para que seja possível alcançar os quatro valores, o manifesto ágil inclui 12 princípios com o intuito de guiar ao sucesso (agilemanifesto, 2001), sendo eles:

1. Satisfaça o cliente através da entrega contínua e adiantada de software valioso.
2. Aceite mudanças nos requisitos, mesmo tardiamente no desenvolvimento. Processos ágeis tiram vantagem das mudanças para proporcionar vantagem competitiva ao cliente.
3. Entregue software funcional frequentemente, com preferência para intervalos menores.
4. Colaboradores de negócio e desenvolvedores devem trabalhar juntos diariamente durante todo o projeto.
5. Construa projetos ao redor de indivíduos motivados. Dê a eles o ambiente e o suporte necessário e confie neles para fazer o trabalho.
6. A forma mais eficiente e eficaz de transmitir informações para e entre uma equipe de desenvolvimento é através de conversa face a face.

7. Software funcional é a medida primária de progresso.
8. Desenvolvimento sustentável, capaz de manter um ritmo constante.
9. Contínua atenção à excelência técnica e bom design.
10. Simplicidade, a arte de maximizar a quantidade de trabalho não realizado - é essencial.
11. As melhores arquiteturas, requisitos e designs emergem de equipes auto-organizáveis.
12. Em intervalos regulares, a equipe reflete sobre como se tornar mais eficaz, então ajusta e otimiza seu comportamento de acordo.

A mudança no paradigma de desenvolvimento de software a partir do método ágil trouxe consigo diversas mudanças, como o controle em processos adaptativos, onde é recomendado um desenvolvimento incremental, construindo e realizando entregas de pequenas partes do projeto, criando a capacidade de lidar com mudanças, atendendo assim ao segundo princípio do manifesto ágil. Dessa forma não é necessário desenvolver todo o projeto para que no fim seja apresentado para o cliente, pois muitas vezes o mesmo não entende desde do início o que realmente precisa e o método ágil possibilita que o mesmo entenda suas necessidades à medida que o projeto é construído, de tal forma que a cada iteração um pedaço do sistema vai ficando pronto e disponível para uso, tornando visível o estado da aplicação.

Figura 2 - Processos no modelo ágil



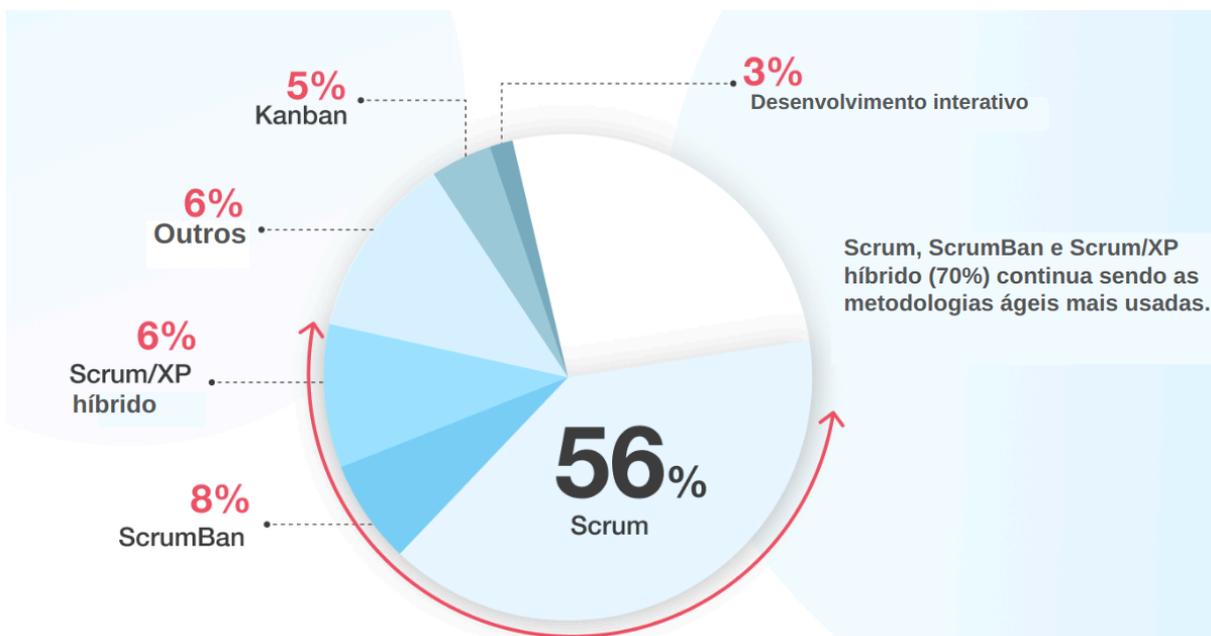
Fonte: Sommerville (2011, p.31)

3.1.3 Scrum

De acordo com Schwaber e Sutherland em “O guia scrum” (2020), o Scrum é uma metodologia ágil que remonta à década de 1990, antecedendo a criação do Manifesto Ágil e é uma estrutura leve que ajuda pessoas, equipes e organizações a gerar valor por meio de soluções adaptativas para problemas complexos. Diferentemente de fornecer uma estrutura rígida, o Scrum propositadamente permite a incorporação de práticas incrementais visando aprimorar continuamente o processo, onde em vez de fornecer instruções detalhadas às pessoas, as suas regras orientam seus relacionamentos e interações.

O artigo intitulado "Relatório sobre o estado do Agile de 2022 - 5 conclusões" publicado pela Scrum Inc. em dezembro de 2022 apresentou resultados da 16ª edição anual da pesquisa sobre técnicas e práticas ágeis, onde o estudo resultou na comprovação de que o Scrum é a metodologia ágil mais amplamente adotada nas organizações, com uma participação de mais de 50%, superando outros métodos ágeis em termos de utilização.

Figura 3 - Métodos ágeis mais usados



Fonte: Simform (2022)

Uma equipe Scrum é composta por um grupo reduzido de indivíduos, frequentemente dez ou menos, onde este time é multifuncional e auto gerenciado, caracterizado pela posse coletiva das habilidades necessárias para criar valor durante a Sprint.

O Scrum define três papéis específicos na equipe:

- **Dono do produto:** É um especialista no negócio do produto, representando a voz do cliente. Sua responsabilidade é maximizar o valor do produto, englobando a criação do *Release Plan*, a gestão do *Backlog* do produto e a definição da priorização das atividades.
- **Scrum Master:** É responsável por assegurar que as práticas do Scrum sejam seguidas durante o desenvolvimento do projeto. Atua como um facilitador, buscando solucionar impedimentos e acompanhando o projeto do time na Sprint.
- **Equipe de desenvolvimento:** São os membros comprometidos em criar um incremento utilizável na Sprint, ficando responsáveis por desenvolver as atividades do *Backlog* do produto.

3.1.3.1 Artefatos

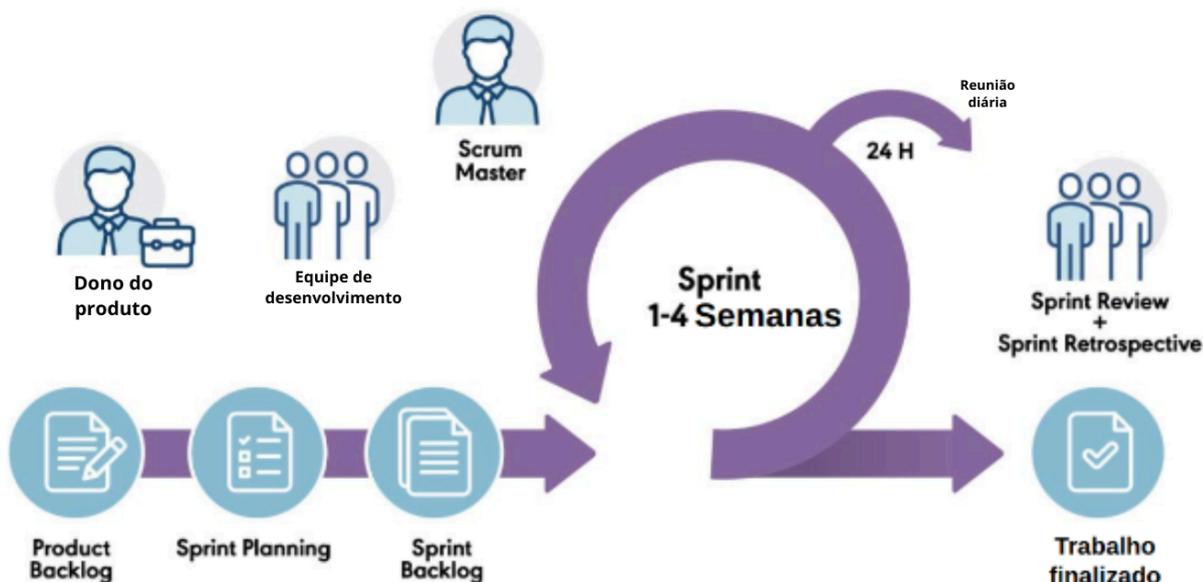
Os artefatos, são elementos tangíveis e visíveis que são criados, mantidos e utilizados durante o desenvolvimento do produto. Segundo Schwaber e Sutherland em “O guia scrum” (2020), “Eles são projetados para maximizar a transparência das informações principais”. No Scrum, existem três artefatos principais são definidos:

- Backlog do produto: É uma lista dinâmica e priorizada do necessário para melhorar o produto. O *Backlog do produto* é mantido e priorizado pelo dono do produto, que é responsável por representar os interesses dos *stakeholders* e garantir que a equipe esteja trabalhando nas atividades mais valiosas.
- Backlog da sprint: O *Backlog da sprint* é uma seleção do *Backlog do produto* para um ciclo de desenvolvimento específico, contendo os itens priorizados que a equipe se compromete a realizar durante a interação. Sendo dinâmico e podendo ser ajustado ao decorrer do seu progresso para acomodar mudanças ou novos insights.
- Incremento: É a soma de todos os itens do *Backlog* do produto que foram completados e atendem à Definição de Pronto no final de um ciclo de desenvolvimento, representando uma versão utilizável e potencialmente entregável do produto que incorpora as adições mais recentes de funcionalidades ou melhorias.

3.1.3.2 Fluxo do processo

O Scrum, fundamentado na filosofia ágil, opera em ciclos iterativos chamados de Sprints, sendo este um fluxo interativo e central para a flexibilidade e adaptabilidade do framework.

Figura 4 - Fluxo do Scrum



Fonte: pm-partners (2024)

Uma interação no Scrum se inicia a partir da priorização dos itens do *Backlog do produto* realizado pelo dono do produto, onde ocorre o primeiro evento do Scrum chamado de *Sprint Planning*. Nele, todo o time participa discutindo os pontos mais importantes do *Backlog* do produto e como eles serão mapeados para construção da meta da *Sprint*. Em seguida é definido o que poderá ser realizado na *Sprint*, onde quanto mais experiência com a metodologia, mais os desenvolvedores terão conhecimento do quanto conseguem desenvolver com base no seu desempenho passado.

Como consequência, o artefato *Backlog da sprint* é gerado, onde os desenvolvedores precisam definir critérios para que as atividades atendam a definição de concluído. A partir deste, o segundo evento chamado *Sprint* é iniciado, com o desenvolvimento dos itens.

Ao longo da *Sprint* a equipe realiza diariamente o terceiro evento, chamado de Daily Scrum, que possui o objetivo de responder às seguintes perguntas:

1. O que foi feito desde a última reunião diária?

2. O que planeja fazer até a próxima reunião diária?

3. Há algum impedimento no caminho?

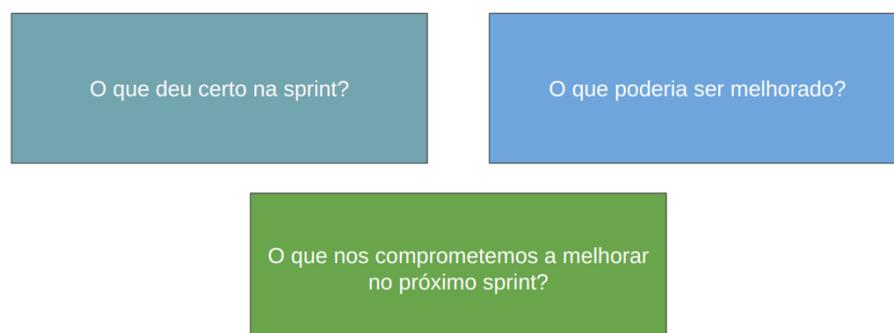
Normalmente essa reunião é limitada entre cinco e quinze minutos de duração, sendo realizada em pé, com o objetivo de criar uma transparência em relação ao que os indivíduos estão trabalhando, identificando obstáculos e facilitando a comunicação necessária.

Ao final do período da Sprint, uma reunião formal é realizada para realizar os dois últimos eventos:

- *Sprint review*: Tem o objetivo de analisar o resultado da *Sprint* e determinar adaptações futuras. É nesse momento onde o time apresenta os resultados desenvolvidos e o progresso em relação ao objetivo do produto.
- *Sprint retrospective*: Analisa como foi a última *Sprint* em relação aos indivíduos, interações, processos e ferramentas, buscando aumentar a qualidade e eficácia. Nesse momento é discutido o que deu certo, quais problemas foram encontrados e como foram (ou não) resolvidos, e buscam identificar possíveis mudanças para melhoria nas próximas *Sprints*.

Figura 5 - Modelo de Sprint retrospective

Um típico modelo de sprint retrospective



Fonte: careerfoundry (2023)

Após a conclusão do último evento da iteração, a experiência adquirida durante a *Sprint* desempenha um papel central no processo de melhoria contínua no

Scrum. Este ciclo de aprendizado constante é essencial para impulsionar não apenas o desenvolvimento do produto, mas também a eficácia e a coesão da equipe.

3.1.4 Planning poker

O *Planning Poker* é uma técnica de planejamento e estimativa baseada em consenso (CAMARGO, 2019). Essa abordagem colaborativa visa promover uma avaliação mais precisa e consensual do tempo e recursos requeridos em cada tarefa. Essa técnica geralmente envolve uma reunião da equipe, durante a qual cada membro recebe um conjunto de cartas numeradas que representam diferentes níveis de esforço ou complexidade. Tipicamente, as cartas são valores de Fibonacci, como 1, 2, 3, 5, 8, 13, etc., o que reflete a ideia de que estimar com precisão tarefas de alta complexidade é mais desafiador. Os desenvolvedores, então, discutem a natureza da atividade em questão e, individualmente, selecionam uma carta que melhor representa sua estimativa de esforço. Após a votação inicial, os membros da equipe compartilham suas escolhas e discutem as discrepâncias. Em seguida, ocorrem rodadas adicionais de votação até que um consenso seja alcançado.

Essa abordagem tem vários benefícios. Em primeiro lugar, ela promove a colaboração e a troca de ideias entre os membros da equipe, permitindo uma compreensão mais holística das tarefas. Além disso, o processo ajuda a minimizar vieses individuais, uma vez que os membros podem ser influenciados pelas opiniões dos outros durante as discussões.

4 MÉTODO DE PESQUISA

A metodologia escolhida para este estudo foi a Observação Direta Intensiva. Este método se mostrou particularmente adequado, pois possibilitou uma análise detalhada e contextualizada dos fenômenos investigados, aproveitando a proximidade com o ambiente pesquisado. Essa abordagem facilitou a coleta de dados diretamente das interações, comportamentos e processos cotidianos, garantindo uma percepção precisa e fiel à realidade da organização.

Para complementar e validar os dados coletados, foram utilizadas informações de outras fontes, como livros, artigos científicos e materiais disponíveis na internet. A combinação da observação prática com a pesquisa teórica proporcionou uma fundamentação sólida ao estudo, resultando em uma análise abrangente e bem embasada. Este estudo pode ser descrito e compreendido com base na imagem apresentada a seguir:

Figura 6 - Diagrama de processos utilizados



Fonte: própria (2023)

4.1 Experiência que motivou o projeto

Para realização desse projeto, foi utilizada a experiência do autor em uma empresa sediada no Porto Digital, onde o desenvolvimento de suas atividades eram através da metodologia Scrum. Essa experiência está relacionada ao período de 2 anos no qual o autor trabalhou na mesma e ao longo desse período identificou problemas que poderiam ser melhorados no gerenciamento dos projetos.

Por conta do autor ser funcionário da mesma, ao longo desse período o mesmo pode ter uma imersão completa em todos os processos e a dinâmica de trabalho da organização, além viver os problemas que impactam diariamente o processo. Essa abordagem detalhada permitiu uma compreensão abrangente dos fluxos de trabalho, das interações entre as equipes e das estratégias adotadas para atingir os objetivos propostos.

4.1.1 Problemas identificados na experiência

A análise aprofundada dos processos empregados na empresa utilizada como base, revelou aspectos que exercem um impacto adverso na eficiência dos integrantes da equipe ao longo do ciclo de atividades e que auxiliam a compreender e abordar as áreas que demandam melhorias. A seguir, será realizada uma descrição mais detalhada desses pontos, visando uma compreensão abrangente de como esses fatores afetam a produtividade durante o ciclo de desenvolvimento.

4.1.1.1 [P1] Múltiplos sistemas de gerenciamento de projeto

Com o aumento da adoção de metodologias ágeis, surgiram diversos sistemas voltados para aprimorar a produtividade nesse processo. No contexto em análise, três sistemas desempenham papéis fundamentais nesse gerenciamento.

- Monday é um sistema para gerenciamento de projetos utilizado por cerca de 180 mil clientes ao redor do mundo (MONDAY, 2024), e no caso em estudo, é utilizado para geração de métricas a partir dos dados das atividades realizadas proporcionando uma visão quantitativa e analítica do progresso .
- Gitlab é uma plataforma com mais de 30 milhões de usuários registrados (GITLAB, 2024), e no contexto da empresa em estudo, é utilizada para o registro e acompanhamento de atividades que serão desenvolvidas durante a *Sprint*, além de centralizar nele qualquer atividade futura que será desenvolvida.

- Scrum-poker auxilia na definição do esforço necessário para cada atividade, contribuindo para estimativas mais precisas e alinhadas com as capacidades da equipe (SCRUM POKER, 2024).

No entanto, vale destacar que no contexto em foco, essas plataformas operam de forma independente, sem integração entre si, e isso implica que quaisquer alterações realizadas em atividades precisam ser replicadas manualmente em cada uma das plataformas, um processo suscetível a falhas humanas e potenciais perda de dados relevantes para os usuários.

4.1.1.2 [P2] Acompanhamento de atividades em diferentes times

Existiam indivíduos que desempenhavam atividades em diferentes times dentro da organização e acabavam enfrentando o desafio de ter que acessar vários quadros de atividades e filtrar suas tarefas repetidamente. Isso cria uma falta de centralização, pois não existe um espaço dedicado onde o mesmo consiga trabalhar sem ter que navegar entre diferentes contextos de equipe.

Essa dispersão impacta a eficiência e a produtividade, uma vez que o usuário precisa constantemente alternar entre diferentes interfaces e contextos de trabalho. Além disso, essa falta de centralização dificulta o acompanhamento do progresso das atividades em cada time, bem como a coordenação entre as diferentes iniciativas em que o usuário está envolvido.

4.1.1.3 [P3] Planejamento de Sprint

No contexto do planejamento dos ciclos de atividades, a prática utilizada na empresa era realizar esse processo diretamente no quadro de desenvolvimento. Isso significa que as atividades são adicionadas conforme necessário, mesmo após o início da Sprint, sem que haja um impedimento formal para tal inclusão.

No entanto, é importante ressaltar que, dentro desse método, não havia um controle rigoroso da capacidade de trabalho da equipe durante a Sprint. Isso significa que não há uma gestão explícita dos pontos de história ou de outras

métricas de capacidade que são comuns em metodologias mais estruturadas, como o Scrum, por exemplo.

Essa falta de gerenciamento quantitativo pode trazer benefícios em termos de flexibilidade e agilidade para lidar com mudanças inesperadas, mas também pode representar desafios na previsibilidade do progresso e na capacidade de entrega dentro do prazo estabelecido.

4.1.1.4 [P4] Análise de Sprints anteriores

Ao concluir um Sprint, era realizada uma retrospectiva para revisar o que foi feito. Durante essa etapa, eram identificadas as atividades realizadas ao longo do ciclo, utilizando relatórios disponíveis na plataforma Monday. Esses relatórios são filtrados para mostrar as atividades relacionadas ao ciclo em questão. No entanto, ao final da reunião de retrospectiva, as atividades que não foram concluídas são transferidas para a próxima Sprint, criando uma dificuldade na análise de Sprints anteriores, pois os relatórios refletem apenas as atividades que estão na Sprint selecionada. Como resultado, não era possível avaliar quantas atividades não foram concluídas em ciclos anteriores, o que limita a análise dos ciclos passados.

4.1.1.5 [P5] Obter duração de atividade

Com a utilização de diferentes plataformas para o gerenciamento das atividades (Gitlab e Monday), a empresa exigia que os membros dos times registrassem o tempo gasto para concluir uma atividade. Alguns membros utilizavam outros sistemas para cronometrar o tempo, enquanto outros calculavam o tempo apenas a partir do início da atividade e alguns não realizavam esse processo por esquecer que era necessário. Isso levava a imprecisões, especialmente quando uma atividade era pausada, resultando em tempo gasto incorreto.

4.1.1.6 Planning poker

A empresa em foco utilizava uma plataforma web que possibilita a realização desse processo, entretanto, enfrenta alguns desafios para a implementar de forma eficaz, sendo esses itens relatados neste tópico.

4.1.1.6.1 [P6] Possibilidade de alteração da votação

Durante a votação de uma atividade, a plataforma utilizada (Scrum-poker) apresenta a vulnerabilidade de qualquer usuário realizar a troca do seu voto logo após a visualização da escolha dos demais indivíduos, dessa forma, o processo de discussão sobre divergências no voto que representa o esforço daquele item não era realizado.

4.1.1.6.2 [P7] Inserção manual dos votos

Após conclusão da determinação do esforço necessário para uma atividade, é necessário inserir manualmente essa informação em todos os sistemas de gerenciamento de atividades utilizados (Gitlab e Monday), o que não apenas consome tempo, mas também aumenta o risco de erros humanos nesse processo.

4.1.1.6.3 [P8] Rastreamento de atividades votadas

A plataforma utilizada (Scrum-poker) não fornece a capacidade de terminar quantas atividades foram avaliadas em um período de tempo ou se faltam itens a serem avaliados, isso porque como a mesma não possui integração com as demais plataformas (Gitlab e Monday), não existe essa informação.

4.1.1.6.4 [P9] Determinar tempo em votação

Durante o processo de determinação do esforço para realização de uma atividade, pode ocorrer divergência de opiniões e dessa forma ocorre a discussão

para se chegar em um consenso. Quando esse processo é realizado para diversas atividades, pode acabar levando um tempo considerável e na plataforma utilizada (Scrum-poker), não é possível avaliar quanto tempo a equipe leva para finalizar esse processo, o que dificulta a análise da eficiência desse item no desenvolvimento do projeto.

5 ESPECIFICAÇÃO DO SISTEMA DESENVOLVIDO

Após a experiência vivenciada na empresa em questão e entender as especificações para atender aos problemas identificados, ficou claro que era crucial criar um ambiente que fosse centralizado. Nesse contexto, é necessário o desenvolvimento de uma plataforma voltada para o gerenciamento de projetos, que ofereça aos usuários a conveniência de realizar todas as etapas necessárias em um único local, e neste tópico será desenvolvido sobre o sistema construído para atender aos desafios a serem superados.

5.1 Sistema Genagile

Como resultado das considerações anteriores, foi desenvolvido um sistema denominado Genagile, com o propósito principal de integrar diferentes plataformas de gerenciamento de projetos e oferecer as etapas do método ágil Scrum em um único ambiente centralizado. Isso significa que os usuários têm a opção de utilizar o Genagile de forma autônoma, sem depender de integrações externas ou de continuar usando outras plataformas, mantendo, ao mesmo tempo, a capacidade de centralizar alterações e métricas em um único ponto.

Essa abordagem oferece flexibilidade aos usuários, permitindo que escolham a melhor forma de gerenciar seus projetos, ao mesmo tempo em que fornece um hub centralizado para otimizar a integração, comunicação e consistência nas atividades relacionadas ao método ágil Scrum. O Genagile busca simplificar e unificar o processo de gerenciamento, oferecendo uma solução às necessidades levantadas no tópico 4.1.1.

5.2 Tecnologias utilizadas

Para conseguir atender as propostas do sistema Genagile, uma série de tecnologias foram utilizadas, seja visando a facilidade na sua implementação ou uma melhor estruturação em sua arquitetura, e nesse tópico, será tratado sobre as principais tecnologias utilizadas para construção do Genagile.

5.2.1 Angular

O Angular, mantido pelo Google, é um framework que possibilita a criação de interfaces de usuário dinâmicas e interativas, onde sua arquitetura se baseia em componentes facilitando a construção de aplicações front-end escaláveis e de fácil manutenção.

Ao adotar o Angular, os desenvolvedores podem aproveitar as vantagens de um sistema de design coeso e consistente, o que amplia as possibilidades de design e funcionalidades disponíveis para os projetos. A combinação de ferramentas como Angular Material e ngx-bootstrap amplia ainda mais essas possibilidades, proporcionando uma base sólida para o desenvolvimento de aplicações web modernas e complexas (El-Masry, 2018).

5.2.2 Java

Java é uma linguagem de programação orientada a objetos que teve sua origem na década de 1990, quando foi desenvolvida pela Sun Microsystems. Segundo Deitel (2020), ela foi criada com o objetivo de ser uma linguagem portátil e de alto desempenho, capaz de ser executada em diferentes plataformas sem a necessidade de recompilação do código-fonte. Uma das razões para a popularidade contínua do Java ao longo dos anos é sua capacidade de adaptação e atualização constante. Como mencionado por Liang (2020), o Java tem evoluído com o tempo, incorporando novos recursos e melhorias de desempenho que mantêm a linguagem relevante e competitiva no cenário da programação.

5.2.3 Spring Boot

É um *framework* que simplifica o desenvolvimento de aplicativos Java, fornecendo um ambiente de configuração simplificado e uma estrutura que facilita a criação de aplicativos robustos. Conforme afirmado por Karanam (2020), o Spring Boot é uma extensão do popular *framework* Spring *Framework*, projetado para reduzir a complexidade da configuração do Spring e acelerar o processo de

desenvolvimento. Isso significa que o Spring Boot adota padrões e convenções que ajudam a reduzir a quantidade de configuração manual necessária por parte do desenvolvedor, permitindo que eles se concentrem mais na lógica de negócios e nas funcionalidades do aplicativo.

5.2.4 Banco de dados Mariadb

Os bancos de dados são sistemas de armazenamento de informações que permitem a recuperação e manipulação de dados de forma estruturada e eficiente. Esses sistemas são extremamente importantes e segundo Danielle Oliveira e Tathiana Rodnarchuki (2024), “são a base da era da informação”, isso porque eles possuem um papel fundamental no dia a dia das pessoas, uma vez que o mundo está cada vez mais orientado por dados.

Sua importância vem oriunda de todas as funcionalidades disponibilizadas por esses sistemas, sendo elas: Armazenamento de dados, recuperação eficiente, consistência e integridade, segurança dos dados, suporte à tomada de decisões, escalabilidade, integração com aplicações, *backup* e recuperação, compartilhamento de dados e gerenciamento de transações.

O MariaDB é um dos tipos de banco de dados existentes no mercado, sendo amplamente utilizado como alternativa ao MySQL e desenvolvido pela comunidade de desenvolvedores após a aquisição do MySQL pela Oracle Corporation. O MariaDB mantém compatibilidade com a linguagem SQL e oferece recursos avançados de desempenho, segurança e escalabilidade.

5.2.5 Gateway

Um *gateway*, ou "porta de entrada", é um ponto central para a comunicação entre diversos recursos, funcionando como uma ponte que conecta diferentes sistemas e aplicações. Ele facilita a integração e o gerenciamento do fluxo de informações, garantindo que a troca de dados seja eficiente, segura e bem coordenada.

No contexto de comunicação entre módulos distribuídos em uma aplicação, um gateway pode trabalhar sendo a porta de entrada para qualquer módulo, resultando em uma comunicação mais ágil e confiável, capaz de suportar o crescimento e gerenciar os dados trafegados das aplicações.

5.2.6 Netflix Eureka

O Netflix Eureka é uma solução que auxilia a comunicação entre aplicações distribuídas, de modo que cada módulo da aplicação ao ficar disponível realiza um registro no *Eureka Server*, onde o servidor mantém registrado todos os módulos da aplicação que estão disponíveis, além de atribuir e realizar o controle de qual porta o mesmo está operando.

Em um ambiente distribuído o controle realizado através do *Eureka Server* é fundamental, uma vez que o mesmo pode ser integrado com um *gateway* e dessa forma ao realizar a requisição para um módulo específico passando através do *gateway*, o mesmo consegue realizar o balanceamento de carga entre as instâncias fazendo a identificação da porta automaticamente.

5.2.7 RabbitMQ

O RabbitMQ é um corretor de mensagens, e através dele é possível transmitir informações de forma assíncrona através de um canal de comunicação que utiliza uma fila para armazenamento dos dados.

Basicamente com o RabbitMQ é possível criar produtores em uma aplicação, que serão responsáveis por pegar um dado e enviar para ele, onde será armazenado em uma fila para processamento assíncrono, e em outra parte do sistema, definir um consumidor, que irá se comunicar com o RabbitMQ e ficará buscando os dados dessa fila, para realizar algum tipo de processamento. Dessa forma, a informação pode ser processada de forma assíncrona.

5.3 Arquitetura

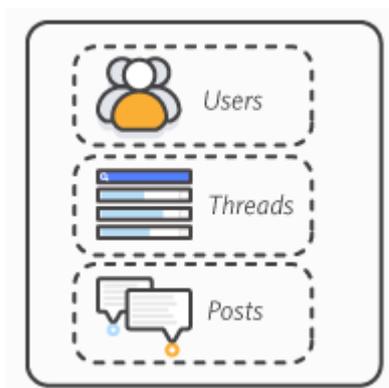
Para a construção de um projeto, a escolha da arquitetura é fundamental para atender ao seu propósito. No entanto, é importante compreender que o desenvolvimento ocorre em fases e adotar uma arquitetura complexa na fase inicial pode aumentar desnecessariamente a complexidade do sistema desde o começo. Já uma abordagem mais equilibrada permite que a complexidade seja introduzida gradualmente, conforme o projeto evolui, evitando sobrecargas prematuras e facilitando a gestão e a adaptação às necessidades emergentes ao longo do desenvolvimento.

Tomando isso como base, esse tópico irá relatar as arquiteturas que foram utilizadas no desenvolvimento do Genagile e o resultado final.

5.3.1 Arquitetura monolítica

A arquitetura monolítica é uma das mais tradicionais utilizada no mercado de desenvolvimento de software, onde basicamente todo o código é contido dentro de um único módulo e esse padrão foi utilizado durante o desenvolvimento inicial do Genagile pois ele não requer um grande planejamento inicial e segundo Newman no seu livro *Criando microsserviços*, essa arquitetura “pode resultar em fluxos de trabalhos muito mais simples para os desenvolvedores e atividade como monitoração, resolução de problemas e testes fim a fim podem ser bastante simplificados, além de simplificar a reutilização do código dentro do sistema.”.

Figura 7 - Arquitetura monolítica



Fonte: Aws Amazon

Ao observar a estrutura dessa arquitetura, nota-se que ao criar um novo processo não é necessário se preocupar com a comunicação com os demais, pois todos os componentes estão integrados em um único módulo de forma que não é necessário aprender e implementar diversas tecnologias para realizar a comunicação entre os processos.

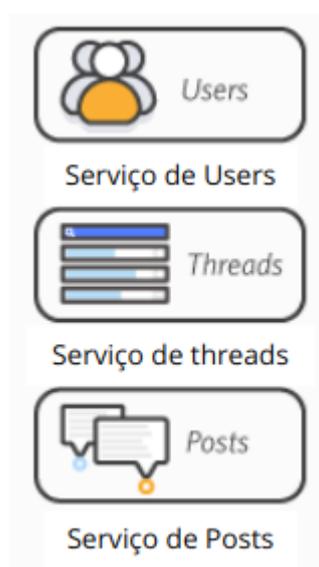
Entretanto, à medida que o sistema cresce em questão de funcionalidades e quantidade de código, a arquitetura monolítica começa a apresentar desafios. A escalabilidade se torna mais difícil, pois o aumento da carga de trabalho em um único processo pode levar a gargalos, além de que a manutenção e a implantação de novas funcionalidades podem se tornar complicadas, já que qualquer alteração no código pode impactar a aplicação inteira visto que tudo está em um único módulo.

5.3.2 Arquitetura de microsserviços

A arquitetura de microsserviços é uma forma de estruturar a aplicação realizando a divisão em módulos menores e segundo Newman em *Criando microsserviços*, “ao combinar os conceitos de ocultação de informações e um design orientado a domínios com a eficácia dos sistemas distribuídos, os microsserviços podem ajudar a proporcionar ganhos significativos em comparação com outras formas de arquiteturas distribuídas.”.

Ao longo do desenvolvimento do Genagile, os problemas da arquitetura monolítica começaram a aparecer, e com isso foi realizada a migração para a arquitetura de microsserviços, aumentando a especialização de cada módulo e facilitando a manutenção e escrita de novas funcionalidades, uma vez que a quantidade de código em um módulo isolado é consideravelmente menor em comparação a uma aplicação monolítica que contém um único módulo com todas as funcionalidades.

Figura 8 - Arquitetura de microsserviços



Fonte: Aws Amazon

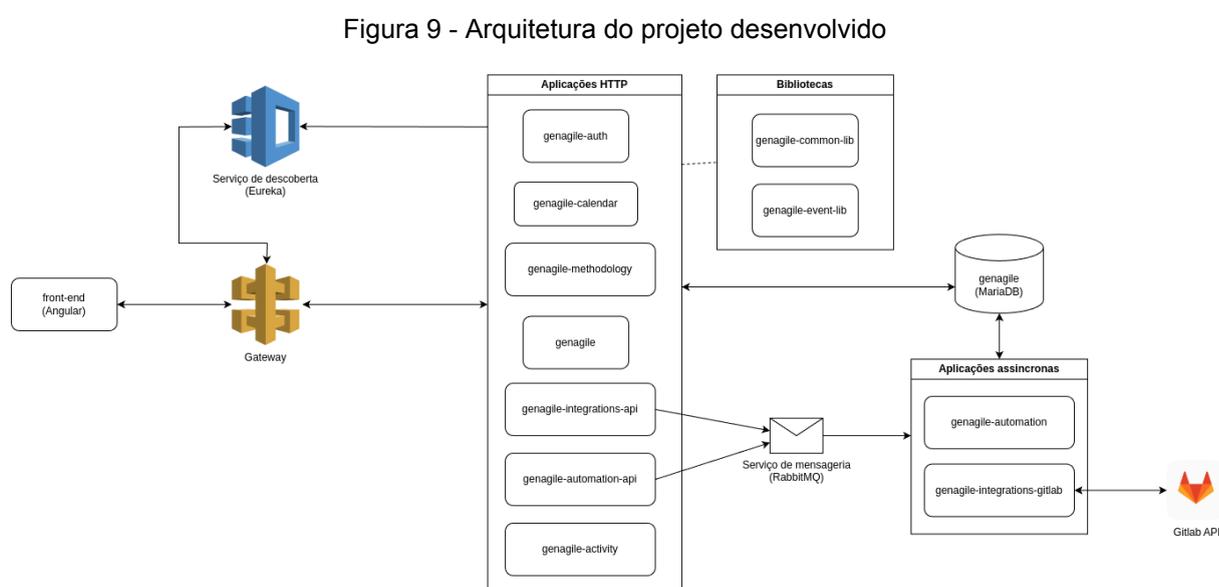
Ao analisar a figura 8 e comparar com a figura 7, é possível notar que na arquitetura de microsserviços cada módulo concentra apenas as funcionalidades pertinentes ao mesmo, e dessa forma, caso precise acessar outra parte do sistema, é necessário buscar as informações em outro módulo da aplicação, criando assim um isolamento entre os domínios.

5.3.3 Arquitetura do sistema desenvolvido

Após a fase inicial do desenvolvimento, a adoção da arquitetura de microsserviços foi essencial para realização da separação da aplicação em módulos

menores e especializados, que centralizam determinadas funções do sistema e definem o que um domínio terá visível para outras aplicações, além de auxiliar na escalabilidade, podendo fazer com que apenas as partes do sistema que possuam maior utilização tenham mais instâncias para suportar a carga de trabalho.

Com isso, a arquitetura final da aplicação ficou como relatada na figura 9 abaixo:



Fonte: própria (2024)

Dessa forma, com o uso das tecnologias informadas no tópico 5.2, foi possível aproveitar dos conceitos da arquitetura de microsserviços e desenvolver uma arquitetura para o Genagile que separa os principais módulos da aplicação, diminuindo o impacto que alterações nesses possam gerar problemas em demais partes do sistema.

5.4 Módulos desenvolvidos

Ao utilizar a arquitetura de microsserviços, diversos módulos foram criados e neste tópico, serão descritos detalhadamente cada um desses módulos e suas respectivas responsabilidades na aplicação.

5.4.1 Genagile

Esse é o módulo principal da aplicação, nele se encontra os itens relacionados a manipulação de usuário e time, sendo a base do restante do sistema, onde o mesmo foi o primeiro módulo desenvolvido na aplicação no início, quando sua arquitetura era a de monolito e ao longo do desenvolvimento, as demais funcionalidades foram retiradas deste e receberam um módulo específico.

5.4.2 Genagile-auth

Este módulo trabalha realizando a centralização de controle de autenticação e permissões no sistema, onde todos os módulos precisam inicialmente enviar uma requisição para o mesmo para autenticar ou validar o token informado pelo usuário.

Seu papel é fundamental ao observar a aplicação do genagile como um todo, pois isso permite que apenas um projeto tenha conhecimento de como realizar a autenticação e o controle das permissões, simplificando e especificando o papel dos demais módulos.

5.4.3 Genagile-activity

O genagile-activity é o módulo que centraliza as operações relacionadas às atividades e além disso, o mesmo é responsável pela geração dos dados que serão utilizados pelos relatórios em tela, uma vez que muitos destes são informações de progresso de conclusão de atividades e o único projeto que possui acesso a esses dados e define o que será visível é o genagile-activity.

5.4.4 Genagile-methodology

O genagile-methodology é responsável por toda a parte de metodologia de gerenciamento configurada em um time, centralizando as ferramentas que cada time possui e suas configurações.

5.4.5 Genagile-calendar

Esse projeto exemplifica a forma ideal de como os módulos devem ser, pois ele centralizada toda a funcionalidade de calendário e faz com que caso alguma outra parte do sistema precisa criar, alterar, busca ou deletar um evento no calendário, seja passado pelo mesmo, onde será realizado o controle de quais informações podem ser visíveis.

O módulo de genagile-calendar é responsável por todas as operações relacionadas a eventos de calendário, incluindo a criação, alteração, busca e exclusão de eventos. Este módulo atua como um ponto centralizado para todas as funcionalidades de calendário, assegurando que as operações sejam realizadas de forma consistente e que o acesso às informações seja devidamente controlado.

5.4.6 Genagile-automation-api

O genagile-automation-api disponibiliza uma *Application Programming Interface* (API), para o gerenciamento de automações no sistema. A partir dele, é possível saber quais automações é possível definir quais as condições para a automação executar e informar o que será feito caso atenda aos itens necessários para execução.

O seu objetivo principal é isolar a API da execução das automações, visto que, em um ambiente altamente escalável, estará ocorrendo mais execuções de automação do que criação ou alterações de automatizações no sistema.

5.4.7 Genagile-automation

Nesse projeto, é centralizada toda a parte de automações que são cadastradas pelos usuários da aplicação, seu objetivo é automatizar ações, para execução de forma assíncrona, que façam o usuário diminuir tarefas repetidas, como por exemplo, ao desabilitar um usuário, todas suas atividades em desenvolvimento podem ser atribuídas a outro desenvolvedor ou apenas ficar sem atribuição.

5.4.8 Genagile-integrations-api

O `genagile-integration-api` fornece um conjunto de *interfaces* e *endpoints* que permitem a configuração de conexões com integrações externas. A partir das informações configuradas, o módulo específico da integração configurada se encarrega de gerenciar a comunicação com serviço externo, como por exemplo, o Gitlab.

5.4.9 Genagile-integrations-gitlab

Esse projeto é responsável por isolar a comunicação com a plataforma Gitlab, fazendo com que demais alterações no sistema não influenciam a forma como o Genagile envia as informações para esse serviço externo.

O `genagile-integrations-gitlab` trabalha recebendo eventos assíncronos de criação, alteração ou deleções de entidades como usuário e atividade, seja estes oriundos do próprio Genagile através de RabbitMQ ou do Gitlab a partir de webhooks, fazendo a devida atualização em ambas plataformas.

5.4.10 Genagile-event-lib

É a biblioteca responsável pelo registro e gerenciamento de eventos que disparam execuções assíncronas de automações. Com isso ela permite que projetos possam informar quando determinadas ações ocorrem e, a partir dessas ações, disparar eventos assíncronos para outros projetos que necessitem ser notificados. Isso é particularmente útil em sistemas complexos, onde múltiplos serviços precisam colaborar e reagir a eventos em tempo real, sem depender de uma comunicação síncrona e direta.

Um exemplo de uso da mesma, é quando ocorre uma alteração em algum atributo em uma determinada atividade e o time possui integração com outras plataformas. É necessário obter esse evento de alteração e disparar uma comunicação assíncrona para o projeto responsável por obter o evento e atualizar na plataforma de destino.

5.4.11 Genagile-common-lib

Com o objetivo de centralizar códigos que estavam se repetindo em diversos projetos e que se comportavam como métodos utilitários, foi criada a biblioteca genagile-common-lib, que auxilia a todos os projetos utilizarem um mesmo código e assim seguindo o padrão estabelecido para tratamento de exceções, validações de atributos e etc.

6 RESULTADOS

Após o desenvolvimento da plataforma Genagile, uma série de funcionalidades foram desenvolvidas buscando a solução dos problemas identificados no tópico 4.1.1, e nesse momento será detalhado cada uma dessas funções e como elas podem auxiliar no gerenciamento de projetos utilizando ou não a metodologia Scrum.

6.1 Definições

Para uma compreensão completa do funcionamento do sistema, é crucial entender como ele organiza seus usuários e projetos, e dentro do Genagile, a estruturação dos usuários e a definição de times desempenham um papel fundamental no gerenciamento eficiente das atividades e na colaboração entre os membros da equipe. Neste tópico, vamos nos aprofundar na organização dos usuários e times.

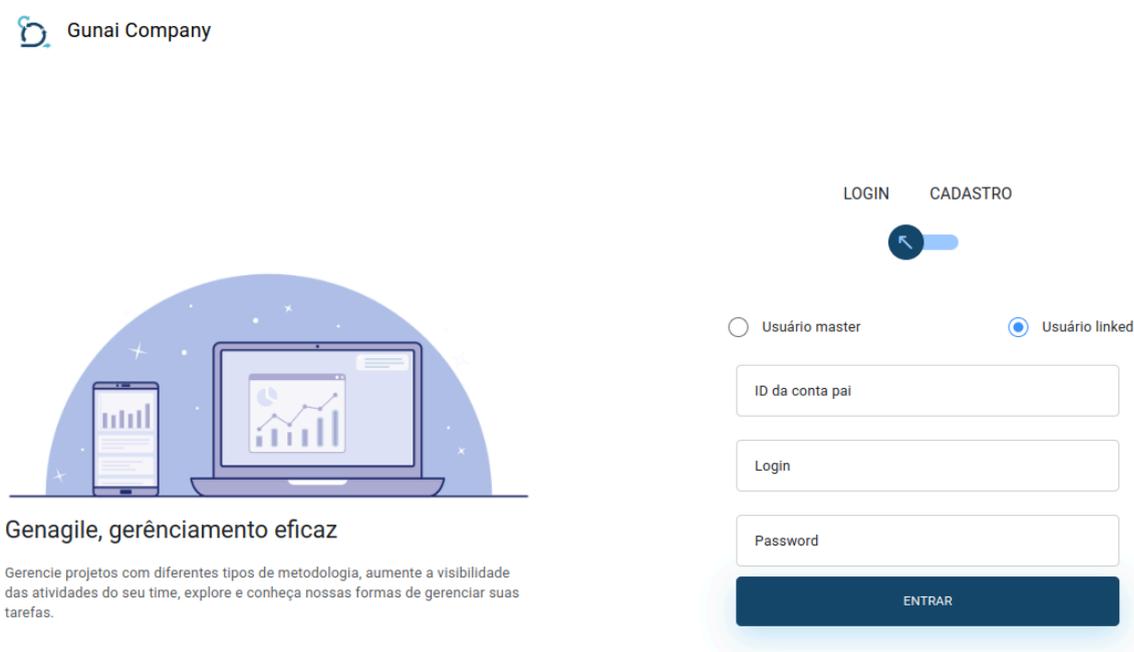
6.1.1 Usuários

Analisando o modelo de acesso da AWS (*Amazon Web Service*), o ambiente é estruturado com a distinção entre usuários *Master* e usuários *Linked*. Este modelo foi adotado visando proporcionar uma segregação de ambientes entre diferentes clientes. Em conformidade com esse modelo, um cliente estabelece sua própria conta de acesso, que atua como uma conta *master*, permitindo que ele configure seu ambiente de acordo com suas necessidades específicas.

A partir desse ponto, o cliente pode criar sub usuários vinculados à sua conta, esses são os usuários *linked*, que terão acesso ao ambiente configurado pela conta *master*. Essa abordagem oferece flexibilidade e controle ao cliente para personalizar e gerenciar seu próprio ambiente, enquanto ainda mantém uma estrutura hierárquica que facilita a administração.

Quando um usuário do tipo *linked* precisa acessar o sistema, é necessário informar o identificador da conta *master*. Essa prática garante a autenticação e autorização adequadas, garantindo que cada usuário tenha acesso apenas às partes do ambiente designadas pela conta *master* correspondente. Essa divisão de contas proporciona uma camada adicional de segurança e organização em ambientes com múltiplos clientes.

Figura 10 - Login usuário linked



The image shows a login interface for a linked user. At the top left is the logo for "Gunai Company". The interface has two tabs: "LOGIN" and "CADASTRO", with "LOGIN" being the active tab. Below the tabs are two radio buttons: "Usuário master" (unselected) and "Usuário linked" (selected). The form contains three input fields: "ID da conta pai", "Login", and "Password". At the bottom is a dark blue button labeled "ENTRAR". To the left of the form is a promotional graphic for "Genagile, gerenciamento eficaz" featuring a smartphone and a laptop displaying charts, with text describing project management capabilities.

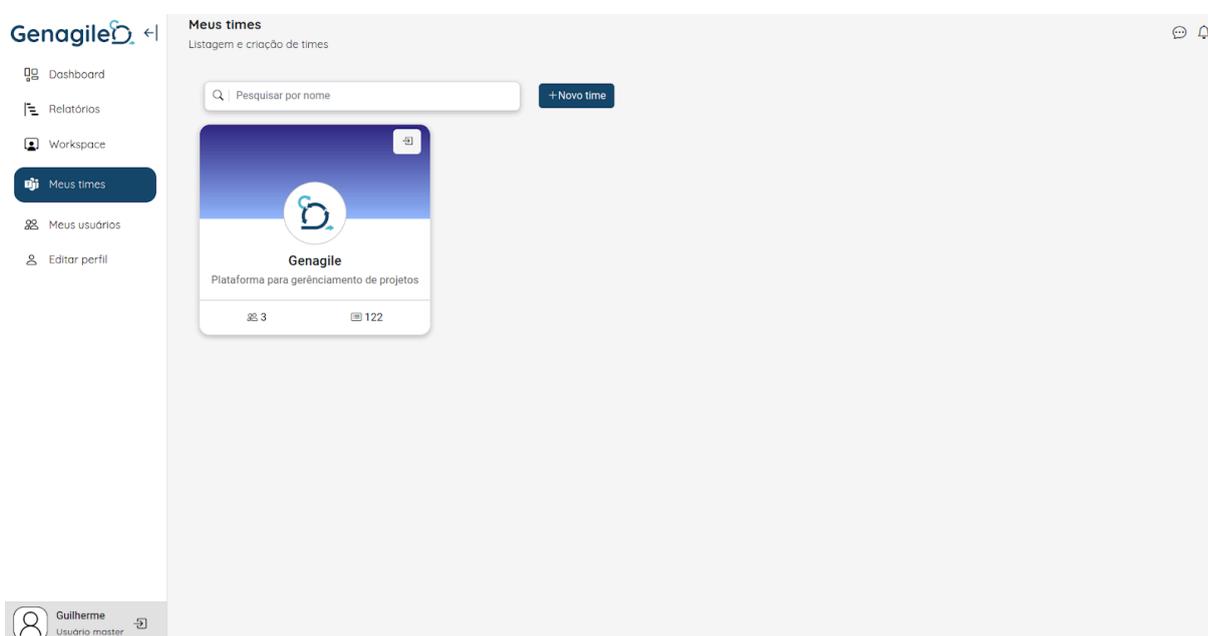
Fonte: própria (2024)

6.1.2 Times

Para segmentar os projetos de forma eficiente e personalizada, foi adotado o conceito de times dentro do Genagile. Isso permite que o usuário *master* tenha o controle para criar um time e configurar seu gerenciamento de acordo com as necessidades específicas da equipe. Ao criar um time, o usuário *master* pode definir não apenas as ferramentas a serem utilizadas, mas também a organização das atividades dentro do time.

Essa abordagem oferece uma flexibilidade significativa, pois cada time pode ter sua própria organização de tarefas, fluxos de trabalho e métodos de acompanhamento de projetos. Dessa forma, cada time pode adotar práticas e ferramentas que melhor se adequem às suas características e objetivos, contribuindo para uma gestão mais eficiente e alinhada com as necessidades específicas de cada equipe.

Figura 11 - Listagem de times



Fonte: própria (2024)

6.2 Funcionalidades desenvolvidas

Com o objetivo de resolver os problemas mencionados no tópico 4.1.1 e considerando o contexto de múltiplos times, foram desenvolvidas funcionalidades que têm aplicações tanto internas quanto externas aos times. Neste ponto, será detalhado essas funcionalidades e sua relevância para a gestão eficaz em ambientes com várias equipes.

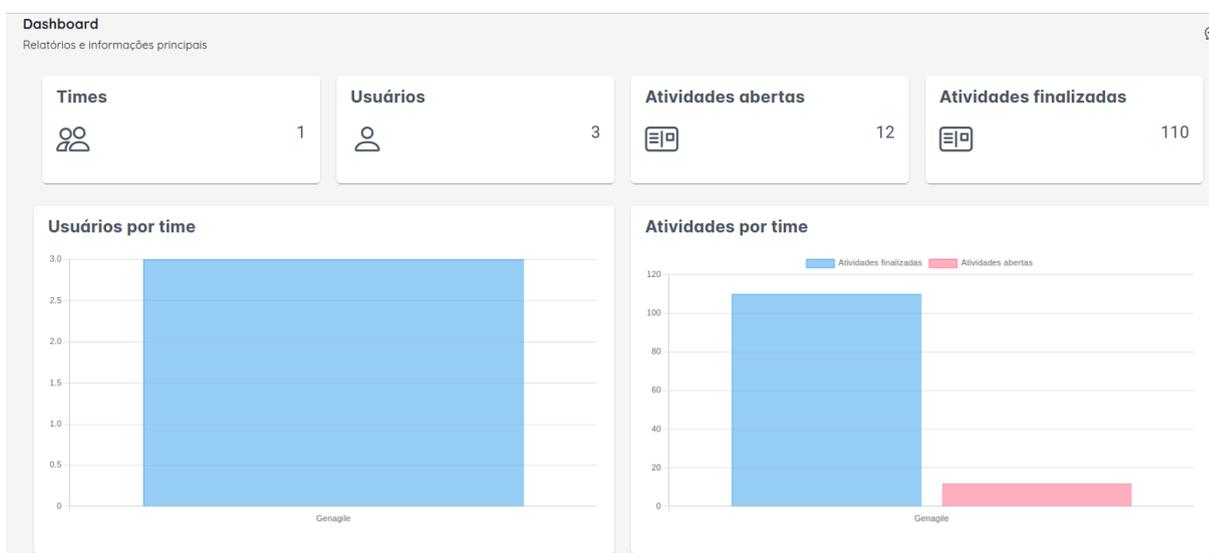
6.2.1 Funcionalidades externas ao time

As funções fora do contexto de um time abrangem a criação de times, gestão de usuários (incluindo criação e edição), personalização do perfil do usuário logado, além de funcionalidades como relatórios e a área de trabalho do usuário. Neste tópico, vamos detalhar duas das principais funcionalidades externas aos times: Relatórios e área de trabalho do usuário.

6.2.1.1 Dashboard geral

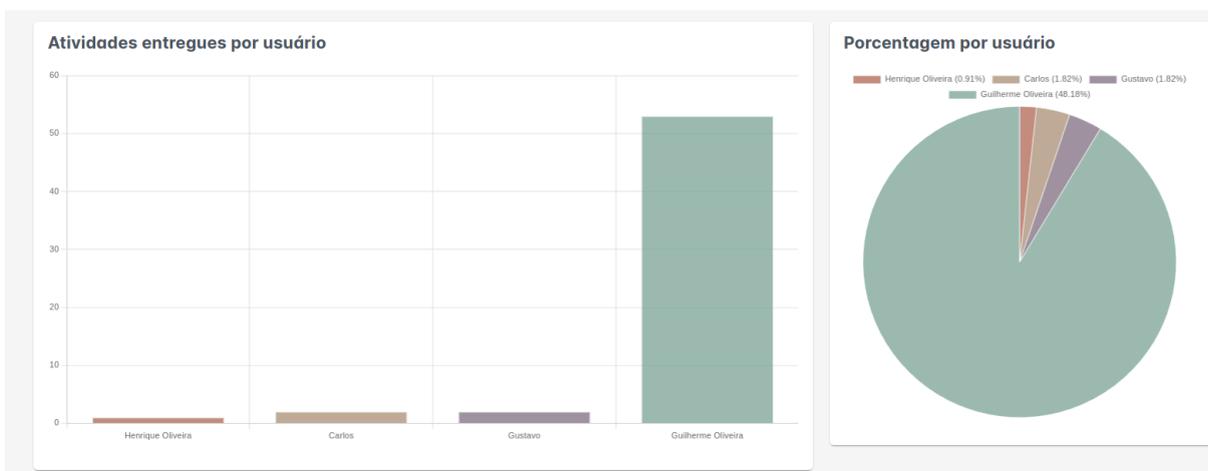
O dashboard geral é um relatório exibido logo ao acessar a plataforma, nele é disponibilizada uma visão geral do sistema, trazendo informações da quantidade de times, quantidade de usuários, quantidade de atividades por time e atividades entregues por usuário.

Figura 12 - Dashboard geral parte 1



Fonte: Própria (2024)

Figura 13 - Dashboard geral parte 2



Fonte: própria (2024)

Essas informações proporcionam uma visão abrangente que ultrapassa o contexto de um time específico, permitindo visualizar dados sobre todo o sistema. Isso possibilita uma análise mais ampla e estratégica das atividades, equipes e desempenho global dentro do Genagile.

6.2.1.2 Área de trabalho do usuário

Visando centralizar as atividades dos usuários em diferentes times e resolver o problema P2, foi desenvolvida a funcionalidade *workspace*. Essa função foi criada para reunir as atividades do usuário em todos os times que o mesmo está vinculado, em um único lugar, facilitando sua visualização sem a necessidade de acessar individualmente cada time. Essa abordagem promove uma maior eficiência ao permitir que o usuário gerencie suas tarefas de forma mais integrada e organizada.

Figura 14 - Área de trabalho do usuário

The screenshot displays a user workspace interface. At the top left, it says "Workspace" and "Área de trabalho pessoal". The date and time are "Sábado, 13 de abril" and "Bom dia, Guilherme Oliveira!".

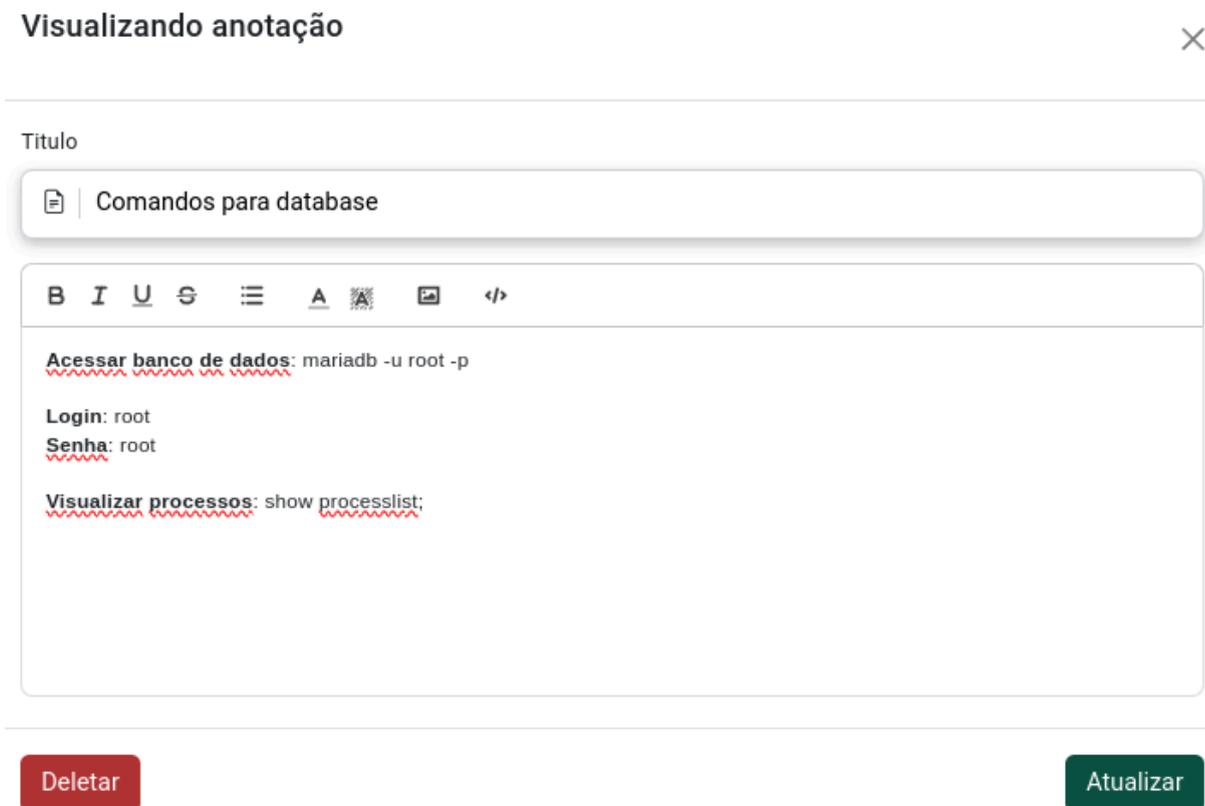
The main content is divided into two sections:

- Bloco de anotações**: A section for personal notes. It includes a sub-header "As anotações são visíveis apenas para o seu usuário". Below this is a table with columns "TÍTULO" and "ÚLTIMA ALTERAÇÃO". One note is visible: "Comandos para database" with a timestamp of "há cerca de 21 horas". A green "+" button is in the top right corner of the notes block. At the bottom right, it says "Items per page: 5" and "1 - 1 of 1".
- Atividades abertas**: A section for open activities, with the sub-header "Suas atividades em progresso separado por time". A tab labeled "Genagile" is selected. Below the tab is a table with columns: "ID", "TÍTULO", "SEÇÃO", "SPRINT", and "PONTUAÇÃO". There are three rows of activities, each with a checkmark icon on the right. The activities are:
 - ID: 4, TÍTULO: Criar demais relatórios, SEÇÃO: Fazendo, PONTUAÇÃO: 5
 - ID: 44, TÍTULO: FEATURE: Criar uma tela específica da atividade, SEÇÃO: Abertas, PONTUAÇÃO: 13
 - ID: 50, TÍTULO: FEATURE: Implementar deleções por atributo "deleted", SEÇÃO: Abertas, PONTUAÇÃO: 1At the bottom right of the table, it says "Items per page: 5" and "1 - 3 of 3".

Fonte: própria (2024)

Além de gerenciar suas atividades, os usuários têm a opção de criar anotações pessoais que são visíveis exclusivamente para eles. Essa funcionalidade permite que os usuários mantenham notas, lembretes ou informações privadas relacionadas ao seu trabalho ou projetos, oferecendo uma função adicional para organização e registro de informações importantes de forma individual e confidencial.

Figura 15 - Visualização de anotação do usuário



The screenshot displays a web interface for viewing a user annotation. At the top, the title "Visualizando anotação" is shown with a close button (X) on the right. Below the title, the annotation's title is "Comandos para database". The main content area contains a rich text editor with a toolbar (Bold, Italic, Underline, Link, List, Text Color, Background Color, Image, Code) and the following text: "Acessar banco de dados: mariadb -u root -p", "Login: root", "Senha: root", and "Visualizar processos: show processlist;". At the bottom, there are two buttons: "Deletar" (red) and "Atualizar" (green).

Fonte: própria (2024)

6.2.2 Funcionalidades interna ao time

As funcionalidades internas ao time estão centradas no gerenciamento de atividades e indivíduos dentro do mesmo. Neste tópico, vamos detalhar como ocorre esse gerenciamento de atividades e como as ferramentas desenvolvidas auxiliam na solução dos problemas mencionados no tópico 4.1.1.

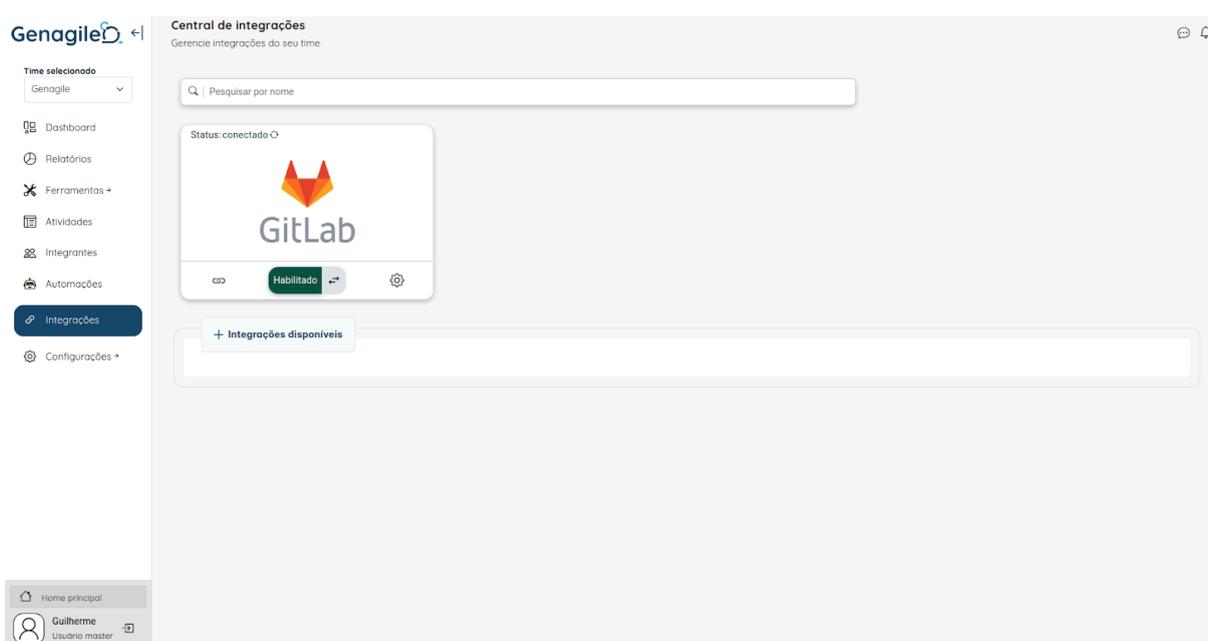
6.2.2.1 Integrações

Para solução do desafio de ter múltiplos sistemas para gerenciamento do projeto, como relatado no problema P1, foi vista a necessidade inicial de centralizar as atividades no genagile através de integrações com as demais plataformas.

Para isso, uma integração pode ser associada a um único time, que, por sua vez, pode ter integração de diferentes plataformas configuradas. O papel fundamental dessa integração no projeto é automatizar alterações em atividades, eliminando a necessidade do usuário replicar manualmente essas mudanças em todos os locais nos quais o gerenciamento do projeto é realizado.

Essa funcionalidade simplifica e agiliza o processo de atualização, garantindo consistência nas informações em diferentes plataformas ou sistemas utilizados na gestão do projeto. Ao centralizar as alterações por meio da integração, reduz-se a possibilidade de erros humanos, economizando tempo e assegurando uma visão coesa e atualizada do projeto em todas as plataformas envolvidas.

Figura 16 - Central de integrações



Fonte: própria (2024)

Para isso foi desenvolvido uma central de integração, entretanto, no projeto foi desenvolvida uma única integração até o momento, e ela vincula a plataforma Gitlab. Nesse cenário, o usuário preenche todos os detalhes do seu projeto, e a partir desse momento, algumas funcionalidades tornam-se disponíveis para utilização. Estas incluem:

- Importar e sincronizar alterações nas atividades.
- Enviar e sincronizar alterações nas atividades.

Figura 17 - Configuração de integração com Gitlab

Atualizar integração ×

Atualize a integração e defina as sincronizações



— Configurações de conexão

URL de acesso

Id do projeto Token Versão da API

— Configurações de sincronização

Importar e sincronizar atividades

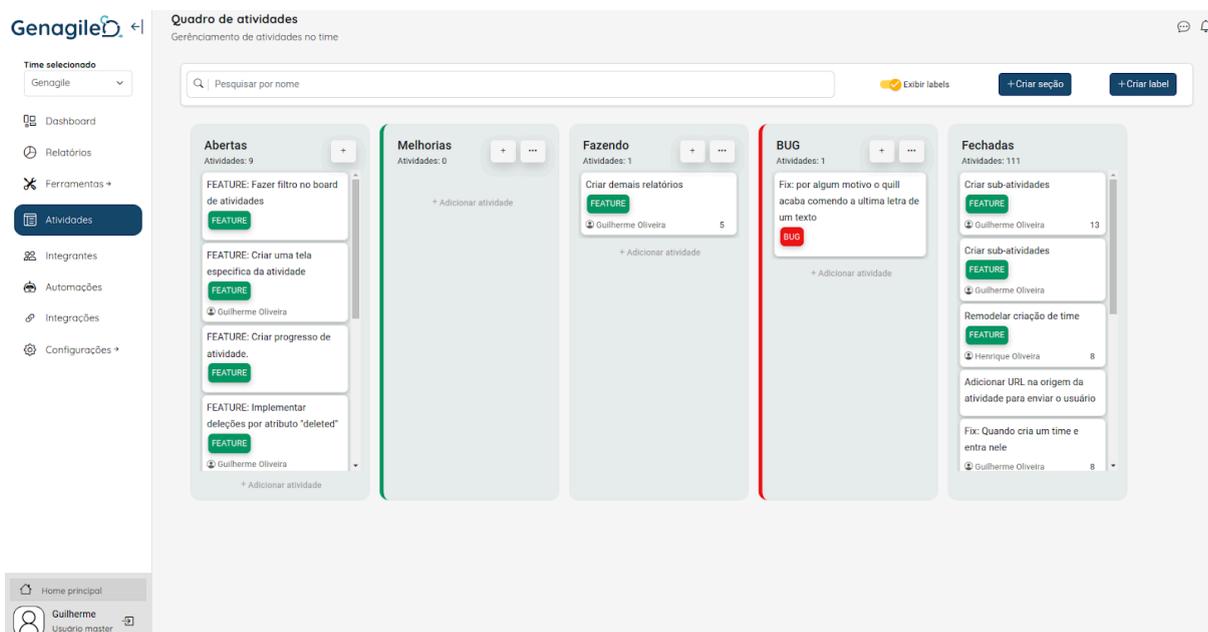
Enviar e sincronizar atividades

Fonte: própria (2024)

6.2.2.2 Quadro de atividades

Também com o objetivo de tratar o problema P1, para gerenciar as atividades, foi adotado um modelo semelhante ao quadro de atividades utilizado em plataformas consolidadas como o Trello. Esse quadro centraliza todas as atividades criadas no Genagile, bem como em outras plataformas, proporcionando uma visão integrada e abrangente das tarefas em andamento. Isso proporciona uma visão unificada e completa das tarefas em andamento, facilitando o acompanhamento e a gestão do trabalho em equipe.

Figura 18 - Quadro de atividades



Fonte: própria (2024)

Dentro do quadro de atividades, temos a capacidade de gerenciar as tarefas de forma abrangente, incluindo a criação e edição das mesmas. Podemos criar novas colunas para organizar as atividades de acordo com as etapas do processo e também criar labels que serão adicionadas às atividades para uma melhor categorização e identificação. Além disso, o quadro de atividades torna fácil acessar informações relevantes, tais como:

- Título da atividade
- Labels atribuídas
- Usuário responsável pela atividade
- Voto, sendo a estimativa de esforço necessária para concluir a atividade

Figura 19 - Visualização simplificada de uma atividade



Fonte: própria (2024)

Além das informações principais mencionadas, ao clicar em uma atividade específica, o sistema exibirá uma visualização detalhada da mesma, oferecendo ao usuário a capacidade de editar e acrescentar itens que não são exibidos na visualização simplificada. Esses itens adicionais incluem:

- Plataforma de origem da atividade.
- Descrição detalhada.
- Comentários relacionados, promovendo a comunicação entre os membros da equipe.
- Lista de sub-atividades associadas.
- Sprint em que a tarefa está inserida, proporcionando contexto e rastreamento do progresso.
- Tempo já gasto, oferecendo insights sobre o andamento e esforço dedicado.
- Integrações em que a atividade está presente.

Figura 20 - Visualização detalhada de uma atividade

Fonte: própria (2024)

E para solução do problema P5, os usuários têm a opção de cronometrar o tempo gasto em suas atividades diretamente pela plataforma. Isso é feito ao clicar no ícone de relógio dentro da visualização detalhada da atividade. O cronômetro é então iniciado, podendo ser interrompido e reiniciado do momento em que parou e, ao concluir a atividade, o usuário pode pará-lo, registrando automaticamente o tempo gasto na atividade.

Essa funcionalidade proporciona um registro preciso e conveniente do tempo dedicado a cada tarefa, fazendo com que o usuário não precise realizar esse processo utilizando ferramentas fora da plataforma.

Figura 21 - Cronometrando atividade

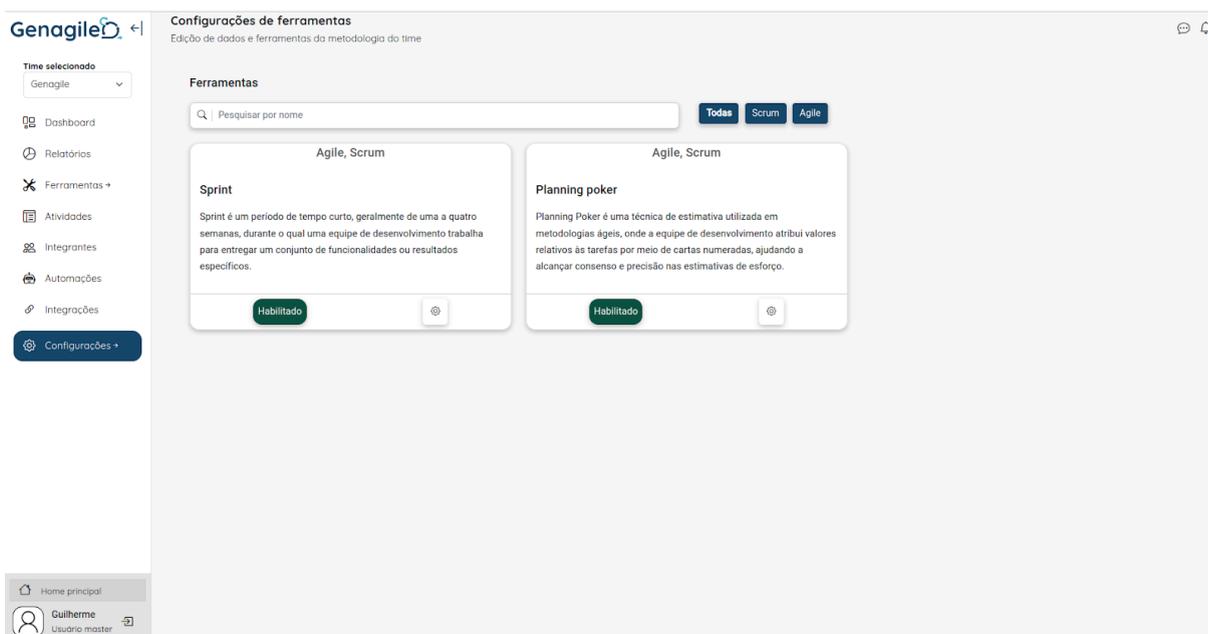
Fonte: própria (2024)

6.2.2.3 Ferramentas ágeis

Com o intuito de simplificar o gerenciamento do projeto e diminuir o uso de demais plataformas, algumas funcionalidades que eram desenvolvidas em outros sistemas foram incorporadas ao Genagile como ferramentas. Inicialmente todas estão desabilitadas, com o objetivo de tornar o sistema mais simples possível no primeiro uso, e a medida que vão sendo habilitadas, mais funções são agregadas.

O projeto conta com duas ferramentas, que são utilizadas ativamente pela empresa, sendo elas: *Sprint* e *planning poker*.

Figura 22 - Ferramentas



Fonte: própria (2024)

6.2.2.3.1 Planning poker

Com o objetivo de solucionar os problemas descritos no tópico 4.1.1.6, viu-se a necessidade do planning poker ser uma funcionalidade interna na aplicação, com o objetivo de diminuir o uso de demais plataformas utilizadas apenas com esse

propósito. Nesse contexto, a funcionalidade foi concebida com o objetivo de otimizar e agilizar significativamente esse processo.

Inicialmente é necessário realizar a configuração dessa ferramenta, onde o usuário master conseguirá informar qual o modelo que o mesmo deseja utilizar.

Figura 23 - Configuração de planning poker



Planning poker

Habilitar ferramenta

Tipo de votação

Fibonacci [0,1,2,3,5,8,13,21]

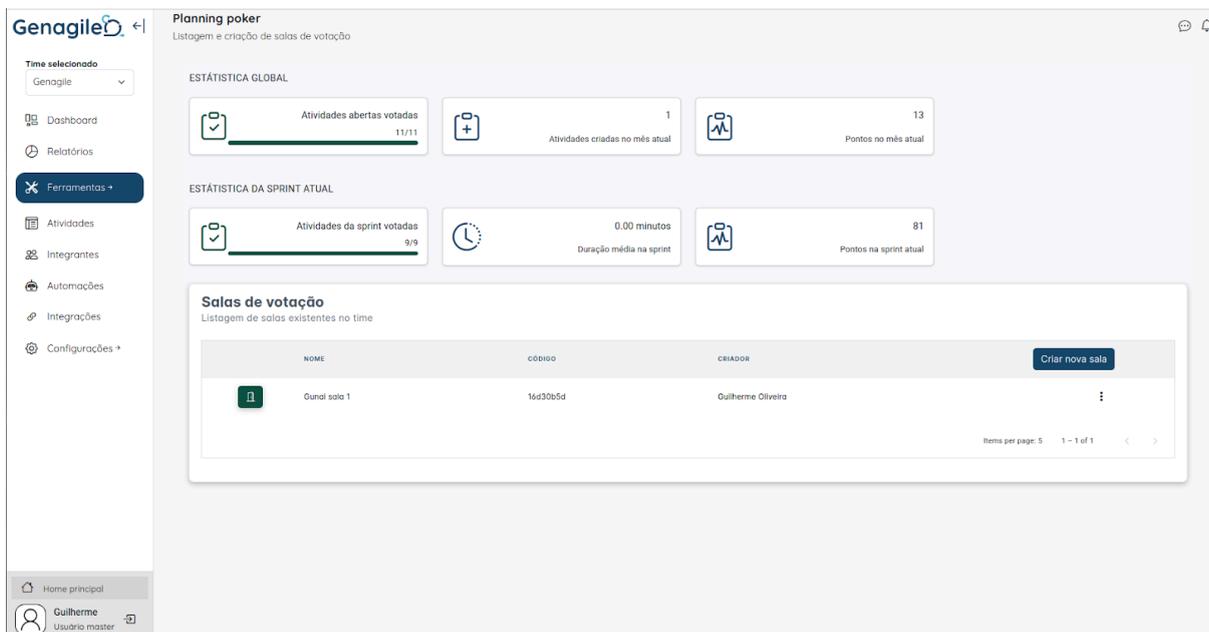
Cancelar Salvar

Fonte: própria (2024)

6.2.2.3.1.1 Informações de votações

Após a realização da configuração, uma nova tela é disponibilizada para visualização no time na qual foi configurada essa ferramenta. Nela é possível acompanhar informações sobre as votações realizadas, se existem atividades em aberto que ainda não foram votadas e criar salas para a realização do planning poker, tratando assim o problema P8.

Figura 24 - Informações do planning poker

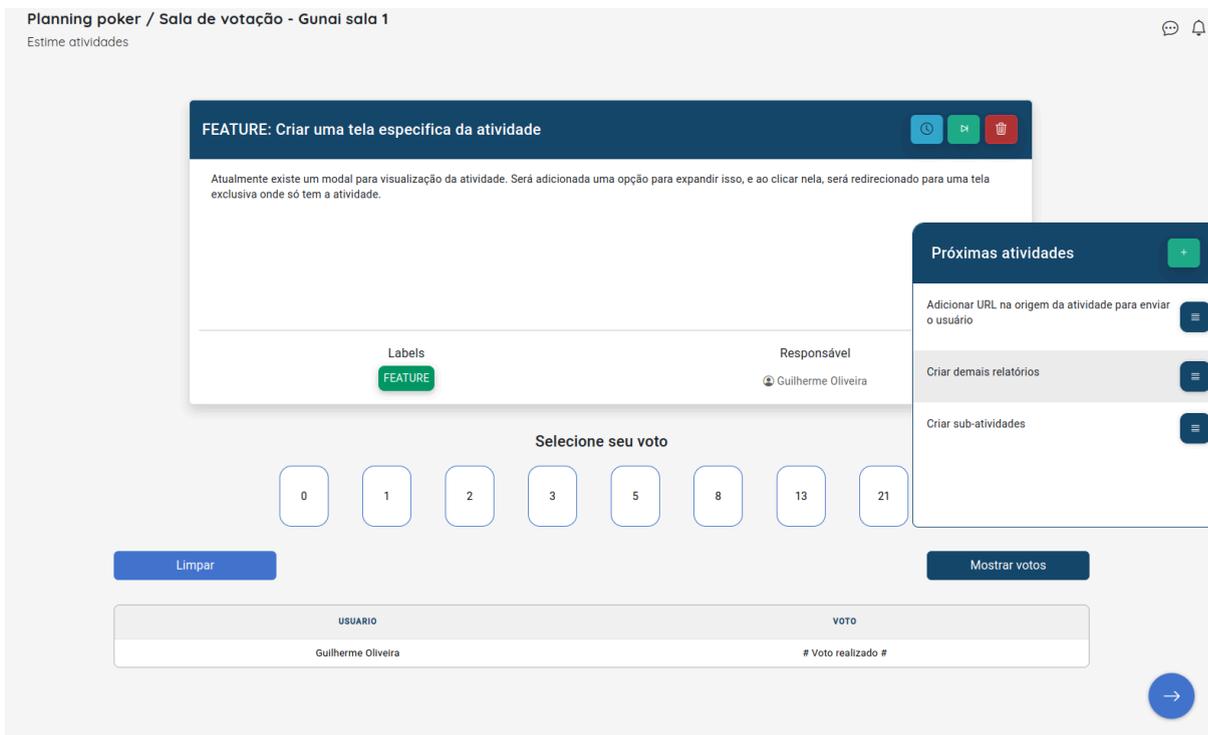


Fonte: própria (2024)

6.2.2.3.1.2 Votação com permissionamento e cronometragem

Com o objetivo de tratar o problema P6, ao acessar uma sala, é possível realizar a votação de uma atividade e durante esse processo, os usuários possuem diferentes níveis de permissões que determinam seu acesso às funcionalidades. Isso implica em uma apresentação diferenciada da tela, dependendo do perfil do usuário. Na figura abaixo, é ilustrada a perspectiva do usuário *master*:

Figura 25 - Planning poker visão de usuário master

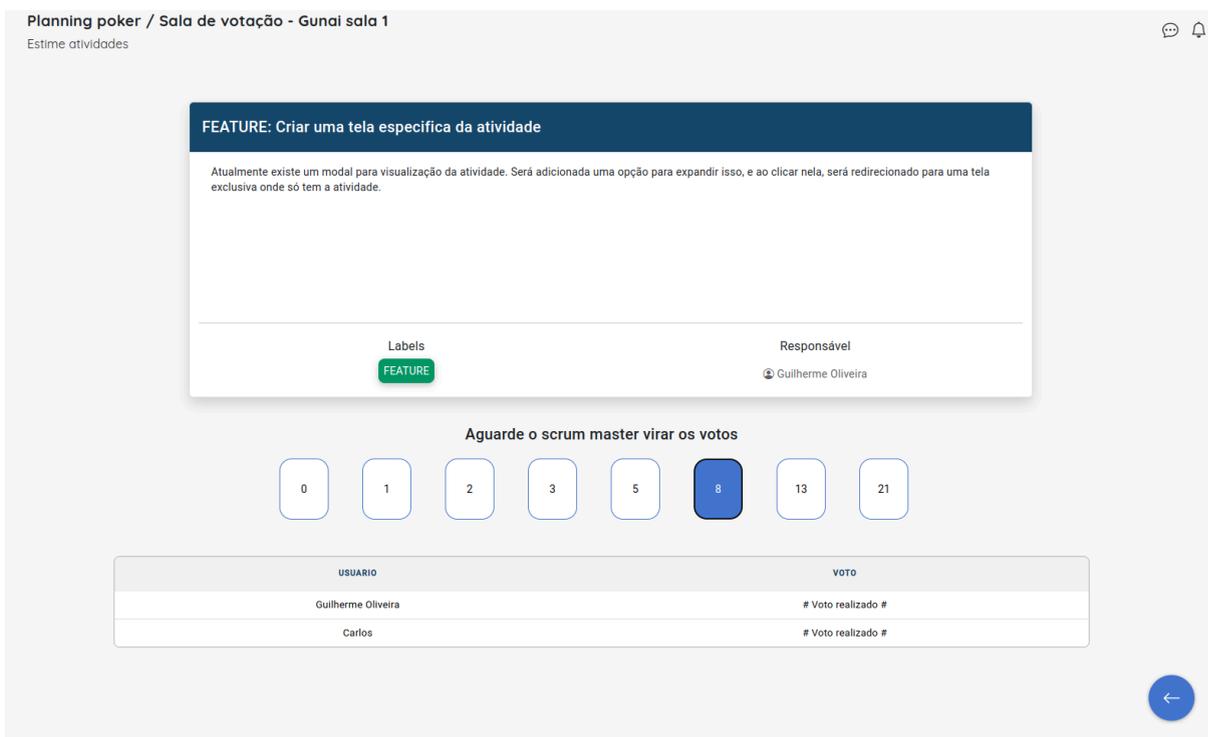


Fonte: própria (2024)

Assim, o usuário *master* é capaz de conduzir de maneira eficiente todo o gerenciamento das votações das atividades, utilizando as funcionalidades de: botão para visualização dos votos, definição das próximas atividades a serem votadas, exclusão de uma atividade da votação, pular uma atividade a ser votada e realização da cronometragem da duração da votação para obtenção de métricas para solucionar o problema P9.

Por outro lado, o usuário sem permissão, não dispõe dessas funcionalidades e capacidades de gestão, limitando-se à simples participação na votação da atividade. Na figura abaixo, é apresentada a perspectiva do usuário sem permissão:

Figura 26 - Planning poker visão de usuário sem permissão

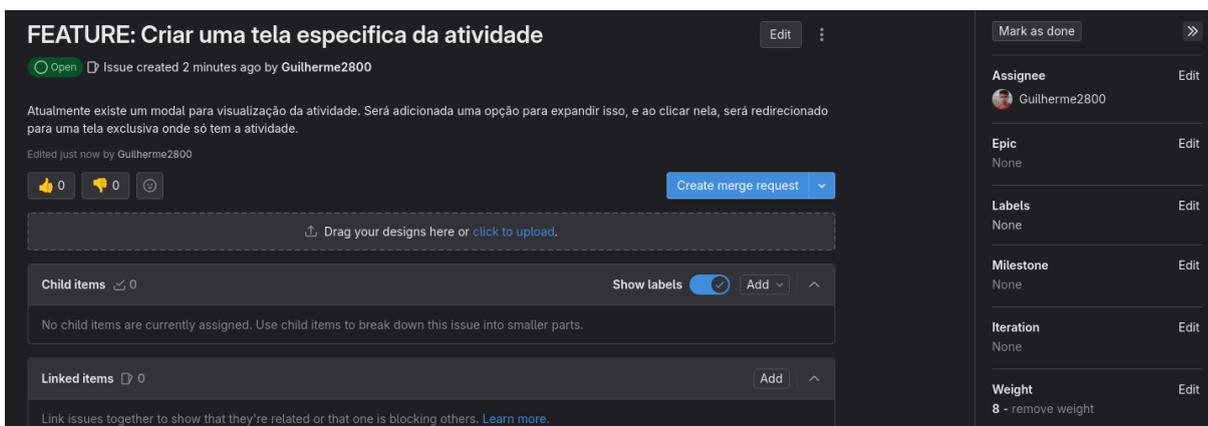


Fonte: própria (2024)

6.2.2.3.1.3 Integração com Gitlab

Se o time tiver integrações configuradas, ao realizar a votação de uma atividade e definir seu voto, esse valor será automaticamente atribuído à atividade em todas as integrações existentes, sanando assim o problema P7. Na figura abaixo, ilustrando uma integração com a plataforma Gitlab, após a conclusão da votação, o voto é adicionado automaticamente no atributo “*Weight*”. Isso elimina possíveis falhas humanas ao evitar a necessidade de replicar manualmente a alteração em várias outras plataformas.

Figura 27 - Alteração de voto no Gitlab



Fonte: própria (2024)

6.2.2.3.2 Sprint

Para abordar as questões identificadas nos problemas P3 e P4, foi reconhecida a necessidade de incorporar uma funcionalidade específica ao sistema para gerenciar os ciclos de atividades no contexto do Scrum, conhecidos como *Sprints*. Esta nova funcionalidade é concebida como uma ferramenta no Genagile, que pode ser habilitada pelo usuário master dentro das configurações do time.

O processo de habilitação dessa ferramenta inicia-se com o usuário *master* acessando as configurações do time, onde terá a opção de definir se todas as *Sprints* terão um *Scrum Master* fixo, ao ativar a opção "*Scrum Master Fixo*", ou se a responsabilidade será rotativa, designando um novo indivíduo a cada ciclo. Essa flexibilidade permite uma adaptação precisa às necessidades e dinâmicas de cada equipe.

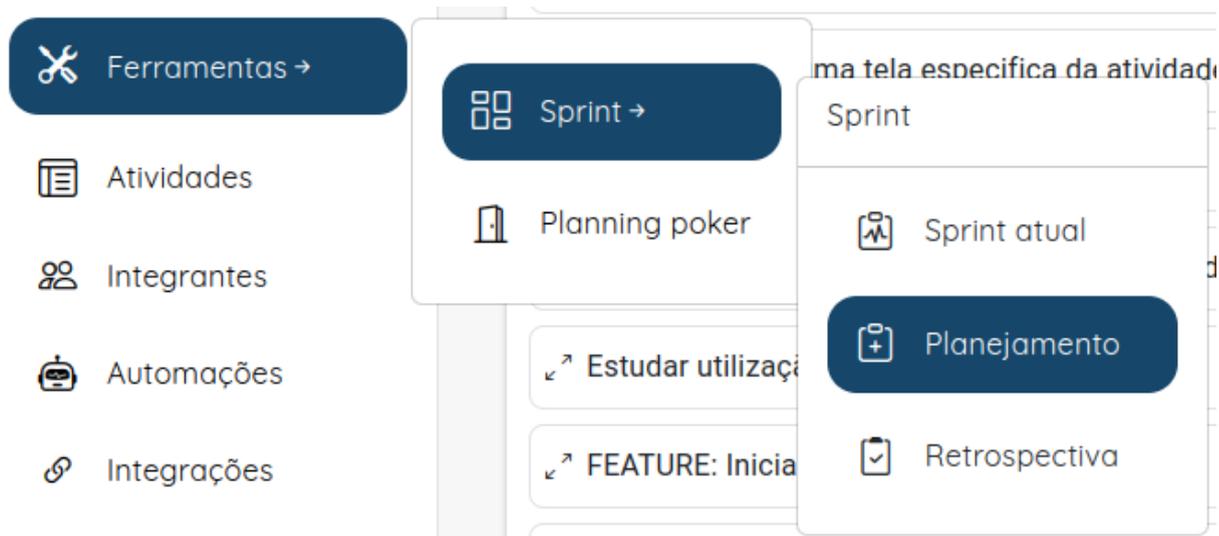
Figura 28 - Configuração de ferramenta Sprint



Fonte: própria (2024)

Ao ativar essa funcionalidade, uma nova seção é adicionada ao menu de ferramentas do time, composta pelas seguintes opções: *Sprint* atual, Planejamento e Retrospectiva. Essas opções são essenciais para o gerenciamento eficaz das atividades durante o ciclo de *Sprint* e para a revisão e melhoria contínua do processo de trabalho da equipe.

Figura 29 - Menu de ferramenta Sprint

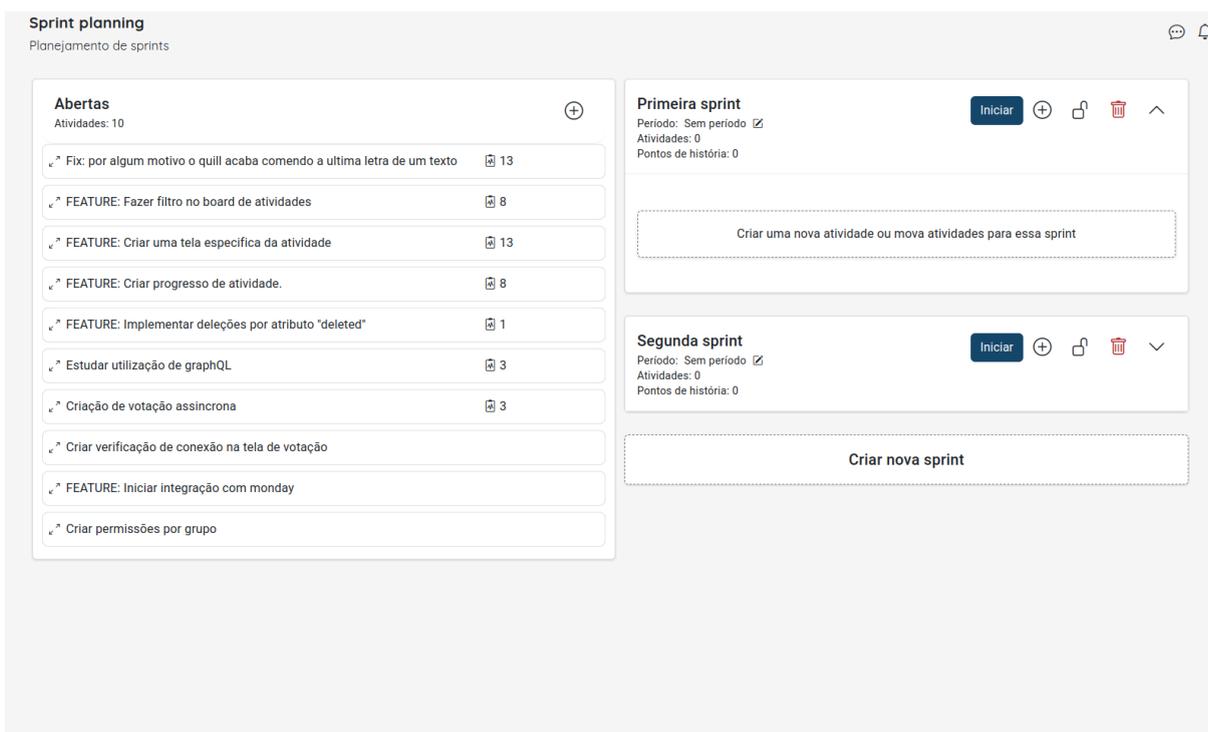


Fonte: própria (2024)

6.2.2.3.2.1 Planejamento da Sprint

A tela de planejamento da *Sprint* é um recurso essencial para o planejamento eficiente das próximas *Sprints* e busca solucionar o problema P3. Ao acessá-la, é exibida uma interface contendo todas as atividades que ainda não foram finalizadas. Onde disponibiliza a capacidade de visualizar o título das atividades e seu voto, além de poder clicar em uma atividade específica para ver todos os detalhes relacionados a ela. Além disso, trás a funcionalidade de criar novas *Sprints* diretamente a partir dessa interface, o que facilita o processo de planejamento e organização das atividades para os próximos ciclos de trabalho.

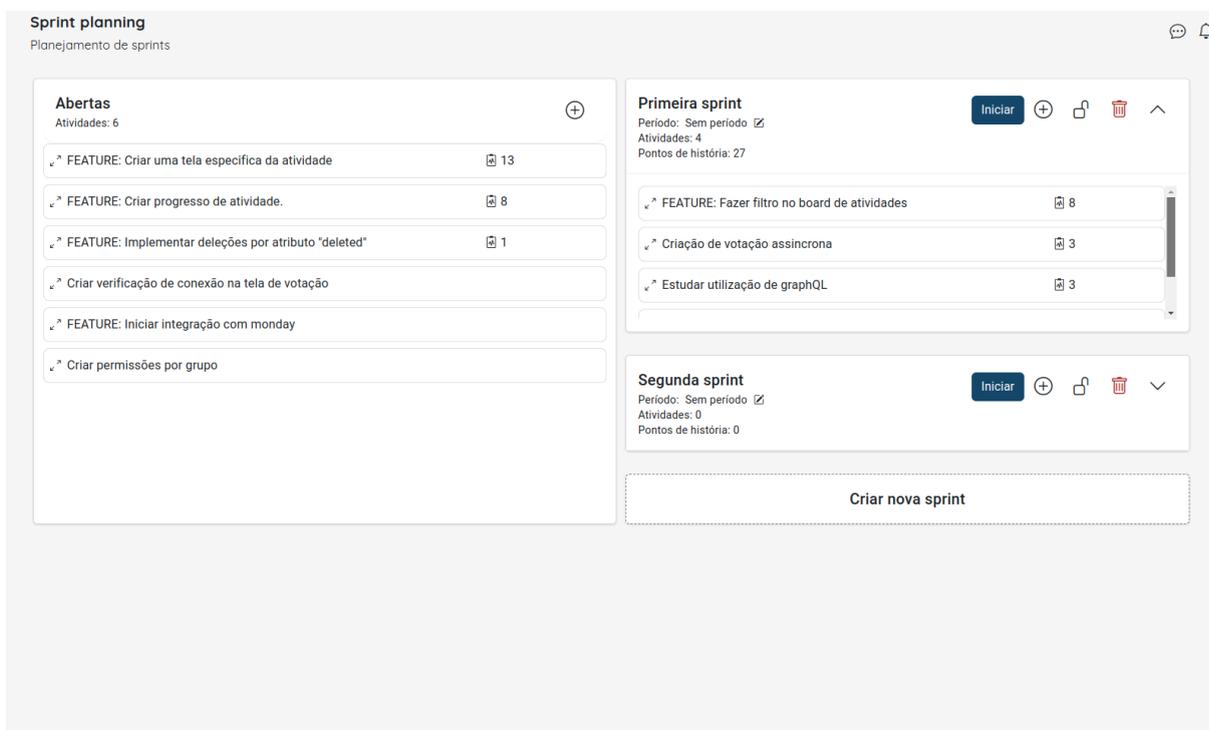
Figura 30 - Planejamento da Sprint



Fonte: própria (2024)

Para adicionar atividades a uma *Sprint* em processo de planejamento, basta mover a tarefa entre os cards ou clicar no botão de "+" dentro do card da *Sprint* desejada para criar uma nova atividade. Após essa ação, o sistema atualiza automaticamente a quantidade de atividades e a pontuação da *Sprint*.

Figura 31 - Adição de atividades em Sprint planejada

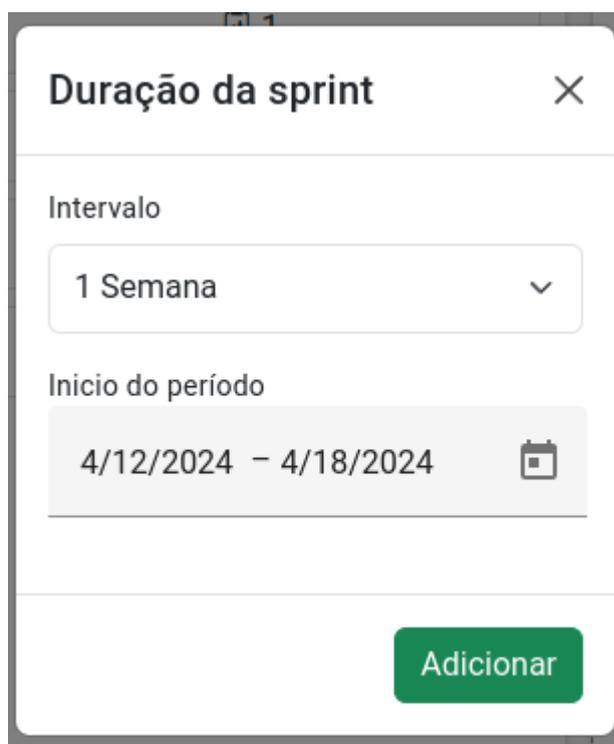


Fonte: própria (2024)

Essa abordagem de planejamento proporciona uma visão clara da capacidade de trabalho da equipe ao longo do ciclo de atividades, permitindo o acompanhamento preciso das tarefas planejadas para a *Sprint* e identificando claramente quais atividades estão fora do escopo das próximas *Sprints*.

Além disso, como uma *Sprint* representa um ciclo de atividades com um período definido, é fundamental configurar sua duração. Para isso é disponibilizada a flexibilidade de escolher entre diferentes períodos pré-definidos, como 1 semana, 2 semanas, 3 semanas, 1 mês, ou até mesmo configurar um período personalizado de acordo com as necessidades específicas do projeto e da equipe. Essa variedade de opções de duração permite uma adaptação precisa às dinâmicas de trabalho de cada equipe.

Figura 32 - Configuração de período da Sprint



The image shows a mobile application dialog box titled "Duração da sprint" with a close button (X) in the top right corner. Below the title, there are two sections: "Intervalo" and "Inicio do período". The "Intervalo" section has a dropdown menu currently set to "1 Semana". The "Inicio do período" section shows a date range "4/12/2024 - 4/18/2024" with a calendar icon to its right. At the bottom right of the dialog is a green button labeled "Adicionar".

Fonte: própria (2024)

Após adicionar as atividades a uma *Sprint* em planejamento, é possível bloquear a adição de novos itens ao pressionar o botão com o ícone de cadeado. Esse recurso permite impedir que novas adições sejam feitas, evitando assim que a capacidade de entrega do time seja ultrapassada durante o ciclo de atividades. Essa funcionalidade é essencial para garantir que a equipe se comprometa apenas com o trabalho que pode ser realizado dentro do período da *Sprint*, contribuindo para uma gestão eficiente do tempo e dos recursos disponíveis.

Figura 33 - Adição de atividade com Sprint bloqueada



Fonte: própria (2024)

Ao concluir o planejamento da *Sprint*, o usuário tem a opção de iniciar uma nova *Sprint*, desde que não haja uma em andamento. Se o time estiver configurado para não ter um *Scrum Master* fixo, ao iniciar uma nova *Sprint* será exibido um modal com as permissões que o novo *Scrum Master* irá obter, juntamente com um campo de seleção para escolher o usuário que assumirá esse papel. Isso oferece uma flexibilidade adicional ao permitir que diferentes membros da equipe assumam a responsabilidade de *Scrum Master* em cada ciclo de *Sprint*, de acordo com as necessidades e dinâmicas do time.

Figura 34 - Configuração de scrum master da Sprint

Scrum master ✕

Selecione o scrum master dessa sprint

Guilherme Oliveira ▾

Permissões:

Sprint

- Remover usuário atribuído à atividade
- Finalizar sprint

Planning poker

- Gerenciar votação

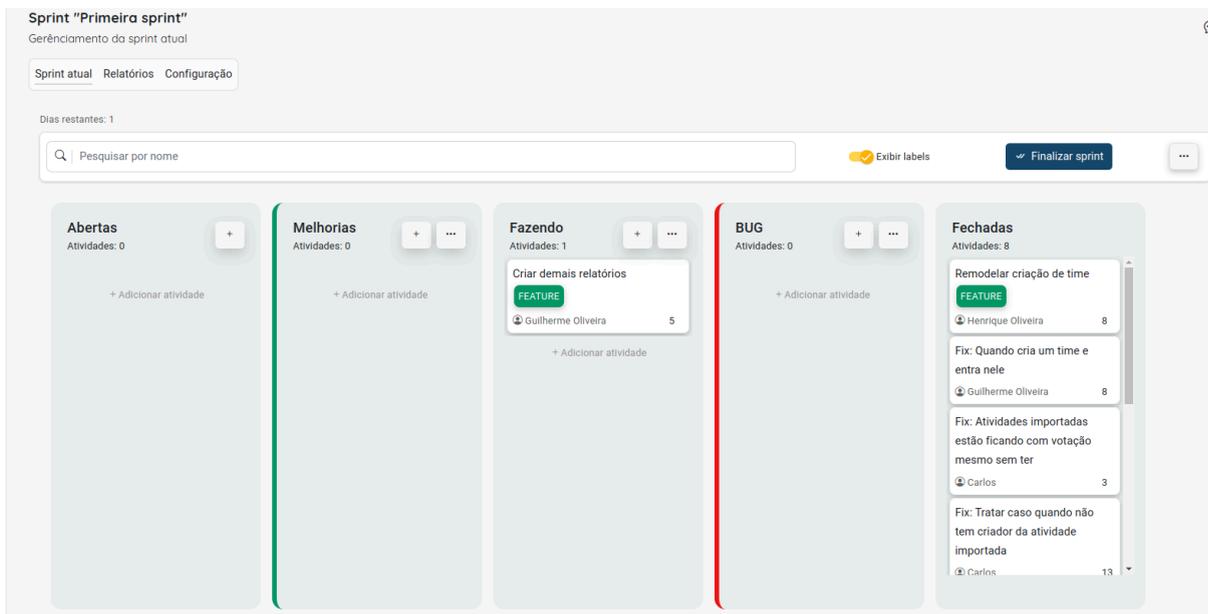
Cancelar Iniciar sprint

Fonte: própria (2024)

6.2.2.3.2.2 Sprint atual

Após iniciar uma *Sprint*, o gerenciamento das atividades é feito por meio da tela "*Sprint Atual*". Nessa tela, é apresentado inicialmente um quadro de atividades contendo apenas os itens que foram adicionados àquele ciclo de atividades específico. Essa abordagem facilita a visualização e o acompanhamento das atividades em desenvolvimento durante a *Sprint*, permitindo que a equipe se concentre nos itens que estão em progresso e forneçam uma visão clara do andamento do trabalho planejado para o ciclo em curso.

Figura 35 - Quadro de atividades da Sprint



Fonte: própria (2024)

Além do quadro de atividades, essa tela fornece funcionalidade de Relatórios da *Sprint* Atual proporciona um acompanhamento detalhado do progresso da *Sprint*, apresentando informações essenciais para a análise e tomada de decisões. Os relatórios disponíveis incluem:

- Histórico de atividades entregues.
- Quantidade e porcentagem de itens concluídos.
- Duração total da Sprint e dias restantes.
- Duração média das tarefas.
- Distribuição de atividades por seção.
- Distribuição de atividades por membros da equipe.

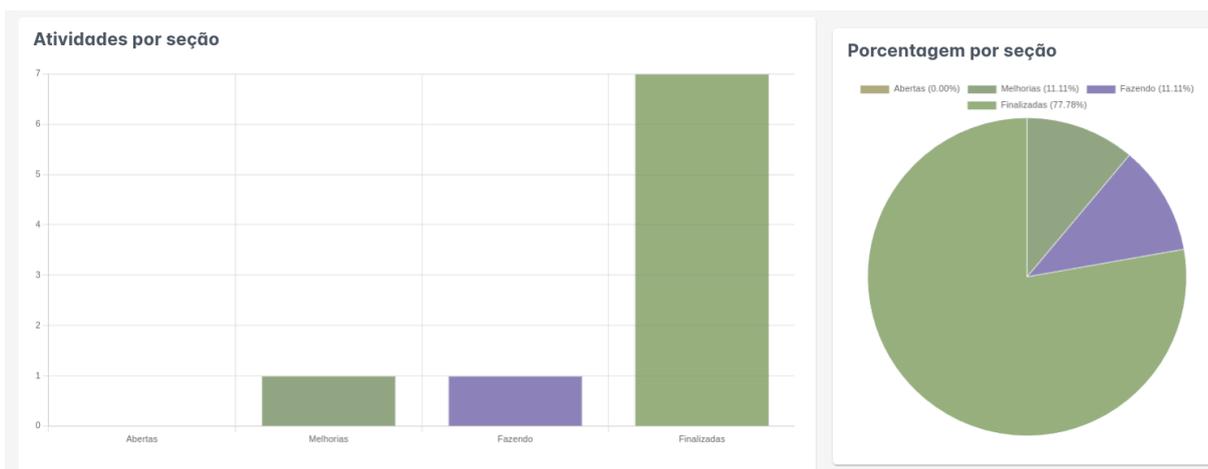
Esses relatórios podem ser gerados com base em diferentes critérios, como o número de atividades ou os votos atribuídos às atividades. No exemplo abaixo, a geração dos relatórios foi realizada com base na quantidade de itens concluídos:

Figura 36 - Relatórios da Sprint atual por quantidade parte 1



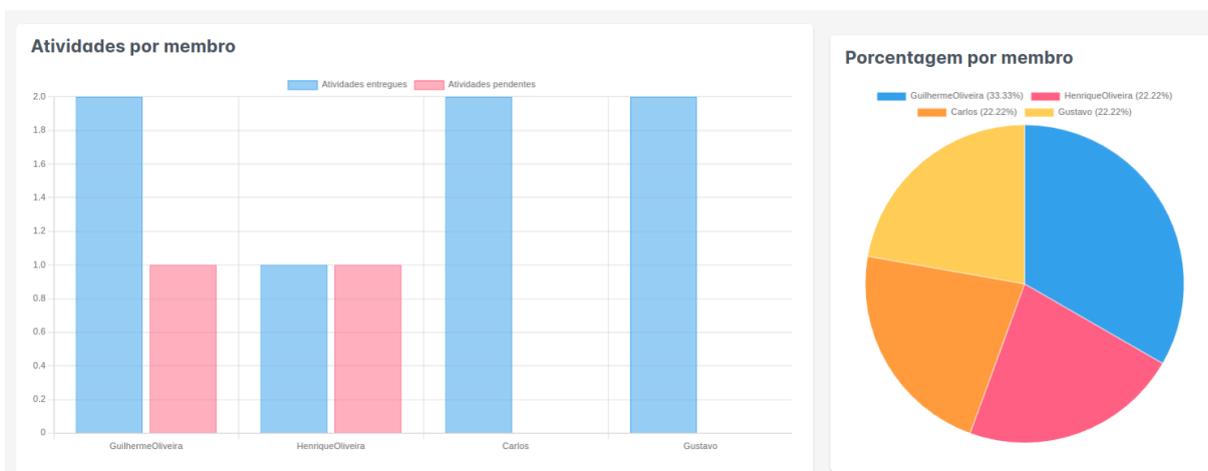
Fonte: própria (2024)

Figura 37 - Relatórios da Sprint atual por quantidade parte 2



Fonte: própria (2024)

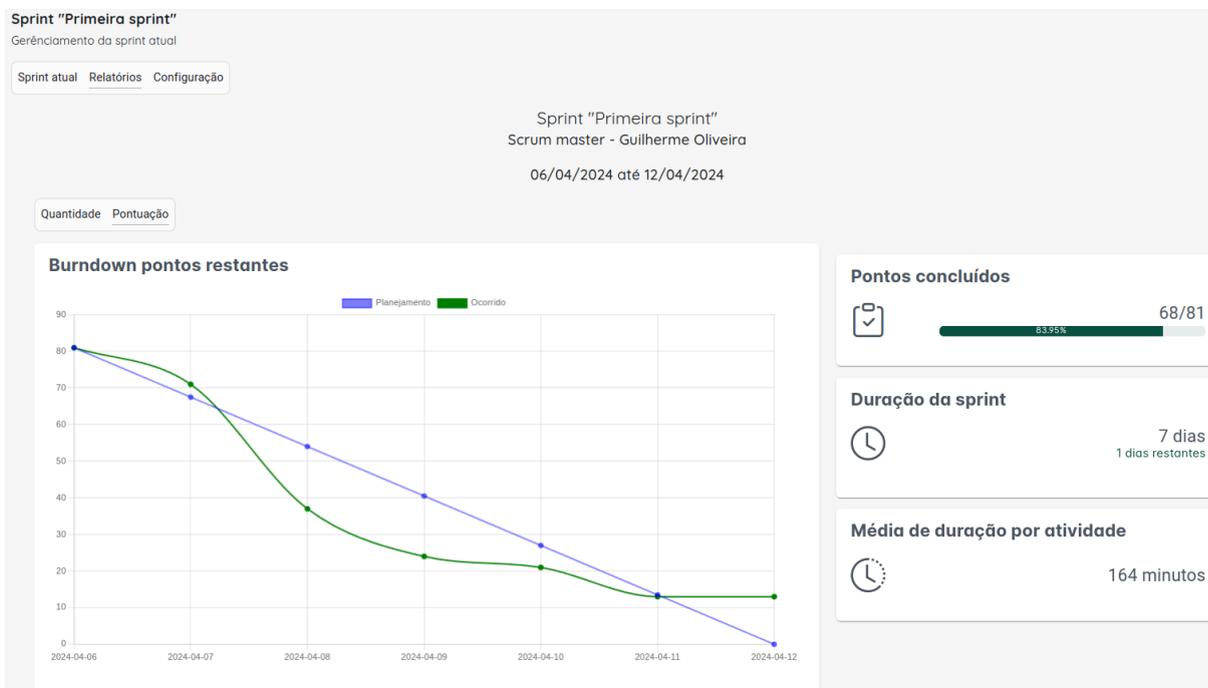
Figura 38 - Relatórios da Sprint atual por quantidade parte 3



Fonte: própria (2024)

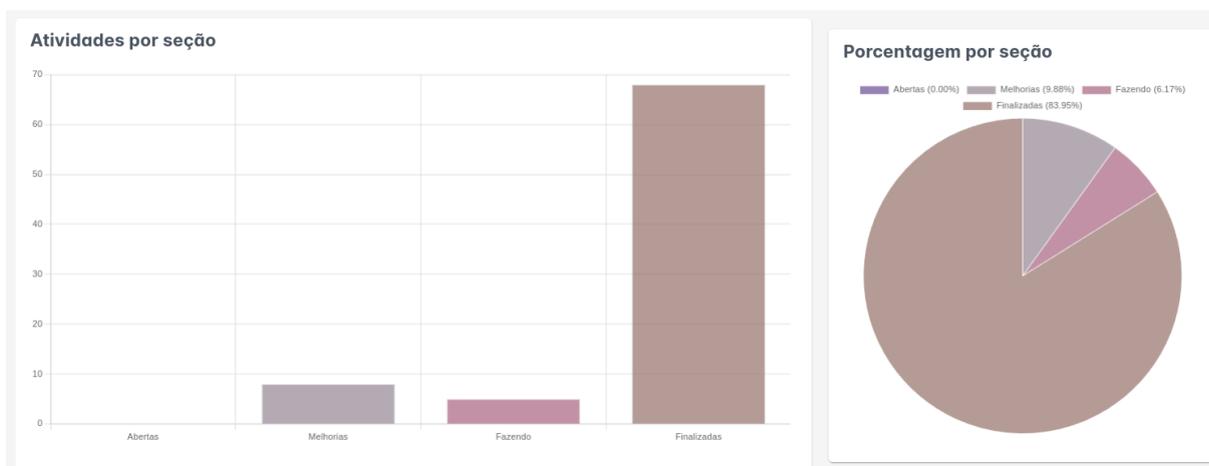
Já nos casos abaixo, o relatório foi gerado utilizando os votos atribuídos às atividades, para a geração das métricas:

Figura 39 - Relatórios da Sprint atual por voto parte 1



Fonte: própria (2024)

Figura 40 - Relatórios da Sprint atual por voto parte 2



Fonte: própria (2024)

Figura 41 - Relatórios da Sprint atual por voto parte 3



Fonte: própria (2024)

Essa diferenciação oferece uma perspectiva mais completa do progresso da equipe. Ao considerar os votos, estamos avaliando a percepção da dificuldade de cada atividade. Isso nos permite compreender a capacidade de entrega do time em relação à complexidade das tarefas, gerando insights valiosos para ajustes e melhorias. Como resultado, os gráficos gerados refletem não apenas o avanço quantitativo, mas também a eficiência e eficácia da equipe ao enfrentar desafios variados, oferecendo uma visão mais detalhada e estratégica do progresso da Sprint.

Além disso, a tela fornece a funcionalidade de configuração da *Sprint* atual, trazendo opções para ajustar de acordo com as necessidades da equipe. Isso inclui a capacidade de:

- Alterar a data final da *Sprint*.
- Bloquear ou desbloquear a adição de novas atividades durante a *Sprint*.
- Renomear a *Sprint*.
- Designar ou alterar o usuário responsável pelo papel de *Scrum Master*.

Essas opções proporcionam flexibilidade e controle para adaptar a *Sprint* conforme as circunstâncias evoluem durante o ciclo de trabalho, garantindo uma gestão eficaz e alinhada com as metas e objetivos do projeto.

Figura 42 - Configuração da Sprint atual

Nome da sprint: Nome default - b13b6007

Backlog: Desbloqueado

Scrum master: Guilherme Oliveira

Data fim da sprint: 4/12/2024

Dias restantes: 0 dias

Salvar

Fonte: própria (2024)

Ao retornar ao quadro de atividades e finalizar a *Sprint*, será aberto no novo modal, onde será possível informar se deseja realizar a retrospectiva e para onde deseja mover as atividades que não foram finalizadas. Essas opções permitem um encerramento organizado e estruturado da *Sprint*, incentivando a reflexão e aprendizado da equipe para futuras iterações, ao mesmo tempo em que garante a continuidade e gestão adequada das atividades pendentes.

Figura 43 - Finalização da Sprint

Finalizar sprint "Primeira Sprint"
✕

✔ Realizar retrospectiva

Atividades restantes:

Mover para backlog

Mover para uma sprint específica

Selecione a sprint

Segunda sprint
▼

Cancelar

Finalizar

Fonte: própria (2024)

6.2.2.3.2.3 Retrospectiva da Sprint

Ao acessar a tela de retrospectiva, é possível realizar uma busca e listar todas as retrospectivas de *Sprints* disponíveis, oferecendo uma visão organizada e acessível das mesmas, permitindo que a equipe revise e analise os resultados e aprendizados de *Sprints* passadas, buscando solucionar o problema P4.

Figura 44 - Listagem de retrospectivas de Sprints

Sprints retrospective
Retrospectiva de sprints concluídas

Nome

Intervalo

Início - Fim 📅

Filtrar

IDENTIFICADOR	NOME DA SPRINT	DATA DE CONCLUSÃO	DURAÇÃO	AÇÕES
e542997b	Nome 1	2024-04-06	0 dias	Acessar relatório
4bd4cdf	Primeira Sprint	2024-04-12	6 dias	📄

Fonte: própria (2024)

Ao abrir o detalhamento de uma retrospectiva, os usuários terão acesso aos mesmos gráficos e informações presentes na seção de "Relatórios" da "Sprint Atual". Isso inclui dados como histórico de atividades entregues, quantidade de itens concluídos, duração da Sprint, entre outros.

Uma característica importante é que mesmo que as atividades não concluídas na *Sprint* sejam passadas para outra, o relatório é capaz de manter as informações corretas sobre os dados da *Sprint* na qual está realizando a retrospectiva, tratando assim o problema identificado no problema P4. Isso garante que as métricas e dados apresentados no relatório reflitam com precisão o desempenho e o progresso da equipe até o encerramento da *Sprint*, independentemente de ajustes feitos posteriormente na lista de atividades. Essa capacidade de manter a integridade dos dados é fundamental para uma análise confiável e consistente do ciclo de trabalho realizado.

Figura 45 - Retrospectiva da Sprint



Fonte: própria (2024)

6.2.2.4 Dashboard do time

O dashboard do time é um relatório apresentado quando o usuário seleciona um time específico. Ele oferece uma visão dos dados para o time selecionado, trazendo informações como:

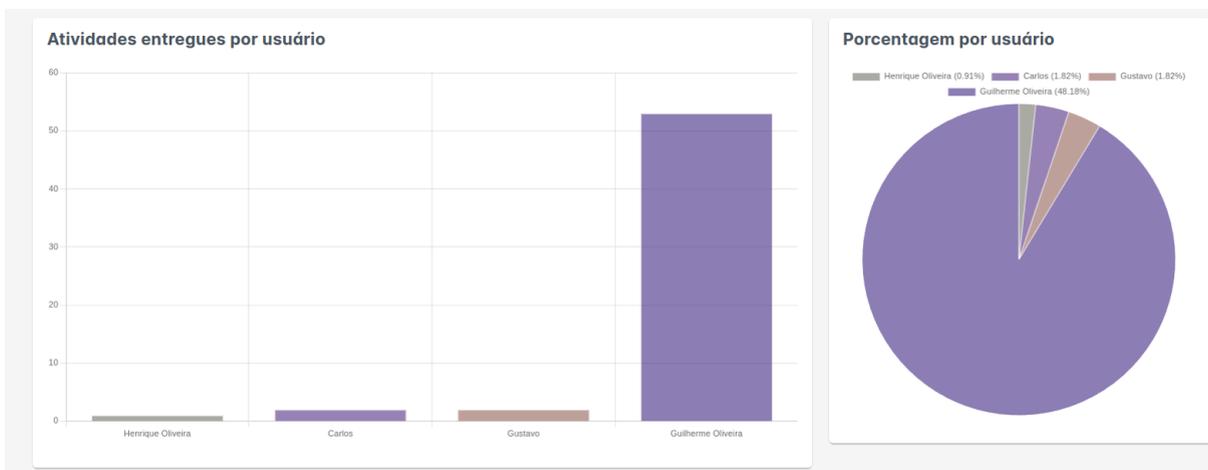
- Quantidade de atividades abertas e finalizadas pelo time.
- Comparação de atividades criadas e concluídas nos últimos 3 meses, permitindo uma análise do desempenho ao longo do tempo.
- Dados sobre atividades entregues por usuário, destacando a produtividade individual dentro do time.
- Histórico de *sprints* anteriores, proporcionando insights sobre o atraso e duração em ciclos de trabalho anteriores.

Figura 46 - Dashboard do time parte 1



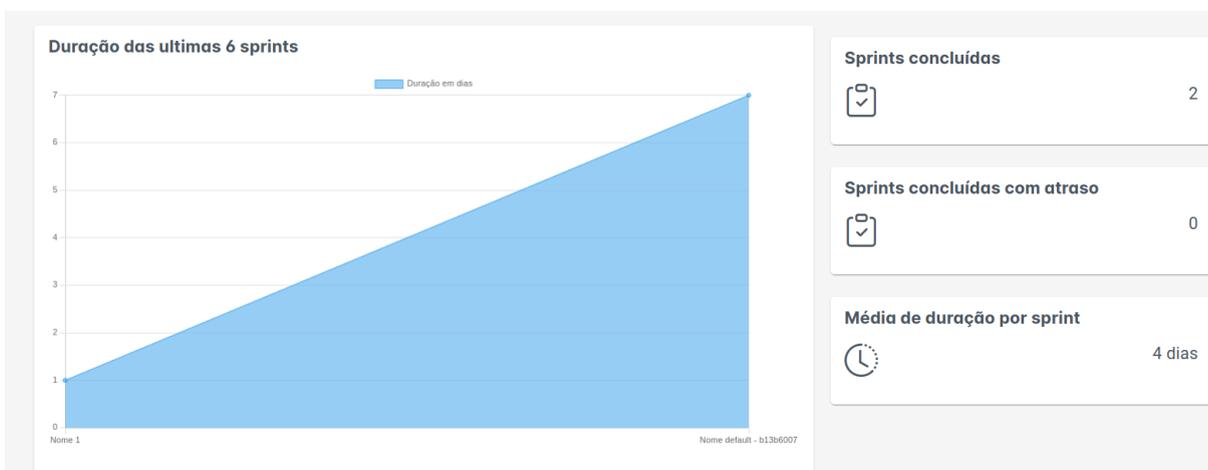
Fonte: própria (2024)

Figura 47 - Dashboard do time parte 2



Fonte: própria (2024)

Figura 48 - Dashboard do time parte 3



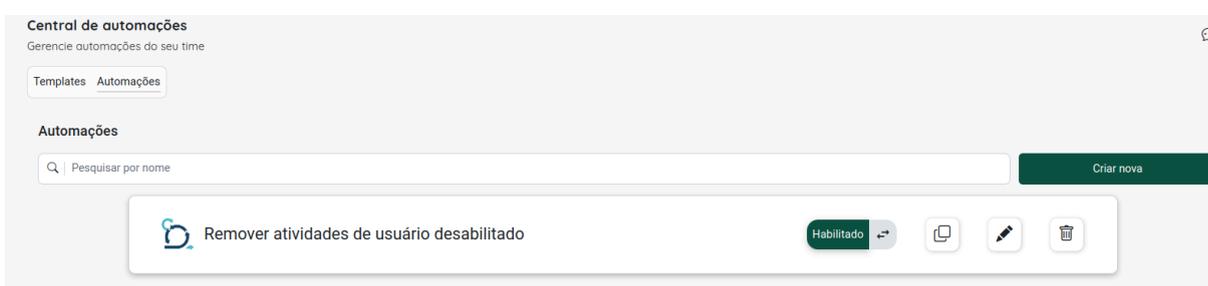
Fonte: própria (2024)

Essas informações reunidas no dashboard do time são essenciais para uma gestão eficaz das atividades e para monitorar o desempenho da equipe ao longo do tempo, permitindo tomadas de decisão mais embasadas e aprimoramento contínuo dos processos de trabalho.

6.2.2.5 Automatização de ações

Durante o desenvolvimento do projeto, foi identificado a repetição de certas ações que impactam a produtividade do time em um ambiente ágil. Para resolver esse problema, foi desenvolvida uma central de automação dentro do Genagile, com o objetivo de automatizar tarefas rotineiras e impulsionar a produtividade da equipe.

Figura 49 - Listagem de automações



Fonte: própria (2024)

Para criação de uma automação, é necessário realizar algumas configurações e para isso, no exemplo a seguinte será configurada uma automação para remover o usuário responsável de atividade que o mesmo esteja relacionado, quando ele for desativado. Primeiramente será necessário configurar o evento, selecionando a entidade que deseja monitorar e qual ação será capturada no evento.

Figura 50 - Criação de automação parte 1

Nova automação ×

Configure uma automação para seu time

Nome da automação

 Mudar responsável da atividade quando usuário for desativado

Evento

Quando um(a) for

- Escolha uma opção
- Atualizado
- Criado
- Deletado

+ Condições (0)

+ Ações (1)

Criar automação

Fonte: própria (2024)

Na segunda etapa, são configuradas as condições que devem ser atendidas para que a ação seja executada. Nesse processo, é selecionado um atributo da entidade escolhida e logo após uma condição, definindo o valor para comparação.

Figura 51 - Criação de automação parte 2

Nova automação ×

Configure uma automação para seu time

Nome da automação

 Mudar responsável da atividade quando usuário for desativado

+ Evento

- Condições (1)

Atributo	Condição	Valor	
Habilitado	EQUALS	false	
+			

+ Ações (1)

Criar automação

Fonte: própria (2024)

Para concluir a configuração da automação, o usuário define a ação que será executada quando o evento ocorrer e atender às condições previamente estabelecidas. Uma lista de ações específicas está disponível para a entidade selecionada. Nesse contexto, será escolhida a opção de trocar o usuário responsável pela atividade como a ação a ser realizada pela automação.

Figura 52 - Criação de automação parte 3

Nova automação ×

Configure uma automação para seu time

Nome da automação

Mudar responsável da atividade quando usuário for desativado

+ Evento

+ Condições (1)

- Ações (1)

Ação

Mudar usuário de atividades em aberto

Valor

Guilherme Oliveira

+ 

Criar automação

Fonte: própria (2024)

Ao finalizar, a automação estará funcionando e disponível na listagem, permitindo ao usuário realizar diversas ações, como habilitar ou desabilitar, editar configurações, duplicar ou excluir automações. Essa funcionalidade oferece flexibilidade ao usuário para configurar automações com base em eventos e condições de sua escolha, proporcionando ao mesmo a capacidade de eliminar a necessidade de realizar certas ações manualmente, o que contribui para aumentar a produtividade do time, onde os membros da equipe podem focar em atividades mais estratégicas e de maior valor para o projeto.

7. CONCLUSÃO E TRABALHOS FUTUROS

Este estudo ofereceu uma abrangente revisão da literatura acerca das metodologias ágeis, que têm como propósito central a busca pela flexibilidade e adaptabilidade. Esse enfoque torna-se particularmente crucial em ambientes permeados por mudanças rápidas e constantes. No âmbito das metodologias ágeis, o Scrum destaca-se como uma das abordagens mais amplamente adotadas, onde, não apenas abraça, mas também internaliza e implementa de maneira consistente os valores fundamentais da agilidade. Dentre esses valores, destacam-se a colaboração efetiva, a prontidão para se adaptar a mudanças e a entrega incremental de valor.

Com base na experiência do autor em uma empresa que utilizava o scrum, identificou-se que a principal dificuldade enfrentada pela equipe para manter o fluxo da *Sprint* otimizado estava relacionada à necessidade de atualizações frequentes em vários sistemas de gerenciamento de projetos, resultando em uma perda de tempo diária. Esses sistemas desempenham um papel crucial no tratamento de diferentes aspectos da *Sprint*, e a exigência constante de atualização emerge como um desafio significativo para preservar a eficiência do processo.

A criação do sistema Genagile, surge oferecendo integração e funcionalidades capazes de abranger os processos da *Sprint* auxiliando o desempenho da equipe, centralizando e otimizando o fluxo de trabalho, tendo como base a solução para os problemas identificados ao longo deste documento. No entanto, as funcionalidades foram desenvolvidas com o objetivo de atender à metodologia Scrum, contudo, existem mais metodologias ágeis como o Kanban e o XP no qual exige melhoria da plataforma para oferecer um suporte a essas práticas.

Para trabalhos futuros, está previsto o desenvolvimento de mais ferramentas de outras metodologias ágeis, além da criação de mais relatórios no qual seja possível ter uma visão da contribuição dos membros das equipes, integrações com outros sistemas de gerenciamento de projetos mais comumente utilizados no mercado como Jira e Trello, e mais opções de automatização de tarefas no ambiente, proporcionando flexibilidade e melhoria da arquitetura atual do sistema, buscando maior escalabilidade e qualidade no projeto. Essa abordagem de evolução

contínua garante que o sistema permaneça alinhado com as melhores práticas e requisitos em constante mudança do cenário ágil

REFERÊNCIAS

AGILE. **Agile Software Development: Hype or Working Practice?** Disponível em: <<https://agilie.com/blog/agile-software-development-hype-or-working-practice>>. Acesso em: 21 dez. 2023.

AGILE ALLIANCE. **O que é ágil?** Disponível em: <<https://www.agilealliance.org/agile101/>>. Acesso em: 14 abr. 2024.

ALURA, Danielle Oliveira, Tathiana Rodnarchuki. **Banco de dados.** Disponível em: <<https://www.alura.com.br/artigos/banco-de-dados?srsItd=AfmBOoomQdQiCZBBblOfb6eZoSZ7GXcE6hy31RJJIOvVKXROkHLwvlAg>>. Acesso em: 30 jul. 2024.

AMAZON WEB SERVICES. **The difference between monolithic and microservices architecture.** Disponível em: <<https://aws.amazon.com/pt/compare/the-difference-between-monolithic-and-microservices-architecture/>>. Acesso em: 30 jul. 2024.

ANGULAR. **Angular Documentation.** Disponível em: <<https://angular.io/docs>>. Acesso em: 20 dez. 2023.

BRITTO, M. **Netflix Eureka: Comunicação entre microservices.** Disponível em: <<https://medium.com/@michellibrito/netflix-eureka-comunica%C3%A7%C3%A3o-ent-re-microservices-383d32d39506>>. Acesso em: 30 jul. 2024.

CAMARGO, Robson. **Manifesto Ágil: entenda como surgiu e conheça os 12 princípios.** Disponível em: <<https://robsoncamargo.com.br/blog/Manifesto-Agil-entenda-como-surgiu-e-conheca-os-12-principios>>. Acesso em: 14 abr. 2024.

CAMARGO, Robson. **Planning poker: saiba por que equipes ágeis gostam de usar.** 2019. Disponível em: <<https://robsoncamargo.com.br/blog/Planning-poker>>. Acesso em: 18 ago. 2024.

CAREERFOUNDRY. **A Product Manager's Guide to Sprint Retrospectives.** Disponível em: <<https://careerfoundry.com/en/blog/product-management/Sprint-retrospective/>>. Acesso em: 14 abr. 2024.

DEITEL, P., DEITEL, H., & DEITEL, A. **Java: como programar.** 12. ed. Pearson, 2020.

DIGITAL.AI. **State of Agile 2022: A global survey on Agile adoption.** Digital.ai, 2022. Disponível em: <https://info.digital.ai/rs/981-LQX-968/images/SOA15.pdf?_ga=2.2071153.1396698264.1663000131-2013811604.1663000131>. Acesso em: 24 nov. 2024.

EL-MASRY, Shady. **Angular 6 for Enterprise-Ready Web Applications: Deliver production-ready and cloud-scale Angular web apps.** Packt Publishing, 2018.

FRENET. **Gateway: O que é e como funciona?** Disponível em: <<https://www.frenet.com.br/blog/gateway-o-que-e-como-funciona/>>. Acesso em: 30 jul. 2024.

GABRIEL, G. **O que é e para o que serve arquitetura monolítica?**. Disponível em: <<https://dev.to/gabrielgcj/o-que-e-e-para-o-que-serve-arquitetura-monolitica-3pp4>>. Acesso em: 30 jul. 2024.

GITLAB. **Plataforma de DevOps para gerenciamento de código**. Disponível em: <<https://gitlab.com>>. Acesso em: 18 ago. 2024.

HORSTMANN, C. S. **Java SE 8 para programadores**. 3. ed. Pearson, 2018.

KARANAM, A., KULKARNI, S., & KODHANDAPANI, M. (2020). **Hands-On Spring Boot: Build and deploy Java applications with confidence**. Packt Publishing.

LIANG, Y. D. **Introdução à programação em Java: Uma abordagem abrangente**. 12. ed. Pearson, 2020.

MONDAY. **Plataforma de gerenciamento de trabalho**. Disponível em: <<https://monday.com>>. Acesso em: 18 ago. 2024.

PM-PATERNES. **What is Scrum? An overview of Scrum and The Agile Journey**. Disponível em: <<https://www.pm-partners.com.au/insights/the-agile-journey-a-scrum-overview/>>. Acesso em: 14 abr. 2024.

RED HAT. **What is agile methodology?** Disponível em: <<https://www.redhat.com/pt-br/devops/what-is-agile-methodology>>. Acesso em: 29 nov. 2023.

ROHIT AKIWATKAR. **Agile Adoption Statistics: How is Software Development changing?** Disponível em: <<https://www.simform.com/blog/state-of-agile-adoption/>>. Acesso em: 14 abr. 2024.

SCHWABER, K.; SUTHERLAND, J. **Scrum guide**. Disponível em: <<https://scrumguides.org/scrum-guide.html>>. Acesso em: 21 dez. 2023.

SCRUM POKER. **Ferramenta para estimativa de tarefas**. Disponível em: <<https://scrumpoker.online>>. Acesso em: 18 ago. 2024.

SIDKY, A.; ARTHUR, J. D.; BOHNER, S. A. **A structured review of the agile software development literature**. ScienceDirect. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0164121212000532>>. Acesso em: 21 dez. 2023.

SPRING. **Spring Boot**. Disponível em: <<https://spring.io/projects/spring-boot>>. Acesso em: 20 dez. 2023.

TECHTARGET. **What is a daily stand-up meeting?** Disponível em: <<https://www.techtarget.com/searchsoftwarequality/definition/daily-stand-up-meeting>>. Acesso em: 21 dez. 2023.

TRENTIM, Mario H. **Adoção do ágil: números e estatísticas de 2022**. LinkedIn, 11 mar. 2022. Disponível em: <<https://www.linkedin.com/pulse/ado%C3%A7%C3%A3o-do-%C3%A1gil-n%C3%BAmeros-e-estat%C3%ADsticas-de-2022-mario-h-trentim//>>. Acesso em: 19 ago. 2024.

WALL, C. (2016). **Spring Boot in Action**. Manning Publications.

WIKIPÉDIA. **Gateway**. Disponível em: <<https://pt.wikipedia.org/wiki/Gateway>>. Acesso em: 30 jul. 2024.