

Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Engenharia de Software

# **Backlog Maker: Uma Ferramenta de Apoio à Construção e à Manutenção do Backlog do Produto**

**Autores: Maria Luiza Ferreira Assumpção Almeida e Natália  
Maria Rodrigues Queiroz**

**Orientador: Profa. Dra. Milene Serrano**

Brasília, DF

2021





Maria Luiza Ferreira Assumpção Almeida e Natália Maria Rodrigues Queiroz

## **Backlog Maker: Uma Ferramenta de Apoio à Construção e à Manutenção do Backlog do Produto**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Profa. Dra. Milene Serrano

Coorientador: Prof. Dr. Maurício Serrano

Brasília, DF

2021

---

Maria Luiza Ferreira Assumpção Almeida e Natália Maria Rodrigues Queiroz  
Backlog Maker: Uma Ferramenta de Apoio à Construção e à Manutenção do  
Backlog do Produto/ Maria Luiza Ferreira Assumpção Almeida e Natália Maria  
Rodrigues Queiroz . – Brasília, DF, 2021-  
177 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Milene Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA , 2021.

1. . 2. . I. Profa. Dra. Milene Serrano. II. Universidade de Brasília. III. Facul-  
dade UnB Gama. IV. Backlog Maker: Uma Ferramenta de Apoio à Construção e  
à Manutenção do Backlog do Produto

CDU 02:141:005.6

---

Maria Luiza Ferreira Assumpção Almeida e Natália Maria Rodrigues Queiroz

## **Backlog Maker: Uma Ferramenta de Apoio à Construção e à Manutenção do Backlog do Produto**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 04 de Novembro de 2021:

---

**Profa. Dra. Milene Serrano**  
Orientador

---

**Prof. Dr. Maurício Serrano**  
Coorientador

---

**Profa. Dra. Carla Silva Rocha Aguiar**  
Examinadora

Brasília, DF  
2021



# Agradecimentos

Agradecemos a Deus, primeiramente, por ter nos dado a força e a coragem necessárias para chegarmos até aqui.

Agradecemos às nossas famílias, que sempre nos incentivaram a estudar e a batalhar pelos nossos estudos.

Agradecemos aos nossos orientadores, Profa. Dra. Milene Serrano e Prof. Dr. Maurício Serrano, que nos orientaram e apoiaram todo o nosso trabalho.

Agradecemos aos nossos amigos, que, de alguma forma, contribuíram para o nosso aprendizado.





# Resumo

Os processos da Engenharia de Software são estabelecidos com o objetivo de fornecer software em tempo hábil e com qualidade adequada para satisfazer seus *stakeholders*. Para uma série de empresas, os modelos ágeis têm sido preferidos como abordagens de desenvolvimento de software e o *Scrum* é um dos métodos mais utilizados. Nesse modelo, a gestão dos requisitos tem como artefato principal o *Backlog* do Produto, que é definido por uma lista de todos os requisitos do produto de *software*. Alguns problemas podem surgir na gestão de requisitos quando se utiliza o *Scrum*, podendo levar à falha de todo o projeto ou à entrega inadequada de incrementos. Dessa forma, é necessário construir um *Backlog* do Produto que seja conciso, com níveis de granularidade bem definidos e que seja entendido igualmente por todos os *stakeholders*. Em vista disso, este Trabalho de Conclusão de Curso tem como produto uma ferramenta que oferece apoio tecnológico à construção e à manutenção do *Backlog* do Produto e, por meio dela, possibilita maneiras de mitigar os problemas identificados nesse processo.

**Palavras-chaves:** *Backlog* do Produto. Modelos ágeis. Gestão de Requisitos. *Scrum*.



# Abstract

Software Engineering processes are established with the objective of providing software in a timely manner and with proper quality to satisfy its stakeholders. For a number of companies, agile models have been preferred as software development approaches and Scrum is one of the most used methods. In this model, requirements management has as its main artifact the Product Backlog, which is defined by a list of all software product requirements. Some problems can arise in the requirements management when using Scrum, which can lead to the failure of the entire project or the inadequate delivery of increments. Thus, it is necessary to build a Product Backlog that is concise, with well-defined levels of granularity and that is equally understood by all stakeholders. With that in mind, this Course Conclusion Work provides a tool that offers technological support to the construction and maintenance of the Product Backlog and, through it, enables ways to mitigate the problems identified in this process.

**Key-words:** Product Backlog. Agile models. Requirements Management. Scrum.



# Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
NoSQL	<i>Not Only SQL</i>
PO	<i>Product Owner</i>
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>
SM	<i>Scrum Master</i>
SQL	<i>Structured Query Language</i>
CRUD	<i>Create, Read, Update, Delete</i>



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	Contextualização	17
1.2	Questões de Pesquisa	19
1.3	Justificativa	19
1.4	Objetivos	20
1.4.1	Objetivos Gerais	20
1.4.2	Objetivos Específicos	20
1.5	Metodologia	21
1.6	Organização do Trabalho	21
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>23</b>
2.1	Engenharia de Software	23
2.2	Modelos Ágeis	24
2.2.1	Scrum	25
2.2.2	Projetos Ágeis de Pequeno e Médio Portes	27
2.3	Engenharia de Requisitos	28
2.3.1	Gestão de Requisitos no Scrum	29
2.3.2	Ferramenta de Apoio Tecnológico à Gestão do <i>Backlog</i> do Produto	31
2.4	Resumo do Capítulo	32
<b>3</b>	<b>SUPORTE TECNOLÓGICO</b>	<b>35</b>
3.1	Tecnologias de Apoio à Aplicação	35
3.1.1	Git 2.31.0	35
3.1.2	GitHub Free Version	35
3.1.3	Docker 20.10	35
3.1.4	Docker Compose 1.28.5	35
3.1.5	JavaScript ECMAScript 2020	36
3.1.6	TypeScript 4.2.3	36
3.1.7	NodeJS 14.16.0	36
3.1.8	Express 4.17.1	36
3.1.9	Firebase Node.js SDKs	36
3.1.10	ReactJS 17.0.1	37
3.1.11	NextJS 10.0.9	37
3.1.12	Tailwind CSS 2.1.4	37
3.1.13	Heroku Heroku-20	37
3.1.14	Vercel Hobby Version	38

3.1.15	ZenHub Free Version . . . . .	38
<b>3.2</b>	<b>Tecnologias de Apoio à Pesquisa e à Comunicação . . . . .</b>	<b>38</b>
3.2.1	Drive . . . . .	38
3.2.2	Zotero 5.0 . . . . .	38
3.2.3	LaTeX LaTeX3 . . . . .	38
3.2.4	Overleaf Community Edition . . . . .	39
3.2.5	WhatsApp 2.21.50 . . . . .	39
3.2.6	Hangouts . . . . .	39
<b>3.3</b>	<b>Resumo do Capítulo . . . . .</b>	<b>39</b>
<b>4</b>	<b>METODOLOGIA . . . . .</b>	<b>41</b>
<b>4.1</b>	<b>Metodologia de Pesquisa . . . . .</b>	<b>41</b>
4.1.1	Abordagem da Pesquisa . . . . .	41
4.1.2	Natureza da Pesquisa . . . . .	41
4.1.3	Objetivos da Pesquisa . . . . .	42
4.1.4	Procedimentos . . . . .	42
4.1.5	Classificação da Pesquisa . . . . .	42
<b>4.2</b>	<b>Metodologia de Desenvolvimento da Proposta . . . . .</b>	<b>43</b>
4.2.1	<i>Scrum</i> . . . . .	43
4.2.2	<i>Kanban</i> . . . . .	44
4.2.3	<i>XP - Extreme Programming</i> . . . . .	45
4.2.4	Metodologia Híbrida . . . . .	46
<b>4.3</b>	<b>Metodologia de Análise . . . . .</b>	<b>47</b>
<b>4.4</b>	<b>Fluxos de Atividades . . . . .</b>	<b>47</b>
4.4.1	Trabalho de Conclusão de Curso 1 . . . . .	48
4.4.2	Trabalho de Conclusão de Curso 2 . . . . .	50
4.4.3	Realização da Revisão Sistemática . . . . .	51
4.4.4	Desenvolvimento da Proposta . . . . .	52
4.4.5	Análise de Resultados . . . . .	54
<b>4.5</b>	<b>Cronogramas . . . . .</b>	<b>55</b>
<b>4.6</b>	<b>Resumo do Capítulo . . . . .</b>	<b>55</b>
<b>5</b>	<b>BACKLOG MAKER . . . . .</b>	<b>57</b>
<b>5.1</b>	<b>Contextualização . . . . .</b>	<b>57</b>
5.1.1	Ferramentas Similares . . . . .	58
<b>5.2</b>	<b><i>Baseline</i> de Requisitos . . . . .</b>	<b>67</b>
<b>5.3</b>	<b>Arquitetura da Solução . . . . .</b>	<b>68</b>
5.3.1	<i>Firestore</i> . . . . .	69
5.3.2	<i>Front-end</i> . . . . .	70
5.3.3	API . . . . .	70



5.3.4	Integração dos Componentes . . . . .	70
<b>5.4</b>	<b>Prova de Conceito . . . . .</b>	<b>71</b>
5.4.1	Protótipo . . . . .	71
5.4.2	Configuração do Ambiente de Desenvolvimento e Integração entre os Com- ponentes da Solução . . . . .	75
<b>5.5</b>	<b>Backlog da Ferramenta . . . . .</b>	<b>76</b>
<b>5.6</b>	<b>A solução . . . . .</b>	<b>76</b>
5.6.1	Visualização e Criação de Projetos . . . . .	77
5.6.2	Visualização e Gerenciamento do <i>Backlog</i> do Produto . . . . .	77
5.6.3	Guia e Textos de Ajuda . . . . .	79
5.6.4	Comentários em Épico, <i>Feature</i> , História de Usuário e <i>Bugs</i> . . . . .	80
<b>5.7</b>	<b>Resumo do Capítulo . . . . .</b>	<b>82</b>
<b>6</b>	<b>RESULTADOS . . . . .</b>	<b>83</b>
<b>6.1</b>	<b>Quadro Problemas vs. Potenciais Soluções . . . . .</b>	<b>83</b>
<b>6.2</b>	<b>Primeiro Ciclo de Pesquisa-Ação . . . . .</b>	<b>84</b>
<b>6.3</b>	<b>Segundo Ciclo de Pesquisa-Ação . . . . .</b>	<b>95</b>
<b>6.4</b>	<b>Operacionalizações dos Ciclos de Pesquisa-Ação . . . . .</b>	<b>101</b>
<b>6.5</b>	<b>Resumo do Capítulo . . . . .</b>	<b>105</b>
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>107</b>
<b>7.1</b>	<b>Trabalhos Futuros . . . . .</b>	<b>109</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>111</b>
	<b>APÊNDICES . . . . .</b>	<b>115</b>
	<b>APÊNDICE A – REVISÃO SISTEMÁTICA DA LITERATURA . . . . .</b>	<b>117</b>
	<b>APÊNDICE B – QUADRO PROBLEMAS VS. POTENCIAIS SO- LUÇÕES . . . . .</b>	<b>119</b>
	<b>APÊNDICE C – PROTÓTIPO . . . . .</b>	<b>125</b>
	<b>APÊNDICE D – AUTOMAÇÃO DO AMBIENTE DE DESENVOL- VIMENTO DO <i>FRONT-END</i> . . . . .</b>	<b>143</b>
	<b>APÊNDICE E – AUTOMAÇÃO DO AMBIENTE DE DESENVOL- VIMENTO DA API . . . . .</b>	<b>145</b>
	<b>APÊNDICE F – INICIALIZAÇÃO DO FIREBASE . . . . .</b>	<b>147</b>

<b>APÊNDICE G – INTEGRAÇÃO ENTRE OS COMPONENTES DO SISTEMA . . . . .</b>	<b>149</b>
<b>APÊNDICE H – TELAS DA FERRAMENTA . . . . .</b>	<b>155</b>

# 1 Introdução

Neste capítulo, pretende-se contextualizar a pesquisa realizada neste Trabalho de Conclusão de Curso, sendo o mesmo centrado em Metodologias Ágeis e no Gerenciamento dos Requisitos de Software. A partir da contextualização, serão apresentadas as questões de pesquisa que norteiam a revisão da literatura bem como são respondidas ao final do trabalho. Dando continuidade, são acordadas as justificativas para a realização deste trabalho, as quais procuram mitigar os típicos problemas encontrados no desenvolvimento orientado por Metodologias Ágeis, e que possuem fronteiras com as atividades inerentes à Engenharia de Requisitos. Adicionalmente, são descritos os principais objetivos atingidos ao longo da pesquisa. Por fim, tem-se a organização da monografia.

## 1.1 Contextualização

A utilização de aplicações que ajudam as organizações a alcançarem suas metas e seus objetivos tem se tornado uma obrigação nos dias atuais. Muitas companhias estão desenvolvendo aplicações que podem ajudar seus usuários a fazerem seus trabalhos diários (ADITAMA; LEGOWO, 2020). A Engenharia de Software surge, então, como uma abordagem sistemática, disciplinada e mensurável para o desenvolvimento, a operação, e a manutenção de software ou de aplicações (PRESSMAN; MAXIM, 2016). Com o objetivo de fornecer software em tempo hábil e com qualidade adequada para satisfazer seus patrocinadores e usuários, os processos da Engenharia de Software são estabelecidos. Um processo não é algo rígido para descrever como construir um software, e sim uma abordagem adaptável que permite pessoas trabalhando para escolher o conjunto apropriado de ações e tarefas a serem realizadas (ADITAMA; LEGOWO, 2020).

Alguns dos modelos de processo existentes são: Cascata, Espiral, Prototipação, Iterativo e Incremental, Desenvolvimento Rápido de Aplicações e os Modelos Ágeis (ANAND; DINAKARAN, 2015). Os modelos ágeis, por sua vez, existem há quase 20 anos e a sua adoção tem crescido significativamente, fazendo com que se tornem as abordagens de desenvolvimento preferidas de uma série de indústrias (KHOMYAKOV; MIRGALIMOVA; SILLITTI, 2020). Ao contrário dos modelos tradicionais, tais abordagens enfatizam a interação com o cliente, de forma que um entregável funcional seja entregue, logo nas primeiras iterações do processo de desenvolvimento, para que um *feedback* bem como uma validação inicial possam ser obtidos junto aos interessados (RAJEEV; HEJIB, 2018a).

Os quatro valores presentes no Manifesto Ágil, nos quais se baseiam os modelos de desenvolvimento ágil, focam no desenvolvimento de software funcional; na priorização de indivíduos e iterações; na colaboração do cliente, e na reação rápida a mudanças (ALQU-

DAH; RAZALI, 2018). Neste trabalho, o método ágil de desenvolvimento em pauta é o *Scrum*. O *Scrum* é um dos métodos ágeis mais utilizados, e tem como principal objetivo a conclusão do projeto com o mínimo de tempo e custo (ADITAMA; LEGOWO, 2020). Aditama e Legowo (2020) destacam, ainda, que as diferenças entre ele e os modelos tradicionais, como o modelo Cascata, residem nos seus papéis, reuniões, características, artefatos e linguagem.

O *Scrum* é uma estrutura de processo que permite entregar produtos com o maior valor possível e lidar com problemas ou situações complexas, utilizando uma abordagem iterativa e incremental para desenvolver produtos com equipes multifuncionais (ANAND; DINAKARAN, 2015). O método compreende uma série de práticas amplamente utilizadas na indústria (ALQUDAH; RAZALI, 2018). Um time *Scrum* é formado pelo *Product Owner* (PO), pelo *Scrum Master* (SM) e pelo Time de Desenvolvimento. Cada equipe realiza suas tarefas por meio de artefatos como o *Backlog* do Produto e o *Backlog* da *Sprint*. Além disso, algumas tarefas devem ser realizadas pela equipe para que seus objetivos sejam atingidos, entre elas estão: o refinamento do *Backlog*; as reuniões diárias regulares; as reuniões de planejamento; a retrospectiva e a revisão considerando cada *sprint* ou iteração (ALQUDAH; RAZALI, 2018).

Por sua vez, a gestão dos requisitos no modelo *Scrum* tem como artefato principal o *Backlog* do Produto, que é definido por uma lista de todos os requisitos desejados no produto. Esse artefato é proposto pelo dono do produto (PO) na forma de histórias de usuário. A história de usuário é feita explicitando o ator da história; seu objetivo ou tarefa que precisa ser realizada; explicação da necessidade e seu valor associado, além de uma prioridade correspondente. Além disso, as histórias são decompostas em tarefas executáveis para que sejam compreendidas pela equipe de desenvolvimento (SANTAMARIA, 2007).

O *Product Owner*, que pode ser o próprio cliente ou um representante do cliente, tem como principais funções a manutenção e a priorização do *Backlog* do Produto, especificando e reunindo as histórias de usuário ou requisitos, assim como os seus critérios de aceitação (HODA; MURUGESAN, 2016). Após a obtenção das prioridades das histórias pelo *Product Owner*, a equipe faz a estimativa e se reúne para detalhar como será a implementação da história de usuário (RAJEEV; HEJIB, 2018b). Com a necessidade de se adaptar às mudanças, a metodologia ágil exige a repriorização do *Backlog* do Produto frequentemente (ENGUM; RACHEVA; DANEVA, 2009).

Alguns problemas podem surgir na gestão de requisitos quando se utiliza o *Scrum* como modelo de desenvolvimento, dentre eles estão: a dificuldade de transformar requisitos em histórias de usuários; problemas na priorização, devido às constantes mudanças propostas pela metodologia ágil; conflito de prioridades com as partes interessadas, e a definição de quando a história pode ser considerada pronta para que esta seja entregue

(ANAND; DINAKARAN, 2015). Segundo (ENGUM; RACHEVA; DANEVA, 2009), dificuldades como a especificação insuficiente da tarefa a ser realizada e a falta de identificação de dependência entre as tarefas também podem atrapalhar no planejamento da *sprint*.

Diante do exposto, com os problemas e desafios encontrados na literatura referentes às dificuldades de construir e manter um *Backlog* do Produto, que seja conciso e de fácil manutenção, surgiu o propósito deste Trabalho de Conclusão de Curso. Além da extensa literatura, somou-se a necessidade de empresas de pequeno e médio porte, que adotam o *Scrum* como modelo de desenvolvimento, em ter um bom *Backlog* do Produto e uma boa gestão de requisitos para alcançar os objetivos desejados. Portanto, foi construída uma ferramenta que oferece apoio a esse desafio, levando em consideração fatores como a granularidade e a rastreabilidade dos requisitos. A granularidade é relevante para organizar o *Backlog* do Produto em níveis adequados, que podem ser desde o nível menos granular (ou seja, Tema), seguido de Épico, *Feature*, História de Usuário e Tarefa, até o mais granular (ou seja, Sub-Tarefa). Já a rastreabilidade permite manter os elos de ligação *forward to* (i.e. das fontes de informação para o *Backlog* do Produto); *forward from* (i.e. do *Backlog* do Produto até o código); *backward from* (i.e. do *Backlog* do Produto para as fontes de informações), e *backward to* (i.e. do código ao *Backlog* do Produto). Nesses casos, considerando que a *Baseline* de Requisitos está especificada no artefato *Backlog* do Produto.

## 1.2 Questões de Pesquisa

Para guiar a revisão da literatura e o embasamento do trabalho, foram definidas as seguintes questões de pesquisa:

Questão 1 - Quais são os problemas enfrentados pelas equipes *Scrum* na definição e na manutenção do *Backlog* do Produto ao longo do ciclo de vida de desenvolvimento de um software?

Questão 2 - Quais são as práticas mais indicadas pela literatura para mitigar os problemas encontrados e guiar a construção de uma ferramenta que apoie a montagem e a manutenção de um bom *Backlog* do Produto?

## 1.3 Justificativa

Conforme (ANAND; DINAKARAN, 2015), o *Backlog* do Produto é um artefato central no *Scrum*. A lista de defeitos, melhorias solicitadas pelo cliente, funcionalidades, atualização de tecnologias, entre outros itens, são concentrados nele. Segundo os autores, a primeira atividade do *Scrum* é a atribuição dos itens do *Backlog* do Produto, e tal

atividade, caso não seja devidamente realizada, pode levar à falha de todo o projeto ou à entrega inadequada de incrementos.

Diferentemente do desenvolvimento tradicional, o *Backlog* do Produto em um projeto ágil muda frequentemente, o que exige constante (re)priorização de requisitos. Um procedimento eficaz faz-se, então, necessário para ajudar tanto o *Product Owner* quanto o gerente de projeto a manter o rastro dos requisitos e das mudanças no *Backlog* do Produto (ENGUM; RACHEVA; DANEVA, 2009).

Com o objetivo de gerenciar os requisitos e manter o *Backlog* do Produto, uma prática comum em companhias ágeis novas e pequenas é a utilização de uma planilha complementada com o uso de *post-its* escritos à mão. Embora simples e fácil de aplicar, esse método pode trazer alguns sérios problemas, entre eles estão: a falta de informação necessária aos desenvolvedores, levando-os a não entender o que deve ser desenvolvido; problemas de ambiguidade, ou seja, quando a interpretação individual se sobressai em relação ao entendimento compartilhado entre os participantes do projeto, e sobrecarga, quando há dependência entre os membros da equipe. Por isso, é necessária uma maior coordenação e compartilhamento de conhecimento. (ENGUM; RACHEVA; DANEVA, 2009)

Em vista disso, com a realização deste Trabalho de Conclusão de Curso, procurou-se desenvolver uma ferramenta que oferece apoio tecnológico à construção e à manutenção do *Backlog* do Produto e, por meio dela, têm-se maneiras de mitigar os problemas identificados nesse processo.

## 1.4 Objetivos

### 1.4.1 Objetivos Gerais

Construir uma ferramenta que auxilie na construção e na manutenção do *Backlog* do Produto, de forma que seja possível mitigar os típicos problemas encontrados pelos desenvolvedores de software no escopo das atividades inerentes, principalmente, à Engenharia de Requisitos. Além disso, essa ferramenta tem como propósito ser acessível às empresas de pequeno e médio portes com equipes pequenas, as quais aplicam o *Scrum* como modelo de desenvolvimento, e que, preferivelmente, tenham o *PO* inserido em seus contextos. Esse objetivo foi atingido, e os resultados são apresentados ao longo desta monografia.

### 1.4.2 Objetivos Específicos

Os objetivos específicos deste trabalho, são:

1. Definir quais são os principais problemas da construção e da manutenção do *Backlog* do Produto, por meio de uma revisão sistemática da literatura;
2. Definir quais são as possíveis soluções para os problemas encontrados na construção e na manutenção do *Backlog* do Produto, por meio do referencial teórico encontrado com a revisão sistemática;
3. Elicitar os requisitos necessários para a construção da ferramenta;
4. Definir a arquitetura do sistema a ser desenvolvido;
5. Desenvolver a ferramenta que auxilie na construção e na manutenção do *Backlog* do Produto, e
6. Realizar pesquisa-ação e documentar os resultados obtidos.

Cabe ressaltar que esses objetivos específicos foram atingidos, sendo cada um deles tratado nesta monografia em detalhe.

## 1.5 Metodologia

Para levantar os materiais sobre Modelos Ágeis, bem como gestão de requisitos no *Scrum* e problemas relacionados à construção e à manutenção do *Backlog* do Produto, foi realizada uma Revisão Sistemática baseada no protocolo de (KITCHENHAM, 2004), que define um processo com três fases: planejamento, execução e análise de resultados. Ademais, foi utilizada a técnica *Snow Ball* definida por (BALDIN N.; MUNHOZ, 2011). Detalhes sobre a Revisão Sistemática podem ser encontrados no Apêndice A.

Adicionalmente, em termos de metodologia de desenvolvimento da ferramenta, fez-se uso de uma adaptação do *Scrum*. Por fim, para análise de resultados, tem-se o uso da modalidade de pesquisa-ação, seguindo uma abordagem, predominantemente, qualitativa. Outros detalhes metodológicos constam no Capítulo 4.

## 1.6 Organização do Trabalho

Este trabalho está dividido nos seguintes capítulos:

**Capítulo 2 - Referencial Teórico:** apresenta as áreas e conceitos relacionados ao tema do trabalho;

**Capítulo 3 - Suporte Tecnológico:** caracteriza as ferramentas, *frameworks* e aparatos tecnológicos necessários para o desenvolvimento do produto proposto;

**Capítulo 4 - Metodologia:** traz a especificação das escolhas metodológicas feitas para a condução do trabalho, desde a idealização e pesquisa até o desenvolvimento do sistema, envolvendo a coleta e a análise de seus resultados;

**Capítulo 5 - *Backlog Maker*:** apresenta as especificações do sistema de construção e de manutenção do *Backlog* do Produto, bem como o detalhamento da arquitetura da solução;

**Capítulo 6 - Resultados:** apresenta os principais resultados obtidos até o momento, e

**Capítulo 7 - Conclusão:** acordando as considerações finais, bem como apresentando possibilidades de trabalhos futuros.



## 2 Referencial Teórico

O escopo deste Trabalho de Conclusão de Curso está associado à identificação de como construir e manter o *Backlog* do Produto. Dado este contexto, este capítulo apresenta o referencial teórico necessário para nortear o entendimento dos conceitos revelantes para o projeto.

Dentre os tópicos destacados neste capítulo, fez-se necessário o entendimento sobre o que é a Engenharia de Software; apresentação da visão geral sobre modelos ágeis; detalhamento do *Scrum*, modelo ágil em foco no trabalho; apresentação geral de projetos ágeis para empresas de pequeno e médio portes, o qual representa o público-alvo do projeto; apresentação das principais atividades inerentes à Engenharia de Requisitos; apresentação da gestão de requisitos no *Scrum*, evidenciando problemas e soluções encontrados na literatura referente à Engenharia de Requisitos em modelos ágeis, e exposição do que a literatura evidencia sobre uma ferramenta que ofereça apoio tecnológico à construção e à manutenção do *Backlog* do Produto.

Com o objetivo de manter a clareza de todo o referencial teórico exposto, foi construída uma lógica mais abrangente desde a Engenharia de Software, perpassando pelo modelo ágil e pelo público-alvo escolhido, até chegar na gestão de requisitos no *Scrum* e na construção de uma ferramenta de apoio a esta atividade, evidenciados na literatura. Por fim, tem-se o resumo do capítulo.

### 2.1 Engenharia de Software

Segundo (PRESSMAN; MAXIM, 2016), alguns fatores devem ser percebidos para o desenvolvimento de software que seja capaz de enfrentar os desafios do século XXI, são eles: 1) deve haver um esforço concentrado para compreender o problema antes de desenvolver uma solução de software; 2) projetar tornou-se uma atividade fundamental; 3) um software deve apresentar qualidade elevada, e 4) um software deve ser passível de manutenção. Tais fatores levam à constatação de que o software, em todas as suas formas e em todos os seus campos de aplicação, deve passar pelos processos de engenharia.

A Engenharia de Software é definida como sendo a aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software. No entanto, uma abordagem "sistemática, disciplinada e quantificável", aplicada por determinada equipe, pode ser um fardo para outra. Da mesma forma que se precisa de disciplina, habilidade de adaptação e agilidade também são necessárias. A Engenharia de Software é, então, uma abordagem em camadas (Figura 1), que deve estar

fundamentada em um comprometimento organizacional com a qualidade (PRESSMAN; MAXIM, 2016).

Figura 1 – Camadas da Engenharia de Software



Fonte: (PRESSMAN; MAXIM, 2016)

A base para a Engenharia de Software é a camada de processos. A Figura 1 indica que um processo não é algo rígido para descrever como construir um software, e sim uma abordagem adaptável que permite pessoas trabalhando para escolher o conjunto apropriado de ações e tarefas a serem realizadas. O objetivo é sempre oferecer software em tempo e qualidade adequados para satisfazer seus patrocinadores e usuários (ADITAMA; LEGOWO, 2020).

A fim de estabelecer o alicerce para um processo completo de Engenharia de Software, existem as chamadas metodologias (ou *frameworks*) de processo, que, por meio da identificação de um pequeno número de atividades estruturais aplicáveis a todos os projetos de software, englobam um conjunto de atividades de apoio aplicáveis em todo o processo de software. Uma metodologia de processo genérica para Engenharia de Software compreende as seguintes atividades: comunicação, planejamento, modelagem, construção e implantação (PRESSMAN; MAXIM, 2016).

## 2.2 Modelos Ágeis

Todos os modelos de processo de software podem acomodar as atividades metodológicas genéricas descritas no final da Seção 2.1. No entanto, cada modelo enfatiza de forma diferente essas atividades, bem como define um fluxo de processo que invoca diferentemente cada atividade metodológica, incluindo as tarefas e ações de Engenharia de Software (PRESSMAN; MAXIM, 2016).

Originalmente, modelos de processo prescritivo foram propostos para coordenar o desenvolvimento de software. Essas abordagens tradicionais contribuíram consideravelmente quanto à estruturação do trabalho de Engenharia de Software (PRESSMAN; MAXIM, 2016). Apesar disso, tais abordagens estabelecem os requisitos e a modelagem do software no início do projeto, impondo rigidez ao processo de mudança. Os requisitos

definidos no início do projeto, no entanto, raramente atendem às necessidades dos usuários finais e isso faz com que o projeto inteiro seja falho (HERDIKA; BUDIARDJO, 2020).

Na tentativa de sanar as fraquezas reais e perceptíveis da Engenharia de Software convencional, foram desenvolvidos os modelos ágeis (PRESSMAN; MAXIM, 2016). Tais modelos são baseados nos valores e princípios presentes no Manifesto Ágil (ALQUDAH; RAZALI, 2018). Em suma, a principal característica dos modelos ágeis, em comparação aos modelos tradicionais, é a habilidade de lidar com incertezas no projeto e produzir software de alta qualidade em um curto período de tempo. A abordagem de desenvolvimento ágil permite o desenvolvimento de software em intervalos regulares, ou seja, iterações, produzindo o software em incrementos, e isso facilita a capacidade de adaptação a mudanças (HERDIKA; BUDIARDJO, 2020).

Alguns dos modelos ágeis de desenvolvimento existentes são: *Scrum*, XP (*Extreme Programming*), *Lean* e *Kanban*. Eles são os modelos ágeis estudados mais populares e são muito diferentes uns dos outros. Enquanto o *Scrum* tem seu foco na gestão do desenvolvimento; o XP auxilia a gestão de mudanças no código; o *Lean* concentra-se na eliminação de desperdícios, e o *Kanban* busca a melhoria do processo de desenvolvimento (HERDIKA; BUDIARDJO, 2020). Este trabalho utiliza o *Scrum*, modelo ágil mais conhecido e amplamente utilizado (ADITAMA; LEGOWO, 2020), como modelo base para o desenvolvimento do apoio tecnológico à construção e à manutenção do *Backlog* do Produto. A Subseção seguinte apresenta os principais conceitos referentes ao *Scrum*.

### 2.2.1 Scrum

Como citado anteriormente, o *Scrum* é um modelo ágil com foco na gestão do desenvolvimento de software. Ele orienta o gerenciamento de tempo e de escopo do projeto; o acompanhamento do progresso, e o desenvolvimento centrado no cliente. Também serve de guia para conduzir o desenvolvimento, regulando vários componentes, como papéis, artefatos e atividades. As organizações que adotam o *Scrum* podem aderir às práticas relevantes do mesmo, adaptando o modelo com base nas próprias características (HERDIKA; BUDIARDJO, 2020).

O *Scrum* adere às regras que ligam os eventos, papéis e artefatos, ao mesmo tempo que controla os relacionamentos e as interações que ocorrem entre eles (ALQUDAH; RAZALI, 2018). As principais atividades compreendidas em um processo orientado pelo *Scrum* são as seguintes:

1. Determinação do *Backlog* do Produto

O *Backlog* do Produto é uma lista que contém os itens de trabalho que serão realizados no desenvolvimento do sistema, normalmente em forma de histórias de usuário

(ADITAMA; LEGOWO, 2020). As histórias de usuário, por sua vez, são a representação dos requisitos do usuário, escritas em frases diretas em um alto nível de comunicação. A definição do *Backlog* do Produto abrange a criação, refinamento, estimativa e priorização dos seus itens com base nos requisitos do usuário (HERDIKA; BUDIARDJO, 2020).

## 2. Reunião de Planejamento da *Sprint*

Uma *sprint* é uma iteração em um determinado período de tempo para o desenvolvimento do sistema (ADITAMA; LEGOWO, 2020). O planejamento da *sprint* visa definir os objetivos da *sprint* (HERDIKA; BUDIARDJO, 2020).

## 3. Determinação do *Backlog* da *Sprint*

O *Backlog* da *Sprint* é o conjunto de itens selecionados a partir do *Backlog* do Produto para serem completados durante a *sprint* (ADITAMA; LEGOWO, 2020).

## 4. Desenvolvimento da *Sprint*

O tempo de duração de uma *sprint* é fixo e, normalmente, varia de 2 a 4 semanas. Em uma *sprint*, o trabalho que deve ser realizado pela equipe está organizado e pronto para ser feito. A equipe trabalha para completá-lo no período de tempo determinado (ADITAMA; LEGOWO, 2020).

## 5. Reunião Diária

Essa reunião não toma mais do que 15 minutos, e é conduzida todos os dias pela equipe (ADITAMA; LEGOWO, 2020). Ela garante que a visão de todos os membros esteja alinhada com os objetivos da *sprint* (HERDIKA; BUDIARDJO, 2020).

## 6. Reuniões de Revisão e Retrospectiva da *Sprint*

A revisão da *sprint* é feita pela demonstração do produto desenvolvido até o momento aos *stakeholders*, para garantir que tudo esteja ocorrendo como definido. A retrospectiva da *sprint* é a discussão, pela equipe, dos aspectos que podem ser melhorados, mantidos ou interrompidos para otimizar o processo de desenvolvimento (HERDIKA; BUDIARDJO, 2020).

Depois das reuniões de revisão e retrospectiva, o processo retorna para o planejamento da *sprint* (ADITAMA; LEGOWO, 2020). Ademais, os papéis usualmente definidos no Scrum são:

- *Product Owner (PO)*: Como responsável pela representação e integração com o cliente, o PO assume a maior responsabilidade na gestão do *Backlog* do Produto (ALQUDAH; RAZALI, 2018);

- *Scrum Master (SM)*: Assume a tarefa de garantir que todas as partes relacionadas entendam o *Scrum* e que o modelo seja empregado de forma eficaz (ALQUDAH; RAZALI, 2018), e
- Equipe de Desenvolvimento: Como responsável pelo desenvolvimento do produto, a equipe de desenvolvimento assume a tarefa de produzir os entregáveis da *sprint* (ALQUDAH; RAZALI, 2018).

O *Product Owner*, o *Scrum Master* e a Equipe de Desenvolvimento formam a equipe clássica do *Scrum*. Espera-se que essa equipe seja auto-organizada, madura e multifuncional, sem depender de pessoas externas (ADITAMA; LEGOWO, 2020).

Além disso, o uso do *Scrum* requer que o tamanho da equipe seja o suficiente para conter sua flexibilidade e, ainda assim, completar uma quantidade considerável de trabalho em uma *sprint*. Em vista disso, o tamanho da equipe de desenvolvimento deve ser de 5 a 11 membros (ALQUDAH; RAZALI, 2018).

### 2.2.2 Projetos Ágeis de Pequeno e Médio Portes

Um projeto pequeno de software deve atingir metas de qualidade tão rigorosas quanto as de projetos grandes, porém com menos membros na equipe (Russ; McGregor, 2000). Segundo (Richardson; Von Wangenheim, 2007), grandes e pequenas empresas enfrentam desafios semelhantes, inerentes à Engenharia de Software, já que empresas menores não são apenas versões em escala reduzida das grandes empresas.

Existem alguns fatores que definem um projeto como pequeno ou médio. Para (LEE S; YONG, 2013), esses fatores são: duração de seis meses ou menos do projeto; a equipe é composta por dez ou menos membros; o projeto envolve um pequeno número de áreas de habilidade; existe um único objetivo e a solução é facilmente alcançada; o escopo e a definição são estreitamente definidos; uma única unidade de negócio é afetada; o projeto possui um único tomador de decisões; soluções automatizadas de fontes externas de projeto não são requeridas; são produzidos resultados diretos com interdependências entre as áreas de habilidade, e o projeto possui financiamento disponível.

Com o objetivo de persistir e crescer no mercado, as pequenas e médias empresas de software precisam de eficiência e de soluções eficazes de Engenharia de Software. Para isso, essas empresas exigem abordagens diferentes devido aos seus modelos de negócios com objetivos específicos, nicho de mercado, tamanho, disponibilidade de recursos (financeiros e humanos), capacidade de gestão e de processo, entre outras coisas. Geralmente, pequenas e médias empresas são extremamente ágeis e flexíveis, porque essa é a vantagem competitiva anunciada (Richardson; Von Wangenheim, 2007).

Com os recursos limitados, empreendimentos menores não têm pessoal suficiente para desenvolver especialidades funcionais que permitem realizar tarefas complexas e nem sempre podem comprar o conhecimento necessário. Portanto, surgem os desafios de elicitar, de organizar e de priorizar os requisitos, pois os membros devem selecionar os requisitos funcionais e não funcionais mais importantes de um conjunto muito grande de requisitos (Richardson; Von Wangenheim, 2007). Para esses autores, é fundamental que a equipe tenha o suporte necessário de ferramentas para o desenvolvimento de um software. Esses aparatos tecnológicos podem auxiliar, entre outras coisas, na gestão de requisitos para empresas de pequeno e médio portes.

## 2.3 Engenharia de Requisitos

A Engenharia de Requisitos é responsável por descrever as atividades relacionadas à produção e à gerência dos requisitos do sistema. A produção de requisitos possui algumas atividades, as quais são: o levantamento; o registro; a validação, e a verificação dos requisitos. Já a gerência possui as seguintes atividades: o controle de mudanças; a rastreabilidade; a gerência de qualidade, e a gerência de configuração dos requisitos (ÁVILA A. L.; SPÍNOLA, 2017). Os mesmos autores definem a Engenharia de Requisitos como sendo o processo base para as demais tarefas associadas ao desenvolvimento de software. Se esse processo não for feito de forma adequada e eficiente, de maneira que os erros possam ser corrigidos nessa etapa, o custo de manutenção do software aumenta significativamente.

A seguir estão os detalhes das atividades de produção e de gerência de requisitos propostos pelos autores (ÁVILA A. L.; SPÍNOLA, 2017).

Após o levantamento e a negociação dos requisitos com o cliente, os requisitos começam a ser documentados para servir de base para o restante do desenvolvimento do software. Compilar toda a informação necessária, adquirida pela elicitação de requisitos, não é uma tarefa fácil, pois o documento que contém a especificação dos requisitos precisa ter um nível apropriado de detalhes, para que qualquer *stakeholder* entenda o que está descrito.

A próxima etapa a ser executada é a de verificação, de forma que não haja inconsistências entre o que foi solicitado e o que está especificado. Feita a verificação, a validação é necessária para que seja obtido o aceite do cliente em relação a uma determinada funcionalidade. Estas duas etapas são de extrema importância, pois o quanto antes se conseguir identificar os erros, menor o custo de manutenção e complexidade das correções.

O desenvolvimento ágil de software tem a capacidade de acomodar mudanças de requisitos durante todo o desenvolvimento do projeto (ENGUM; RACHEVA; DANEVA, 2009). Devido à volatilidade dos requisitos, ÁVILA A. L.; SPÍNOLA (2017) evidenciam a necessidade de gerência de requisitos, de forma que as alterações sejam feitas de maneira

controlada e ordenada, a fim de cumprir o prazo e o custo do projeto.

A primeira atividade da gerência de requisitos é o controle de mudanças, que precisa ter o controle do requisito acordado no início (original), o requisito introduzido e o requisito descartado. A segunda atividade é a gerência de configuração, que existe no intuito de definir critérios que permitam realizar as modificações mantendo a consistência e a integridade do software com as especificações.

A próxima atividade é a rastreabilidade de requisitos, que significa acompanhar o requisito em todo o seu ciclo de vida. Nesta atividade, encontra-se uma dificuldade que é o grande volume de informações geradas, necessitando do auxílio de ferramentas adequadas. Um artefato que é comumente utilizado é a matriz de rastreabilidade, que tem como objetivo mapear os rastros dos requisitos descritos na especificação. Para os autores (ÁVILA A. L.; SPÍNOLA, 2017), existem dois tipos de rastros de requisitos, que são:

- Pré-Rastreabilidade - os rastros que fundamentaram a criação do requisito, e
- Pós-Rastreabilidade - os rastros que se formaram a partir do requisito criado.

A última atividade é a gerência de qualidade dos requisitos, que deve atender a alguns fatores, que são: os requisitos reais do usuário devem coincidir com os requisitos identificados; os requisitos devem ser interpretados pelos *stakeholders* de uma única maneira; o requisito é dito completo quando descreve todas as demandas de interesse dos usuários; o requisito deve ser consistente, não podendo entrar em conflito com os outros requisitos do sistema; o requisito deve ser verificável de forma que indique que o sistema cumpre o requisito, e o requisito deve ser modificável quando as alterações podem ser realizadas de forma simples e consistente com os demais requisitos.

### 2.3.1 Gestão de Requisitos no Scrum

No processo de desenvolvimento ágil, as atividades de Engenharia de Requisitos são realizadas continuamente ao longo do desenvolvimento em concordância com a abordagem iterativa (RUBIN HILLEL; RUBIN, 2011).

Para melhor compreensão dos requisitos no *Scrum*, é necessário haver níveis de granularidade destes requisitos, que são Épicas, *Features*, Histórias de Usuário e Tarefas. Caroli e Aguiar (2021) explicam que o Épico é uma explicação textual de um trabalho mais abrangente e que seja relacionado, que pode ser realizado em dias, semanas ou meses. Uma *Feature* é a descrição de uma ação ou interação de um usuário com o produto. A História de Usuário é uma explicação textual de uma pequena parte da *Feature* desejada por um usuário final, que pode ser realizada em horas ou poucos dias. Já as tarefas são pedaços ainda menores sobre o que deve ser feito em cada História de Usuário. Essas tarefas contemplam detalhes técnicos para a equipe de desenvolvimento.

Santamaria (2007) diz que cada história de usuário pode possuir vários campos, incluindo: 1) o ator que deseja aquele artefato; 2) o objetivo ou tarefa que precisa ser realizada; 3) a explicação de "por que" ou "para que" o artefato é necessário; 4) o valor associado, e 5) a prioridade correspondente. Para facilitar a visualização e o gerenciamento das histórias de usuário, os modelos ágeis utilizam o quadro *Kanban* (HERDIKA; BUDIARDJO, 2020).

Para (COHN, 2004), as histórias de usuário são flexíveis, de maneira que é fácil alterar os seus detalhes, os quais podem ser negociados entre o cliente e o time de desenvolvimento. Portanto, durante a escrita da história, é possível anexar ao *card* dessa história tudo o que for pertinente para a sua realização. Apesar da flexibilidade na escrita da história, Herdika e Budiardjo (2020) dizem que a história de usuário descreve de forma insuficiente o requisito do produto. Esse fato acontece, pois o modelo ágil enfatiza a comunicação em tempo real, de preferência face a face, ao invés de muitos documentos escritos e processos rígidos (SANTAMARIA, 2007).

Tal confiança na comunicação e na colaboração, defendida pelos modelos ágeis, pode se tornar crítica. Isso porque é difícil fazer com que os *stakeholders* e as equipes de desenvolvimento se comuniquem em uma linguagem comum e possuam um entendimento compartilhado a respeito da solução, o que pode gerar desafios para o desenvolvimento da especificação de requisitos (HERDIKA; BUDIARDJO, 2020). Ao mesmo tempo que benefícios como maior flexibilidade, maior eficiência de planejamento e melhora na comunicação são percebidos com a utilização dos modelos ágeis, várias áreas de deficiência podem ser identificadas na gestão de requisitos em tais modelos (KASAULI et al., 2021).

O Quadro 16, exposto no Apêndice B, apresenta os principais problemas/desafios encontrados na literatura para a gestão de requisitos a partir da utilização de modelos ágeis como o *Scrum*. Para cada problema/desafio identificado, foram endereçadas as práticas encontradas na literatura que podem representar soluções em potencial para enfrentar tais problemas/desafios.

Herdika e Budiardjo (2020) afirmam que os requisitos precisos de uma simples história de usuário possibilitam o desenvolvimento do software correto. O problema reside em como escrever os requisitos corretos em histórias de usuário. Segundo (KASAULI et al., 2021), não existe uma resolução universal para a utilização da Engenharia de Requisitos em abordagens ágeis. No entanto, a adoção de práticas que visam mitigar os problemas enfrentados pode aumentar as chances de sucesso no desenvolvimento da especificação de requisitos (HERDIKA; BUDIARDJO, 2020).

No Scrum, a lista ordenada de tudo o que pode ser necessário e a única fonte de requisitos para qualquer mudança a ser feita no produto configura-se no *Backlog* do Produto. Nele, estão concentradas todas as histórias de usuário ou quaisquer representações das necessidades do produto (HVATUM; WIRFS-BROCK, 2015).



A equipe *Scrum* deve organizar a preparação do *Backlog*, especificando o item do *Backlog* do Produto em épicos específicos, e dividir o item grande em histórias de usuário menores, afim de priorizá-los. Organizar a história de usuário em épicos ajudará a equipe a rastrear quais recursos ainda não foram concluídos. Cada história de usuário estimada deve ter critérios de aceitação definidos. No planejamento da *sprint*, a equipe *Scrum* dividirá a história de usuário priorizada em tarefas que serão desenvolvidas na *sprint*, e o desenvolvedor da equipe irá estimar a tarefa. Após a estimativa feita pelos desenvolvedores, as tarefas são colocadas no *Backlog* da *Sprint*. Os recursos desenvolvidos são apresentados aos *stakeholders* e recebem *feedback*. Este *feedback* é importante para priorizar novamente o *Backlog* do Produto (HERDIKA; BUDIARDJO, 2020).

Uma vez que o *Backlog* do Produto representa a única fonte para conduzir o desenvolvimento do sistema, é importante que ele seja capaz de oferecer uma imagem completa dos requisitos do produto. Isso implica que o *Backlog* seja desenvolvido de forma a mostrar as relações entre os requisitos de negócio e as histórias de usuário; explicitar as dependências entre requisitos de mesmo nível de granularidade; incorporar requisitos técnicos e de qualidade; manter os resultados das discussões com os *stakeholders*; mostrar o estado do desenvolvimento e dos testes, e incluir defeitos, que também representam itens de trabalho. Dessa forma, a estrutura do *Backlog* deve ser planejada com cuidado, evoluída e mantida para que ele possa oferecer o nível apropriado de clareza necessária para uma correta implementação do sistema (HVATUM; WIRFS-BROCK, 2015).

### 2.3.2 Ferramenta de Apoio Tecnológico à Gestão do *Backlog* do Produto

O fluxo de atividades do *Scrum* é contínuo e alimenta o *Backlog* com o conteúdo derivado da elicitação de requisitos. Os resultados da elicitação não são as únicas entradas para o processo de definição de requisitos e criação do *Backlog*. Assim que o produto começa a ganhar forma e maturidade, o *Backlog* será influenciado pela experiência do usuário, arquitetura do sistema, estratégia de testes, *bugs* encontrados, entre outros aspectos. A evolução contínua do *Backlog* requer esforço e precisa ser feita com cautela (HVATUM; WIRFS-BROCK, 2015).

Em vista disso, um procedimento efetivo faz-se necessário para ajudar tanto o PO quanto o gerente de projeto a manter o rastro dos requisitos e das mudanças no *Backlog*. Para (ENGUM; RACHEVA; DANEVA, 2009), o *Backlog* deve ter uma descrição de tarefas de forma clara em uma ferramenta de rastreamento. Uma ferramenta de apoio tecnológico a esse propósito pode fornecer uma série de benefícios, incluindo: (1) disponibilização de uma única interface tanto para o *Backlog* do Produto quanto para o *Backlog* da *Sprint*; (2) aumento considerável da quantidade de informação explícita; (3) maior facilidade na identificação das dependências entre membros das equipes; (4) viabilização da análise de riscos e da comunicação, e (5) melhor compartilhamento de experiência entre os membros

da equipe.

Segundo (HVATUM; WIRFS-BROCK, 2015) e (ENGUM; RACHEVA; DANEVA, 2009), os seguintes critérios devem ser alcançados por uma ferramenta tecnológica que confira apoio à gestão de um bom *Backlog* do Produto:

- Possibilitar uma visão geral do produto;
- Possibilitar uma visão técnica do produto;
- Apoiar navegação;
- Apoiar planejamento;
- Apoiar respostas sobre o progresso;
- Oferecer capacidade de associar requisitos em coleções;
- Apoiar especificação detalhada das tarefas;
- Capturar critérios de qualidade para requisitos individuais;
- Capturar critérios de qualidade para as coleções de requisitos;
- Possibilitar que qualquer parte interessada possa adicionar um item ao *Backlog* do Produto;
- Possibilitar de forma fácil a pesquisa, o filtro e a classificação dos problemas, para que estes possam ser encontrados e comparados;
- Possibilitar a priorização dos itens do *Backlog* do Produto;
- Possibilitar a estimativa dos itens do *Backlog* do Produto;
- Fornecer uma única e fácil interface que seja acessível e transparente para o *Backlog* do Produto e para o *Backlog* da *Sprint*;
- Possibilitar de forma fácil a identificação de dependência entre tarefas, e
- Possibilitar a comunicação entre os membros da equipe.

## 2.4 Resumo do Capítulo

Neste capítulo, foram apresentados os principais conceitos necessários ao entendimento do contexto em que se insere este Trabalho de Conclusão de Curso: a construção de uma ferramenta que oferece apoio à construção e à manutenção do *Backlog* do Produto, principal documento de requisitos do Scrum, que está presente também em inúmeros modelos de desenvolvimento ágil.

Para a apresentação desses conceitos, fez-se necessária uma breve explicação sobre a Engenharia de Software, bem como sobre os modelos ágeis, detalhando o Scrum, suas práticas, papéis e artefatos, a fim de dar uma visão geral a respeito do modelo ágil em foco neste trabalho. Definições a respeito de projetos ágeis de pequeno e médio portes também foram apresentadas, com o objetivo de explanar os pontos que dizem respeito ao público-alvo definido para o escopo de atuação deste trabalho.

Em seguida, foram apresentados os conceitos relacionados à Engenharia de Requisitos e à forma como se dá a gestão de requisitos no Scrum, incluindo a identificação dos principais problemas/desafios enfrentados nessa etapa, com o endereçamento das soluções em potencial, encontradas na literatura, que podem auxiliar na mitigação dos problemas/desafios identificados.

Por fim, foram dispostos os principais pontos identificados na literatura a respeito do desenvolvimento de uma ferramenta que apoie a construção e a manutenção do *Backlog* do Produto, encerrando, assim, o capítulo com a bagagem teórica utilizada para viabilizar este trabalho.



## 3 Suporte Tecnológico

Este capítulo tem como objetivo apresentar e justificar a utilização de ferramentas para o desenvolvimento deste Trabalho de Conclusão de Curso. Serão especificadas as tecnologias utilizadas na construção da ferramenta de apoio à criação e à manutenção do *Backlog* do Produto, bem como as tecnologias necessárias para ambos, desenvolvimento da pesquisa e comunicação entre as autoras do trabalho. Por fim, tem-se o resumo do capítulo.

### 3.1 Tecnologias de Apoio à Aplicação

#### 3.1.1 Git 2.31.0

O *Git* é um sistema de controle de versão gratuito e de código aberto, desenhado para lidar com projetos tanto de grande quanto de pequena escala (GIT, 2021). Ele foi utilizado para fazer o versionamento do código fonte da aplicação.

#### 3.1.2 GitHub Free Version

O *GitHub* é um serviço baseado em nuvem que hospeda o sistema de controle de versão do *Git*. Ele permite a colaboração entre desenvolvedores, por meio da modificação de projetos compartilhados e da manutenção do registro detalhado de progresso (L., 2021). A ferramenta foi utilizada para hospedar o código fonte do projeto, além de melhorar o fluxo de trabalho e a colaboração entre as autoras.

#### 3.1.3 Docker 20.10

O *Docker* é uma plataforma aberta para criação, execução e publicação de *containers*. Um *container* é a forma de empacotar uma aplicação e suas dependências de forma padronizada (GEEKHUNTER, 2019). Ela foi utilizada na prova de conceito (Seção 5.4), para validação da configuração dos ambientes de desenvolvimento da aplicação.

#### 3.1.4 Docker Compose 1.28.5

O *Docker Compose* é uma ferramenta para criação e execução de aplicações que utilizam vários *containers Docker*. Ele permite, de forma simples, a configuração e a realização dos serviços necessários à execução da aplicação (DOCKER, 2021). A orquestração dos *containers* criados com o *Docker*, utilizados para a configuração dos ambientes de de-

desenvolvimento da aplicação na prova de conceito (Seção 5.4), deu-se com o uso do *Docker Compose*.

### 3.1.5 JavaScript ECMAScript 2020

*JavaScript* (JS) é uma linguagem dinâmica, interpretada e multiparadigma, que suporta estilos orientados a objetos, imperativos e declarativos. Essa linguagem é comumente utilizada para páginas da Web, ou seja, para o desenvolvimento *front-end*. Porém, ambientes que não dependem do navegador também a utilizam, como é o exemplo do NodeJS (DOCS, 2021). Para o desenvolvimento da aplicação, a linguagem foi utilizada a partir do emprego do *NodeJS* no *back-end*.

### 3.1.6 TypeScript 4.2.3

*TypeScript* é uma linguagem de programação fortemente tipada que se baseia em *JavaScript* (TYPESCRIPT, 2021). Ela foi utilizada no *front-end* da aplicação, como linguagem empregada para o uso do *ReactJS*, juntamente com o *NextJS*.

### 3.1.7 NodeJS 14.16.0

Como um ambiente de execução de código *JavaScript*, assíncrono e orientado a eventos, o *NodeJS* é uma plataforma projetada para o desenvolvimento de aplicações escaláveis de rede (NODEJS, 2021). Arquitetura, flexibilidade e baixo custo foram os fatores que determinaram a utilização do Node para a implementação da API que faz a comunicação entre as camadas da aplicação.

### 3.1.8 Express 4.17.1

O *Express* é um *framework* para aplicações desenvolvidas em *NodeJS*. É um *framework* minimalista e flexível, que fornece um conjunto robusto de recursos para aplicativos *web* e *mobile* (EXPRESS, 2021). O *Express* foi utilizado no desenvolvimento do *back-end* da aplicação em razão de seu bom escopo de métodos utilitários *HTTP* e *middleware*, e por conta da facilidade de criação de APIs que ele fornece.

### 3.1.9 Firebase Node.js SDKs

O *Firebase* fornece as ferramentas para desenvolver aplicações *web* e *mobile* de alta qualidade. Ele oferece diversos serviços, como: a autenticação e armazenamento de usuários, via e-mail ou redes sociais, utilizando o *Firebase Authentication*; o armazenamento de dados, por meio do *Cloud Firestore*, que é um banco de dados em nuvem *NoSQL* flexível e escalável; e o armazenamento de arquivos por meio do *Cloud Storage*. Além desses

serviços, o *Firebase* oferece diversos outros recursos para os seus usuários (FIREBASE, 2021c). O *Firebase* foi escolhido devido à grande quantidade de possibilidades oferecidas pela ferramenta, além da oportunidade de escalar a aplicação utilizando serviços de alta qualidade.

### 3.1.10 ReactJS 17.0.1

O *React* é uma biblioteca *JavaScript* usada para construir interfaces de usuário interativas. Ele utiliza visualizações declarativas, que torna o código mais previsível e fácil de depurar. Os seus componentes são encapsulados, os quais gerenciam seu próprio estado e auxiliam na criação de interfaces de usuário complexas (REACT, 2021). O *React* tem se tornado muito popular entre as grandes empresas atuais, e foi escolhido por conta do seu bom desempenho com o virtual *DOM* (*Document Object Model*) e pela facilidade de criação de componentes que ele fornece.

### 3.1.11 NextJS 10.0.9

O *Next* é um *framework* para *React*, que adiciona funcionalidades, como: renderização estática pelo lado do servidor e o tratamento de rotas. Ele utiliza o conceito de *Server-Side-Rendering*, ou seja, entregar as páginas prontas para o *Browser* a partir da renderização no servidor (NEXT, 2021). Esse *framework* foi escolhido porque tem a capacidade de tornar a aplicação *React* mais performática.

### 3.1.12 Tailwind CSS 2.1.4

O *Tailwind CSS* é um *framework* CSS, com foco na utilidade, que utiliza classes que podem ser compostas para construir qualquer *design*, diretamente do HTML (CSS, 2021). A ferramenta foi utilizada para ajudar na construção das telas da aplicação, facilitando a codificação dos componentes e deixando o visual da ferramenta mais bonito e organizado.

### 3.1.13 Heroku Heroku-20

O *Heroku* é uma plataforma em nuvem que permite construir, entregar, monitorar e escalar aplicações, oferecendo serviços de infraestrutura para aplicações construídas em diversas tecnologias (HEROKU, 2021). A pilha de tecnologias do *Heroku-20* é baseada no *Ubuntu* 20.04, e foi utilizada para dar apoio à implantação e entrega dos módulos do sistema, de forma a descarregar a preocupação acerca dos problemas de infraestrutura que podem surgir durante esse processo.

### 3.1.14 Vercel Hobby Version

*Vercel* é uma plataforma de hospedagem de aplicações *front-end*. Ela permite a implantação para *frameworks* e *sites* estáticos sem a necessidade de configuração (VERCEL, 2021). Por conta da facilidade e da escalabilidade fornecidas pela ferramenta, a implantação do *front-end* da aplicação foi feita por meio dela.

### 3.1.15 ZenHub Free Version

O *ZenHub* é uma ferramenta de gerenciamento de projetos ágeis integrada ao *GitHub* (ZENHUB, 2021). Sua utilização se deu pela facilidade em transformar issues dos repositórios do *GitHub* em épicos e histórias de usuário, além de tornar o planejamento mais fácil e rápido, concentrando as atividades de desenvolvimento em uma única plataforma.

## 3.2 Tecnologias de Apoio à Pesquisa e à Comunicação

### 3.2.1 Drive

O *Drive* é um serviço da *Google* que permite armazenar e compartilhar arquivos em forma de documentos, planilhas ou apresentações (GOOGLE, 2021). Essa ferramenta foi utilizada para a realização de planilhas e documentos colaborativos, que contribuem para a organização das tarefas e o compartilhamento de conhecimento entre as desenvolvedoras do projeto.

### 3.2.2 Zotero 5.0

O *Zotero* é uma ferramenta gratuita que auxilia na coleta, organização, citação e compartilhamento do material de pesquisa. A ferramenta permite o trabalho colaborativo, além de sincronizar os dados armazenados em diferentes dispositivos (ZOTERO, 2021). O *Zotero* foi escolhido como ferramenta de apoio à pesquisa e à revisão da literatura que fundamentou este trabalho, porque facilita a gestão do material, assim como sua análise.

### 3.2.3 LaTeX LaTeX3

O *LaTeX* é uma composição tipográfica de alta qualidade, frequentemente usada para produção de documentos técnicos ou científicos (LATEX, 2021). O *LaTeX* foi utilizado neste trabalho porque auxilia na formatação do documento, viabilizando centrar esforços no conteúdo escrito.



### 3.2.4 Overleaf Community Edition

O *Overleaf* é um editor *online* LaTeX, colaborativo em tempo real e com saída compilada e produzida automaticamente (OVERLEAF, 2021). Essa ferramenta foi utilizada devido à facilidade de edição e de produção de texto científico. Além de ser fácil compartilhar as atualizações do documento, pois é possível fazer edições simultâneas e conseguir gerar o arquivo compilado em tempo real.

### 3.2.5 WhatsApp 2.21.50

O *WhatsApp* é um aplicativo, disponível nas versões *mobile* e *web*, para o compartilhamento instantâneo de mensagens. Além de suportar o envio de mensagens de texto, de voz, de mídia e documentos, o aplicativo possibilita a realização de chamadas de voz e de vídeo (WHATSAPP, 2021). Ele foi utilizado como principal ferramenta de comunicação, na qual foram discutidas as ideias, sendo armazenado o histórico de mensagens que dizem respeito à execução deste trabalho.

### 3.2.6 Hangouts

O *Hangouts* é uma ferramenta de comunicação da Google que permite a troca de mensagens *online*, a efetuação de ligações telefônicas e a realização de videochamadas com até 25 pessoas (LOUBAK, 2020). Ela foi utilizada como principal ferramenta para realização de reuniões remotas entre as desenvolvedoras do projeto.

## 3.3 Resumo do Capítulo

Neste capítulo, foram apresentadas as principais tecnologias definidas para auxiliar o desenvolvimento da ferramenta de apoio à criação e à manutenção do *Backlog* do Produto, resultante deste Trabalho de Conclusão de Curso, bem como as ferramentas que foram utilizadas para dar apoio à pesquisa e à comunicação necessárias para a realização do mesmo.

As ferramentas de apoio ao desenvolvimento foram dispostas de maneira a descrever como foram utilizadas para auxiliar desde o gerenciamento do código fonte até o desenvolvimento do sistema propriamente dito, envolvendo as camadas de processamento de dados (*back-end*) e de interface com o usuário (*front-end*).

Além disso, as ferramentas de apoio à pesquisa foram descritas, de forma a indicar como foram utilizadas para auxiliar o processo de pesquisa, referência e escrita desta monografia. Em adição, as principais ferramentas definidas para facilitar a comunicação entre as partes desenvolvedoras foram descritas, finalizando, assim, a apresentação do suporte tecnológico necessário em todo o processo de realização do trabalho.



## 4 Metodologia

Este capítulo tem como objetivo apresentar quais foram as escolhas metodológicas feitas para o desenvolvimento deste Trabalho de Conclusão de Curso. Para isso, são apresentadas as metodologias escolhidas para cada fase do trabalho. Primeiro, têm-se as classificações da pesquisa quanto à abordagem, à natureza, aos objetivos e quais são os procedimentos executados para a realização desta pesquisa. Em seguida, são definidas as metodologias utilizadas, respectivamente, no desenvolvimento da ferramenta e na análise de resultados. Após as definições, são acordados os fluxos de atividades e os cronogramas para ambos, Trabalho de Conclusão de Curso 1 e 2. Por fim, tem-se o resumo do capítulo.

### 4.1 Metodologia de Pesquisa

Para a investigação científica, são necessários procedimentos intelectuais e técnicos para alcançar o objetivo esperado. Esses procedimentos são chamados de métodos científicos, que são adotados no processo de pesquisa (PRODANOV C. C. ; FREITAS, 2013). A proposta desta seção é classificar a pesquisa realizada quanto à abordagem, à natureza, aos objetivos e aos procedimentos, como exposto por Gerhardt T. E.; Silveira (2009).

#### 4.1.1 Abordagem da Pesquisa

A abordagem de pesquisa deste trabalho pode ser considerada como **pesquisa qualitativa**, a qual tem preocupação com aspectos da realidade que não podem ser quantificados, ou seja, não requer o uso de métodos e técnicas estatísticas (GERHARDT T. E.; SILVEIRA, 2009).

Ao se falar da construção do *Backlog* do Produto no contexto de modelos ágeis, é necessário levar em consideração, além do contexto em si, o caráter interativo entre todos os papéis desempenhados dentro do modelo ágil *Scrum*, os quais são subjetivos. Deve-se considerar, também, o caráter descritivo dos problemas e das soluções ao construir e manter o *Backlog* do Produto. Todas essas observâncias são relevantes para a construção da ferramenta, que tem como objetivo a mitigação de várias das dificuldades encontradas, ao longo do desenvolvimento de software, em relação ao *Backlog* do Produto.

#### 4.1.2 Natureza da Pesquisa

Quanto à natureza da pesquisa, este trabalho pode ser definido como uma **pesquisa aplicada**. Para (GERHARDT T. E.; SILVEIRA, 2009), essa natureza é conduzida à resolução de problemas específicos. Tal classificação entra em concordância com o prin-

cipal objetivo deste trabalho, o qual está focado em resolver problemas relacionados à construção e à manutenção do *Backlog* do Produto, dentro do contexto de projetos que utilizam o *Scrum* como metodologia de desenvolvimento.

### 4.1.3 Objetivos da Pesquisa

Quanto ao objetivo da pesquisa, este trabalho pode ser definido como **pesquisa exploratória**. Segundo (GIL, 2008), as pesquisas exploratórias permitem que os investigadores se aproximem mais da proposta escolhida e têm como finalidade a formulação de problemas ou hipóteses. Gil (2008) recomenda a pesquisa exploratória quando o tema escolhido é genérico e necessita de mais investigação, podendo utilizar a revisão da literatura para esclarecimentos.

Os objetivos da pesquisa exploratória encaixam-se bem com o escopo deste trabalho, devido à necessidade de investigação de problemas e soluções para a construção e a manutenção do *Backlog* do Produto. Além do caráter subjetivo, que depende do contexto e da interação dos indivíduos, o que demanda bastante investigação e pontos de vista diferentes.

### 4.1.4 Procedimentos

Quanto aos procedimentos utilizados para realização deste trabalho, foram definidos quatro procedimentos para etapas diferentes. Para a primeira etapa, foram utilizados os procedimentos de **Revisão Sistemática da Literatura**, *Snow Ball* e **Levantamento Bibliográfico**. Esses procedimentos foram necessários para a formação de um referencial bibliográfico relacionado ao tema da pesquisa, com o objetivo de formar um forte embasamento sobre o assunto. A Revisão Sistemática da Literatura foi feita com base no protocolo de Kitchenham (2004), e está descrita com mais detalhes no Apêndice A.

Para a segunda etapa, na fase de análise dos resultados, foi definido o procedimento de **Pesquisa-Ação**, no qual os pesquisadores desempenham um papel ativo na realidade observada, na solução dos problemas encontrados, no acompanhamento e na avaliação das ações (PRODANOV C. C. ; FREITAS, 2013). Em vista disso, esse procedimento contempla a análise de resultados deste projeto com o objetivo de corrigir e aperfeiçoar a ferramenta para a resolução de problemas encontrados na construção e na manutenção de um *Backlog* do Produto. Os detalhes desse procedimento estão na Seção 4.3.

### 4.1.5 Classificação da Pesquisa

A classificação da pesquisa quanto à abordagem, à natureza, aos objetivos e aos procedimentos está descrita no Quadro 1.

Quadro 1 – Classificação da Pesquisa

<b>Abordagem</b>	Qualitativa
<b>Natureza</b>	Aplicada
<b>Objetivo</b>	Exploratória
<b>Procedimentos</b>	Revisão Sistemática, <i>Snow Ball</i> , Levantamento Bibliográfico e Pesquisa-Ação

Fonte: autoria própria

## 4.2 Metodologia de Desenvolvimento da Proposta

A fim de estabelecer o alicerce para um processo completo de Engenharia de Software, existem as chamadas metodologias (ou *frameworks*) de processo, que, por meio da identificação de um pequeno número de atividades estruturais aplicáveis a todos os projetos de software, englobam um conjunto de atividades de apoio aplicáveis em todo o processo de software (PRESSMAN; MAXIM, 2016). Em vista disso, esta seção tem como objetivo a apresentação da metodologia que guiou as atividades de desenvolvimento da ferramenta construída neste Trabalho de Conclusão de Curso.

É importante destacar que a escolha das atividades, práticas e artefatos, que forneceram apoio ao processo de desenvolvimento, se deu com a utilização dos modelos ágeis *Scrum*, *Kanban* e *XP (Extreme Programming)*. Sendo assim, serão apresentados, primeiramente, os modelos originais, dos quais oriundam as atividades, práticas e artefatos selecionados, e, posteriormente, a metodologia híbrida resultante.

### 4.2.1 *Scrum*

O *Scrum* é um modelo ágil com foco na gestão do desenvolvimento de software. Ele pode ser usado para orientar o gerenciamento de tempo e de escopo do projeto. Também serve de guia para conduzir o desenvolvimento, regulando vários componentes, como atividades, papéis e artefatos (HERDIKA; BUDIARDJO, 2020).

As principais atividades, práticas e artefatos do *Scrum*, segundo (ALQUDAH; RAZALI, 2018), estão definidas a seguir:

- **Backlog do Produto:** uma lista que contém todas as funcionalidades e requisitos de implementação do sistema;
- **Backlog da Sprint:** os itens do *Backlog* do Produto selecionados para serem implementados durante uma *sprint* ou iteração;
- **Gráfico de Burndown da Sprint:** o gráfico que representa o restante do trabalho da *sprint*. Ele é atualizado diariamente, e mede o progresso em relação aos objetivos da iteração;

- **Estimativa de Esforço:** todos os membros da equipe utilizam o nível de dificuldade e esforço para estimar o trabalho necessário para a realização dos itens do *Backlog*. Geralmente, são usados os números da série de Fibonacci para representar o esforço atribuído a cada item;
- ***Sprint*:** a *Sprint* é o centro do *Scrum*, e compreende um intervalo de tempo com duração entre uma semana e um mês. Durante esse tempo, espera-se a construção de um produto acabado, funcional e entregável.
- **Reunião de Planejamento da *Sprint*:** A reunião de Planejamento é, principalmente, para decidir acerca dos itens do *Backlog* do Produto que podem ser realizados dentro da *sprint* atual, e como as tarefas atribuídas devem ser realizadas até o final do período da *sprint*;
- **Reuniões Diárias:** as reuniões diárias devem ter duração de até 15 minutos e contribuem para a análise do progresso da *sprint*, por meio do compartilhamento de *feedback* dos membros da equipe de desenvolvimento acerca do andamento das atividades atribuídas a cada um;
- **Reunião de Revisão da *Sprint*:** os *stakeholders* reúnem-se para a revisão da *sprint* que passou, com as funcionalidades da nova entrega sendo apresentadas, e
- **Reunião de Retrospectiva da *Sprint*:** os membros da equipe discutem sobre os problemas enfrentados ao longo da *sprint*, e como esses problemas podem ser abordados nas próximas iterações.

### 4.2.2 *Kanban*

O *Kanban* é um modelo ágil com foco na melhoria do processo de desenvolvimento de software. Ele ajuda a equipe de desenvolvimento a ter uma melhor visão do projeto em andamento, utilizando o quadro *Kanban* como ferramenta de visualização do estado atual do desenvolvimento (HERDIKA; BUDIARDJO, 2020).

As principais práticas do *Kanban*, segundo (ALQUDAH; RAZALI, 2018), estão definidas a seguir:

- **Visualização do Fluxo de Trabalho:** fazer com que o trabalho necessário fique visível para todos os membros da equipe, considerando que o trabalho não visível pode levar a riscos para a implementação do projeto. O quadro *Kanban* fornece clara visualização do fluxo de trabalho;
- **Limite de Trabalho em Andamento:** a redução da quantidade de tarefas ocorrendo em paralelo provoca a redução do tempo de entrega dos itens de trabalho;

- **Gerenciamento e Medição do Fluxo:** todos os estados dentro do fluxo de trabalho são supervisionados. Tal supervisão gera movimentos suaves e eficientes, os quais geram valor apropriado, o que reduz os riscos e previne custos de adiamento;
- **Políticas de Trabalho Explícitas:** dadas as incontáveis formas de trabalho, a maioria das empresas possui políticas implícitas. Por outro lado, para seu uso constante, as equipes *Kanban* apresentam explicitamente as políticas de trabalho;
- **Implementação de *Feedback*:** o *Kanban* requer ciclos de *feedback* para funcionar. Conseqüentemente, são empregadas reuniões "em pé", além de uma revisão da entrega do serviço. Isso permite a comparação entre os resultados planejados e os resultados reais, e a execução dos ajustes necessários, e
- **Melhoria Contínua:** implica uma consciência esclarecida e universal das teorias que se relacionam ao trabalho, fluxo de trabalho, processo, bem como risco. Tal consciência auxilia os membros no desenvolvimento de um discernimento mútuo sobre os problemas, ao mesmo tempo que propõe melhorias solidamente acordadas.

#### 4.2.3 XP - *Extreme Programming*

O XP é um modelo ágil que ajuda o time de desenvolvimento a gerenciar as mudanças de código no desenvolvimento de software. As principais práticas do XP, segundo (HERDIKA; BUDIARDJO, 2020), assim como sua explicação para apoiar as mudanças no desenvolvimento realizado em cada iteração, estão definidas a seguir:

- **Desenvolvimento Orientado a Testes:** o desenvolvimento dos testes acontece antes da codificação dos componentes. Isso ajuda o desenvolvedor a construir um padrão de codificação que todos os membros entendam, e adotem os testes automatizados para cada mudança;
- **Programação em Pares:** a programação em duplas reduz a probabilidade de erros no código;
- **Integração Contínua:** cada nova parte desenvolvida é integrada ao código de maneira que haja a garantia de que as mudanças não interferem no que já estava desenvolvido;
- **Planejamento:** o planejamento ocorre com a colaboração entre os *stakeholders* na priorização de histórias de usuário e na definição do planejamento da *Release*;
- **Metáforas:** o uso de metáforas na definição do trabalho de uma iteração implica simplicidade na construção das histórias de usuário;

- **Releases Curtas:** dentro da iteração, os resultados validados são implantados com frequência, mas de forma reduzida;
- **Espaço de Trabalho Aberto:** O processo de desenvolvimento envolve o cliente e todos os membros da equipe no local para facilitar *feedbacks* e avaliações mais naturais, e
- **Refatoração:** é o processo de mudança no código de acordo com *feedbacks* ou mudança de requisitos. Todos na equipe de desenvolvimento devem poder fazer as mudanças porque o XP incentiva a propriedade coletiva.

#### 4.2.4 Metodologia Híbrida

Uma metodologia híbrida guiou as atividades de desenvolvimento da ferramenta construída neste Trabalho de Conclusão de Curso. Tal metodologia combina as atividades, práticas e artefatos dos modelos ágeis *Scrum*, *Kanban* e XP. A seguir, é apresentada a relação de atividades, práticas e artefatos selecionados a partir dos modelos citados para a formação da metodologia híbrida de desenvolvimento.

As atividades, práticas e artefatos selecionados a partir do *Scrum* foram:

- *Backlog* do Produto;
- *Backlog* da *Sprint*;
- Estimativa de Esforço;
- *Sprint*, com duração de duas semanas;
- Reunião de Planejamento da *Sprint*;
- Reunião de Revisão da *Sprint*, e
- Reunião de Retrospectiva da *Sprint*.

As práticas selecionadas a partir do *Kanban* foram:

- Visualização do Fluxo de Trabalho;
- Implementação de *Feedback*, e
- Melhoria Contínua.

As práticas selecionadas a partir do XP foram:

- Programação em Pares;



- Integração Contínua;
- *Releases* Curtas, e
- Refatoração.

É bom destacar que a equipe responsável pelo desenvolvimento e gerenciamento do sistema é formada pelas autoras desta monografia.

### 4.3 Metodologia de Análise

Para a análise de resultados e o aperfeiçoamento deste trabalho, foi utilizada a metodologia de **Pesquisa-Ação**. Essa metodologia requer uma participação planejada do pesquisador no problema a ser investigado. O papel do pesquisador passa a ser participativo e não somente observador (GERHARDT T. E.; SILVEIRA, 2009). A Pesquisa-Ação tem como objetivo atender dois propósitos: o prático e o do conhecimento. Nesse sentido, o prático tem a contribuição na solução do problema. Já o conhecimento é o que se adquire a partir da solução encontrada (THIOLLENT M. ; SILVA, 2007).

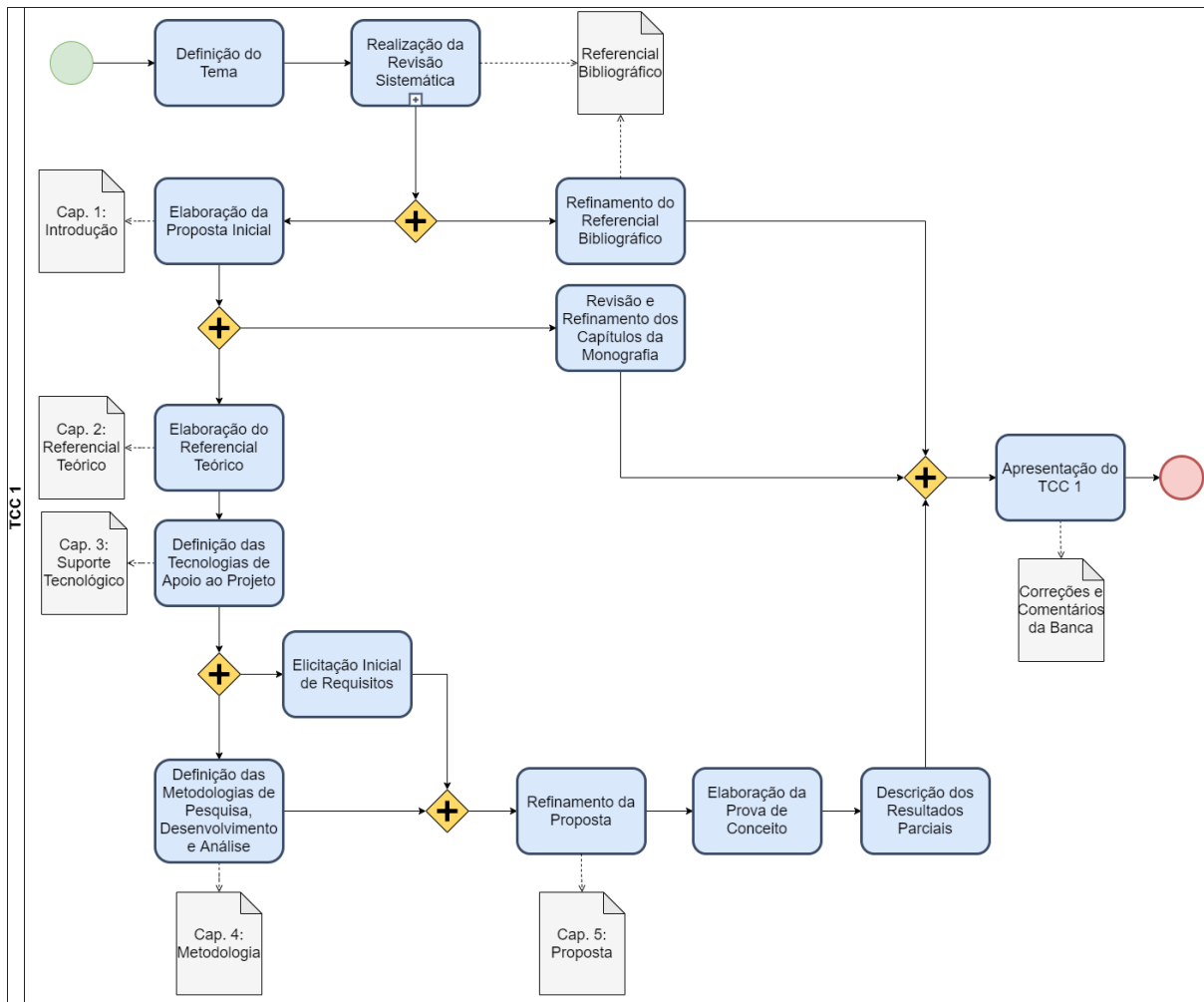
Dentre as características da Pesquisa-Ação, são destacadas três, sendo: a primeira, uma metodologia situacional, ou seja, a qual procura diagnosticar um problema específico numa situação também específica; a segunda, uma metodologia auto-avaliativa, ou seja, cada modificação introduzida é avaliada e monitorada para que haja, se necessário, redefinições do processo, e a terceira, uma metodologia cíclica, ou seja, as fases finais são usadas para aprimorar os resultados das fases anteriores (ENGEL, 2000).

O fluxo de atividades que ilustra as etapas seguidas na análise de resultados, com o protocolo de Pesquisa-Ação, assim como as descrições dessas atividades, está apresentado na Seção 4.4.5.

### 4.4 Fluxos de Atividades

O fluxo de atividades para a realização deste Trabalho de Conclusão de Curso foi dividido em duas etapas, as quais são referentes ao TCC 1 e ao TCC 2, como representadas nas Figuras 2 e 3, respectivamente. O fluxo de atividades da Realização da Revisão Sistemática, representado na Figura 4, é derivado do fluxo do TCC 1; enquanto os fluxos do Desenvolvimento da Proposta e da Análise dos Resultados, representados nas Figuras 5 e 6, respectivamente, são derivados do fluxo do TCC 2. As atividades referentes a cada um dos fluxos estão descritas a seguir.

Figura 2 – Fluxo de Atividades do TCC 1



Fonte: autoria própria

#### 4.4.1 Trabalho de Conclusão de Curso 1

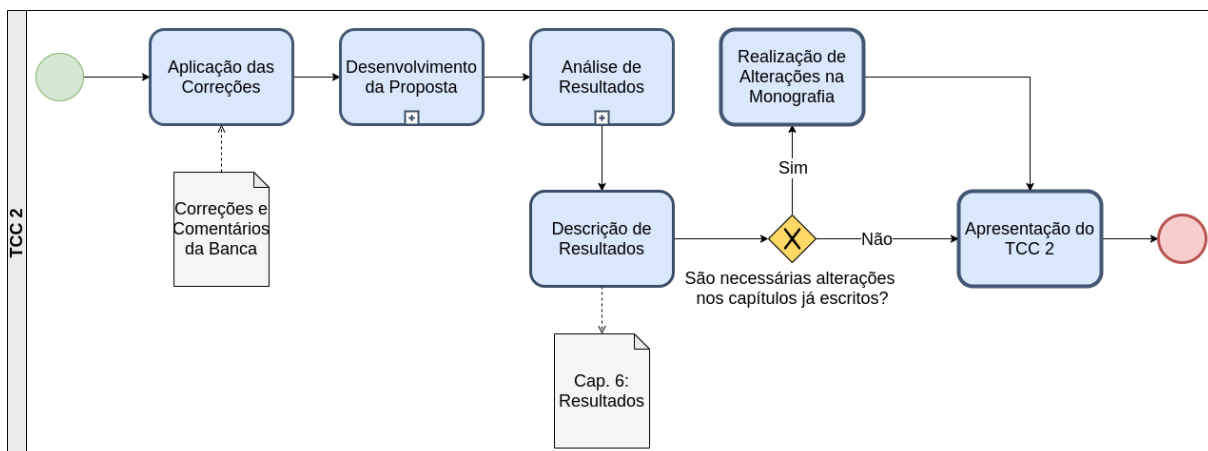
- **Definição do Tema:** Esta atividade envolve a definição da área geral em que ocorre a pesquisa e o Trabalho de Conclusão de Curso, a definição de escopo inicial, e a pesquisa de estudos relacionados;
- **Realização da Revisão Sistemática:** Esta etapa envolve a realização da Revisão Sistemática da Literatura, feita para levantar os estudos relacionados ao tema definido, e contribuir na formação de um referencial bibliográfico para embasamento do trabalho. O fluxo de atividades referente a esta etapa está ilustrado na Figura 4, e suas atividades estão descritas na Seção 4.4.3;
- **Elaboração da Proposta Inicial:** Esta atividade envolve a definição e contextualização do escopo do Trabalho de Conclusão de Curso, definição dos objetivos e da justificativa do projeto. O resultado desta atividade é a documentação do Capítulo 1;

- **Refinamento do Referencial Bibliográfico:** Esta atividade ocorre durante quase todo o fluxo do TCC 1, como ilustrado na Figura 2. Ela é responsável pelo refinamento e aprofundamento da pesquisa, buscando melhorar o referencial bibliográfico e suprir necessidades que podem surgir ao longo do desenvolvimento do trabalho, com a utilização da técnica de *Snow Ball*, a partir dos artigos já presentes no referencial, ou apenas fazendo um Levantamento Bibliográfico acerca dos temas sobre os quais o referencial bibliográfico apresentar algum tipo de lacuna;
- **Elaboração do Referencial Teórico:** Esta atividade envolve a sintetização e a apresentação dos dados presentes no referencial bibliográfico, os quais são relevantes para o escopo da pesquisa. Nesta atividade, são definidos os conceitos utilizados para a construção da proposta. O resultado desta atividade é a documentação do Capítulo 2;
- **Revisão e Refinamento dos Capítulos da Monografia:** Esta atividade ocorre durante toda a escrita do Trabalho de Conclusão de Curso. Ela envolve a revisão e o refinamento dos capítulos que são escritos;
- **Definição das Tecnologias de Apoio ao Projeto:** Esta atividade envolve a definição das tecnologias que servirão de apoio ao desenvolvimento de todo o projeto. Em outras palavras, as tecnologias que foram utilizadas para a pesquisa, o desenvolvimento e a análise de resultados do projeto. O resultado desta atividade é a documentação do Capítulo 3;
- **Definição das Metodologias de Pesquisa, Desenvolvimento e Análise:** Esta atividade envolve a definição das metodologias utilizadas para a pesquisa, para o desenvolvimento da ferramenta, e para a análise dos resultados obtidos. O resultado desta atividade é a documentação do Capítulo 4;
- **Elicitação Inicial de Requisitos:** Esta atividade envolve a elicitacão inicial dos requisitos da ferramenta, que ocorreu de forma paralela à execução da atividade anterior;
- **Refinamento da Proposta:** Esta atividade envolve o refinamento do projeto, a partir da compreensão das atividades anteriores e da proposta inicial. O resultado desta atividade é o Capítulo 5;
- **Elaboração da Prova de Conceito:** Esta atividade envolve a elaboração da prova de conceito para examinar a viabilidade da proposta. Ela foi realizada com a elaboração de um protótipo, e com a configuração do ambiente de desenvolvimento da ferramenta, utilizando as tecnologias definidas;

- **Descrição dos Resultados Parciais:** Esta atividade envolve a descrição dos resultados parciais obtidos no Trabalho de Conclusão de Curso 1, e a descrição das perspectivas para o Trabalho de Conclusão de Curso 2, e
- **Apresentação do TCC 1:** Esta atividade envolve a finalização do Trabalho de Conclusão de Curso 1 com a apresentação final de todo o projeto. O resultado desta atividade são as correções e os comentários da banca examinadora.

#### 4.4.2 Trabalho de Conclusão de Curso 2

Figura 3 – Fluxo de Atividades do TCC 2



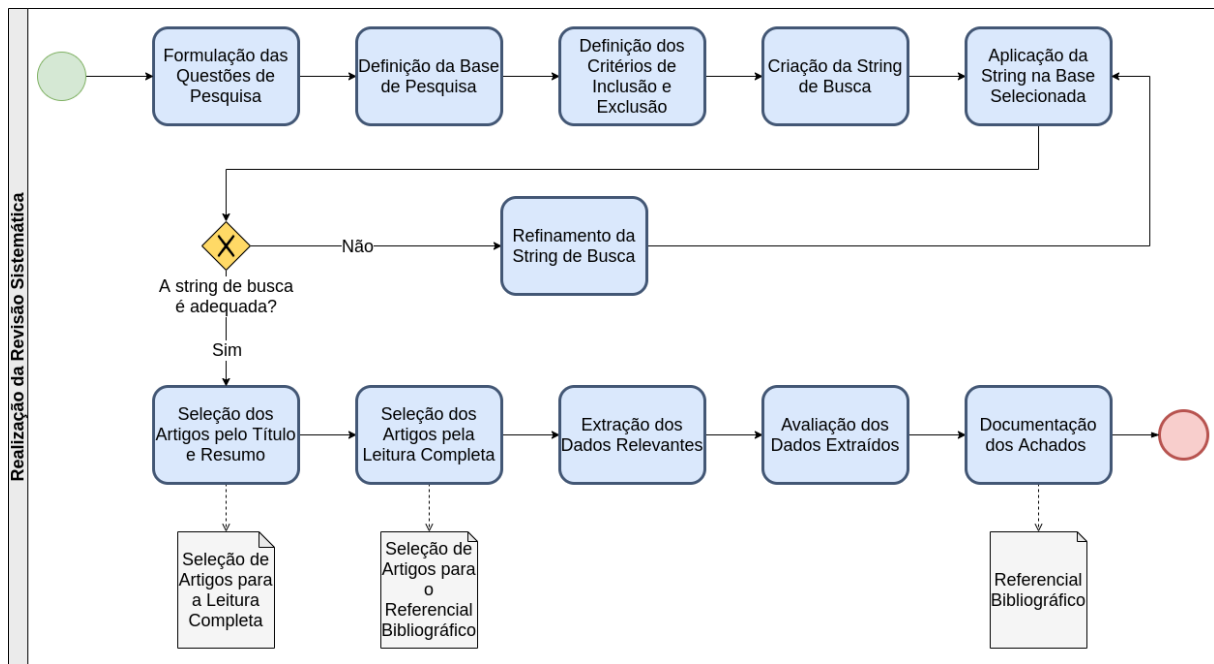
Fonte: autoria própria

- **Aplicação das Correções:** Esta atividade envolve a aplicação de correções apontadas pela banca examinadora na apresentação do Trabalho de Conclusão de Curso 1;
- **Análise de Resultados:** Esta etapa envolve as atividades relacionadas à análise de resultados do trabalho, realizada com o protocolo de Pesquisa-Ação. O fluxo de atividades referente a esta etapa está ilustrado na Figura 6, e as descrições das atividades presentes nesse fluxo encontram-se na Seção 4.4.5;
- **Descrição de Resultados:** Esta atividade envolve a descrição dos resultados obtidos após o desenvolvimento e a análise da proposta. São evidenciados os objetivos alcançados e quais serão os possíveis trabalhos futuros. O resultado desta atividade é a documentação do Capítulo 6;
- **Realização de Alterações na Monografia:** Esta atividade envolve a realização de alterações nos capítulos escritos anteriormente, caso sejam necessárias, e
- **Apresentação do TCC 2:** Esta atividade envolve a apresentação do Trabalho de Conclusão de Curso 2 à banca examinadora.

### 4.4.3 Realização da Revisão Sistemática

Como citado na Seção 4.1.4, o procedimento de Revisão Sistemática da Literatura foi utilizado para contribuir com a formação do referencial bibliográfico utilizado para o embasamento deste Trabalho de Conclusão de Curso. Uma melhor descrição da Revisão Sistemática realizada pode ser encontrada no Apêndice A, e o fluxo de atividades referente à realização desse procedimento está ilustrado na Figura 4. A seguir, são descritas as atividades presentes nesse fluxo.

Figura 4 – Fluxo de Atividades da Realização da Revisão Sistemática



Fonte: autoria própria

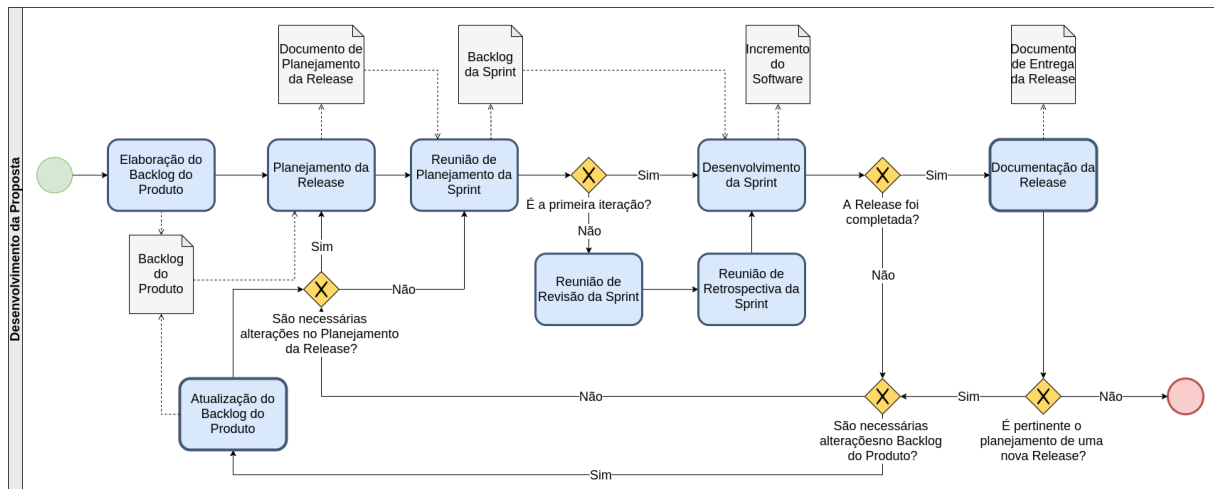
- **Formulação das Questões de Pesquisa:** Esta atividade diz respeito à definição das questões que guiarão a realização da Revisão Sistemática da Literatura. Elas indicam quais pontos deverão ser investigados, para uma melhor compreensão do problema foco deste trabalho;
- **Definição da Base de Pesquisa:** Esta atividade envolve a avaliação das bases de pesquisa disponíveis e a decisão de qual, entre as bases consideradas, deveria ser utilizada para a realização da pesquisa;
- **Definição dos Critérios de Inclusão e Exclusão:** Esta atividade é referente à elaboração dos critérios de inclusão e exclusão dos artigos encontrados na pesquisa. Tais critérios foram utilizados nas atividades de seleção de quais artigos deveriam ser lidos e quais deveriam ser descartados, dentre os artigos resultantes da busca;

- **Criação da *String* de Busca:** Esta atividade envolve a definição e a estruturação das palavras-chave e dos operadores lógicos empregados para a formação da *string* utilizada na busca;
- **Aplicação da *String* na Base Seleccionada:** Esta atividade diz respeito à aplicação da *string*, criada na atividade anterior, na base de pesquisa selecionada para a busca. Neste ponto, os artigos resultantes foram avaliados, com o intuito de decidir se a *string* utilizada foi ou não adequada. Se sim, seguiu-se para as etapas de seleção dos artigos encontrados. Caso contrário, a *string* teve que ser refinada;
- **Refinamento da *String* de Busca:** Esta atividade diz respeito à reformulação da *string* de busca, com base nos resultados da aplicação da mesma na base de pesquisa selecionada. Uma quantidade muito grande de artigos resultantes indica uma *string* muito genérica; enquanto uma quantidade muito pequena indica uma *string* muito específica. Tudo depende dos objetivos da pesquisa, mas a intenção foi encontrar um ponto de equilíbrio para que a revisão pudesse ser concluída de maneira eficiente;
- **Seleção dos Artigos pelo Título e Resumo:** Esta atividade envolve a leitura dos títulos e dos resumos dos artigos resultantes na busca, aplicando os critérios de inclusão e exclusão definidos, para escolher os artigos que deveriam passar para a próxima fase de seleção;
- **Seleção dos Artigos pela Leitura Completa:** Esta atividade envolve a leitura completa dos artigos selecionados na atividade anterior, a fim de estabelecer os artigos que contribuem com o objeto de pesquisa;
- **Extração dos Dados Relevantes:** Esta atividade é referente à extração, dentre os artigos estabelecidos na fase anterior, dos dados úteis para o embasamento do trabalho;
- **Avaliação dos Dados Extraídos:** Esta atividade envolve a análise dos dados extraídos na atividade anterior, de forma a definir de que maneira eles poderiam contribuir no embasamento do trabalho, e
- **Documentação dos Achados:** Esta atividade é referente à documentação, tanto dos artigos utilizados quanto dos dados extraídos, e resultou na criação do referencial bibliográfico que contribuiu para o embasamento do trabalho.

#### 4.4.4 Desenvolvimento da Proposta

- **Elaboração do *Backlog* do Produto:** Esta atividade envolve a elaboração do *Backlog* do Produto, na qual ocorre o levantamento e a priorização dos requisitos da ferramenta para o direcionamento do desenvolvimento da proposta. O resultado desta atividade é a documentação do próprio *Backlog* do Produto;

Figura 5 – Fluxo de Atividades do Desenvolvimento da Proposta



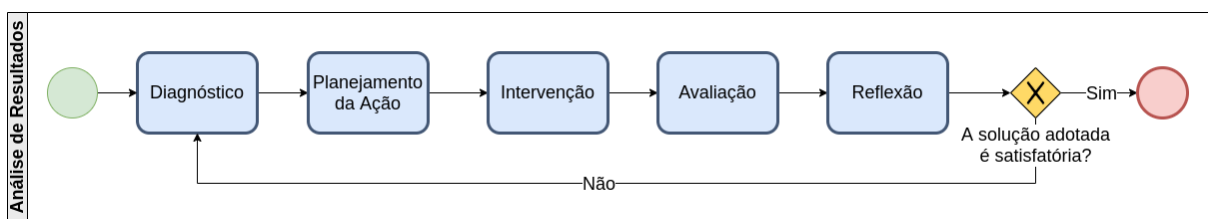
Fonte: autoria própria

- **Planejamento da *Release*:** Esta atividade envolve o planejamento da *Realease*, o qual é a seleção de um conjunto de atividades do *Backlog* do Produto para serem concluídas no período da *Realease*. O resultado desta atividade é a documentação do planejamento da *Realease*;
- **Reunião de Planejamento da *Sprint*:** Esta atividade envolve o planejamento da *Sprint*, o qual é a seleção das atividades planejadas na *Realease* e colocadas no Quadro *Kaban*, para serem concluídas no período da *Sprint*. O resultado desta atividade é a documentação do *Backlog* da *Sprint*;
- **Reunião de Revisão da *Sprint*:** Esta atividade envolve a revisão da *Sprint* para que sejam apresentadas as mudanças feitas ao final da *Sprint*, a fim de que a dupla esteja alinhada com as funcionalidades entregues;
- **Reunião de Retrospectiva da *Sprint*:** Esta atividade envolve a retrospectiva da *Sprint*, sendo que esta reunião é feita para que a dupla discuta quais foram os problemas relacionados à *Sprint* concluída, e como solucioná-los nas próximas iterações;
- **Desenvolvimento da *Sprint*:** Esta atividade envolve o desenvolvimento das atividades selecionadas para serem executadas no período da *Sprint*. O resultado desta atividade é o incremento das funcionalidades do Software;
- **Documentação da *Release*:** Esta atividade envolve a documentação do que foi executado ao final da *Release*, e
- **Atualização do *Backlog* do Produto:** Esta atividade envolve a atualização do *Backlog* do Produto, caso o documento necessite de alterações.

#### 4.4.5 Análise de Resultados

Como definido na Seção 4.3, a metodologia que guiou o processo de análise dos resultados do trabalho foi a Pesquisa-Ação. O fluxo de atividades definido está de acordo com o processo apresentado por Filippo (2011). Segundo o autor, a Pesquisa-Ação é tipicamente realizada em ciclos iterativos que, sucessivamente, refinam o conhecimento adquirido nos ciclos anteriores. A representação do processo é um ciclo formado por várias etapas, que podem ser visualizadas no fluxo de atividades da Análise de Resultados (Figura 6).

Figura 6 – Fluxo de Atividades da Análise de Resultados.



Fonte: (FILIPPO, 2011), adaptado pelas autoras

- **Diagnóstico:** Esta atividade compreende a identificação e a análise dos problemas que motivam ações de melhoria na proposta. Ela envolve uma visão abrangente de todo o contexto e problema. Nos ciclos seguintes, a atividade de diagnóstico deve ser realizada com base nos resultados e reflexões do ciclo anterior;
- **Planejamento da Ação:** Esta atividade envolve o planejamento das intervenções que serão realizadas para solucionar ou reduzir os problemas identificados na atividade anterior. O planejamento da ação deve abranger as formas de coleta e análise de dados;
- **Intervenção:** Nesta atividade, são realizadas as ações ou tentativas de solução que causam modificações na proposta concebida anteriormente. Nela, os dados que influenciam na avaliação da efetividade da ação são também coletados;
- **Avaliação:** Nesta atividade, os resultados obtidos na etapa anterior são avaliados diante dos objetivos definidos, buscando identificar os efeitos decorrentes das ações e até que ponto os problemas foram resolvidos, e
- **Reflexão:** Esta é a última atividade do ciclo. Nela, é feita uma reflexão das atividades e resultados obtidos na Pesquisa-Ação até o momento, e é decidido se um novo ciclo deve ser iniciado. O conhecimento adquirido é explicitado nesta atividade.



## 4.5 Cronogramas

Os Quadros 2 e 3 apresentam os cronogramas do Trabalho de Conclusão de Curso 1 e 2, respectivamente. Nos quadros, são descritas as atividades com as suas respectivas datas de execução, de forma a contemplar todas as atividades descritas na Seção 4.4.

Quadro 2 – Cronograma TCC 1

<b>Atividade</b>	<b>Fevereiro</b>	<b>Março</b>	<b>Abril</b>	<b>Maiο</b>
Definição do Tema	X			
Revisão Sistemática da Literatura	X			
Introdução	X			
Referencial Teórico		X		
Suporte Tecnológico		X		
Metodologia		X	X	
Proposta		X	X	
Considerações Finais		X		
Finalização Geral				X
Prévia da Apresentação				X
Apresentação Final				X

Fonte: autoria própria

Quadro 3 – Cronograma TCC 2

<b>Atividade</b>	<b>Julho</b>	<b>Agosto</b>	<b>Setembro</b>	<b>Otubro</b>	<b>Novembro</b>
Aplicação das Correções da Banca	X				
Atualização da Escrita do TCC	X				
Levantamento dos Requisitos	X	X			
Implementação da Ferramenta		X	X	X	
Pesquisa-Ação			X	X	
Documentação dos Resultados Obtidos				X	
Finalização da Monografia				X	
Apresentação Final					X

Fonte: autoria própria

## 4.6 Resumo do Capítulo

Neste capítulo, foram apresentadas as escolhas metodológicas definidas para o desenvolvimento deste Trabalho de Conclusão de Curso. Para isso, foram apresentadas as metodologias da fase de pesquisa, de desenvolvimento da proposta, e de análise dos resultados, expondo e classificando os pontos importantes em cada uma delas. Em seguida, com o objetivo de retratar as etapas de desenvolvimento do trabalho, foram apresentados os fluxos de atividades que tem guiado o desenvolvimento, e as respectivas descrições das

atividades ilustradas nesses fluxos. Por fim, os cronogramas das duas fases de realização do trabalho (TCC 1 e TCC 2) foram apresentados.

## 5 Backlog Maker

*Backlog Maker* é o nome dado à ferramenta de construção e manutenção do *Backlog* do Produto, desenvolvida neste Trabalho de Conclusão de Curso. Neste capítulo, há o detalhamento dessa ferramenta.

Sendo assim, a primeira seção apresenta a contextualização, incluindo o estudo realizado a respeito das ferramentas similares; a segunda seção apresenta uma primeira versão da *Baseline* de Requisitos da ferramenta, requisitos de alto nível; na terceira seção, é detalhada a arquitetura da solução; na quarta seção, é apresentada a prova de conceito, que abrange tanto a construção do protótipo de alta fidelidade da ferramenta quanto a configuração de ambiente, com as tecnologias destacadas na Seção 3.1; na quinta seção, é exibido o *Backlog* da ferramenta, com requisitos mais detalhados; na sexta seção, são apresentadas a solução e a versão final da ferramenta, e por fim, tem-se o resumo do capítulo.

### 5.1 Contextualização

A principal justificativa para o desenvolvimento da ferramenta envolve a busca por mitigar os problemas enfrentados na gestão de requisitos em organizações que adotam o *Scrum* como modelo ágil de desenvolvimento. Sendo o *Backlog* do Produto o artefato central do *Scrum*, onde está concentrada a lista de defeitos, melhorias solicitadas pelo cliente, funcionalidades, atualização de tecnologias, entre outros itens que expressam os requisitos do produto, a atividade de atribuição desses itens, caso não seja devidamente realizada, pode resultar na falha de todo o projeto ou na entrega inadequada de incrementos (ANAND; DINAKARAN, 2015). Dessa forma, em vista da volatilidade de requisitos em um projeto ágil, e da necessidade de manutenção do rastro desses requisitos e das mudanças no *Backlog* do Produto, foi considerada a elaboração de uma ferramenta que oferece apoio tecnológico à construção e à manutenção do *Backlog* do Produto, com o intuito de possibilitar um meio para facilitar e melhorar esse processo. O público-alvo dessa ferramenta está concentrado em empresas de pequeno e médio portes, com equipes pequenas, nas quais o *PO* esteja inserido.

A seguir, é apresentado um estudo feito acerca das ferramentas existentes no mercado, que podem ser utilizadas com o mesmo propósito da ferramenta acordada neste trabalho, considerando os aspectos que poderiam influenciar no desenvolvimento da mesma.

### 5.1.1 Ferramentas Similares

Com o objetivo de analisar os sistemas similares à ferramenta *Backlog Maker*, foram levantados os principais pontos positivos e negativos, bem como as funcionalidades que envolvem a criação de um *Backlog* do Produto nos sistemas existentes no mercado. A escolha das ferramentas foi feita com base em sua popularidade, em sua função no gerenciamento de projetos de software, e na utilização de algumas dessas pelas autoras do projeto. As ferramentas analisadas foram *Trello*, *Clickup*, *Jira*, *Monday*, *Asana*, *Wrike*, *Hansoft*, *Taiga* e *ZenHub*, descritas, respectivamente, nos Quadros de 4 a 12.

Quadro 4 – Análise de Ferramentas Similares: *Trello*

<b>FERRAMENTA</b>	<i>Trello</i>
<b>DESCRIÇÃO</b>	Ferramenta de gerenciamento de projetos que utiliza um esquema de listas, cartões e quadros para organizar atividades dentro de um projeto.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Facilidade de movimentar as tarefas entre as listas do projeto, e</li> <li>- Facilidade de personalização para o gerenciamento do projeto.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Versão gratuita com limitações;</li> <li>- Não é especializada no gerenciamento <i>Scrum</i>, embora tenha funcionalidades que permitem a personalização para este gerenciamento, e</li> <li>- Não há direcionamento para a descrição de histórias de usuários.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação de cartões para as tarefas;</li> <li>- Permite a personalização do <i>Kaban</i>;</li> <li>- Permite a criação de <i>tags</i> para os cartões;</li> <li>- Permite a associação de membros da equipe, e a adição de estimativa e de data de entrega;</li> <li>- Permite a criação de <i>checklists</i> para a realização da tarefa, além da anexação arquivos;</li> <li>- Permite a adição de comentários nas tarefas, e</li> <li>- Permite a visualização da completude da tarefa com base na <i>checklist</i>.</li> </ul>

Fonte: autoria própria

Quadro 5 – Análise de Ferramentas Similares: *Clickup*

<b>FERRAMENTA</b>	<i>Clickup</i>
<b>DESCRIÇÃO</b>	Ferramenta gratuita que apoia o gerenciamento de projetos ágeis.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Possibilita a criação de listas de atividades;</li> <li>- Auxilia no gerenciamento de projetos, com a disponibilização de quadros <i>Kanban</i>;</li> <li>- Possibilita a criação de documentos e <i>wikis</i>, e</li> <li>- Fornece integração com ferramentas de planilhas, calendários, dentre outros.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Versão gratuita limitada;</li> <li>- Para criar níveis de granularidade (Épico-<i>Feature-Histórias</i> de Usuário), é necessário utilizar a nomenclatura da ferramenta, pois esta não tem estas divisões por padrão;</li> <li>- Não é direcionado para o gerenciamento <i>Scrum</i>, embora tenha a opção de personalização para que seja possível esse gerenciamento, e</li> <li>- Muitas funcionalidades a serem personalizadas, o que pode causar confusão na hora de gerenciar.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite até 5 níveis de indentação para as <i>checklists</i>;</li> <li>- Apresenta o histórico de mudança nas tarefas e permite a configuração de quem pode ver o histórico;</li> <li>- Permite a adição de comentários nas tarefas, com marcação de pessoas;</li> <li>- Permite diferentes tipos de visualização das tarefas (<i>List, Board, Gantt, Calendar, Chart, Doc, etc</i>);</li> <li>- Permite a importação de listas de tarefas, por exemplo do <i>Excel</i>;</li> <li>- Permite a adição de pessoas responsáveis, data de vencimento e prioridades nas tarefas;</li> <li>- Permite a criação de classificações personalizadas para as tarefas, por exemplo, a classificação do <i>MOSCOW</i>;</li> <li>- Permite a seleção de várias tarefas de uma vez;</li> <li>- Permite a filtragem das tarefas;</li> <li>- Permite a criação de <i>templates</i> para as tarefas;</li> <li>- Permite o aninhamento de tarefas para colocar níveis de Épico até tarefas, e</li> <li>- Permite a visualização do progresso das tarefas.</li> </ul>

Fonte: autoria própria

Quadro 6 – Análise de Ferramentas Similares: *Jira*

<b>FERRAMENTA</b>	<i>Jira</i>
<b>DESCRIÇÃO</b>	Ferramenta para gerenciamento do desenvolvimento de software.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Tutoriais de uso espalhados pela ferramenta;</li> <li>- Geração de relatórios, em tempo real, do desempenho da equipe e da porcentagem de conclusão para entrega;</li> <li>- Possibilita o acompanhamento de tarefas, e</li> <li>- Auxilia no gerenciamento de <i>Releases</i>.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Muitas funcionalidades que podem ser personalizadas, o que pode causar confusão na hora de gerenciar;</li> <li>- Não possui um direcionamento para a criação dos itens, e</li> <li>- Funcionalidades limitadas no plano gratuito.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a seleção de opções que customizam o projeto;</li> <li>- Permite a seleção do <i>Kanban</i> ou do <i>Scrum</i> para iniciar o projeto;</li> <li>- Disponibiliza um <i>quickstart</i> na ferramenta;</li> <li>- Permite a criação e a visualização de Épicos;</li> <li>- Permite a criação e a visualização das tarefas e dos estados das tarefas, para cada Épico;</li> <li>- Permite a edição e a personalização das tarefas (que podem ser em formato de histórias de usuário);</li> <li>- Permite a adição de tarefas que causam dependência;</li> <li>- Permite a criação de subtarefas;</li> <li>- Permite a visualização dos Épicos e tarefas em um fluxo de trabalho, indicando o período;</li> <li>- Permite a visualização dos Épicos e tarefas em <i>Backlog</i> (lista);</li> <li>- Permite a criação de <i>Sprints</i>;</li> <li>- Disponibiliza o quadro <i>Kanban</i> da <i>Sprint</i>, com as colunas "não iniciado", "em andamento" e "concluído";</li> <li>- Permite a personalização e acompanhamento do quadro <i>Kanban</i> da <i>Sprint</i>;</li> <li>- Permite a integração com ferramentas de gerenciamento de código (<i>Gitlab</i>, <i>Github</i>, <i>Bitbucket</i>, entre outros) para visualização de informações a respeito do código, e</li> <li>- Permite a criação de <i>wikis</i> para compartilhamento de conhecimento.</li> </ul>

Fonte: autoria própria

Quadro 7 – Análise de Ferramentas Similares: *Monday*

<b>FERRAMENTA</b>	<i>Monday</i>
<b>DESCRIÇÃO</b>	Ferramenta que auxilia na criação de fluxos de trabalho e possibilita que as equipes gerenciem seus processos, projetos e trabalhos diários.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Ótima interface de usuário;</li> <li>- Fornece uma visão rápida de todas as tarefas;</li> <li>- Poderosa capacidade de geração de relatórios, e</li> <li>- Geração de notificações.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Disponível apenas no plano <i>Enterprise</i> (apenas 14 dias de uso gratuito), e</li> <li>- Não possui funcionalidades tão ricas quanto um software dedicado ao gerenciamento de projetos.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação de grupos de atividades;</li> <li>- Permite a adição de colunas (ou campos) de vários tipos para as atividades de um grupo;</li> <li>- Permite a adição de atividades;</li> <li>- Permite a adição de diferentes tipos de visualização das atividades (padrão, <i>kanban</i>, linha do tempo, tabela, etc.);</li> <li>- Permite a aplicação de diferentes filtros nas atividades em visualização;</li> <li>- Permite a integração com diferentes ferramentas, possibilitando a concentração das atividades em um só lugar;</li> <li>- Permite a criação de áreas de trabalho diferentes, cada uma com seus próprios grupos e atividades;</li> <li>- Permite a visualização das atividades da semana atribuídas para o usuário;</li> <li>- Permite a visualização das notificações recebidas pelo usuário, e</li> <li>- Permite a configuração da visualização e das ações relacionadas às atividades da semana do usuário.</li> </ul>

Fonte: autoria própria

Quadro 8 – Análise de Ferramentas Similares: *Asana*

<b>FERRAMENTA</b>	<i>Asana</i>
<b>DESCRIÇÃO</b>	Solução de gerenciamento de trabalho e de projetos para equipes.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Ótimo para gerenciar listas de tarefas;</li> <li>- Interface de usuário robusta e customizável, e</li> <li>- Poderosas funcionalidades de colaboração.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- As notificações por <i>e-mail</i> não são sempre úteis;</li> <li>- Uma tarefa pode ser marcada como feita sem ter sido realmente concluída, e</li> <li>- Disponível apenas no plano <i>Premium</i> (apenas 30 dias de uso gratuito).</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação de um <i>brief</i> do projeto (visão compartilhada);</li> <li>- Permite a adição de <i>links</i> e arquivos ao projeto (visão compartilhada);</li> <li>- Permite a atualização de <i>status</i> do projeto (um ponto de controle, com campos definidos que permitem o detalhamento da atualização, ex: "Anexos", "Resumo", "O que já realizamos", entre outros, e possibilidade de adição de campos);</li> <li>- Permite a criação e edição de marcos aos projetos;</li> <li>- Permite a criação de grupos de atividades;</li> <li>- Permite a adição de campos de vários tipos para as atividades de um grupo, inclusive a recuperação de campos salvos em bibliotecas;</li> <li>- Permite a adição de atividades;</li> <li>- Permite a visualização das atividades em lista, quadro ou cronograma;</li> <li>- Permite a visualização de um calendário com as atividades do projeto;</li> <li>- Permite a visualização de um painel com as métricas/informações (ou gráficos) do projeto;</li> <li>- Permite o envio de mensagens aos membros do projeto;</li> <li>- Permite a criação de formulários para padronizar as solicitações, transformando-as automaticamente em tarefas, e</li> <li>- Permite a visualização de todos os arquivos anexados nas tarefas ou mensagens do projeto.</li> </ul>

Fonte: autoria própria



Quadro 9 – Análise de Ferramentas Similares: *Wrike*

<b>FERRAMENTA</b>	<i>Wrike</i>
<b>DESCRIÇÃO</b>	Ferramenta colaborativa de gerenciamento de projetos que auxilia os usuários a gerenciar os projetos do início ao fim.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Customizável para as necessidades específicas dos usuários;</li> <li>- Visibilidade em tempo real das tarefas e seus respectivos estados, e</li> <li>- Suporte para integrações.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Pode ser complicada para usuários iniciantes;</li> <li>- Funcionalidade de pesquisa não é muito eficiente;</li> <li>- Não é especializada no gerenciamento <i>Scrum</i>, embora tenha funcionalidades que permitem a personalização para este gerenciamento, e</li> <li>- Interface confusa.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Possui quadros <i>Kanban</i>;</li> <li>- Permite a personalização dos fluxos de trabalhos de um projeto;</li> <li>- Permite a associação de membros da equipe, e a adição de estimativa e de data de entrega;</li> <li>- Permite a criação de <i>checklists</i> para a realização da tarefa, além da anexação de arquivos;</li> <li>- Permite a adição de comentários nas tarefas;</li> <li>- Permite vários tipos de visualização das tarefas, e</li> <li>- Permite a associação de tarefas como subtarefas de outras tarefas.</li> </ul>

Fonte: autoria própria

Quadro 10 – Análise de Ferramentas Similares: *Hansoft*

<b>FERRAMENTA</b>	<i>Hansoft</i>
<b>DESCRIÇÃO</b>	Ferramenta para planejamento de projetos ágeis.
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Viabiliza a criação e o gerenciamento do <i>Backlog</i> do Produto;</li> <li>- É flexível e customizável;</li> <li>- Interface simples;</li> <li>- Possui um panorama geral do <i>Backlog</i>, e</li> <li>- Permite o gerenciamento de grandes projetos.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Versão gratuita com limitações;</li> <li>- Não é especializado no gerenciamento <i>Scrum</i>, embora tenha funcionalidades que permitem este gerenciamento, e</li> <li>- Não há direcionamento para a descrição de histórias de usuários.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação, a manutenção, a estruturação e a priorização de um <i>Backlog</i> do Produto;</li> <li>- Permite a criação de níveis de granularidade para o <i>Backlog</i> do Produto;</li> <li>- Permite a designação de tarefas para pessoas específicas;</li> <li>- Permite a anexação de arquivos em histórias de usuário;</li> <li>- Permite a customização e adição de estimativa aos itens do <i>Backlog</i>;</li> <li>- Permite a criação de listas de desejos de forma fácil;</li> <li>- Permite que os itens do <i>Backlog</i> sejam arrastados para a uma <i>Sprint</i> ou outra lista;</li> <li>- Possui <i>Dashboards</i> para acompanhamento das entregas da equipe, e</li> <li>- Possui <i>chat</i> para conversas entre os membros da equipe.</li> </ul>

Fonte: autoria própria

Quadro 11 – Análise de Ferramentas Similares: *Taiga*

<b>FERRAMENTA</b>	<i>Taiga</i>
<b>DESCRIÇÃO</b>	Ferramenta <i>open source</i> para planejamento de projetos ágeis
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Viabiliza o gerenciamento do <i>Backlog</i> do Produto;</li> <li>- Versão gratuita permite acesso a todas as funcionalidades, apenas com algumas limitações de capacidade;</li> <li>- Personalização dos níveis de granularidade para o <i>Backlog</i> (épicos, histórias de usuário e tarefas);</li> <li>- Fácil de aprender e usar, e</li> <li>- Customizável.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- A versão gratuita permite apenas um projeto privado;</li> <li>- Remove do <i>Backlog</i> histórias adicionadas às <i>sprints</i>, e</li> <li>- Não possui orientações que direcionam o usuário na criação das histórias.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação de projetos <i>Scrum</i> ou <i>Kanban</i>;</li> <li>- Permite a adição de membros aos projetos criados;</li> <li>- Permite a visualização da atividade do projeto (histórico de atualização);</li> <li>- Permite a visualização do <i>Backlog</i> do projeto;</li> <li>- Permite a criação de histórias de usuário para o projeto;</li> <li>- Permite o anexo de arquivos nas histórias criadas;</li> <li>- Permite a criação de <i>tasks</i> nas histórias criadas;</li> <li>- Permite a adição de comentários às histórias criadas;</li> <li>- Permite a atribuição de pontos (esforço) por tipo;</li> <li>- Exibe o total de pontos de cada história;</li> <li>- Permite a atribuição da história a membros do projeto;</li> <li>- Permite a adição de observadores à história;</li> <li>- Permite a criação de issues no projeto;</li> <li>- Permite a criação de uma <i>wiki</i> para o projeto;</li> <li>- Permite a criação de <i>sprints</i>;</li> <li>- Permite a visualização de um <i>kanban</i> para as <i>sprints</i>, e</li> <li>- Permite a personalização do projeto (permissões, atributos, entre outros).</li> </ul>

Fonte: autoria própria

Quadro 12 – Análise de Ferramentas Similares: *ZenHub*

<b>FERRAMENTA</b>	<i>ZenHub</i>
<b>DESCRIÇÃO</b>	Ferramenta para planejamento e gerenciamento de projetos ágeis integrada ao GitHub
<b>PONTOS POSITIVOS</b>	<ul style="list-style-type: none"> <li>- Permite a visualização de um <i>Kanban</i> diretamente em uma aba do <i>GitHub</i> (por meio da instalação de uma extensão no <i>browser</i>);</li> <li>- Extrai os cards do <i>Kanban</i> (histórias de usuário) diretamente das issues cadastradas no repositório, evitando o retrabalho e facilitando o gerenciamento dos itens de trabalho;</li> <li>- Permite a adição de vários repositórios, e</li> <li>- Versão gratuita com acesso a todas as funcionalidades.</li> </ul>
<b>PONTOS NEGATIVOS</b>	<ul style="list-style-type: none"> <li>- Não é específica para o <i>Scrum</i>;</li> <li>- Não possui orientações que direcionam o usuário na criação de issues e itens nos demais níveis de granularidade possíveis;</li> <li>- Níveis de granularidade engessados, e</li> <li>- <i>Issues</i> no repositório podem não representar histórias de usuário. As histórias de usuário na ferramenta sendo representadas apenas pelas <i>issues</i> do repositório (e vice-versa), em determinados contextos, pode atrapalhar o gerenciamento dos itens de trabalho.</li> </ul>
<b>FUNCIONALIDADES</b>	<ul style="list-style-type: none"> <li>- Permite a criação de <i>workspaces</i>;</li> <li>- Permite a adição de repositórios aos <i>workspaces</i> criados;</li> <li>- Permite a adição de membros aos <i>workspaces</i> criados;</li> <li>- Permite a criação de issues, que representam as histórias de usuário, e são apresentadas no repositório;</li> <li>- Permite a criação de Épicos;</li> <li>- Permite a criação de <i>Milestones</i>;</li> <li>- Permite a criação de <i>Releases</i>;</li> <li>- Permite a definição das datas para as <i>Sprints</i>;</li> <li>- Permite a visualização de um quadro <i>Kanban</i> com o fluxo de trabalho e cada card representando uma issue;</li> <li>- Permite a configuração do fluxo de trabalho para o <i>Kanban</i> (listas e ordem das listas);</li> <li>- Permite a geração de relatórios e gráficos como <i>burn-down</i> e <i>velocity</i>;</li> <li>- Permite a criação de um <i>Roadmap</i>, e</li> <li>- Permite a visualização de notificações.</li> </ul>

Fonte: autoria própria

## 5.2 *Baseline de Requisitos*

Nesta seção, é apresentada a *Baseline* de Requisitos, com o objetivo de listar as principais funcionalidades que a ferramenta de criação e de manutenção de um *Backlog* do Produto deve ter. Para a realização desta atividade, foram levados em consideração o levantamento feito no Capítulo 2 e os dados levantados das ferramentas similares, registrados na Seção 5.1.1. Portanto, para melhor compreensão, os requisitos foram divididos em alguns módulos com suas respectivas funcionalidades, descritos a seguir.

O primeiro módulo está relacionado ao cadastro de diferentes tipos de usuários, como o perfil de *Product Owner*, de *Scrum Master*, de cliente e de desenvolvedor. Os detalhes sobre cada perfil estão estabelecidos na sequência:

- Perfil de *Product Owner* e de *Scrum Master*: Este tipo de perfil tem a responsabilidade de criar e dar acesso aos participantes dos projetos, além de revisar e aprovar os itens do *Backlog* quando forem criados por outros membros da equipe;
- Perfil de desenvolvedor: Este tipo de perfil tem acesso à visualização e à edição dos itens do *Backlog*, e
- Perfil de Cliente: Este tipo de perfil tem acesso à visualização e à criação de comentários nos itens do *Backlog*.

O segundo módulo está associado à autenticação e à autorização dos usuários na ferramenta. Este módulo permite a realização do *login* com o *G-mail* ou utilizando o *E-mail* e Senha.

O terceiro módulo está associado à criação, à atualização, à exclusão e à visualização dos projetos. Algumas funcionalidades que são desejáveis para este módulo estão descritas a seguir:

- Criar e atualizar o nome e a descrição do projeto;
- Os gerentes do projeto (*Product Owner* ou *Scrum Master*) podem inserir membros ao projeto;
- É possível criar um resumo do projeto com perguntas direcionadas para que haja melhor compreensão do seu propósito;
- Anexar arquivos e *links* relevantes para que possam gerar uma visão compartilhada do projeto, e
- Adicionar e atualizar a prioridade do projeto com relação a outros projetos.

O quarto módulo está associado à criação, à atualização, à exclusão e à visualização dos itens do *Backlog* do Produto. Algumas funcionalidades que são desejáveis para este módulo estão descritas a seguir:

- É necessário que haja a criação, a visualização, a exclusão e a atualização dos Épicos;
- É necessário que haja a criação, a visualização, a exclusão e a atualização das *Features*;
- É necessário que haja a criação, a visualização, a exclusão e a atualização das Histórias, e
- Para cada História, é possível adicionar revisores; adicionar Histórias bloqueantes; atribuir membros do projeto; atribuir estimativa e priorização; adicionar *tags*; manter o histórico de atualizações da História; adicionar critérios de aceitação e tarefas a serem executadas; atribuir comentários na História, e contatar o cliente quando necessário.

O quinto módulo está associado à criação, à atualização, à exclusão e à visualização do *Backlog* da *Sprint*, bem como à visualização por meio de um quadro *Kanban*.

O sexto módulo está associado a uma lista de desejos para o projeto. Os itens desta lista podem ser sugeridos e utilizados pelo usuário na criação dos itens do *Backlog* do Produto.

O sétimo módulo está associado à criação de uma página de *wiki* para que os usuários consigam criar documentações pertinentes ao projeto.

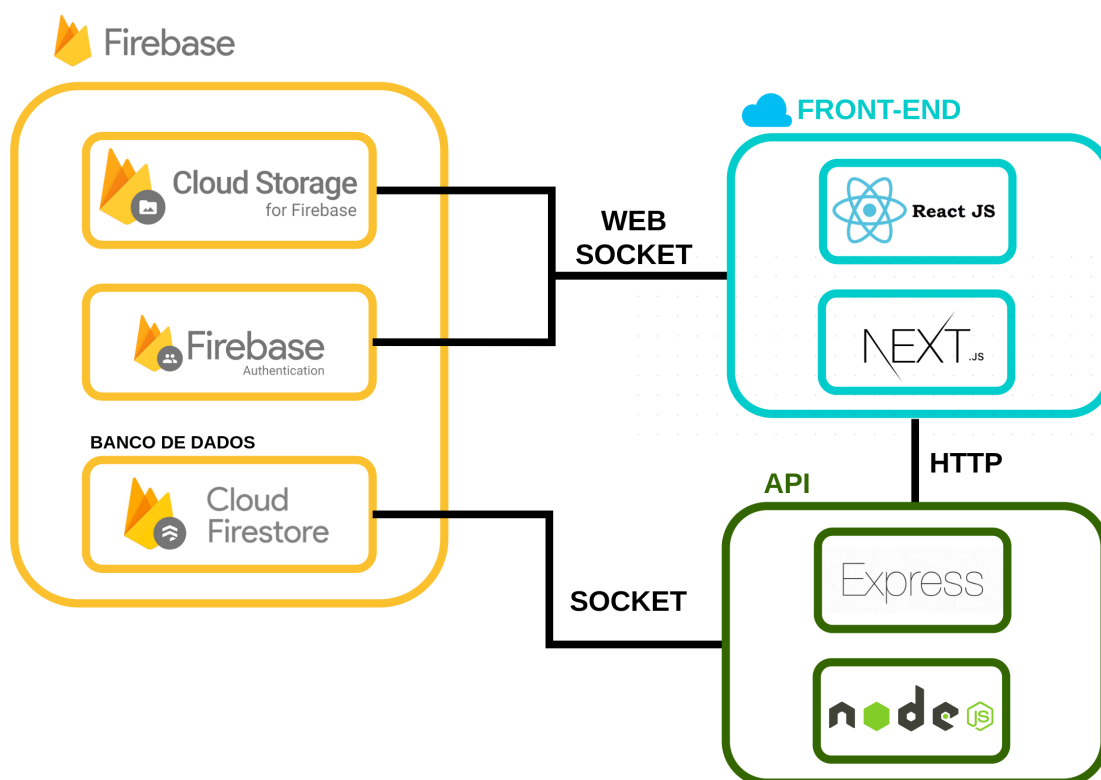
O oitavo módulo está associado às orientações e às ajudas necessárias aos usuários. O objetivo é instruir o usuário na inserção de dados para a criação de um *Backlog* do Produto.

O nono módulo está associado às configurações necessárias e relacionadas aos diferentes perfis de usuário da ferramenta; à personalização de priorização e estimativa desejadas pelos usuários, e à criação de novas *tags* para associar aos itens do *Backlog* do Produto.

### 5.3 Arquitetura da Solução

A arquitetura da ferramenta compreende três componentes diferentes, sendo: o *Firebase*, com os serviços oferecidos pela ferramenta; o *front-end*, utilizando o *React* com o *framework Next*, e a API, utilizando o *Node.js* com o *framework Express*. Todas essas tecnologias estão definidas no Capítulo 3. Os detalhes de cada componente estão descritos

Figura 7 – Componentes da arquitetura da solução



Fonte: autoria própria

nas subseções seguintes e representados na Figura 7, em um Diagrama de Relações ou Contexto.

### 5.3.1 *Firestore*

Este componente é utilizado como um provedor de serviços, fornecido pela infraestrutura do *Google* para auxiliar desenvolvedores. Os serviços utilizados do *Firestore* são:

- *Authentication* - Este serviço fornece um sistema completo de autenticação de usuários para a aplicação. Ele oferece apoio à autenticação por meio de senhas, números de telefone e provedores, como *Google*, *Facebook*, entre outros. Quando o usuário realiza o *login*, as credenciais fornecidas são verificadas por meio do Kit de Desenvolvimento de Software (*Software Development Kit - SDK*) do *Firestore*, que confirma a autenticidade e obtém a resposta no cliente (FIREBASE, 2021d);
- *Storage* - Este serviço fornece o armazenamento de arquivos gerados pelo usuário. Os arquivos são armazenados em um repositório do *Google Cloud Storage*, e o usuário pode fazer *upload* e *download* dos arquivos por meio do Kit de Desenvolvimento de Software (*Software Development Kit - SDK*) do *Firestore* (FIREBASE, 2021b), e

- *Firestore* - Este serviço fornece um banco de dados na nuvem, orientado a documentos, que transfere e recebe dados em tempo real. Quando um dado é modificado, este é sincronizado entre todos os clientes da aplicação. Cada cliente utiliza o Kit de Desenvolvimento de Software (*Software Development Kit* - SDK) disponibilizado pela plataforma. Além da transferência de dados em tempo real, o *Firebase Cloud Firestore* possui escalonamento entre vários bancos de dados. A estruturação dos dados é feita no formato *JSON*, o qual permite maior flexibilidade e diferentes estruturas na construção do banco da aplicação (FIREBASE, 2021a).

### 5.3.2 Front-end

Este componente é responsável pela interface da ferramenta utilizando recursos *HTML* e *CSS* com a biblioteca *React* e com o *framework Next*. Essas tecnologias foram escolhidas devido ao vasto suporte que elas oferecem para a construção de interfaces de usuário interativas, e são descritas com mais detalhes no Capítulo 3.

### 5.3.3 API

Este componente é responsável pela conexão com o banco de dados, o *Firestore*, e pela realização do tratamento de dados necessário. Para a construção da *API RESTful*, foi escolhida a tecnologia *Node.js* com o *framework Express*, com o objetivo de realizar a comunicação entre as camadas de banco de dados e de *front-end*. Os detalhes dessas tecnologias são descritos no Capítulo 3.

### 5.3.4 Integração dos Componentes

Os serviços do *Firebase* são acessíveis por meio do Kit de Desenvolvimento de Software (*Software Development Kit* - SDK) disponibilizado pela ferramenta, mantendo uma conexão via *socket* entre dois diferentes processos. A conexão com o serviço de banco de dados (*Firebase Cloud Firestore*) é feita pela *API*, e a conexão com os serviços de autenticação (*Firebase Authentication*) e de armazenamento de arquivos (*Firebase Storage*) é feita diretamente pelo *front-end*, via *web-socket*, utilizando o navegador para sua execução.

A comunicação entre a *API* e o *front-end* ocorre por meio do protocolo *HTTP*, mantendo a arquitetura Cliente-Servidor. A *API* será *RESTful*, a qual aplica os princípios *Representational State Transfer (REST)*, que permitem diretrizes específicas na comunicação entre as duas camadas, e definem métodos para a realização das requisições, entre os quais os principais são:

- *POST* - criação de dados;



- *GET* - leitura de dados;
- *DELETE* - exclusão de dados, e
- *PUT* - atualização de dados.

## 5.4 Prova de Conceito

Esta seção é responsável por demonstrar a prova de conceito do projeto deste Trabalho de Conclusão de Curso, realizada na primeira etapa de construção da ferramenta *Backlog Maker*. Com esse objetivo, foi decidido fazer um protótipo com as principais telas da ferramenta e a configuração da automação dos ambientes de desenvolvimento do *front-end* e da API, bem como a inicialização do *Firebase* em ambos os módulos (*front-end* e API), e a integração entre eles. Esse esforço permitiu conferir uma visão mais prática sobre o presente trabalho, bem como demonstrou, na época, que o desenvolvimento encontrava-se em andamento, juntamente com os estudos mais conceituais e de levantamento teórico, conforme orientado pela metodologia de desenvolvimento definida para o projeto, e detalhada no Capítulo 4, Seção 4.2.

### 5.4.1 Protótipo

Nesta seção, é apresentado o protótipo da ferramenta. Dessa forma, a seguir, são apresentadas as principais telas do mesmo (Figuras 8 a 11). O fluxo completo de telas do protótipo pode ser encontrado no Apêndice C.

Primeiramente, é apresentada a lista dos itens do *Backlog* do Produto, com os Épicos e as respectivas *Features* associadas. Posteriormente, são apresentadas as telas de visualização das informações de um Épico, de uma *Feature* e de uma História.

Figura 8 – Tela do *Backlog* do Produto no Protótipo de Alta Fidelidade

The screenshot displays a web application interface for managing a product backlog. On the left is a dark sidebar with navigation icons and labels: PROJÉTOS, BACKLOG, SPRINTS, KANBAN, and DOCUMENTAÇÃO. The top header includes the text 'NOME DA FERRAMENTA' with a menu icon, a dropdown menu for 'Projeto X', the text 'Informações do Projeto', and user links for 'Perfil' and 'Sair'. Below the header, there are two blue buttons labeled 'Épico' and 'Feature'. A search bar contains the text 'Pesquisar Épico ou Feature'. The main content area is titled 'Backlog do Produto' and features three filter dropdowns: 'Prioridade: Todas', 'Completação: Todos', and 'Ordenar por: Maior Prioridade'. The backlog is organized into three sections: 'Épico 1' (25% progress, 65 points, high priority), 'Épico 2' (25% progress, 35 points, medium priority), and 'Épico 3' (25% progress, 40 points, low priority). Each epic section contains a list of features with their respective progress percentages and point values.

Item	Progress	Prioridade	Pontos
Épico 1	25%	Prioridade Alta	65 pontos
Feature 1	10%	Prioridade Alta	15 pontos
Feature 2	10%	Prioridade Média	10 pontos
Feature 3	0%	Prioridade Baixa	20 pontos
Feature 4	0%	Prioridade Baixa	20 pontos
Épico 2	25%	Prioridade Média	35 pontos
Épico 3	25%	Prioridade Baixa	40 pontos

1 2 ... >

CONFIGURAÇÕES

Fonte: autoria própria

Figura 9 – Tela das Informações de um Épico no Protótipo de Alta Fidelidade

The screenshot displays a web interface for managing a project. At the top, there is a navigation bar with a user profile and a 'Sair' (Logout) button. Below this, a dropdown menu shows 'Projeto X' and the page title 'Informações do Projeto'. A sidebar on the left contains navigation options: 'PROJETOS', 'BACKLOG', 'SPRINTS', 'KANBAN', and 'DOCUMENTAÇÃO', along with a 'CONFIGURAÇÕES' (Settings) button at the bottom.

The main content area is titled 'Épico 1' and contains a text description: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.' Below the description, a red bar indicates 'Prioridade Alta' (High Priority).

There are two columns of links and files. The 'Links' column contains two URLs: 'https://www.link.com.br', each with a red 'x' icon. The 'Arquivos' (Files) column contains 'file.pdf' and 'image.png', each with a red 'x' icon. A '+' button is located below each column.

The 'Features' section includes a 'Feature' button, filters for 'Prioridade' (Priority) and 'Completude' (Completion), and a 'Ordenar por' (Sort by) dropdown set to 'Maior Prioridade' (Highest Priority). A table lists four features with their respective priorities and point values:

Feature	Progress	Priority	Points
Feature 1	100%	Prioridade Alta	15 pontos
Feature 2	100%	Prioridade Média	10 pontos
Feature 3	50%	Prioridade Baixa	20 pontos
Feature 4	50%	Prioridade Baixa	20 pontos

At the bottom, there is a 'Comentários' (Comments) section with two user comments (User X and User Y) and a text input field for adding a new comment.

Fonte: autoria própria

Figura 10 – Tela das Informações de uma *Feature* no Protótipo de Alta Fidelidade

The screenshot displays a user interface for managing project features. On the left, a dark sidebar contains navigation items: PROJÉTOS, BACKLOG, SPRINTS, KANBAN, DOCUMENTAÇÃO, and CONFIGURAÇÕES. The top header shows 'Projeto X' and 'Informações do Projeto', along with 'Perfil' and 'Sair' options. The main content area is titled 'Feature 1' and contains a text description, a 'Prioridade Alta' label, a list of 'Features Bloqueantes' (Feature X and Feature Y), and sections for 'Links' and 'Arquivos'. Below these is a 'Histórias' section with filters for priority, sprint, and status, and a list of user stories with their respective priorities and statuses. At the bottom, there is a 'Comentários' section with a text input field.

Fonte: autoria própria

Figura 11 – Tela das Informações de uma História no Protótipo de Alta Fidelidade

Eu, como  , desejo  ,  
para que eu possa  .

**Prioridade Alta**

Links:

<a href="https://www.link.com.br">https://www.link.com.br</a>	x
<a href="https://www.link.com.br">https://www.link.com.br</a>	x

Arquivos:

<a href="#">file.pdf</a>	x
<a href="#">image.png</a>	x

Histórias Bloqueantes

<a href="#">História X</a>	x
<a href="#">História Y</a>	x

Revisores:

User X	x
User X	x

**Critérios de Aceitação**

<input type="checkbox"/> Critério 1	x
<input type="checkbox"/> Critério 2	x

**Tarefas Técnicas**

<input type="checkbox"/> Tarefa 1	x
<input type="checkbox"/> Tarefa 2	x

**Comentários**

Fonte: autoria própria

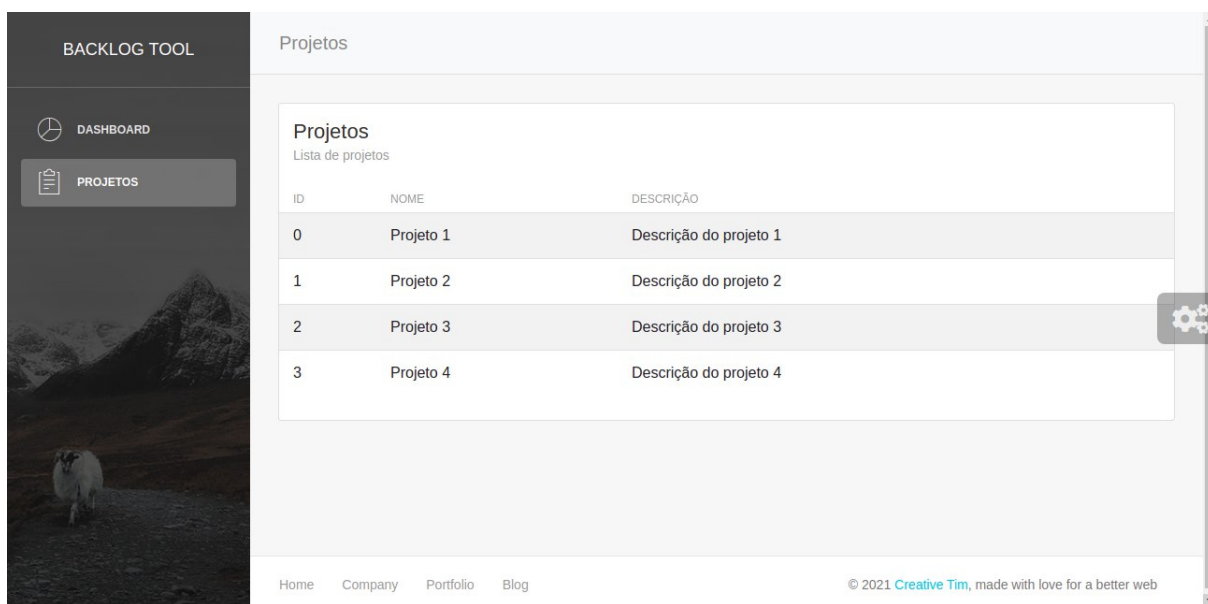
#### 5.4.2 Configuração do Ambiente de Desenvolvimento e Integração entre os Componentes da Solução

Com o objetivo de fazer a configuração inicial do projeto, foi realizada a automação dos ambientes de desenvolvimento do *front-end*, sendo este desenvolvido com a utilização do *React*, e da API, utilizando o *Node.js*. Essa automação foi feita com o apoio das ferramentas *Docker* e *Docker Compose*, como citado na Seção 3.1.

Além disso, a fim de validar a comunicação entre os componentes da solução, foi realizada a integração entre os mesmos. A Figura 12 indica uma lista de projetos recuperada do banco de dados, por meio da API, e apresentada em uma página do *front-end*. Para fazer essa integração, foi necessária a configuração do projeto no *Firebase*, e a inicialização da ferramenta por meio do SDK, tanto no *front-end* quanto na API.

Os códigos das automações dos ambientes de desenvolvimento podem ser encontrados nos Apêndices D e E; os códigos da inicialização do *Firebase* no *front-end* e na API, no Apêndice F; e os códigos da integração entre os componentes, no Apêndice G.

Figura 12 – Lista de projetos recuperada por meio da API e apresentada no *front-end*



Fonte: autoria própria

## 5.5 Backlog da Ferramenta

A versão final do *Backlog* do Produto da ferramenta *Backlog Maker* está disponível e pode ser acessada por meio da URL: <<https://docs.google.com/spreadsheets/d/1cub3ZT9ne4eChJjegt4B0nLRaKZw86eIXfGP0FjlJTU/edit?usp=sharing>>.

## 5.6 A solução

Com base nos insumos e requisitos levantados, foi desenvolvida a ferramenta *Backlog Maker*, que está disponível e pode ser acessada por meio da URL: <<https://backlog-maker.vercel.app>>.

Esta seção apresenta, detalhadamente, as funcionalidades que a ferramenta *Backlog Maker* proporciona aos seus usuários. Todas as telas, atualmente disponíveis na aplicação, podem ser visualizadas no Apêndice H.

### 5.6.1 Visualização e Criação de Projetos

O primeiro passo que o usuário deve fazer na ferramenta é criar um projeto. Para isso, ele deve descrever detalhadamente o projeto com as seguintes informações: nome, descrição, prioridade, *links* e arquivos pertinentes ao projeto, o papel do criador do projeto (*Product Owner*, *Scrum Master*, desenvolvedor ou cliente). Esses campos podem ser visualizados na Figura 13. Ademais, com o objetivo de deixar a descrição mais completa, há uma seção para que o usuário escreva o resumo do projeto. Dessa forma, qualquer novo integrante da equipe irá entender o que é o projeto, qual o seu objetivo, de onde surgiu e qual o seu público-alvo. Esta seção está apresentada na Figura 14.

Figura 13 – Principais Campos de Criação de um Projeto no *Backlog Maker*

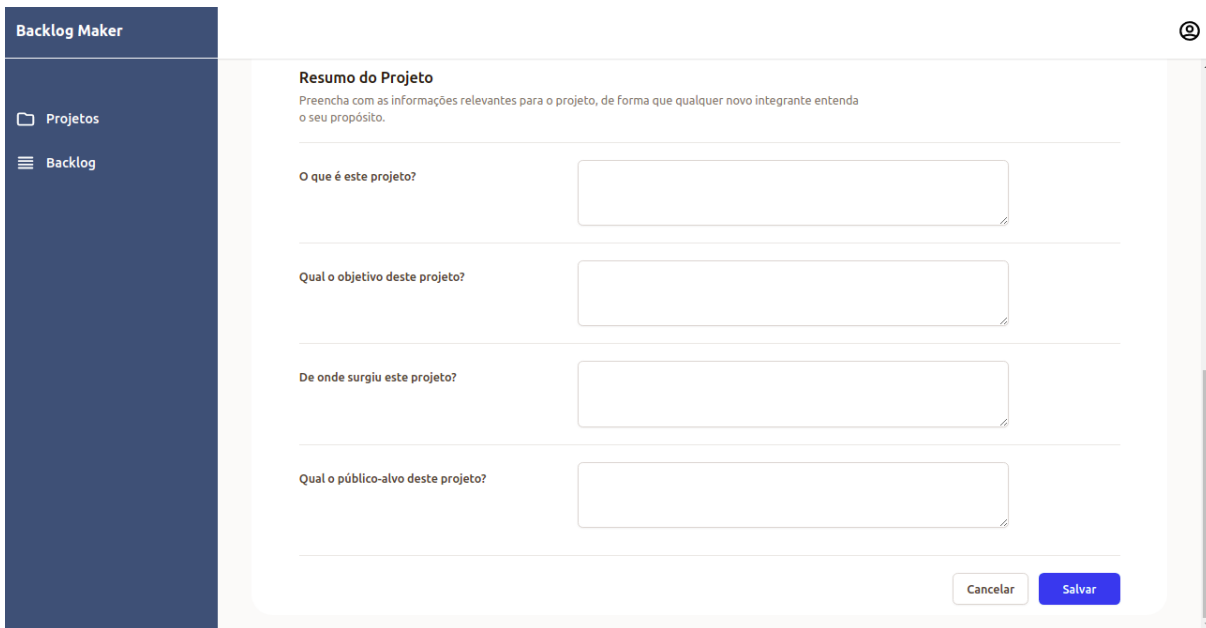
A imagem mostra a interface de usuário para criar um novo projeto no sistema Backlog Maker. O formulário é dividido em seções horizontais. No topo, há um campo de texto para o nome do projeto. Abaixo dele, uma área de texto maior para a descrição, com uma dica para incluir as principais informações. Segue um menu suspenso para a prioridade, atualmente configurado para 'Alta'. Depois, um campo de texto para links, com um ícone de adição (+) abaixo. A seção de arquivos apresenta uma área de upload com um ícone de pasta e uma dica para clicar ou arrastar arquivos (PDF, DOC, DOCX, PNG) menores que 10MB. Por fim, há um menu suspenso para definir o papel do usuário, com 'Product Owner' selecionado.

Fonte: autoria própria

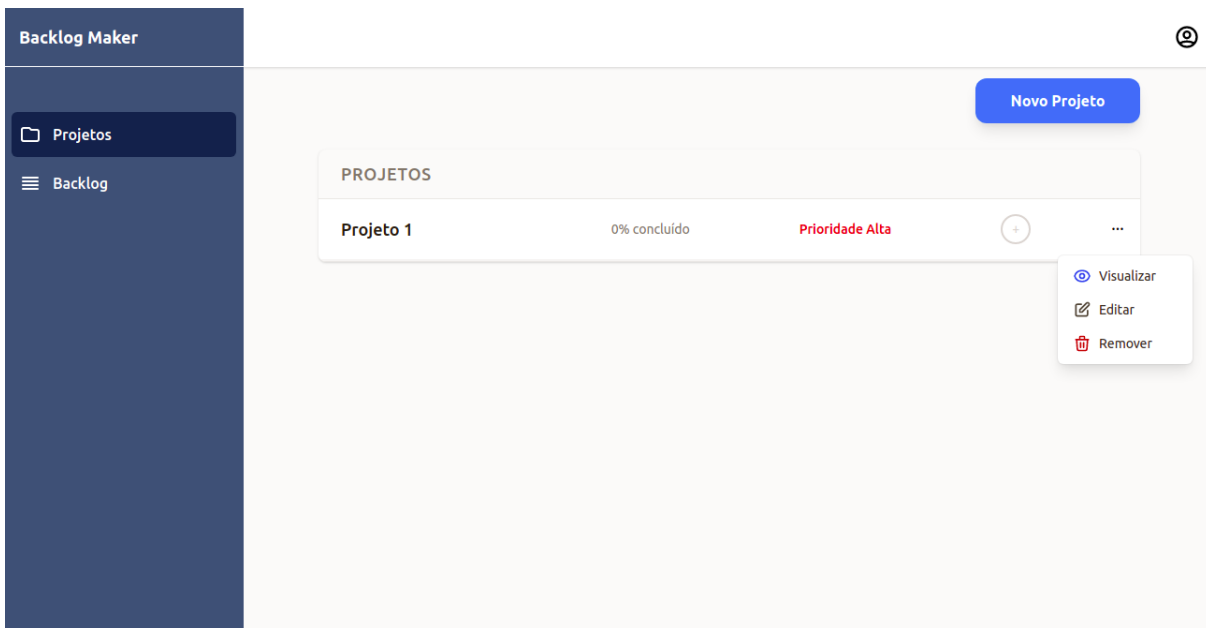
A visualização dos projetos cadastrados pelo usuário é feita por meio de uma lista, na qual o usuário pode visualizar o nome do projeto, a sua completude, a sua prioridade, adicionar novos membros e realizar algumas ações, como visualizar, editar e remover projeto. A Figura 15 mostra essa lista de projetos e o botão de criação de novos projetos. No Apêndice H, está apresentada a Figura 51 que exibe a tela de adição de novos membros ao projeto, inserindo o *e-mail* e o papel do novo usuário. Esta ação dispara um serviço para o envio do *e-mail* para que a pessoa possa aceitar o convite, tornando-se membro do projeto. No Apêndice H, a Figura 52 apresenta o *e-mail* enviado para o convidado.

### 5.6.2 Visualização e Gerenciamento do *Backlog* do Produto

Para visualizar o *Backlog* do Produto de um projeto específico, o usuário deve clicar em um dos itens da lista de projetos, apresentada na tela de visualização dos projetos (Figura 15), ou na guia "*Backlog*", disponível na *sidebar* da ferramenta. Também é possível

Figura 14 – Tela de Criação do Resumo do Projeto no *Backlog Maker*

Fonte: autoria própria

Figura 15 – Tela de Visualização de Projetos no *Backlog Maker*

Fonte: autoria própria

fazer a alteração do projeto atual por meio do campo disponível na *navbar*, que mostra a lista de projetos e permite que o usuário selecione um deles, este campo é representado na Figura 58.

A tela de visualização do *Backlog* do Produto (Figura 16) mostra todos os itens do *Backlog* em uma lista, na qual os itens aninhados são mostrados em seus respectivos níveis, ou seja, épicos podem ser expandidos para visualização de suas *features* que, por sua vez,



podem ser expandidas para visualização de suas histórias. Além disso, o usuário pode visualizar *bugs* cadastrados no *Backlog* do Produto. Nessa mesma tela, o usuário pode iniciar a criação de um novo épico ou de um novo *bug* ou realizar as ações disponíveis em cada item do *Backlog*, tais como: editar, visualizar e excluir itens do *Backlog* (épico, *feature* ou história ou *bug*), criar uma nova *feature* a partir de um épico específico ou uma nova história a partir de uma *feature* específica.

Os formulários de criação e edição dos épicos possuem os campos de nome, descrição e prioridade, além de permitir a adição de *links* e arquivos pertinentes ao épico sendo criado ou editado; os formulários de criação e edição das *features* possuem os mesmos campos presentes nos formulários dos épicos, mais a possibilidade de adição de *features* bloqueantes. As histórias possuem os campos de descrição da história (com ator, ação e propósito), de observações, prioridade e estimativa, além da possibilidade de adição de histórias bloqueantes, revisores, *links* e arquivos, da adição de membros e da criação de critérios de aceitação e tarefas técnicas. Além disso, há a opção de Concluir/Fechar História ou Abrir História, com esta opção o usuário pode finalizar ou não a história quando for pertinente.

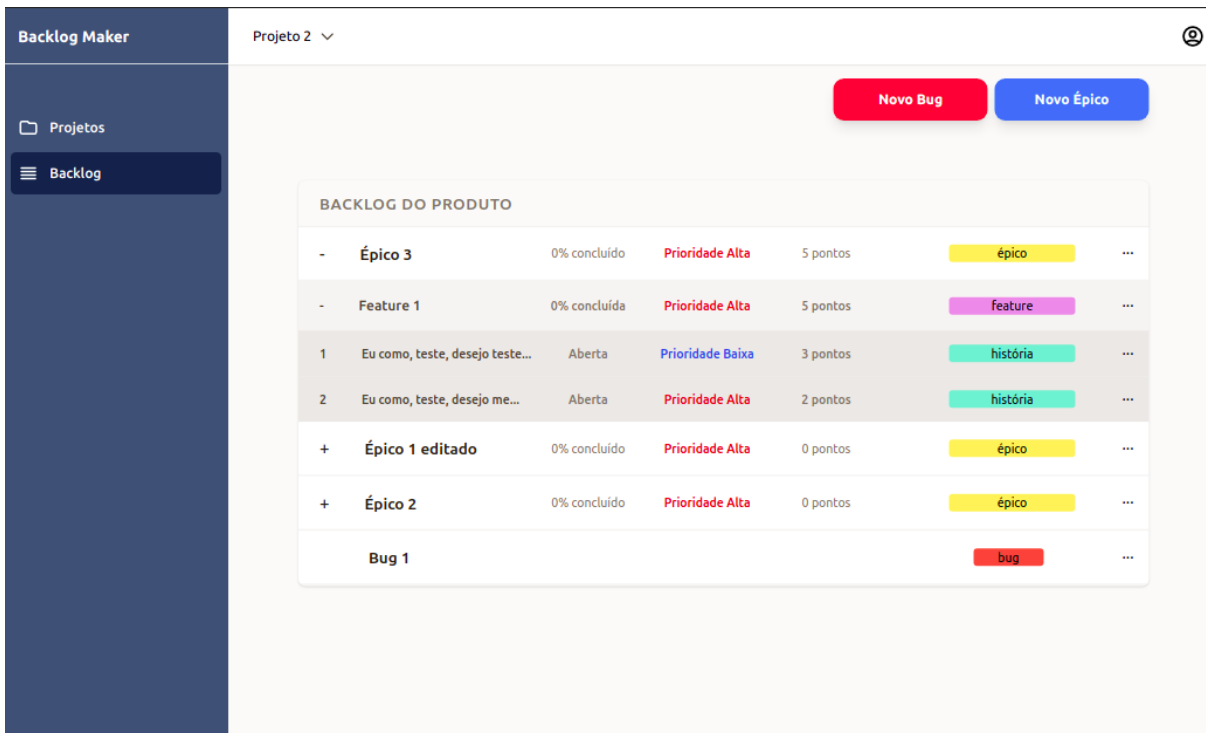
O formulário de criação e edição de *bugs* possui os campos de nome, resultado esperado, resultado obtido, *links* e adição de arquivos que sejam relevantes para explicar a ocorrência do *bug*.

Cada item do *Backlog* do Produto possui o histórico de edições para que o rastro do requisito seja mantido para todos os membros do projeto. As telas da aplicação que mostram essas funcionalidades citadas podem ser vistas no Apêndice H.

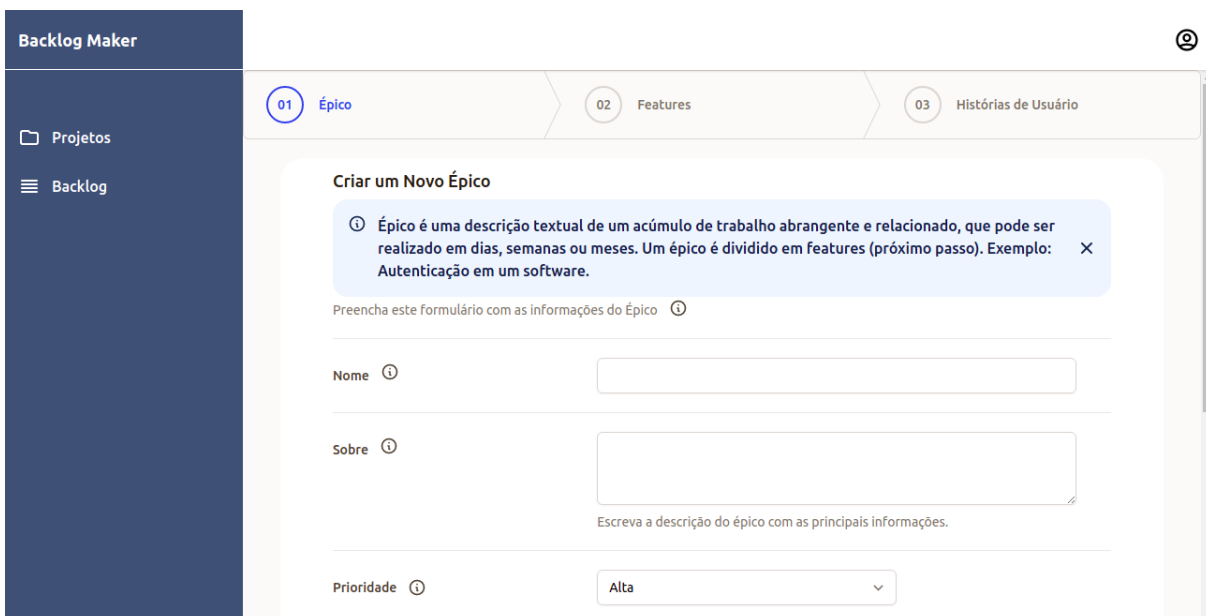
### 5.6.3 Guia e Textos de Ajuda

Com o objetivo de guiar o usuário a criar um *Backlog* que seja conciso e eficiente para um determinado projeto, foi criado o guia de passo a passo para que o usuário gere níveis de granularidade corretos. Além disso, foram criados ícones de ajuda, que, quando clicados, mostram informações que auxiliam o usuário sobre o que ele deve inserir nos campos relacionados. Estes textos de ajuda foram baseados no livro *Product Backlog Building* de Caroli e Aguiar (2021).

As Figuras 17, 18 e 19 exibem o guia com o passo a passo para a criação do *Backlog* com três níveis de granularidade (Épico, *Feature* e Histórias de Usuário), sendo que no nível de Histórias de Usuário, pode-se criar as tarefas técnicas necessárias. As telas apresentadas demonstram os textos de ajuda, que são apresentados quando o usuário clica nos ícones de informação. Essas descrições têm como objetivo auxiliar o usuário sobre o que é cada nível e cada campo dos formulários.

Figura 16 – Tela de Visualização do *Backlog* do Produto no *Backlog Maker*

Fonte: autoria própria

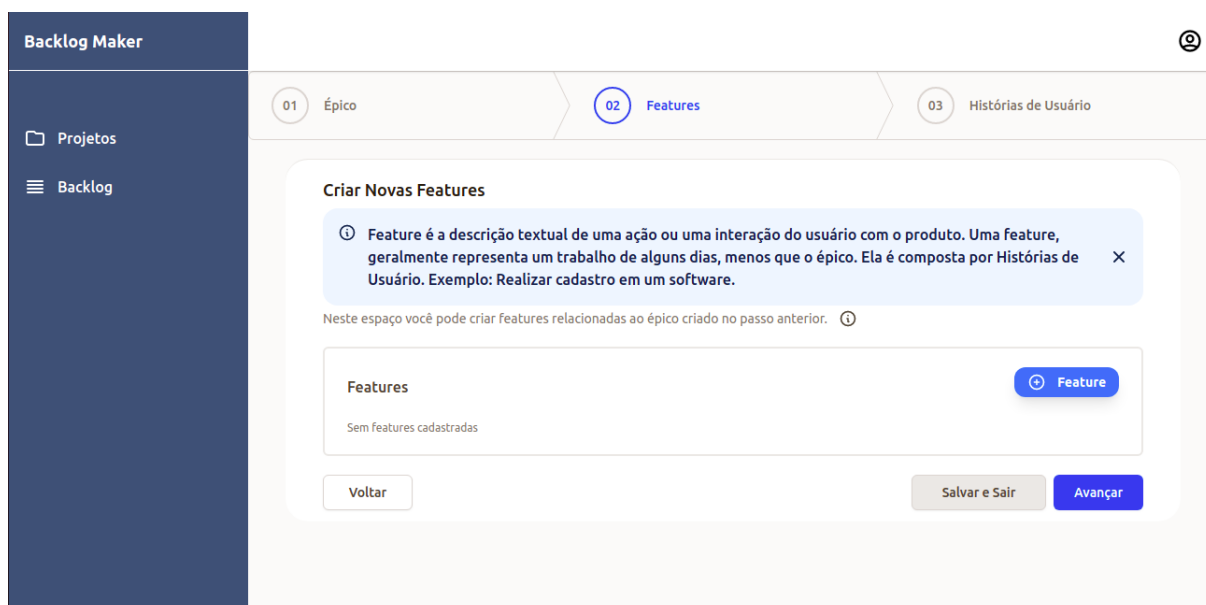
Figura 17 – Guia com o passo a passo para a criação de um *Backlog* - Passo 1 (Criação de um Épico) - Versão Final da Ferramenta

Fonte: autoria própria

#### 5.6.4 Comentários em Épico, *Feature*, História de Usuário e *Bugs*

A Figura 20 apresenta a visualização de comentários que os membros do projeto podem inserir em épicos, *features*, histórias de usuário e *bugs*, com o objetivo de manter

Figura 18 – Guia com o passo a passo para a criação de *Features* - Passo 2 (Criação de *Features*) - Versão Final da Ferramenta



Fonte: autoria própria

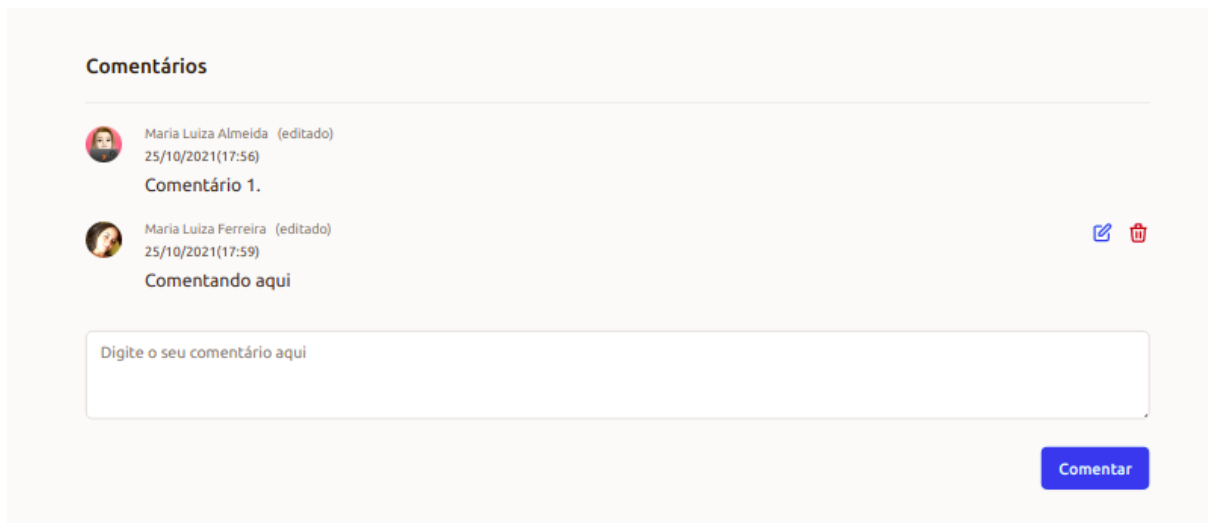
Figura 19 – Guia com o passo a passo para a criação de Histórias de Usuário - Passo 3 (Criação de Histórias) - Versão Final da Ferramenta



Fonte: autoria própria

o rastro dos requisitos. O usuário pode inserir, editar e excluir comentário. Além disso, pode visualizar o seu comentário e o de outros membros do projeto. Cada comentário é seguido com o avatar do criador, o seu nome, a data e o horário de criação ou edição, bem como há a indicação quando este é editado.

Figura 20 – Comentários de épico, *feature*, história de usuário e *bug* - Versão Final da Ferramenta



Fonte: autoria própria

## 5.7 Resumo do Capítulo

Neste capítulo, foi detalhada a ferramenta *Backlog Maker*, obtida neste Trabalho de Conclusão de Curso, sendo essa uma ferramenta que confere apoio à construção e à manutenção do *Backlog* do Produto. Para isso, foi apresentada uma contextualização, com foco no estudo realizado acerca das ferramentas similares existentes no mercado; a versão inicial da *Baseline* de Requisitos; a arquitetura da solução, com detalhamento dos componentes presentes na mesma; a prova de conceito, que consistiu na construção de um protótipo de alta fidelidade, e na configuração dos ambientes de desenvolvimento, juntamente com a integração entre os módulos da solução; o *Backlog* da ferramenta, com requisitos mais detalhados, e, por fim, a solução e a versão final da ferramenta.

## 6 Resultados

Este capítulo tem por objetivo apresentar os resultados deste trabalho, bem como descrever as atividades de análise aplicadas sobre a ferramenta *Backlog Maker*, descrita no capítulo anterior (Capítulo 5), de forma a apresentar os resultados dessas atividades, e as ações de melhoria identificadas.

A Seção 6.1 expõe o resultado do empenho teórico realizado na primeira fase deste trabalho, que culminou na construção do Quadro de Problemas vs. Potenciais Soluções (Quadro 16, presente no Apêndice B).

Já o procedimento de análise, utilizado na última fase deste trabalho, foi fundamentado na metodologia de Pesquisa-Ação, como acordado no Capítulo 4. Dois ciclos de análise foram realizados, sendo apresentadas a forma como se deu a coleta e a interpretação dos dados, além das operacionalizações advindas dessas atividades, com a indicação de quais delas foram incorporadas ao produto final.

O primeiro ciclo da Pesquisa-Ação, cujos detalhes estão descritos na Seção 6.2, deu-se pela realização de uma autoanálise, sendo esta realizada pelas próprias autoras do projeto, baseando-se no referencial teórico (Capítulo 2) e na *Baseline* de Requisitos apresentada no Capítulo 5, Seção 5.2. Já o segundo ciclo, apresentado na Seção 6.3, foi aplicado sobre uma amostra de voluntários com as características do Público-Alvo deste projeto, o que possibilitou a coleta de *feedback* externo, transmitido por meio de um questionário, preenchido pelos participantes após a utilização da ferramenta *Backlog Maker*. Os resultados dos dois ciclos de análise foram combinados e convertidos em operacionalizações concretas, que possibilitam a aplicação das melhorias identificadas, em ambos os ciclos, na ferramenta *Backlog Maker*. Tais operacionalizações são apresentadas na Seção 6.4. Por fim, tem-se o resumo do capítulo.

### 6.1 Quadro Problemas vs. Potenciais Soluções

Buscando responder às questões de pesquisa, acordadas no Capítulo 1, Seção 1.2, e cumprir com dois dos objetivos específicos do trabalho, acordados na Seção 1.4.2 do mesmo Capítulo, uma análise criteriosa da literatura foi realizada e, como resultado dessa análise, o Quadro 16 foi construído. Nele, são identificados os principais problemas, apontados pela literatura, na construção e na manutenção do *Backlog* do Produto, bem como são endereçadas, a cada um dos problemas apresentados, as práticas que podem configurar potenciais soluções para esses problemas. O Quadro, presente no Apêndice B, é considerado um resultado da primeira fase deste trabalho, tendo em vista que serviu para

o embasamento do trabalho como um todo, e ofereceu apoio às atividades de elicitação dos requisitos da ferramenta construída: o *Backlog Maker*.

## 6.2 Primeiro Ciclo de Pesquisa-Ação

Como mencionado anteriormente, o primeiro ciclo foi realizado por meio de uma autoanálise, feita pelas próprias autoras do projeto. Para isso, foram estabelecidos critérios de análise, oriundos do Referencial Teórico deste trabalho (Capítulo 2) e do capítulo anterior (Capítulo 5).

O quadro a seguir apresenta os critérios adotados, assim como o resultado da análise e as ações sugeridas a partir disso. Os critérios de C01 a C12 são provenientes do quadro de Problemas vs. Potenciais Soluções, presente no Apêndice B; os critérios de C13 a C28 são os presentes na Seção 2.3.2 do Capítulo 2, e os demais critérios (C29 a C43) vieram da *Baseline* de Requisitos, estabelecida na Seção 5.2 do Capítulo 5.

Quadro 13 – Resultados da Análise no Primeiro Ciclo

ID	CRITÉRIO	ANÁLISE	AÇÕES
C01	Envolvimento do cliente	Ainda não está claro o envolvimento do cliente dentro da plataforma. O que foi desenvolvido até agora permite que o cliente apenas visualize o <i>Backlog</i> , sem nenhuma interação com a equipe. As funcionalidades pertinentes à adição de comentários nos vários níveis do <i>Backlog</i> ajudariam a resolver o problema.	- Criar seções de comentários nos Épicos, <i>Features</i> e Histórias, com possibilidade de interação do cliente com todos da equipe.
C02	Documentação dos itens do <i>Backlog</i>	A documentação dos itens do <i>Backlog</i> é possível por meio dos campos de descrição, links e arquivos disponibilizados na plataforma.	-

Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C03	Coleta de <i>feedback</i>	A coleta de <i>feedback</i> não está explícita na plataforma. A adição de comentários pode ajudar com isso. Seria interessante a disponibilização de campos específicos para essa finalidade, para uso de clientes e revisores.	- Criar seções de comentários nos Épicos, <i>Features</i> e Histórias, com a possibilidade de coletar o <i>feedback</i> dos envolvidos por meio dessas seções, e - Criar campos para coleta de <i>feedback</i> de clientes e revisores.
C04	Revisão das Histórias de Usuário de acordo com critérios de aceitação	A ferramenta disponibiliza a adição tanto de critérios de aceitação quanto de revisores às Histórias.	-
C05	Níveis de granularidade do <i>Backlog</i>	Os níveis de granularidade do <i>Backlog</i> na ferramenta configuram-se em Épico- <i>Feature</i> -História, sendo uma estrutura engessada, sem possibilidade de mudança. Tanto a literatura quanto o conhecimento advindo da experiência das autoras apontam essa como sendo uma configuração ideal para o <i>Backlog</i> do Produto. No entanto, seria interessante que a equipe tivesse a liberdade de escolher com qual estrutura deseja trabalhar, tendo em vista a variedade de cenários possíveis.	- Possibilitar a escolha dos níveis de granularidade do <i>Backlog</i> .

Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C06	Rastreabilidade dos requisitos por meio dos itens do <i>Backlog</i>	Os campos que permitem a documentação dos itens do <i>Backlog</i> ajudam também na manutenção do rastro dos requisitos. No entanto, essa rastreabilidade poderia ser melhorada com a disponibilização do histórico de edição de cada item.	- Disponibilizar o histórico de edição dos itens do <i>Backlog</i> .
C07	Identificação dos responsáveis pelo desenvolvimento das Histórias	A ferramenta não permite que sejam adicionados membros às Histórias.	- Permitir a adição de membros do projeto às Histórias, para que os responsáveis sejam identificados.
C08	Identificação de dependência entre os itens do <i>Backlog</i>	A ferramenta permite que sejam adicionadas as <i>Features</i> e as Histórias bloqueantes nos mesmos níveis de granularidade. A mesma funcionalidade não existe para os Épicos, o que poderia ser acrescentado. Seria interessante que o usuário pudesse adicionar itens bloqueantes de níveis de granularidade diferentes, para casos em que ocorra dependência entre dois itens de níveis diferentes.	- Permitir a adição de itens bloqueantes aos Épicos, e - Permitir a adição de itens bloqueantes de níveis diferentes.
Continua na próxima página			



Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C09	Estimativa dos itens do <i>Backlog</i>	A ferramenta permite que as Histórias de usuário sejam estimadas de acordo com a sequência de <i>Fibonacci</i> . Como o padrão do <i>Scrum</i> é a estimativa utilizando os números dessa sequência, a ferramenta cobre bem a necessidade, mas seria interessante que a equipe tivesse a liberdade de escolher como estimar os itens do <i>Backlog</i> .	- Permitir a escolha das estimativas possíveis para os itens do <i>Backlog</i> .
C10	Priorização dos itens do <i>Backlog</i>	A ferramenta permite que todos os itens do <i>Backlog</i> , nos seus diferentes níveis (Épicos, <i>Features</i> e Histórias), possam ser priorizados apenas com prioridade "alta", "média" ou "baixa". Seria interessante que a equipe tivesse a liberdade de escolher os graus de prioridade para os itens do <i>Backlog</i> .	- Permitir a escolha dos graus de priorização dos itens do <i>Backlog</i> .
Continua na próxima página			

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C11	Compartilhamento de conhecimento	Tanto os projetos quanto os itens do <i>Backlog</i> possuem campos que permitem a documentação dos mesmos, o que contribui para o compartilhamento de conhecimento entre a equipe. No entanto, as adições de uma <i>wiki</i> aos projetos e de comentários aos itens do <i>Backlog</i> poderiam melhorar ainda mais a disseminação do conhecimento por toda a equipe.	<ul style="list-style-type: none"> <li>- Disponibilizar uma <i>wiki</i> para os projetos, e</li> <li>- Criar seções de comentários nos Épicos, <i>Features</i> e Histórias, de modo a facilitar a disseminação de conhecimento entre a equipe.</li> </ul>
C12	Separação entre requisitos técnicos e de qualidade	A ferramenta não permite que os itens do <i>Backlog</i> sejam separados em requisitos técnicos e de qualidade. A adição de <i>tags</i> aos itens do <i>Backlog</i> permitiriam essa separação. Além disso, a possibilidade de criação de <i>bugs</i> tanto ajudaria nessa separação, quanto agregaria valor ao <i>Backlog</i> criado, pela identificação de itens que representam falhas do produto em desenvolvimento.	<ul style="list-style-type: none"> <li>- Possibilitar a adição de <i>tags</i> aos itens do <i>Backlog</i>, e</li> <li>- Possibilitar a adição de <i>bugs</i> ao <i>Backlog</i>.</li> </ul>
			Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C13	Fornecimento de uma visão geral do produto	É possível que se tenha uma visão geral do produto na tela de <i>Backlog</i> , onde são mostrados os Épicos e <i>Features</i> criadas. No entanto, ainda não há uma exibição das Histórias na mesma tela, o que agregaria na visualização das funcionalidades do produto. Além disso, a separação da exibição de itens concluídos e não concluídos não existe ainda, e a <i>wiki</i> do projeto ajudaria para uma documentação que desse uma visão geral do produto.	<ul style="list-style-type: none"> <li>- Adicionar a visualização das Histórias na tela de <i>Backlog</i>;</li> <li>- Exibir os Épicos concluídos e não concluídos com <i>tags</i> diferentes na tela de <i>Backlog</i>;</li> <li>- Exibir as <i>Features</i> concluídas e não concluídas com <i>tags</i> diferentes na tela de <i>Backlog</i>;</li> <li>Exibir as Histórias concluídas e não concluídas com <i>tags</i> diferentes na tela de <i>Backlog</i>, e</li> <li>- Disponibilizar uma <i>wiki</i> para os projetos.</li> </ul>
			Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C14	Fornecimento de um visão técnica do produto	A visão técnica do produto é possível ser obtida a partir das tarefas cadastradas em cada História e da documentação dos itens do <i>Backlog</i> . No entanto, isso não é de fácil percepção na ferramenta. Uma maneira de contornar o problema seria a criação de uma seção dedicada às informações técnicas do produto (sendo no próprio projeto ou até mesmo em uma <i>wiki</i> do mesmo). A integração com ferramentas de gerenciamento de código para mostrar os detalhes de implementação do produto também ajudaria nessa questão.	<ul style="list-style-type: none"> <li>- Possibilitar a adição de <i>bugs</i> ao <i>Backlog</i>;</li> <li>- Disponibilizar uma <i>wiki</i> para os projetos, com seção dedicada a informações técnicas do produto, e</li> <li>- Fazer a integração com as plataformas de gerenciamento de código (<i>GitHub</i>, <i>GitLab</i>, <i>Bitbucket</i>, entre outras), de maneira a mostrar na plataformas detalhes de implementação tais como <i>issues</i> do repositório, estado de <i>build</i>, cobertura de testes, informações de <i>branches</i>, entre outros.</li> </ul>
Continua na próxima página			

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C15	Apoio à navegação	<p>A navegação na ferramenta está um pouco dificultada pelo tempo de resposta das páginas às ações do usuário. Há uma percepção de que as ações não foram concluídas quando, na verdade, os dados estão sendo carregados para que haja a mudança na página. Seria interessante a adição de um componente "loading" nas páginas onde isso ocorre.</p> <p>Além disso, a navegação por entre os itens do <i>Backlog</i> está confusa. <i>Features</i> sendo criadas apenas em páginas de edição de Épicos e Histórias em páginas de edição de <i>Features</i> é um exemplo disso. De maneira geral, o fluxo está bagunçado e não é intuitivo. A organização do fluxo de navegação entre os itens e a adição de um guia para o usuário são imprescindíveis para mitigar esse problema.</p>	<ul style="list-style-type: none"> <li>- Adicionar o componente de "loading" nas páginas onde há carregamento de dados após interações dos usuários;</li> <li>- Organizar o fluxo de criação, visualização e edição dos itens do <i>Backlog</i>, e</li> <li>- Adicionar um guia de navegação para facilitar a criação, visualização e edição dos itens do <i>Backlog</i>.</li> </ul>
C16	Apoio ao planejamento	<p>De uma maneira geral, a ferramenta fornece apoio para o planejamento do <i>Backlog</i>. No entanto, faltam as funcionalidades de planejamento de <i>Sprint</i>, com a criação do <i>Backlog</i> e visualização do <i>Kanban</i> da <i>Sprint</i>.</p>	<ul style="list-style-type: none"> <li>- Possibilitar a criação do <i>Backlog</i> da <i>Sprint</i>, e</li> <li>- Disponibilizar a visualização e alteração do <i>Kanban</i> da <i>Sprint</i>.</li> </ul>

Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C17	Possibilidade de respostas sobre o progresso	Análogo ao item C03	-
C18	Possibilidade de agrupamento de requisitos em coleções	Os níveis de granularidade do <i>Backlog</i> configurados na ferramenta cobrem essa necessidade.	-
C19	Apoio à especificação detalhada das tarefas	A especificação das tarefas é disponibilizada na ferramenta por meio dos níveis de granularidade do <i>Backlog</i> , dos campos de documentação dos itens do <i>Backlog</i> e pelas tarefas que podem ser adicionadas em cada História. Os comentários em cada item do <i>Backlog</i> poderiam ser úteis para esse tipo de especificação também.	- Criar seções de comentários nos Épicos, <i>Features</i> e Histórias, de forma a facilitar a especificação das tarefas.
C20	Definição de critérios de aceitação	Análogo ao item C04.	-
C21	Definição de critérios de qualidade para os níveis mais altos de granularidade	A adição de critérios de qualidade é possível apenas nas Histórias de Usuário, por meio dos seus critérios de aceitação.	- Adicionar critérios de qualidade/aceitação em <i>Features</i> e Épicos.
C22	Possibilidade de criação de itens no <i>Backlog</i> a todas as partes interessadas	A criação de itens no <i>Backlog</i> só é permitida a usuários administradores do projeto e aos membros do projeto que têm o papel de PO, SM ou Desenvolvedor, não sendo permitida a clientes.	- Permitir que clientes criem itens no <i>Backlog</i> .

Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C23	Possibilidade de pesquisa e filtro dos itens do <i>Backlog</i>	A ferramenta ainda não permite a busca e a filtragem dos itens do <i>Backlog</i> .	- Permitir busca e filtragem dos itens do <i>Backlog</i> .
C24	Priorização dos itens do <i>Backlog</i>	Análogo ao item C10.	-
C25	Estimativa dos itens do <i>Backlog</i>	Análogo ao item C09.	-
C26	Interface padronizada para o <i>Backlog</i> do Produto e da <i>Sprint</i>	A ferramenta ainda não permite a criação do <i>Backlog</i> da <i>Sprint</i> .	- Possibilitar a criação do <i>Backlog</i> da <i>Sprint</i> , e - Disponibilizar a visualização do <i>Backlog</i> da <i>Sprint</i> de forma padronizada com o <i>Backlog</i> do Produto.
C27	Identificação de dependência entre os itens do <i>Backlog</i>	Análogo ao item C08.	-
C28	Comunicação entre os membros da equipe	A ferramenta ainda não permite a comunicação explícita entre os membros da equipe. A adição de comentários nos itens do <i>Backlog</i> ajudaria a mitigar esse problema.	- Criar seções de comentários nos Épicos, <i>Features</i> e Histórias, de forma a possibilitar a comunicação entre os membros da equipe.
C29	Tipos de perfis diferentes para SM, PO, Desenvolvedores e Clientes	A depender do projeto, o usuário tem permissões diferentes no <i>Backlog</i> , definidas pelo seu papel. Administradores do projeto, independentemente do papel, tem todas as permissões de alteração do <i>Backlog</i> . Essa adaptação fez-se necessária por conta da possibilidade de um mesmo usuário ter papéis diferentes em diferentes projetos.	-

Continua na próxima página

Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C30	Autenticação com Google ou utilizando E-mail e Senha	A autenticação na ferramenta é simples e pode ser feita utilizando <i>E-mail</i> e senha ou Google.	-
C31	CRUD de projetos	Os projetos podem ser criados, editados, visualizados e excluídos na ferramenta.	-
C32	Adição de resumo aos projetos	Os projetos contam com campos de resumo.	-
C33	Adição de <i>links</i> e arquivos aos projetos	Os projetos contam com campos de adição de <i>links</i> e arquivos.	-
C34	CRUD de Épicos	Os Épicos podem ser criados, editados, visualizados e excluídos na ferramenta.	-
C35	CRUD de <i>Features</i>	As <i>Features</i> podem ser criadas, editadas, visualizadas e excluídas na ferramenta.	-
C36	CRUD de Histórias	As Histórias podem ser criadas, editadas, visualizadas e excluídas na ferramenta.	-
C37	Informações das Histórias (revisores, Histórias bloqueantes, membros, estimativa, priorização, <i>tags</i> , histórico de atualização, critérios de aceitação, tarefas, comentários, contato com o cliente)	As Histórias contam com a adição de revisores, Histórias bloqueantes, estimativa, priorização, critérios de aceitação e tarefas, mas não contam ainda com a adição de membros, <i>tags</i> , visualização de histórico de atualização, comentários e contato com clientes.	<ul style="list-style-type: none"> <li>- Permitir a adição de membros do projeto às Histórias;</li> <li>- Possibilitar a adição de <i>tags</i> aos itens do <i>Backlog</i>;</li> <li>- Disponibilizar o histórico de atualização das Histórias, e</li> <li>- Criar seções de comentários nas Histórias, de forma a possibilitar o contato com os clientes e com outros membros da equipe.</li> </ul>
C38	<i>Backlog</i> da <i>Sprint</i>	A ferramenta ainda não possibilita a criação do <i>Backlog</i> da <i>Sprint</i> .	- Possibilitar a criação do <i>Backlog</i> da <i>Sprint</i> .

Continua na próxima página



Quadro 13 – continuação da página anterior

ID	CRITÉRIO	ANÁLISE	AÇÕES
C39	Visualização do <i>Kanban</i> da <i>Sprint</i>	A ferramenta ainda não possibilita a visualização do <i>Kanban</i> da <i>Sprint</i> .	- Disponibilizar a visualização e alteração do <i>Kanban</i> da <i>Sprint</i> .
C40	Lista de desejos dos projetos	A ferramenta ainda não possibilita a criação da lista de desejos para os projetos.	- Possibilitar a criação de uma lista de desejos para os projetos.
C41	<i>Wiki</i> para os projetos	A ferramenta ainda não disponibiliza uma <i>wiki</i> para os projetos.	- Disponibilizar uma <i>wiki</i> para os projetos.
C42	Orientações e ajudas para a construção do <i>Backlog</i>	As orientações e ajudas para a construção do <i>Backlog</i> não estão disponíveis em todos os níveis.	- Adicionar textos de ajuda e orientação para a construção do <i>Backlog</i> em todos os níveis
C43	Configurações (personalização de priorização e estimativas, criação de <i>tags</i> , entre outros)	A ferramenta ainda não disponibiliza uma seção de configurações para os projetos.	- Disponibilizar uma seção de configurações para os projetos, com possibilidade de personalização de priorização, estimativas e criação de <i>tags</i> .

Fonte: autoria própria

## 6.3 Segundo Ciclo de Pesquisa-Ação

O segundo ciclo da Pesquisa-Ação foi realizado com um grupo de voluntários com as características do Público-Alvo deste projeto, com o objetivo de obter *feedbacks* da comunidade externa. A pesquisa foi realizada da seguinte forma: foi dado o acesso para cada participante ao *Backlog Maker* para que eles criassem um *Backlog* com a utilização da ferramenta e respondessem um formulário para a coleta do *feedback*. Para facilitar o entendimento dos usuários, as autoras deste trabalho providenciaram um contexto de projeto para que eles pudessem criar o *Backlog* a partir da circunstância dada, disponível em <[https://docs.google.com/document/d/11F1FQZ6Fh2cvOZ\\_ZfpwlfBusYvg\\_eHeyv2o\\_eN1AEok/edit?usp=sharing](https://docs.google.com/document/d/11F1FQZ6Fh2cvOZ_ZfpwlfBusYvg_eHeyv2o_eN1AEok/edit?usp=sharing)>.

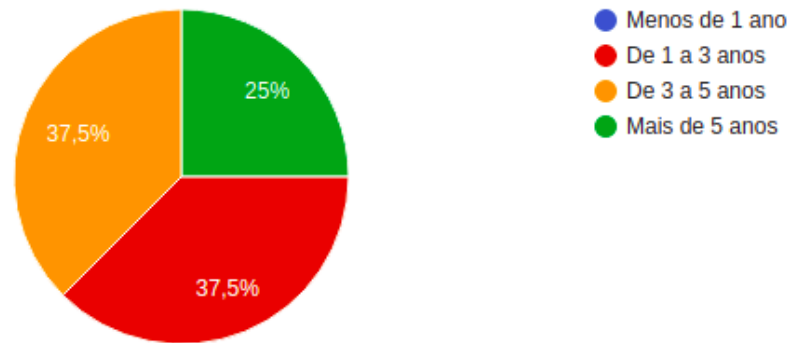
O total foi de oito participantes da pesquisa. A fim de obter o perfil dessas pessoas, havia no questionário perguntas sobre o tempo de experiência em desenvolvimento de *software*, experiência com o *Scrum*, se já haviam utilizado alguma ferramenta de ge-

renciamento de *Backlog* do Produto, e quais ferramentas já utilizaram. As Figuras de 21 a 24 apresentam as respostas para esses questionamentos.

Figura 21 – Respostas sobre o tempo de experiência com o desenvolvimento de *software*

Você tem quanto tempo de experiência no desenvolvimento de Software?

8 respostas

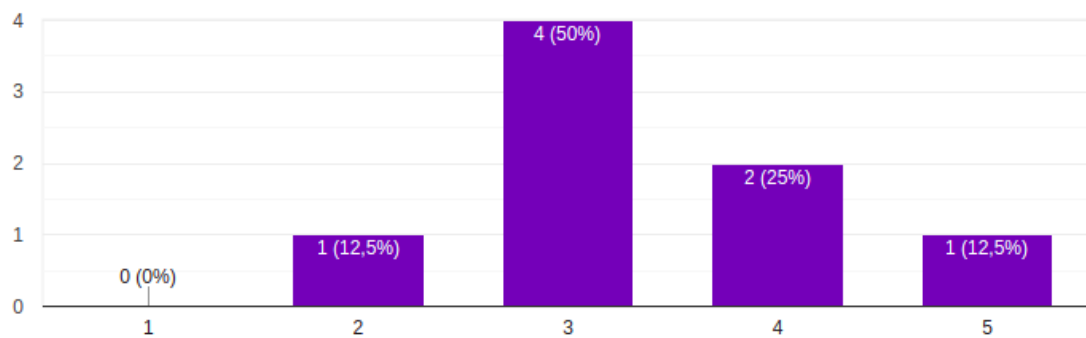


Fonte: autoria própria

Figura 22 – Respostas sobre o tempo de experiência com o *Scrum*

Qual o seu nível de experiência com o Scrum?

8 respostas

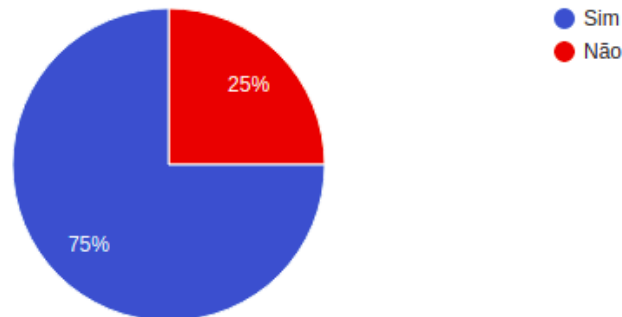


Fonte: autoria própria

Figura 23 – Respostas sobre a utilização de ferramentas para o gerenciamento do *Backlog* do Produto

Você já utilizou alguma ferramenta específica para o gerenciamento de Backlog do Produto?

8 respostas



Fonte: autoria própria

Figura 24 – Respostas sobre quais ferramentas utilizaram para o gerenciamento do *Backlog* do Produto

Se sim, qual ou quais ferramenta(s) você utilizou?

6 respostas

Google sheets
Azure DevOps, ClickUp
JIRA, ClickUp...
Jira, Taiga
Jira
Jira, Clickup, clubhouse

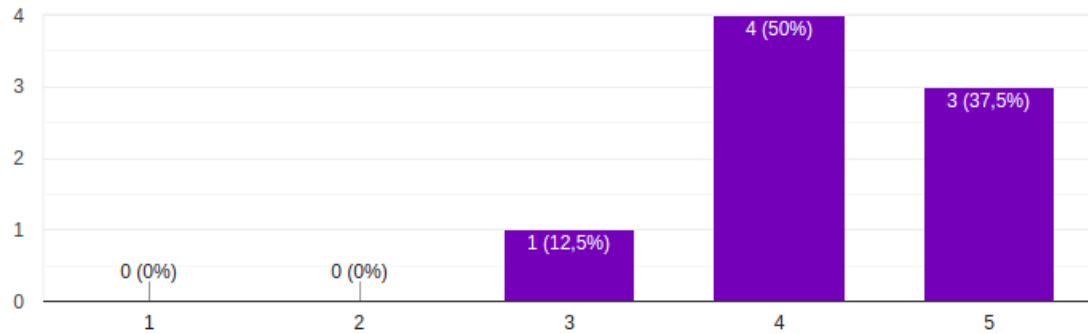
Fonte: autoria própria

Após a utilização da ferramenta, foi solicitado que cada participante descrevesse as impressões e os *feedbacks* sobre a sua experiência de uso. As Figuras de 25 a 28 apresentam as respostas dos usuários. Além disso, foi requerido que dessem sugestões para melhoria da ferramenta, detalhadas no Quadro 14.

Figura 25 – Respostas sobre a intuição do fluxo de navegação da ferramenta (5 representa muito intuitiva e 1 pouco intuitiva)

O que você achou do fluxo de navegação da ferramenta?

8 respostas

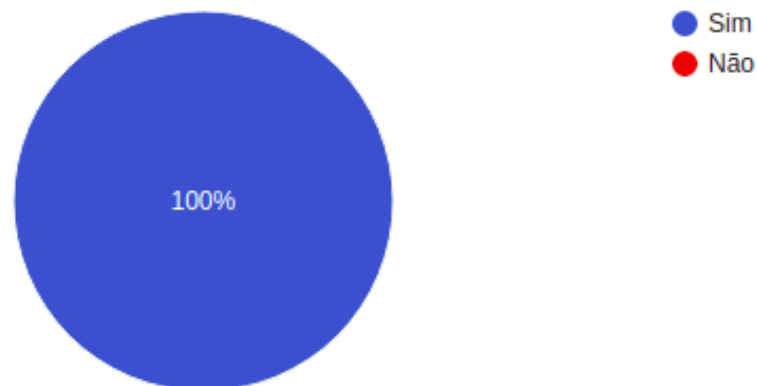


Fonte: autoria própria

Figura 26 – Respostas sobre a criação do *Backlog* de forma desejada

Você conseguiu criar o Backlog do Produto da maneira como esperava?

8 respostas



Fonte: autoria própria

Quadro 14 – Resultados da Análise no Segundo Ciclo

ID	SUGESTÕES	ANÁLISE	AÇÕES
S01	Em " <i>Backlog</i> ", seria melhor se o campo de selecionar projetos fosse maior, e estivesse em maior destaque.	A ferramenta possui um <i>dropdown</i> para trocar de projeto em " <i>Backlog</i> ", porém não encontra-se em destaque.	- Dar destaque ao campo de seleção dos projetos.

Continua na próxima página

Quadro 14 – continuação da página anterior

ID	SUGESTÕES	ANÁLISE	AÇÕES
S02	Melhorar a visualização do progresso do projeto.	A ferramenta não possui uma separação entre itens concluídos e não concluídos o que não fica claro o progresso do projeto.	- A separação entre itens concluídos e não concluídos
S03	Exibir na tela em que são adicionadas as histórias de usuário qual a <i>Feature</i> relacionada.	Na tela para criação de histórias de usuário, não existe o nome da <i>Feature</i> que o usuário está criando as histórias.	- Colocar o nome indicando qual <i>Feature</i> relacionada na tela de criação das histórias de usuário
S04	Em " <i>Backlog</i> ", os botões de "Visualizar" e "Editar" possuem o mesmo comportamento. Isso fica confuso ao usuário, devendo ter apenas uma opção para isso; ou, em "Visualizar", deveria ter as informações sem estarem habilitadas para edições.	Na ferramenta, não existe tela de visualização sem edição dos itens do <i>Backlog</i> .	- Criar tela de visualização sem edição para os itens do <i>Backlog</i> .
S05	Melhorar o entendimento de como dar progresso às tarefas do projeto.	Na ferramenta, não há botão para o usuário finalizar uma história, o que pode causar confusão na hora de dar progresso aos itens do <i>Backlog</i> .	- Criar botão para finalizar histórias de usuário.
Continua na próxima página			

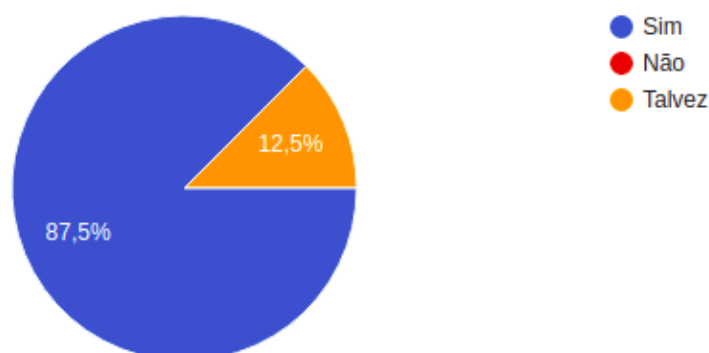
Quadro 14 – continuação da página anterior

ID	SUGESTÕES	ANÁLISE	AÇÕES
S06	Criar outros tipos de visualização, como gráficos, ou uma organização diferente de lista, que permita um entendimento mais amplo do projeto como um todo. Ou seja, tudo que falta ser feito para atingir o resultado final.	Na ferramenta, só existe um tipo de visualização para o <i>Backlog</i> , que é em forma de tabela.	- Criar gráficos para o entendimento do projeto como um todo.
S07	Criar quadro para a visualização do projeto com <i>deadline</i> .	Na ferramenta, não existe local para informar datas de início e fim de um item do <i>Backlog</i> . Consequentemente, não há visualização por <i>deadline</i> .	- Criar datas para o início e fim tanto do projeto quanto dos itens do <i>Backlog</i> , e - Criar quadros para a visualização dos <i>deadlines</i> .
S08	Exibir na tela em que são adicionadas as <i>Features</i> qual o Épico relacionado.	Na tela para criação de <i>Features</i> , não existe o nome do Épico que o usuário está criando as <i>Features</i> .	- Colocar o nome indicando qual o Épico relacionado na tela de criação das <i>Features</i> .
S09	Poderiam ter <i>tags</i> criadas para dividir em times.	Na ferramenta, não existe a opção de criar <i>tags</i> personalizadas pelos os usuários.	- Permitir a criação de <i>tags</i> personalizadas pelos usuários.
S10	Melhorar o fluxo de criação de histórias para <i>Features</i> diferentes.	Na ferramenta, para criar histórias de usuário de várias <i>Features</i> é necessário ficar voltando para selecionar uma <i>Feature</i> específica.	- Permitir que o usuário selecione outra <i>Feature</i> na tela de criação de histórias de usuário.

Figura 27 – Respostas sobre a orientação que a ferramenta proporciona

A respeito dos textos de orientação presentes na ferramenta, você acha que eles te ajudaram na hora de criar o seu Backlog do Produto?

8 respostas

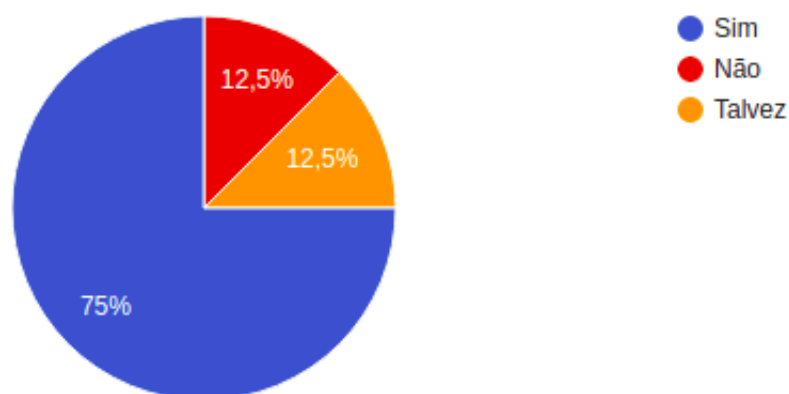


Fonte: autoria própria

Figura 28 – Respostas sobre a utilização da ferramenta em projetos futuros

Você utilizaria nossa ferramenta em projetos futuros?

8 respostas



Fonte: autoria própria

## 6.4 Operacionalizações dos Ciclos de Pesquisa-Ação

Após realizar os dois ciclos de Pesquisa-Ação, analisar cada critério e sugestão, e propor as ações necessárias, foi feito o Quadro 15 com as operacionalizações, sejam essas implementadas ou não na ferramenta *Backlog Maker*. A análise do que seria viável ou não implementar foi feita visando o escopo e o tempo para a realização deste projeto. As operacionalizações de OP01 a OP32 são provenientes do Primeiro Ciclo, presente na

Seção 6.2, e as demais (OP33 a OP43) são referentes ao Segundo Ciclo, presente na Seção 6.3.

Quadro 15 – Operacionalizações resultantes dos ciclos de Pesquisa-Ação

ID	OPERACIONALIZAÇÃO	IMPLEMENTAÇÃO
OP01	Adicionar comentários nos Épicos, <i>Features</i> e Histórias	Implementado
OP02	Criar campos específicos para coleta de <i>feedback</i>	Não implementado
OP03	Adicionar personalização dos níveis de granularidade do <i>Backlog</i> para cada projeto	Não implementado
OP04	Disponibilizar histórico de edição dos Épicos, <i>Features</i> e Histórias	Implementado
OP05	Permitir a adição de membros às Histórias	Implementado
OP06	Permitir a adição de itens bloqueantes aos Épicos	Não implementado
OP07	Permitir a adição de itens bloqueantes de diferentes níveis de granularidade aos Épicos, <i>Features</i> e Histórias	Não implementado
OP08	Adicionar seção de personalização/configuração para cada projeto	Não implementado
OP09	Adicionar personalização das estimativas para as Histórias	Não implementado
OP10	Adicionar personalização dos graus de prioridade para os Épicos, <i>Features</i> e Histórias de cada projeto	Não implementado
OP11	Disponibilizar uma <i>wiki</i> para cada projeto	Não implementado
OP12	Disponibilizar uma seção para registro de informações técnicas na <i>wiki</i> de cada projeto	Não implementado
OP13	Permitir a adição de <i>tags</i> aos Épicos, <i>Features</i> e Histórias	Não implementado
Continua na próxima página		



Quadro 15 – continuação da página anterior

ID	OPERACIONALIZAÇÃO	IMPLEMENTAÇÃO
OP14	Adicionar a visualização das Histórias na tela de <i>Backlog</i> (que, atualmente, mostra apenas Épicos e <i>Features</i> aninhados)	Implementado
OP15	Permitir a marcação de Histórias concluídas	Implementado
OP16	Exibir os Épicos concluídos e não concluídos com <i>tags</i> diferentes na tela de <i>Backlog</i>	Implementado
OP17	Exibir as <i>Features</i> concluídas e não concluídas com <i>tags</i> diferentes na tela de <i>Backlog</i>	Implementado
OP18	Exibir as Histórias concluídas e não concluídas com <i>tags</i> diferentes na tela de <i>Backlog</i>	Implementado
OP19	Fazer integração com plataformas de gerenciamento de código ( <i>GitHub</i> , <i>GitLab</i> , <i>Bitbucket</i> , entre outras) para mostrar detalhes da implementação do produto	Não implementado
OP20	Adicionar o componente " <i>loading</i> " nas páginas onde há carregamento de dados após interações dos usuários	Implementado
OP21	Organizar o fluxo de criação, visualização e edição dos itens do <i>Backlog</i>	Implementado
OP22	Adicionar um guia de navegação para facilitar a criação, visualização e edição dos itens do <i>Backlog</i>	Implementado
OP23	Permitir a seleção de itens para criar o <i>Backlog</i> da <i>Sprint</i>	Não implementado
OP24	Disponibilizar a visualização e alteração do <i>Kanban</i> da <i>Sprint</i>	Não implementado
OP25	Adicionar critérios de qualidade/aceitação aos Épicos e <i>Features</i>	Não implementado
Continua na próxima página		

Quadro 15 – continuação da página anterior

ID	OPERACIONALIZAÇÃO	IMPLEMENTAÇÃO
OP26	Permitir que clientes criem itens no <i>Backlog</i> do Produto	Não implementado
OP27	Permitir a busca dos itens do <i>Backlog</i>	Não implementado
OP28	Permitir a filtragem dos itens do <i>Backlog</i>	Não implementado
OP29	Criar tela de visualização do <i>Backlog</i> da Sprint semelhante a de visualização do <i>Backlog</i> do Produto	Não implementado
OP30	Permitir a criação de uma lista de desejos para cada projeto	Não implementado
OP31	Adicionar textos de ajuda e orientação para a construção do <i>Backlog</i> em todos os níveis	Implementado
OP32	Adicionar <i>bugs</i> ao <i>Backlog</i>	Implementado
OP33	Criar tela de visualização sem edição para os itens do <i>Backlog</i> .	Implementado
OP34	A separação entre itens concluídos e não concluídos.	Implementado
OP35	Dar destaque ao campo de seleção dos projetos.	Implementado
OP36	Colocar o nome indicando qual <i>Feature</i> está sendo criada a História na tela de criação das Histórias de Usuário.	Implementado
OP37	Criar outros tipos de visualização para o entendimento do projeto como um todo.	Não implementado
OP38	Criar datas para o início e fim tanto do projeto quanto dos itens do <i>Backlog</i> .	Não implementado
OP39	Criar quadros para a visualização das <i>deadlines</i> .	Não implementado
OP40	Permitir que o usuário selecione outra <i>Feature</i> na tela de criação de Histórias de Usuário.	Implementado
Continua na próxima página		

Quadro 15 – continuação da página anterior

ID	OPERACIONALIZAÇÃO	IMPLEMENTAÇÃO
OP41	Colocar o nome indicando qual o Épico relacionado na tela de criação das <i>Features</i> .	Implementado
OP42	Ao clicar em "Criar Histórias" na tabela de <i>Backlog</i> , abrir o formulário de Histórias.	Não implementado
OP43	Ao clicar em "Criar <i>Features</i> " na tabela de <i>Backlog</i> , abrir o formulário de <i>Features</i> .	Não implementado

Fonte: autoria própria

As operacionalizações que se encontram implementadas já podem ser usufruídas pelos usuários nessa primeira versão da ferramenta. As que se encontram em andamento, são oferecidas de forma parcial nessa primeira versão, conforme detalhamento da terceira coluna do quadro. As operacionalizações que não foram implementadas podem ser consideradas em projetos futuros, para melhoria da ferramenta *Backlog Maker*.

## 6.5 Resumo do Capítulo

Neste capítulo, foram apresentados os resultados obtidos nas duas fases de realização deste trabalho. Primeiramente, foi exposto o Quadro 16, como resultado do empenho teórico da primeira fase e, logo em seguida, foram descritas as atividades de análise aplicadas sobre a ferramenta *Backlog Maker*. Como resultado da aplicação dessas atividades, na segunda fase do trabalho, foram, também, descritos os resultados obtidos, bem como as ações de melhoria identificadas a partir deles.

Foi utilizada a metodologia de Pesquisa-Ação com dois ciclos de análise, para os quais foram apresentadas a forma da coleta e a interpretação dos dados. Ao final, foi feito o Quadro 15 das operacionalizações obtidas a partir dessas atividades, com a sinalização de quais delas foram incorporadas ao produto final.



## 7 Conclusão

Conforme discutido nos Capítulos 1 e 2, as dificuldades inerentes à construção e à manutenção de um *Backlog* do Produto são evidenciadas nas equipes que adotam o *Scrum* como metodologia de desenvolvimento (ANAND; DINAKARAN, 2015). Uma das problemáticas enfrentadas nesse contexto consiste na dificuldade de transformar os requisitos em histórias de usuários, causando confusão na compreensão dos requisitos.

A gestão de requisitos em modelos ágeis é feita de forma contínua e flexível durante todo o processo de desenvolvimento. Dessa forma, é um desafio ter que lidar com a flexibilidade que o modelo ágil proporciona, e garantir que a comunicação entre os *stakeholders* seja eficiente e compreendida igualmente por todos (KASAULI et al., 2021).

Tendo em vista a problemática apresentada, na gestão de requisitos no *Scrum*, este Trabalho de Conclusão de Curso teve como objetivo geral construir uma ferramenta que auxiliasse na construção e na manutenção do *Backlog* do Produto, de forma que fosse possível mitigar os típicos problemas encontrados.

Em vista disso, foi criada a ferramenta *Backlog Maker*. A solução proposta é apresentada em detalhes no Capítulo 5, e as principais funcionalidades da ferramenta desenvolvida são:

- Realizar cadastro e *login* utilizando o Google ou *e-mail* e senha;
- Criar, visualizar, editar e excluir projetos;
- Convidar membros para participar de um determinado projeto;
- Criar *Backlog* do Produto com três níveis de granularidade (Épicos, *Features* e Histórias de Usuário), sendo que é possível criar tarefas técnicas em Histórias;
- Criar, visualizar, editar e excluir Épicos;
- Criar, visualizar, editar e excluir *Features*;
- Criar, visualizar, editar e excluir *Bugs*;
- Criar, visualizar, editar e excluir Histórias de Usuário;
- Criar, visualizar, editar e excluir comentários dos itens do *Backlog* do Produto;
- Visualizar Histórico de Edições dos itens do *Backlog* do Produto;
- Guia com o passo a passo para a criação dos níveis de granularidade, e

- Textos de ajuda para cada nível de granularidade e cada campo dos formulários.

Após o desenvolvimento das principais funcionalidades, foram feitos dois ciclos de Pesquisa-Ação, como acordado no Capítulo 4. O primeiro ciclo foi realizado com uma autoanálise pelas próprias autoras do projeto. Já o segundo ciclo foi feito com uma amostra de voluntários com as características do Público-Alvo deste trabalho. Após a análise dos resultados, foi feito o Quadro 15 com as operacionalizações que foram implementadas ou não no projeto. Os detalhes dos resultados estão descritos no Capítulo 6.

As principais funcionalidades do projeto foram desenvolvidas e testadas com os usuários no segundo ciclo de Pesquisa-Ação. Porém, em virtude do tempo, algumas funcionalidades, que seriam secundárias ao escopo do projeto acordado, não foram implementadas e ficarão como sugestões para trabalhos futuros.

Cabe ressaltar que os objetivos específicos estabelecidos foram atingidos, sendo os mesmos:

1. **Definir quais são os principais problemas da construção e da manutenção do *Backlog* do Produto, por meio de uma revisão sistemática da literatura:** Os principais problemas encontrados na construção e na manutenção do *Backlog* do Produto foram elencados no Apêndice B. O Quadro 16 apresenta, com base nos resultados da revisão sistemática, uma lista desses problemas;
2. **Definir quais são as possíveis soluções para os problemas encontrados na construção e na manutenção do *Backlog* do Produto, por meio do referencial teórico encontrado com a revisão sistemática:** No Apêndice B, são elencadas as possíveis soluções para os problemas encontrados na construção e na manutenção do *Backlog* do Produto. O mesmo Quadro 16, que apresenta a lista dos problemas, endereça a cada um deles as possíveis soluções para tratá-los;
3. **Elicitar os requisitos necessários para a construção da ferramenta:** Os requisitos definidos para a construção da ferramenta *Backlog Maker* podem ser encontrados no Capítulo 5, tanto na *Baseline de Requisitos*, apresentada na Seção 5.2, quanto no *Backlog* da ferramenta, apresentado na Seção 5.5;
4. **Definir a arquitetura do sistema a ser desenvolvido:** O detalhamento da arquitetura da ferramenta *Backlog Maker* pode ser encontrado no Capítulo 5, na Seção 5.3;
5. **Desenvolver a ferramenta que auxilie na construção e na manutenção do *Backlog* do Produto:** O detalhamento da solução desenvolvida pode ser encontrado no Capítulo 5, Seção 5.6, e

6. **Realizar pesquisa-ação e documentar os resultados obtidos:** Os detalhes da realização da Pesquisa-Ação, bem como os resultados obtidos até o momento, estão presentes no Capítulo 6.

Dessa forma, pode-se concluir que os Objetivos Gerais foram cumpridos, sendo a ferramenta *Backlog Maker* construída, e mais focada em pequenas e médias empresas, uma vez que empresas de grande porte costumam ter seus processos de desenvolvimento com base em arcabouços tecnológicos ainda mais criteriosos.

Por fim, visando responder as questões de pesquisa, pode-se mencionar:

- A Questão 1 - Quais são os problemas enfrentados pelas equipes *Scrum* na definição e na manutenção do *Backlog* do Produto ao longo do ciclo de vida de desenvolvimento de um software? - foi respondida ao longo do Capítulo 2, no qual foi acordada toda a bagagem teórica necessária para entender e identificar tais problemas. Além disso, o Apêndice B apresenta, de maneira mais objetiva, a resposta a esse questionamento, ao elencar, no Quadro 16, os principais problemas, encontrados na literatura, inerentes à construção e à manutenção do *Backlog* do Produto, e
- A Questão 2 - Quais são as práticas mais indicadas pela literatura para mitigar os problemas encontrados e guiar a construção de uma ferramenta que apoie a montagem e a manutenção de um bom *Backlog* do Produto? - foi respondida tanto no Apêndice B quanto no Capítulo 2. Além das possíveis soluções elencadas no Quadro 16, a Seção 2.3.2 apresenta uma série de critérios a serem observados em uma ferramenta tecnológica que confira apoio à gestão de um bom *Backlog* do Produto.

## 7.1 Trabalhos Futuros

Em se tratando de uma ferramenta que busca lidar com uma das maiores lacunas na gestão de requisitos em modelos ágeis atualmente, o escopo do *Backlog Maker* pode tomar proporções tão grandes quanto se queira, devido à grande variedade de fatores que influenciam na criação e na manutenção do *Backlog* do Produto ao longo do desenvolvimento de *software*. Funcionalidades como as de gestão de equipes e projetos, por exemplo, poderiam ser adicionadas aos requisitos da aplicação, fazendo com que sua abrangência seja ainda maior. Esses vislumbres, no entanto, dizem respeito a uma versão da ferramenta com alguns graus de maturidade a mais do que a versão atual.

Nesse caso, pensando em um futuro mais próximo, algumas das funcionalidades que são também importantes no contexto de atuação da *Backlog Maker*, considerando um escopo bem menor, deixaram de ser priorizadas durante o desenvolvimento deste

trabalho, por conta do tempo e da dificuldade de implementação, e são os primeiros pontos a serem abordados em uma evolução da versão atual da ferramenta. Algumas delas já estão presentes na versão final do *Backlog* da aplicação, enquanto outras foram ações de melhoria identificadas apenas a partir dos ciclos de análise, apresentadas no Capítulo 6. Tais funcionalidades são:

- Seção de configurações de cada projeto, com possibilidade de personalização dos níveis de granularidade do *Backlog*, das estimativas, dos graus de priorização de itens, e de criação de *tags*, além do gerenciamento dos membros de cada projeto;
- Disponibilização de uma *wiki* para os projetos;
- Busca de projetos e de itens do *Backlog*;
- Filtragem dos projetos e dos itens do *Backlog*;
- Adição de *tags* aos projetos e aos itens do *Backlog*;
- Adição de datas de início e fim aos projetos e aos itens do *Backlog*;
- Melhoria dos históricos de edições dos itens do *Backlog*;
- Possibilidade de visualização do Backlog em outros arranjos visuais que não somente o de lista (quadros, calendário, entre outros);
- Possibilidade de gerenciar as *Sprints* e seus respectivos *Backlogs*;
- Visualização e possibilidade de interação com um quadro *Kanban*, que possibilite o acompanhamento do progresso do trabalho em cada *Sprint*, e
- Integração com plataformas de gerenciamento de código (GitHub, GitLab, Bitbucket, entre outras) para mostrar detalhes da implementação do produto.



## Referências

- ADITAMA, A.; LEGOWO, N. Combination of scrum methodology and ibm design thinking for dashboard and report system development. *International Journal of Advanced Trends in Computer Science and Engineering*, v. 9, n. 5, p. 7241–7250, 2020. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 6 vezes nas páginas 17, 18, 24, 25, 26 e 27.
- ALQUDAH, M.; RAZALI, R. An empirical study of scrumban formation based on the selection of scrum and kanban practices. *International Journal on Advanced Science, Engineering and Information Technology*, v. 8, n. 6, p. 2315–2322, 2018. Cited By :3. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 6 vezes nas páginas 18, 25, 26, 27, 43 e 44.
- ANAND, R. V.; DINAKARAN, M. Issues in scrum agile development principles and practices in software development. *Indian Journal of Science and Technology*, v. 8, n. 35, 2015. Cited By :4. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 5 vezes nas páginas 17, 18, 19, 57 e 107.
- BALDIN N.; MUNHOZ, E. M. B. Snowball (bola de neve): uma técnica metodológica para pesquisa em educação ambiental comunitária. 2011. Citado na página 21.
- CAROLI, P.; AGUIAR, F. *Product Backlog Building: Um guia prático para criação e refinamento de backlog para produtos de sucesso*. [S.l.]: Editora Caroli, 2021. Citado 2 vezes nas páginas 29 e 79.
- COHN, M. Applied user stories development for agile software. 2004. Citado na página 30.
- CSS, T. *Tailwind CSS - Rapidly build modern websites without ever leaving your HTML*. 2021. Disponível em: <<https://tailwindcss.com/>>. Citado na página 37.
- DOCKER. *Overview of Docker Compose*. 2021. Disponível em: <<https://docs.docker.com/compose/>>. Citado na página 35.
- DOCS, M. W. *MDN WEB DOCS*. 2021. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Citado na página 36.
- ENGEL, G. I. Pesquisa-ação. 2000. Citado na página 47.
- ENGUM, E. A.; RACHEVA, Z.; DANEVA, M. Sprint planning with a digital aid tool: Lessons learnt. In: *Conference Proceedings of the EUROMICRO*. [s.n.], 2009. p. 259–262. Cited By :2. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 10 vezes nas páginas 18, 19, 20, 28, 31, 32, 120, 121, 122 e 123.
- EXPRESS. *Express*. 2021. Disponível em: <<https://expressjs.com/>>. Citado na página 36.
- FILIPPO, D. Pesquisa-ação em sistemas colaborativos. *Sistemas Colaborativos (capítulo 26)*. Rio de Janeiro: SBC/Elsevier., 2011. Citado na página 54.
- FIREBASE. *Cloud Firestore*. 2021. Disponível em: <<https://firebase.google.com/docs/firestore?hl=pt-br>>. Citado na página 70.

- FIREBASE. *Cloud Storage para Firebase*. 2021. Disponível em: <<https://firebase.google.com/docs/storage>>. Citado na página 69.
- FIREBASE. *Firebase*. 2021. Disponível em: <<https://firebase.google.com/docs/>>. Citado na página 37.
- FIREBASE. *Firebase Authentication*. 2021. Disponível em: <<https://firebase.google.com/docs/auth>>. Citado na página 69.
- GEEKHUNTER. *Como construir uma aplicação com Docker?* 2019. Disponível em: <<https://blog.geekhunter.com.br/docker-na-pratica-como-construir-uma-aplicacao/>>. Citado na página 35.
- GERHARDT T. E.; SILVEIRA, D. T. Metodologia de pesquisa. 2009. Citado 2 vezes nas páginas 41 e 47.
- GIL, A. C. Métodos e técnicas de pesquisa social. 2008. Citado na página 42.
- GIT. *Git*. 2021. Disponível em: <<https://git-scm.com/>>. Citado na página 35.
- GOOGLE. *Google Drive*. 2021. Disponível em: <<https://www.google.com.br/drive/apps.html>>. Citado na página 38.
- HEIKKILÄ, V. T. et al. Managing the requirements flow from strategy to release in large-scale agile development: a case study at ericsson. *Empirical Software Engineering*, Springer, v. 22, n. 6, p. 2892–2936, 2017. Citado 4 vezes nas páginas 119, 121, 122 e 123.
- HERDIKA, H. R.; BUDIARDJO, E. K. Variability and commonality requirement specification on agile software development: Scrum, xp, lean, and kanban. In: *2020 3rd International Conference on Computer and Informatics Engineering, IC2IE 2020*. [s.n.], 2020. p. 323–329. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 10 vezes nas páginas 25, 26, 30, 31, 43, 44, 45, 119, 120 e 122.
- HEROKU. *What is Heroku*. 2021. Disponível em: <<https://www.heroku.com/what>>. Citado na página 37.
- HODA, R.; MURUGESAN, L. Multi-level agile project management challenges: A self-organizing team perspective. *Journal of Systems and Software*, v. 117, 03 2016. Citado na página 18.
- HVATUM, L. B.; WIRFS-BROCK, R. Patterns to build the magic backlog. In: *ACM International Conference Proceeding Series*. [s.n.], 2015. v. 08-12-July-2015. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 4 vezes nas páginas 30, 31, 32 e 121.
- HVATUM, L. B.; WIRFS-BROCK, R. A program backlog story with patterns: Expanding the magic backlog pattern collection. In: *ACM International Conference Proceeding Series*. [s.n.], 2018. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado na página 119.
- KASAULI, R. et al. Requirements engineering challenges and practices in large-scale agile system development. *Journal of Systems and Software*, v. 172, 2021. Cited By :1. Disponível em: <[www.scopus.com](http://www.scopus.com)>. Citado 6 vezes nas páginas 30, 107, 119, 120, 121 e 123.

KHOMYAKOV, I.; MIRGALIMOVA, R.; SILLITTI, A. An investigation of the project management approaches of agile and plan-based companies. In: *Proceedings of the 35th Annual ACM Symposium on Applied Computing*. [S.l.: s.n.], 2020. p. 1662–1665. Citado na página 17.

KITCHENHAM, B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, v. 33, n. 2004, p. 1–26, 2004. Citado 2 vezes nas páginas 21 e 42.

L., A. *O Que é GitHub e Para Que é Usado?* 2021. Disponível em: <<https://www.hostinger.com.br/tutoriais/o-que-github>>. Citado na página 35.

LATEX. *LaTeX*. 2021. Disponível em: <<https://www.latex-project.org/about/>>. Citado na página 38.

LEE S; YONG, H. Agile software development framework in a small project environment. 2013. Citado na página 27.

LOUBAK, A. L. *Como funciona o Hangouts? Conheça ferramenta do Google para videochamadas*. 2020. Disponível em: <<https://www.techtudo.com.br/noticias/2020/03/como-funciona-o-hangouts-conheca-ferramenta-do-google-para-videochamadas.ghtml>>. Citado na página 39.

NEXT. *Next*. 2021. Disponível em: <<https://nextjs.org/docs/>>. Citado na página 37.

NODEJS. *Sobre o Node.js*. 2021. Disponível em: <<https://nodejs.org/about/>>. Citado na página 36.

OVERLEAF. *Overleaf*. 2021. Disponível em: <<https://www.overleaf.com/for/authors>>. Citado na página 39.

PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado 5 vezes nas páginas 17, 23, 24, 25 e 43.

PRODANOV C. C. ; FREITAS, E. C. Metodologia do trabalho científico: Métodos e técnicas da pesquisa e do trabalho acadêmico. 2013. Citado 2 vezes nas páginas 41 e 42.

RAJEEV, B.; HEJIB, V. Control based management to self organizing agile teams: a case study. In: *Proceedings of the 13th International Conference on Global Software Engineering*. [S.l.: s.n.], 2018. p. 16–20. Citado na página 17.

RAJEEV, B. V.; HEJIB, V. Control based management to self organizing agile teams: A case study. In: *Proceedings of the 13th International Conference on Global Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2018. (ICGSE '18), p. 16–20. ISBN 9781450357173. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3196369.3196394>>. Citado na página 18.

REACT. *React*. 2021. Disponível em: <<https://reactjs.org/>>. Citado na página 37.

REYES, J. E. A.; INTAL, G. L. D. Agile scrum adoption of the application development projects of company c. In: *Proceedings of the 2020 2nd International Conference on Management Science and Industrial Engineering*. [S.l.: s.n.], 2020. p. 176–183. Citado 5 vezes nas páginas 119, 120, 121, 122 e 123.

- Richardson, I.; Von Wangenheim, C. G. Guest editors' introduction: Why are small software organizations different? *IEEE Software*, v. 24, n. 1, p. 18–22, 2007. Citado 2 vezes nas páginas 27 e 28.
- RUBIN HILLEL; RUBIN, E. Supporting agile software development through active documentation. 2011. Citado na página 29.
- Russ, M. L.; McGregor, J. D. A software development process for small projects. *IEEE Software*, v. 17, n. 5, p. 96–101, 2000. Citado na página 27.
- SANTAMARIA, M. G. Agile & scrum: What are these methodologies and how will they impact qa/testing roles? 2007. Citado 2 vezes nas páginas 18 e 30.
- SCHÖN, E.-M. et al. Key challenges in agile requirements engineering. In: SPRINGER, CHAM. *International Conference on Agile Software Development*. [S.l.], 2017. p. 37–51. Citado na página 122.
- THIOLLENT M. ; SILVA, G. O. Metodologia de pesquisa-ação na área de gestão de problemas ambientais. 2007. Citado na página 47.
- TYPESCRIPT. *TypeScript: JavaScript With Syntax For Types*. 2021. Disponível em: <<https://www.typescriptlang.org/>>. Citado na página 36.
- VERCEL. *Develop. Preview. Ship. For the best frontend teams*. 2021. Disponível em: <<https://vercel.com/>>. Citado na página 38.
- WHATSAPP. *WhatsApp*. 2021. Disponível em: <<https://www.whatsapp.com/>>. Citado na página 39.
- ZENHUB. *ZenHub - Agile Project Management for GitHub*. 2021. Disponível em: <<https://www.zenhub.com/>>. Citado na página 38.
- ZOTERO. *Zotero*. 2021. Disponível em: <<https://www.zotero.org/>>. Citado na página 38.
- ÁVILA A. L.; SPÍNOLA, R. Introdução à engenharia d e requisitos. 2017. Citado 2 vezes nas páginas 28 e 29.

# Apêndices



# APÊNDICE A – Revisão Sistemática da Literatura

## 1. Fase de Planejamento

Com o objetivo de investigar quais são os problemas relacionados à construção e à manutenção de um *Backlog* do Produto, foram criadas as seguintes questões de pesquisa:

- Questão 1 - Quais são as dificuldades na criação do *backlog* do produto e no planejamento das *sprints*?
- Questão 2 - Quais são os problemas gerados a partir dessas dificuldades?
- Questão 3 - Quais as consequências desses problemas?

A base que foi utilizada para a pesquisa foi a *SCOPUS(Elsevier)* para busca de artigos relevantes para a engenharia de software. Os seguintes itens são definidos como critérios de exclusão e inclusão:

- Excluem-se artigos em idiomas diferentes de inglês e português.
- Excluem-se artigos que não citam características de equipes ágeis.
- Excluem-se artigos que não são relevantes para metodologias ágeis.
- Excluem-se artigos que não se aplicam ao contexto de software.
- Excluem-se artigos que não estão disponíveis.
- Incluem-se artigos que falam sobre planejamento e organização do *backlog* ou requisitos de um produto.
- Incluem-se artigos que falam sobre as dificuldades no planejamento e manutenção do *backlog* ou requisitos.

A fase de seleção ocorreu em duas etapas. Na primeira etapa foi feita a leitura dos títulos e dos *abstracts* para encontrar os artigos relevantes para o objetivo da pesquisa. Na segunda etapa foi feita a leitura completa do artigo para execução da revisão sistemática.

Foi criada a *string* de busca, que foi definida por:

```
( ALL ( agile AND teams ) AND ALL ( agile AND software AND development )  
AND ALL ( agile AND project AND management ) AND ALL ( planning ) OR  
ALL ( prioritization ) AND ALL ( agile AND practices ) AND ALL ( product AND
```

backlog ) AND ALL ( sprint ) OR ALL ( size AND measurement ) OR ALL ( risk AND management ) )

## 2. Fase de Execução

A *string* de busca foi executada na base escolhida na fase de planejamento. Foi obtido um resultado de 42 estudos na base *SCOPUS (Elsevier)* de acordo com a *string* de busca definida. Ao aplicar os critérios de inclusão e exclusão foram obtidos 21 artigos remanescentes.

## 3. Fase de Análise de Resultados

Foram feitas as análises de resultados para cada artigo lido e feita a extração de dados relevantes para o entendimento e embasamento do Trabalho de Conclusão de Curso. Estes resultados foram armazenados em um Documento no *Google Drive* e no *Zotero* pelas autoras do projeto, com o objetivo de serem utilizados ao longo do desenvolvimento deste trabalho.



## APÊNDICE B – Quadro Problemas vs. Potenciais Soluções

A seguir, é apresentado o quadro no qual foram identificados os principais problemas, apontados pela literatura, na construção e na manutenção do *Backlog* do Produto, bem como as soluções em potencial para cada um desses problemas.

Quadro 16 – Problemas vs. Potenciais Soluções na Construção e na Manutenção do Backlog do Produto

PROBLEMA/DESAFIO	POTENCIAIS SOLUÇÕES
Falta de envolvimento dos <i>stakeholders</i> (HERDIKA; BUDIARDJO, 2020)	<ul style="list-style-type: none"> <li>- Manter o foco no valor para o cliente em cada <i>sprint</i> (KASAULI et al., 2021);</li> <li>- Envolver o cliente em todos os níveis (KASAULI et al., 2021);</li> <li>- O <i>PO</i> ou alguém que assume o papel deve estar altamente envolvido (REYES; INTAL, 2020);</li> <li>- O <i>PO</i> deve sempre se reunir com a equipe quando houver necessidade de ajuda (REYES; INTAL, 2020), e</li> <li>- Frequente validação com o cliente para melhorar a compreensão dos requisitos (HEIKKILÄ et al., 2017).</li> </ul>
Documentação mínima (HERDIKA; BUDIARDJO, 2020)	<ul style="list-style-type: none"> <li>- Ter um bom <i>Backlog</i>, com conteúdo e estrutura correta (HVATUM; WIRFS-BROCK, 2018);</li> <li>- Governança dedicada de requisitos em todos os níveis (KASAULI et al., 2021);</li> <li>- Documentação técnica e de usuário deve ser incluída para determinar se uma história está completa (REYES; INTAL, 2020), e</li> </ul>
Continua na próxima página	

Quadro 16 – continuação da página anterior

PROBLEMA/DESAFIO	POTENCIAIS SOLUÇÕES
	<ul style="list-style-type: none"> <li>- A documentação deve ser escrita de forma fácil para novos desenvolvedores ou para membros que não estavam envolvidos anteriormente no projeto (REYES; INTAL, 2020).</li> </ul>
<p>Histórias de usuário erradas (entendimento errado do problema/solução) (HERDIKA; BUDIARDJO, 2020)</p>	<ul style="list-style-type: none"> <li>- Os requisitos de usuário devem ser especificados em <i>features</i> a serem inseridas no <i>Backlog</i> do Produto (HERDIKA; BUDIARDJO, 2020), e</li> <li>- <i>Feedback</i> contínuo pela verificação e validação com testes e revisão dos <i>stakeholders</i> e planejamento contínuo de acordo com o <i>feedback</i> obtido (HERDIKA; BUDIARDJO, 2020).</li> </ul>
<p>Falta de informações nas histórias de usuário (HERDIKA; BUDIARDJO, 2020)</p>	<ul style="list-style-type: none"> <li>- Definição de critérios de aceitação (HERDIKA; BUDIARDJO, 2020);</li> <li>- Usar modelos para interfaces e comportamentos, resumos de texto adicionais (KASAULI et al., 2021);</li> <li>- Usar modelos para analisar os requisitos (KASAULI et al., 2021);</li> <li>- Definição clara de concluído das tarefas que são completadas (REYES; INTAL, 2020);</li> <li>- Desenvolvedores e testadores devem trabalhar juntos na elaboração de critérios de aceitação (REYES; INTAL, 2020);</li> <li>- A história deve ser considerada concluída quando totalmente desenvolvida, testada e aceita pelo <i>PO</i> (REYES; INTAL, 2020);</li> <li>- Descrição das histórias de forma clara (ENGUM; RACHEVA; DANEVA, 2009);</li> <li>- Utilização de uma ferramenta para especificação das histórias (ENGUM; RACHEVA; DANEVA, 2009), e</li> </ul>
<p>Continua na próxima página</p>	

Quadro 16 – continuação da página anterior

PROBLEMA/DESAFIO	POTENCIAIS SOLUÇÕES
	<ul style="list-style-type: none"> <li>- Pareamento para análise de requisitos (HEIKKILÄ et al., 2017).</li> </ul>
<p>Falta de granularidade dos requisitos (HVATUM; WIRFS-BROCK, 2015)</p>	<ul style="list-style-type: none"> <li>- Possibilitar aos <i>stakeholders</i> especificar em qualquer nível de abstração, ex. por meio de rastreabilidade e estrutura de requisitos (KASAULI et al., 2021);</li> <li>- Dar suporte à análise distribuída de requisitos (KASAULI et al., 2021);</li> <li>- Modelo de informação de requisitos (épico-<i>feature-story</i>): transportar a visão do <i>stakeholder</i> para os componentes (KASAULI et al., 2021);</li> <li>- Requisitos devem ser documentados também em alto nível (REYES; INTAL, 2020), e</li> <li>- O lançamento do produto deve ser planejado em um alto nível (REYES; INTAL, 2020).</li> </ul>
<p>Falta de rastreamento (ENGUM; RACHEVA; DANEVA, 2009)</p>	<ul style="list-style-type: none"> <li>- Reutilizar <i>features</i> (grupos de requisitos) (KASAULI et al., 2021);</li> <li>- Módulos reutilizáveis (requisitos e soluções) (KASAULI et al., 2021);</li> <li>- Descrever a solução (documentação) (KASAULI et al., 2021);</li> <li>- Link para as páginas de <i>wiki</i> (KASAULI et al., 2021);</li> <li>- Link dos itens do <i>Backlog</i> com seus ancestrais, máx. 3 níveis (KASAULI et al., 2021), e</li> <li>- Utilização de uma ferramenta para rastreamento (ENGUM; RACHEVA; DANEVA, 2009).</li> </ul>
<p>Falta de ferramentas adequadas para o gerenciamento do <i>Backlog</i> (ENGUM; RACHEVA; DANEVA, 2009)</p>	<ul style="list-style-type: none"> <li>- Utilização de ferramentas como quadros, imagens, <i>wikis</i> e planilhas para grandes <i>Backlogs</i> de produto (KASAULI et al., 2021).</li> </ul>
<p>Continua na próxima página</p>	

Quadro 16 – continuação da página anterior

PROBLEMA/DESAFIO	POTENCIAIS SOLUÇÕES
Inexistência das reuniões de planejamento de <i>sprint</i> (REYES; INTAL, 2020)	- Utilização de uma ferramenta para criar <i>Backlog</i> da <i>Sprint</i> (ENGUM; RACHEVA; DANEVA, 2009), e - Planejamento contínuo (HEIKKILÄ et al., 2017).
Falta de identificação de dependência entre as tarefas e pessoas (ENGUM; RACHEVA; DANEVA, 2009)	- Utilização de uma ferramenta de fácil identificação de dependências das tarefas e das pessoas (ENGUM; RACHEVA; DANEVA, 2009).
As histórias não são divididas e pequenas o suficiente para serem feitas em uma iteração (REYES; INTAL, 2020)	- Os requisitos de usuário devem ser especificados em <i>features</i> a serem inseridos no <i>Backlog</i> do Produto (HERDIKA; BUDIARDJO, 2020); - Utilizar as técnicas de mapeamento de história, mapeamento de impacto e mapeamento mental para dividir a <i>feature</i> em uma história de usuário específica (HERDIKA; BUDIARDJO, 2020), e - As histórias devem ser bem quantificadas e alcançáveis dentro do período de tempo da <i>sprint</i> (REYES; INTAL, 2020).
Não estimar o <i>Backlog</i> do Produto (SCHÖN et al., 2017)	- Utilização de uma ferramenta para estimar as histórias (ENGUM; RACHEVA; DANEVA, 2009).
Falta de revisão frequente das prioridades dos itens do <i>Backlog</i> do Produto (REYES; INTAL, 2020)	- Organizar as histórias de usuário em épicos ajuda o time a acompanhar quais <i>features</i> ainda não foram concluídas (HERDIKA; BUDIARDJO, 2020); - Tratamento de histórias incompletas (REYES; INTAL, 2020); - <i>Bugs</i> /Defeitos devem ser continuamente monitorados (REYES; INTAL, 2020);
Continua na próxima página	

Quadro 16 – continuação da página anterior

PROBLEMA/DESAFIO	POTENCIAIS SOLUÇÕES
	<ul style="list-style-type: none"> <li>- A qualidade do produto, <i>bugs</i> ou defeitos devem ser tratados como entradas no <i>Backlog</i> do Produto/<i>Sprint</i> para fins de planejamento ou rastreamento declarado (REYES; INTAL, 2020), e</li> <li>- Utilização de uma ferramenta para priorizar o <i>Backlog</i> do Produto (ENGUM; RACHEVA; DANEVA, 2009).</li> </ul>
<p>Conflito de interesse entre os <i>stakeholders</i> (o que pode dificultar a priorização) (HEIKKILÄ et al., 2017)</p>	<ul style="list-style-type: none"> <li>- Priorização dos requisitos (HEIKKILÄ et al., 2017).</li> </ul>
<p>Mudanças de requisitos e definições de cronograma e custo do projeto podem ser difíceis de serem conciliados (HEIKKILÄ et al., 2017)</p>	<ul style="list-style-type: none"> <li>- Gerenciamento de mudanças (HEIKKILÄ et al., 2017).</li> </ul>
<p>Curto tempo de planejamento pode resultar em uma arquitetura inadequada e dívidas técnicas (HEIKKILÄ et al., 2017)</p>	<ul style="list-style-type: none"> <li>- Revisões frequentes de requisitos em relação aos resultados da <i>Sprint</i> (KASAULI et al., 2021);</li> <li>- Qualidade interna (KASAULI et al., 2021), e</li> <li>- Reduzir o tempo de chegada ao mercado: mapeamento do fluxo de valor (KASAULI et al., 2021).</li> </ul>
<p>Requisitos não funcionais podem ser subestimados (HEIKKILÄ et al., 2017)</p>	<ul style="list-style-type: none"> <li>- <i>Trade-offs</i> apresentados pela equipe e revisados por colegas e gerentes de sistema (KASAULI et al., 2021).</li> </ul>
<p>Falta de habilidade técnica da equipe para executar as tarefas (REYES; INTAL, 2020)</p>	<ul style="list-style-type: none"> <li>- Equipes multidisciplinares (HEIKKILÄ et al., 2017), e</li> <li>- Conhecimento compartilhado (HEIKKILÄ et al., 2017).</li> </ul>
<p>Inexistência de programação por pares (REYES; INTAL, 2020)</p>	<ul style="list-style-type: none"> <li>- Os membros da equipe devem estar atentos às tarefas uns dos outros e sejam voluntários para ajudar uns aos outros a alcançar o sucesso do projeto, se necessário (REYES; INTAL, 2020).</li> </ul>

Fonte: autoria própria

# APÊNDICE C – Protótipo

A seguir, está apresentado o fluxo de telas do protótipo da ferramenta de criação e manutenção do *Backlog* do Produto.

## 1. Cadastro e *Login*

Figura 29 – Tela do Cadastro no Protótipo de Alta Fidelidade

LOGO E NOME DA FERRAMENTA

### Cadastre-se

Nome

E-mail

Senha

Confirmação de Senha

**CADASTRAR**

Fonte: autoria própria

Figura 30 – Tela do *Login* no Protótipo de Alta Fidelidade

LOGO E NOME DA FERRAMENTA

f Continuar com o Facebook

Continuar com o Google

✉ Cadastrar com o seu E-mail

Entre com o seu e-mail e senha

E-mail

Digite o seu e-mail aqui

Senha

Digite o sua senha aqui

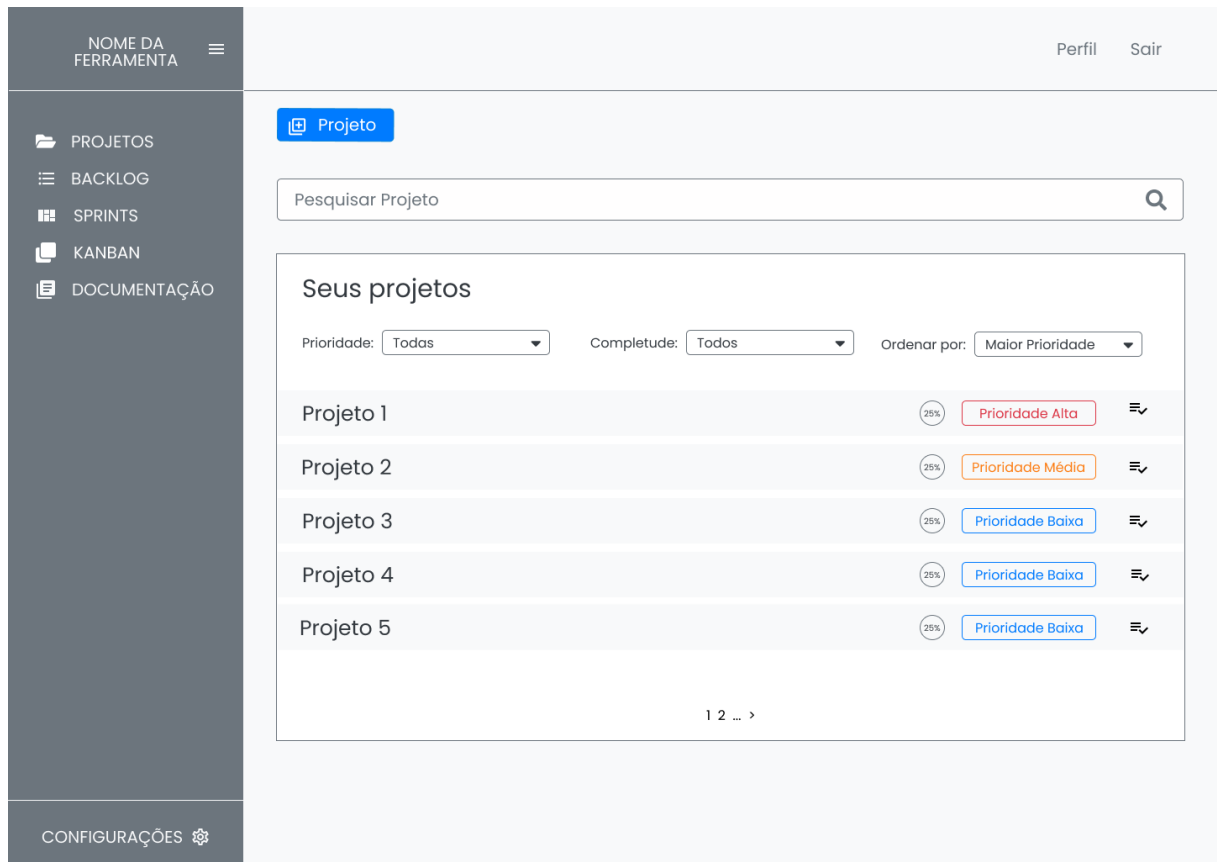
LOGIN

Fonte: autoria própria



## 2. Projeto

Figura 31 – Tela dos Projetos no Protótipo de Alta Fidelidade



Fonte: autoria própria

Figura 32 – Tela das Informações do Projeto no Protótipo de Alta Fidelidade

NOME DA FERRAMENTA

Projeto X Informações do Projeto Perfil Sair

PROJETOS  
BACKLOG  
SPRINTS  
KANBAN  
DOCUMENTAÇÃO

CONFIGURAÇÕES

Projeto 1

Members

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

Prioridade Alta

Links: Arquivos:

<https://www.link.com.br> x [file.pdf](#) x

<https://www.link.com.br> x [image.png](#) x

+ +

Resumo do Projeto

O que é este projeto?

Qual o objetivo deste projeto?

De onde surgiu este projeto?

Qual o público alvo deste projeto?

Fonte: autoria própria

### 3. *Backlog* do Produto

Figura 33 – Tela do *Backlog* do Produto no Protótipo de Alta Fidelidade

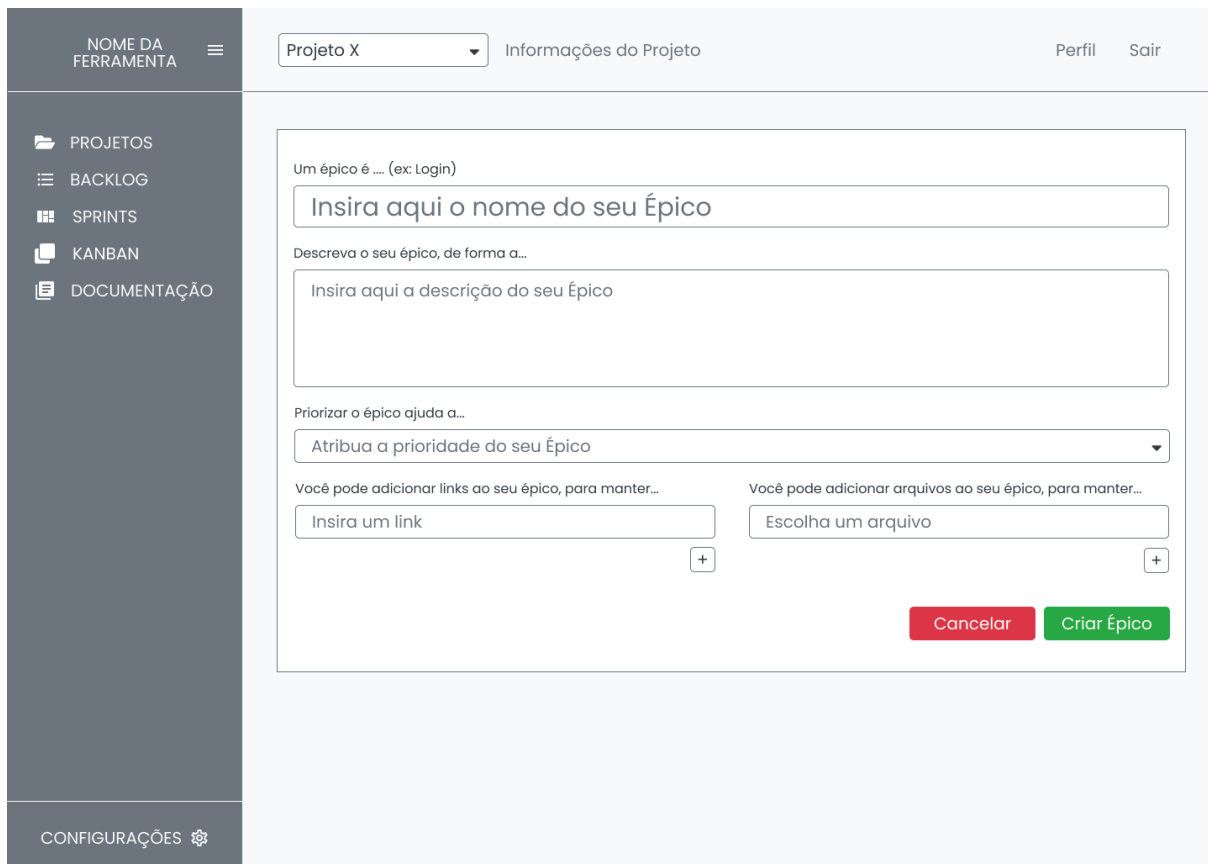
The screenshot displays a web application interface for managing a product backlog. On the left is a dark sidebar with navigation links: 'PROJETOS', 'BACKLOG', 'SPRINTS', 'KANBAN', and 'DOCUMENTAÇÃO'. At the bottom of the sidebar is 'CONFIGURAÇÕES'. The main content area has a header with 'NOME DA FERRAMENTA' and a dropdown for 'Projeto X'. Below the header are buttons for 'Épico' and 'Feature', and a search bar labeled 'Pesquisar Épico ou Feature'. The main section is titled 'Backlog do Produto' and includes filters for 'Prioridade' (Todas), 'Compleitude' (Todos), and 'Ordenar por' (Maior Prioridade). The backlog items are grouped into three 'Épico' sections:

- Épico 1** (25% completion, Prioridade Alta, 65 pontos):
  - Feature 1 (10% completion, Prioridade Alta, 15 pontos)
  - Feature 2 (10% completion, Prioridade Média, 10 pontos)
  - Feature 3 (0% completion, Prioridade Baixa, 20 pontos)
  - Feature 4 (0% completion, Prioridade Baixa, 20 pontos)
- Épico 2** (25% completion, Prioridade Média, 35 pontos)
- Épico 3** (25% completion, Prioridade Baixa, 40 pontos)

At the bottom of the backlog list, there is a pagination indicator '1 2 ... >'.

Fonte: autoria própria

Figura 34 – Tela da Criação de um Épico no Protótipo de Alta Fidelidade



Um épico é ... (ex: Login)

Insira aqui o nome do seu Épico

Descreva o seu épico, de forma a...

Insira aqui a descrição do seu Épico

Priorizar o épico ajuda a...

Atribua a prioridade do seu Épico

Você pode adicionar links ao seu épico, para manter...

Insira um link

Você pode adicionar arquivos ao seu épico, para manter...

Escolha um arquivo

Cancelar Criar Épico

CONFIGURAÇÕES

Nome da Ferramenta: Projeto X

Informações do Projeto

Perfil Sair

Menu Lateral: PROJÉTOS, BACKLOG, SPRINTS, KANBAN, DOCUMENTAÇÃO

Fonte: autoria própria

Figura 35 – Tela das Informações de um Épico no Protótipo de Alta Fidelidade

The screenshot displays a user interface for managing an Epic in a project. The top navigation bar shows the project name 'Projeto X' and the page title 'Informações do Projeto'. A sidebar on the left provides navigation for 'PROJETOS', 'BACKLOG', 'SPRINTS', 'KANBAN', and 'DOCUMENTAÇÃO'. The main content area is divided into several sections:

- Epic Title:** 'Épico 1' with edit and refresh icons.
- Description:** A text area containing placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.'
- Priority:** A red-bordered box labeled 'Prioridade Alta'.
- Links:** A list of two URLs: 'https://www.link.com.br', each with a red 'x' icon and a '+' button below.
- Arquivos:** A list of two files: 'file.pdf' and 'image.png', each with a red 'x' icon and a '+' button below.
- Features:** A section with a 'Feature' button, filters for 'Prioridade' (Todas) and 'Compleitude' (Todas), and a sort option 'Ordenar por: Maior Prioridade'. It contains a table of features:
 

Feature	Progress	Priority	Points
Feature 1	0%	Prioridade Alta	15 pontos
Feature 2	0%	Prioridade Média	10 pontos
Feature 3	0%	Prioridade Baixa	20 pontos
Feature 4	0%	Prioridade Baixa	20 pontos
- Comentários:** A section for user comments, showing two entries from 'User X' and 'User Y' with placeholder text, and a text input field for 'Escrever um comentário...'.

Fonte: autoria própria

Figura 36 – Tela da Edição de um Épico no Protótipo de Alta Fidelidade

NOME DA FERRAMENTA
Projeto X
Informações do Projeto
Perfil
Sair

- PROJETOS
- BACKLOG
- SPRINTS
- KANBAN
- DOCUMENTAÇÃO

CONFIGURAÇÕES

Um épico é ... (ex: Login)

Épico 1

Descreva o seu épico, de forma a...

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

Priorizar o épico ajuda a...

Prioridade Alta

Cancelar
Salvar Alterações

Links:

<https://www.link.com.br> x

<https://www.link.com.br> x

+

Arquivos:

[file.pdf](#) x

[image.png](#) x

+

**Features** + Feature

Prioridade: Todas    Completude: Todas    Ordenar por: Maior Prioridade

Feature 1	10%	Prioridade Alta	15 pontos
Feature 2	10%	Prioridade Média	10 pontos
Feature 3	0%	Prioridade Baixa	20 pontos
Feature 4	0%	Prioridade Baixa	20 pontos

1 2 ... >

**Comentários**

User X

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

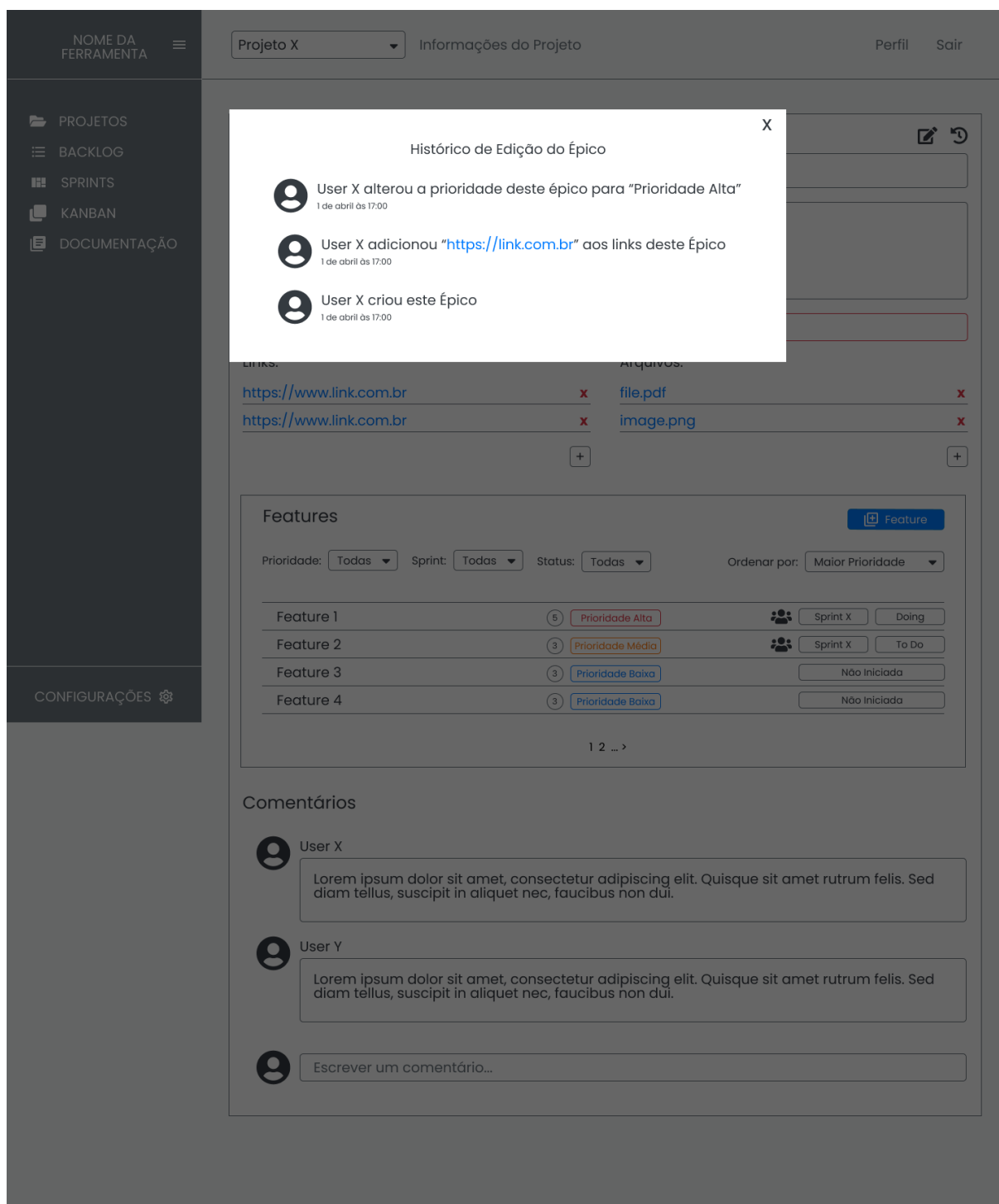
User Y

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

Escrever um comentário...

Fonte: autoria própria

Figura 37 – Tela do Histórico de Atualizações do Épico no Protótipo de Alta Fidelidade



Fonte: autoria própria

Figura 38 – Tela da Criação de uma *Feature* no Protótipo de Alta Fidelidade

Uma Feature é ... (ex: Cadastro de Usuário)

Insira aqui o nome da sua Feature

Descreva a sua feature, de forma a...

Insira aqui a descrição da sua Feature

Priorizar a feature ajuda a...

Atribua a prioridade da sua Feature

Você pode inserir as Features que devem ser completadas antes da sua Feature, para evitar...

Insira uma Feature Bloqueante

Você pode adicionar links à sua Feature, para manter...

Insira um link

Você pode adicionar arquivos à sua Feature, para manter...

Escolha um arquivo

Histórias de Usuário

Eu, como usuário, desejo...

Inserir história

Cancelar Criar Feature

Comentários

User X

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

User Y

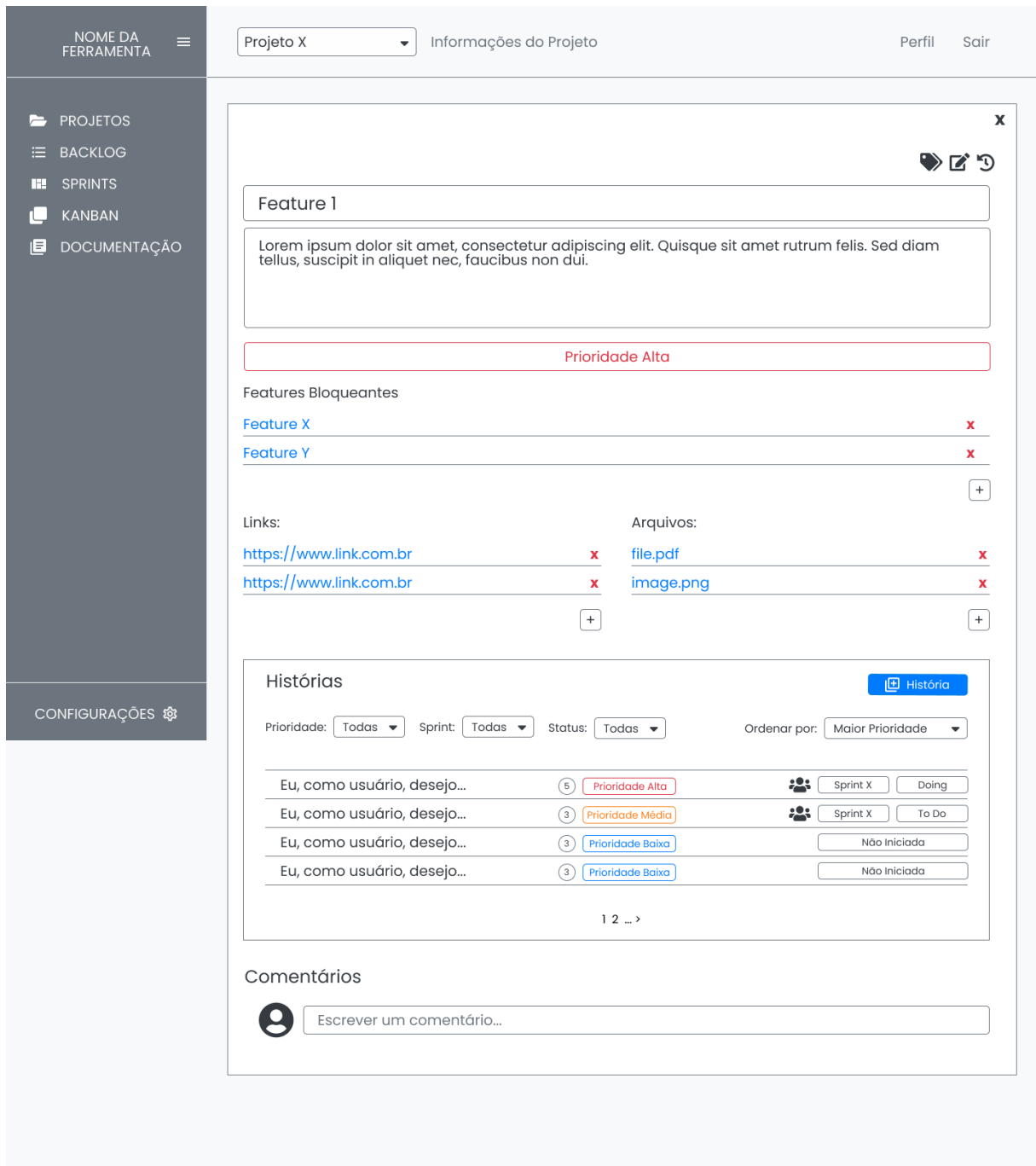
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque sit amet rutrum felis. Sed diam tellus, suscipit in aliquet nec, faucibus non dui.

Escrever um comentário...

Fonte: autoria própria



Figura 39 – Tela das Informações de uma *Feature* no Protótipo de Alta Fidelidade



Fonte: autoria própria

Figura 40 – Tela da Criação de uma História no Protótipo de Alta Fidelidade

Nome da Ferramenta

Membros

Descreva a sua História, de forma a...

Eu, como  , desejo  ,  
para que eu possa

Priorizar a História ajuda a...

Atribua a prioridade da sua História

Estimar o esforço necessário para a realização da Feature é...

Atribua a estimativa de esforço da História

Você pode adicionar links à sua História, para manter...

Insira um link

Você pode adicionar arquivos à sua História, para manter...

Escolha um arquivo

Você pode inserir as Histórias que devem ser completadas antes desta, para evitar...

Insira uma Feature Bloqueante

Você pode adicionar revisores à sua História, para que...

Insira um revisor

**Critérios de Aceitação**

Critério 1

Inserir critério

**Tarefas Técnicas**

Inserir uma tarefa técnica

Inserir tarefa

Cancelar Criar História

User Y

Escrever um comentário...

Fonte: autoria própria

Figura 41 – Tela das Informações de uma História no Protótipo de Alta Fidelidade

Eu, como  , desejo  ,  
para que eu possa  .

**Prioridade Alta**

Links:

<a href="https://www.link.com.br">https://www.link.com.br</a>	x
<a href="https://www.link.com.br">https://www.link.com.br</a>	x

Arquivos:

<a href="#">file.pdf</a>	x
<a href="#">image.png</a>	x

Histórias Bloqueantes

<a href="#">História X</a>	x
<a href="#">História Y</a>	x

Revisores:

User X	x
User X	x

**Critérios de Aceitação**

<input type="checkbox"/> Critério 1	x
<input type="checkbox"/> Critério 2	x

**Tarefas Técnicas**

<input type="checkbox"/> Tarefa 1	x
<input type="checkbox"/> Tarefa 2	x

**Comentários**

Fonte: autoria própria

## 4. Backlog da Sprint

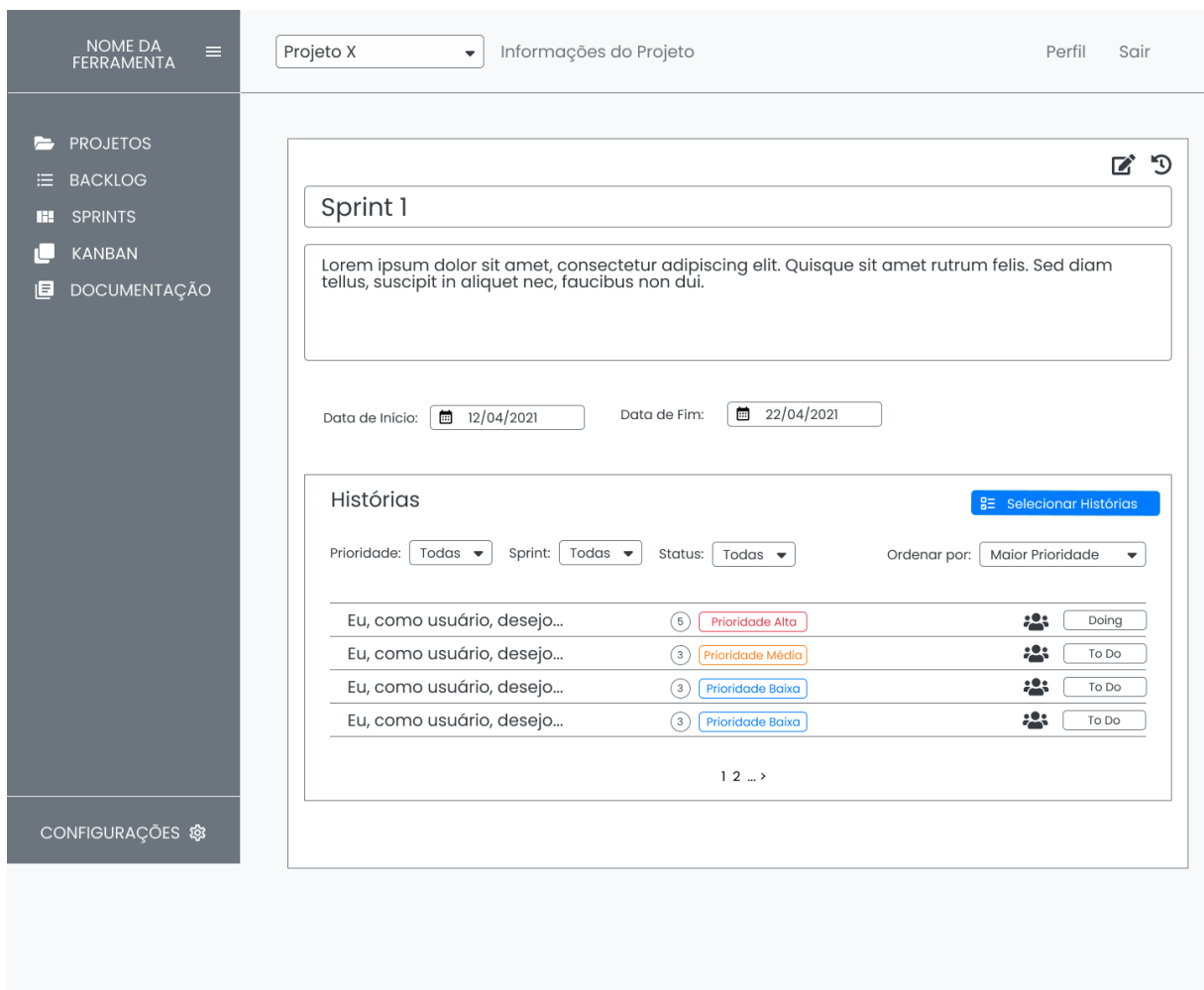
Figura 42 – Tela da Visualização das *Sprints* no Protótipo de Alta Fidelidade

The screenshot displays a web application interface for managing sprints. On the left is a dark sidebar with navigation links: PROJÉTOS, BACKLOG, SPRINTS, KANBAN, and DOCUMENTAÇÃO. At the bottom of the sidebar is a CONFIGURAÇÕES link with a gear icon. The top header shows 'NOME DA FERRAMENTA' with a menu icon, a dropdown menu for 'Projeto X', and 'Informações do Projeto'. On the right of the header are 'Perfil' and 'Sair' links. Below the header is a blue 'Sprint' button. The main content area is titled 'Sprints' and features two dropdown menus: 'Compleitude: Todos' and 'Ordenar por: Mais recente'. Below these is a table with the following data:

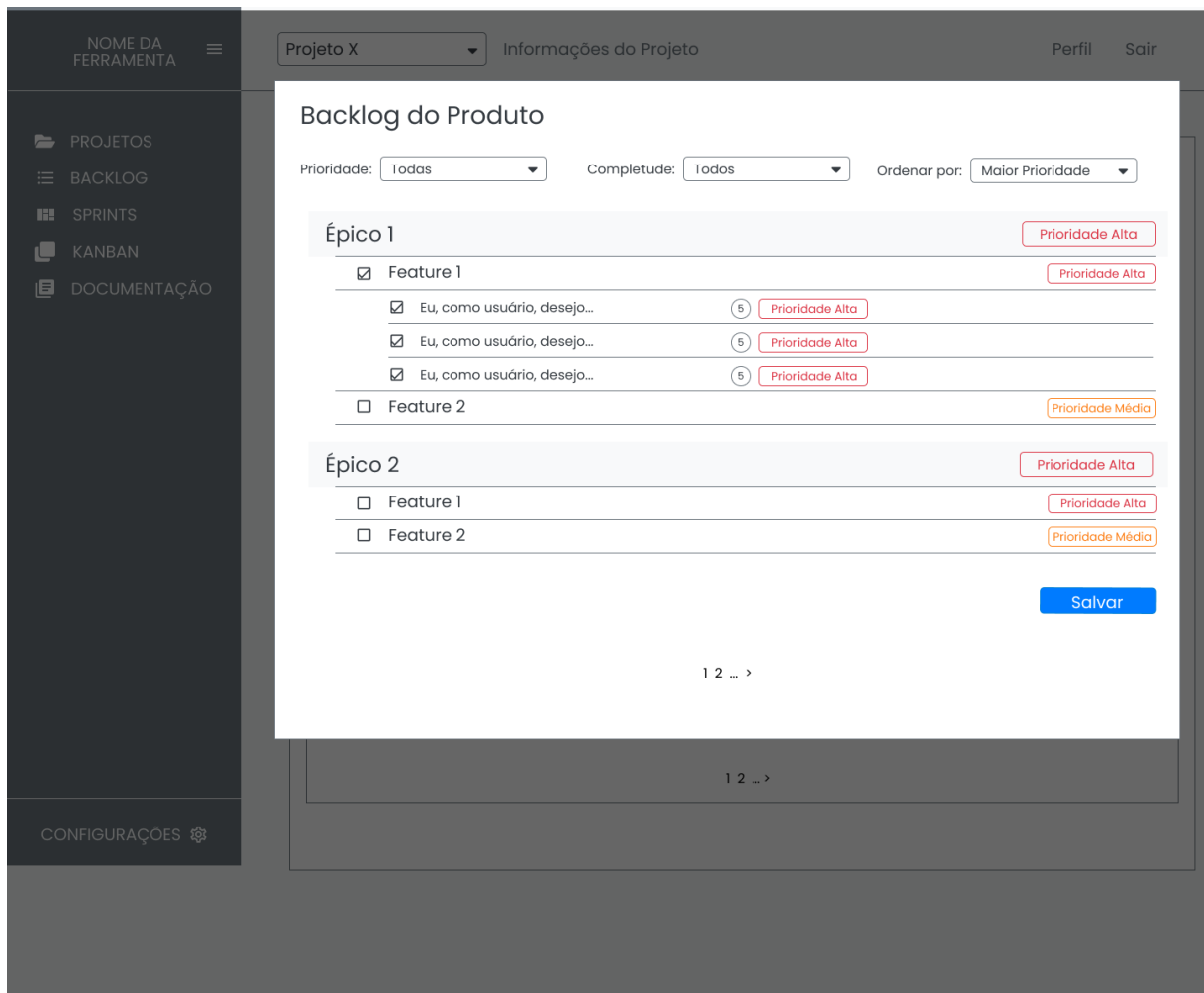
			Início	Fim
Sprint 1	ATUAL	25%	12/04/2021	22/04/2021
Sprint 2	ENCERRADA	25%	12/04/2021	22/04/2021
Sprint 3	ENCERRADA	25%	12/04/2021	22/04/2021
Sprint 4	ENCERRADA	25%	12/04/2021	22/04/2021
Sprint 5	ENCERRADA	25%	12/04/2021	22/04/2021

At the bottom of the table area, there is a pagination control showing '1 2 ... >'.

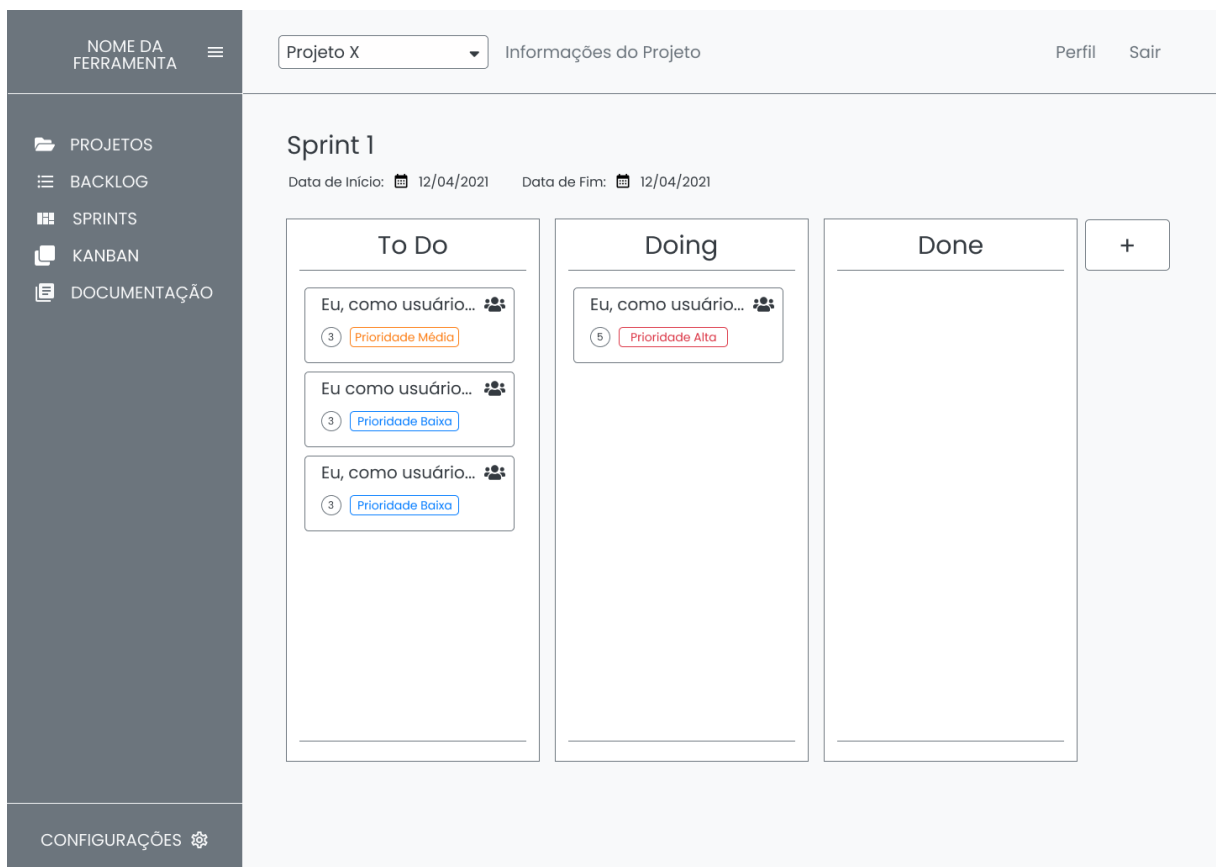
Fonte: autoria própria

Figura 43 – Tela das Informações de uma *Sprint* no Protótipo de Alta Fidelidade

Fonte: autoria própria

Figura 44 – Tela de Seleção de Histórias Para a *Sprint* no Protótipo de Alta Fidelidade

Fonte: autoria própria

Figura 45 – Tela da Visualização do *Kanban* no Protótipo de Alta Fidelidade

Fonte: autoria própria





## APÊNDICE D – Automação do Ambiente de Desenvolvimento do *Front-end*

Para dar apoio à automação do ambiente de desenvolvimento do *front-end* do sistema, foram utilizadas as ferramentas *Docker* e *Docker Compose*. A seguir, estão apresentados, respectivamente, os arquivos de configuração utilizados nas duas ferramentas.

- *Dockerfile*:

Listing D.1 – Arquivo de configuração do *Docker* no *front-end*

```
1
2 FROM node:14
3
4 WORKDIR /app
5
6 ENV PATH /app/node_modules/.bin:$PATH
7
8 COPY package.json ./
9 COPY package-lock.json ./
10 RUN npm install --silent
11
12 COPY . ./
13
14 CMD ["npm", "start"]
15
```

Fonte: autoria própria

- *docker-compose.yml*:

Listing D.2 – Arquivo de configuração do *Docker Compose* no *front-end*

```
1
2 version: '3.8'
3
4 services:
5   app:
6     container_name: app
7     build:
8       context: .
9       dockerfile: Dockerfile
10    volumes:
11      - ./app
12      - /app/node_modules
```

```
13   ports:  
14     - 3000:3000  
15
```

Fonte: autoria própria

# APÊNDICE E – Automação do Ambiente de Desenvolvimento da API

Para dar apoio à automação do ambiente de desenvolvimento da API, foram utilizadas as ferramentas *Docker* e *Docker Compose*. A seguir, estão apresentados, respectivamente, os arquivos de configuração utilizados nas duas ferramentas.

- *Dockerfile*:

Listing E.1 – Arquivo de configuração do *Docker* na API

```
1
2 FROM node:14-alpine as base
3
4 WORKDIR /app-node-report
5
6 COPY package*.json ./
7
8 RUN npm install
9
10 COPY . .
11
12 EXPOSE 8000
13
14 CMD npm run dev
15
```

Fonte: autoria própria

- *docker-compose.yml*:

Listing E.2 – Arquivo de configuração do *Docker Compose* na API

```
1
2 version: '3.8'
3 services:
4   backend:
5     build:
6       context: ./
7     volumes:
8       - ./app
9     ports:
10      - "8000:8000"
11
```

Fonte: autoria própria



## APÊNDICE F – Inicialização do Firebase

O *Firebase* será utilizado no projeto para dar apoio ao sistema de autenticação e gerenciamento de usuários, armazenamento de arquivos e fornecimento do banco de dados da aplicação.

A seguir, estão apresentados os arquivos de inicialização do *firebase* no *front-end* e na API:

- *firebase.js* (*front-end*):

Listing F.1 – Arquivo de inicialização do *Firebase* no *front-end*

```
1
2 import firebase from "firebase/app";
3
4 import "firebase/auth";
5 import "firebase/storage";
6
7 const firebaseConfig = {
8   apiKey: process.env.REACT_APP_API_KEY,
9   authDomain: process.env.REACT_APP_AUTH_DOMAIN,
10  projectId: process.env.REACT_APP_PROJECT_ID,
11  storageBucket: process.env.REACT_APP_STORAGE_BUCKET,
12  messagingSenderId: process.env.REACT_APP_MESSAGING_SENDER_ID,
13  appId: process.env.REACT_APP_APP_ID,
14  measurementId: process.env.REACT_APP_MEASUREMENT_ID,
15 };
16
17 firebase.initializeApp(firebaseConfig);
18
19 export const firebaseAuth = firebase.auth();
20 export const storage = firebase.storage();
21
```

Fonte: autoria própria

- *firebase.js* (API):

Listing F.2 – Arquivo de inicialização do *Firebase* na API

```
1
2 import firebase from "firebase/app";
3
4 import "firebase/firestore";
5 import "firebase/storage";
```

```
6
7 import ENVIRONMENT from "./env";
8
9 const firebaseConfig = {
10   apiKey: ENVIRONMENT.FIREBASE_API_KEY,
11   authDomain: ENVIRONMENT.FIREBASE_AUTH_DOMAIN,
12   projectId: ENVIRONMENT.FIREBASE_PROJECT_ID,
13   storageBucket: ENVIRONMENT.FIREBASE_STORAGE_BUCKET,
14   messagingSenderId: ENVIRONMENT.FIREBASE_MESSAGING_SENDER_ID,
15   appId: ENVIRONMENT.FIREBASE_APP_ID,
16   measurementId: ENVIRONMENT.FIREBASE_MEASUREMENT_ID,
17 };
18
19 firebase.initializeApp(firebaseConfig);
20
21 export const firestore = firebase.firestore();
22
```

Fonte: autoria própria

## APÊNDICE G – Integração Entre os Componentes do Sistema

Para realizar a integração entre os sistemas, uma página de projetos foi construída no *front-end* com o objetivo de recuperar, por meio da API, os dados de alguns projetos cadastrados diretamente no *Cloud Firestore*, banco de dados fornecido pelo *Firebase*, como citado na seção 3.1.9.

A seguir, estão apresentados os principais arquivos do *front-end*, utilizados para apresentar os dados dos projetos cadastrados:

- *Projects.js*:

Listing G.1 – Arquivo de renderização da página de projetos no *front-end*

```
1
2 import React from "react";
3
4 import { getProjects } from "../handlers/api-handler";
5
6 // react-bootstrap components
7 import {
8   Badge,
9   Button,
10  Card,
11  Navbar,
12  Nav,
13  Table,
14  Container,
15  Row,
16  Col,
17 } from "react-bootstrap";
18
19 export default class TableList extends React.Component {
20   state = {
21     rows: []
22   };
23
24   setProjects = async () => {
25     let projects = await getProjects();
26     let rows = [];
27     for(let i = 0; i < projects.length; i++) {
28       rows.push(
```

```
29     <tr key={projects[i].id}>
30       <td>{i}</td>
31       <td>{projects[i].name}</td>
32       <td>{projects[i].description}</td>
33     </tr>
34   )
35 }
36
37   this.setState({ rows });
38 };
39
40 render() {
41   this.setProjects();
42   return (
43     <>
44       <Container fluid>
45         <Row>
46           <Col md="12">
47             <Card className="strpied-tabled-with-hover">
48               <Card.Header>
49                 <Card.Title as="h4">Projetos</Card.Title>
50                 <p className="card-category">
51                   Lista de projetos
52                 </p>
53               </Card.Header>
54               <Card.Body className="table-full-width table-
55 responsive px-0">
56                 <Table className="table-hover table-striped">
57                   <thead>
58                     <tr>
59                       <th className="border-0">ID</th>
60                       <th className="border-0">Nome</th>
61                       <th className="border-0">Descricao</th>
62                     </tr>
63                   </thead>
64                   <tbody>
65                     {this.state.rows}
66                   </tbody>
67                 </Table>
68               </Card.Body>
69             </Card>
70           </Col>
71         </Row>
72       </Container>
73     </>
74   );
75 }
```



```
75  
76 }  
77
```

Fonte: autoria própria

- *api-handler.js*:

Listing G.2 – Arquivo de requisições à API no *front-end*

```
1  
2 import axios from "axios";  
3  
4 const localApiURL = "http://localhost:8000/";  
5  
6 export async function getProjects() {  
7   try {  
8     const result = await axios.get(localApiURL + "projects");  
9     return result.data.data;  
10  } catch(error) {  
11    console.log(error);  
12    return [];  
13  }  
14 }  
15
```

Fonte: autoria própria

A seguir, estão apresentados os principais arquivos da API, utilizados para tratar a requisição oriunda do *front-end*, e recuperar e tratar os dados do *Cloud Firestore*, para retorná-los ao *front-end*:

- *projects.controller.js*:

Listing G.3 – Arquivo de tratamento e resposta à requisição na API

```
1  
2 import { createSuccessResponse, createErrorResponse } from "../util  
  /response.util";  
3 import projectsService from "../services/projects.service";  
4 import { MESSAGE_CODE } from "../message"  
5 import { matchedData } from "express-validator";  
6  
7 const ProjectsController = {  
8  
9   getAll: async (req, res) => {  
10    try {  
11      const data = await projectsService.getAll();  
12      return res.status(200).json(createSuccessResponse(data));  
13    }  
14  }  
15
```

```
13     } catch(e) {
14         return res.status(500).json(createErrorResponse(MESSAGE_CODE.
INTERNAL_ERROR, e));
15     }
16 }
17 }
18
19 export default ProjectsController;
20
```

Fonte: autoria própria

- *projects.service.js*:

Listing G.4 – Arquivo intermediário para a recuperação dos dados dos projetos na API

```
1
2 import firestoreService from "../firestore.service";
3
4 const ProjectsService = {
5
6     getAll: async () => {
7         const request = await firestoreService.get("projects");
8         return request.data;
9     },
10
11 }
12
13 export default ProjectsService;
14
```

Fonte: autoria própria

- *firestore.service.js*:

Listing G.5 – Arquivo de recuperação de dados do *Firestore* na API

```
1
2 import { firestore } from "../firebase";
3
4 const FirestoreService = {
5
6     get: async (collection) => {
7         const response = await firestore.collection(collection)
8             .orderBy('name', 'asc')
9             .get();
10
11         let data = [];
12         response.forEach((doc) => {
13             data.push({
```

```
14         id: doc.id,  
15         ...doc.data()  
16     });  
17 });  
18  
19     return { data };  
20 }  
21  
22 }  
23  
24 export default FirestoreService;  
25
```

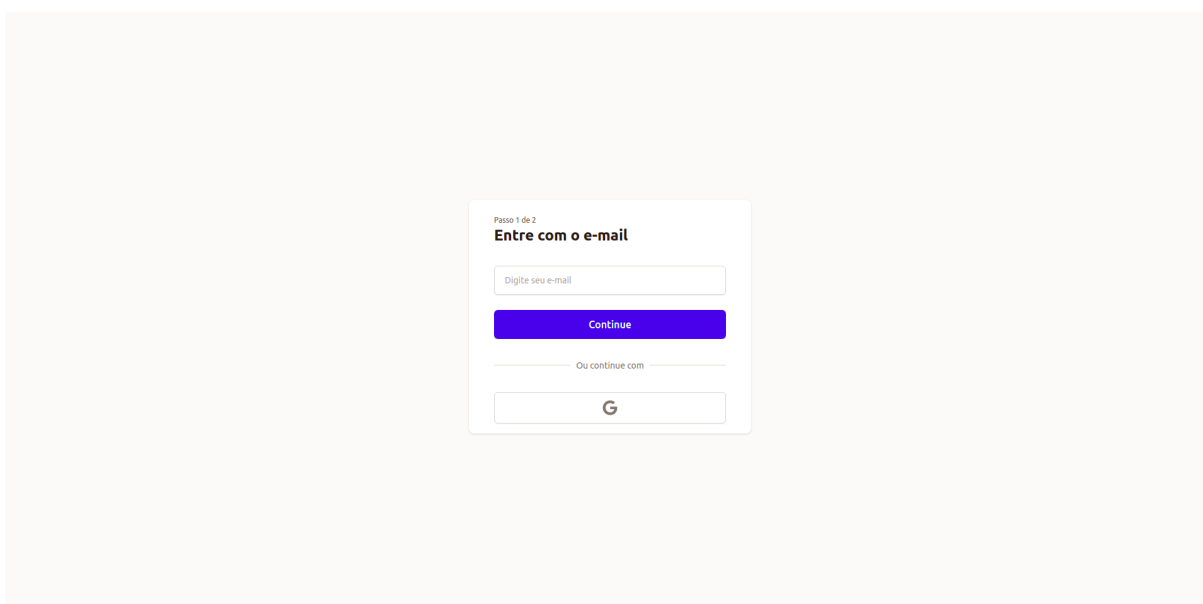
Fonte: autoria própria



## APÊNDICE H – Telas da Ferramenta

A seguir, são apresentadas as telas da ferramenta *Backlog Maker*:

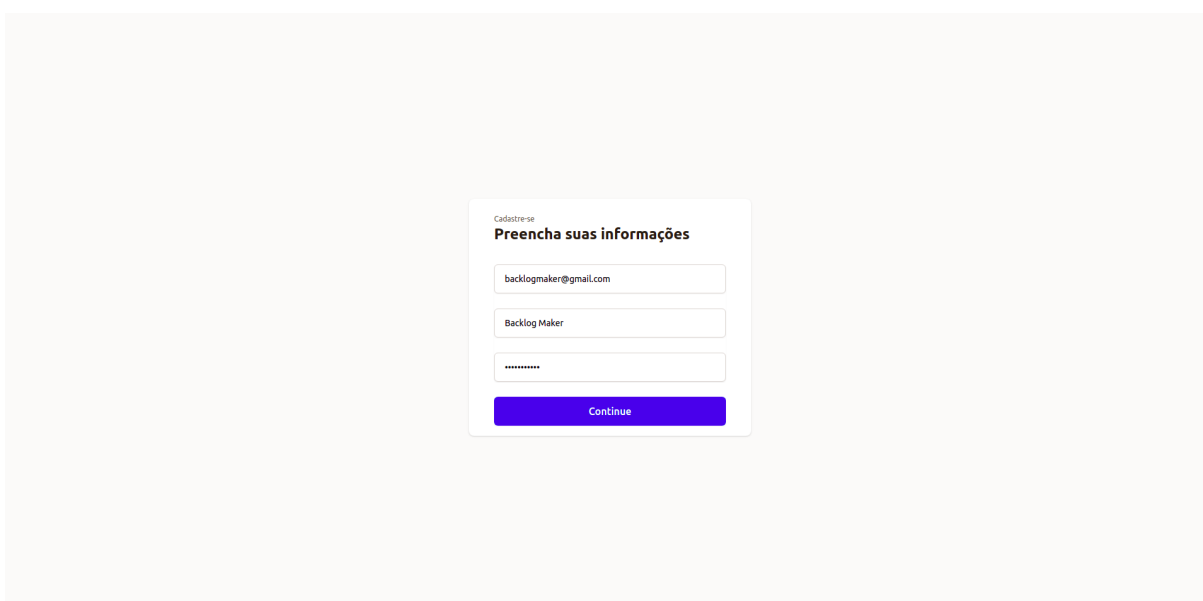
Figura 46 – Tela do Passo 1 do Login no *Backlog Maker*



The screenshot shows a login form titled "Passo 1 de 2" and "Entre com o e-mail". It features a text input field with the placeholder "Digite seu e-mail", a blue "Continue" button, and a section for social login with the text "Ou continue com" and a Google "G" icon button.

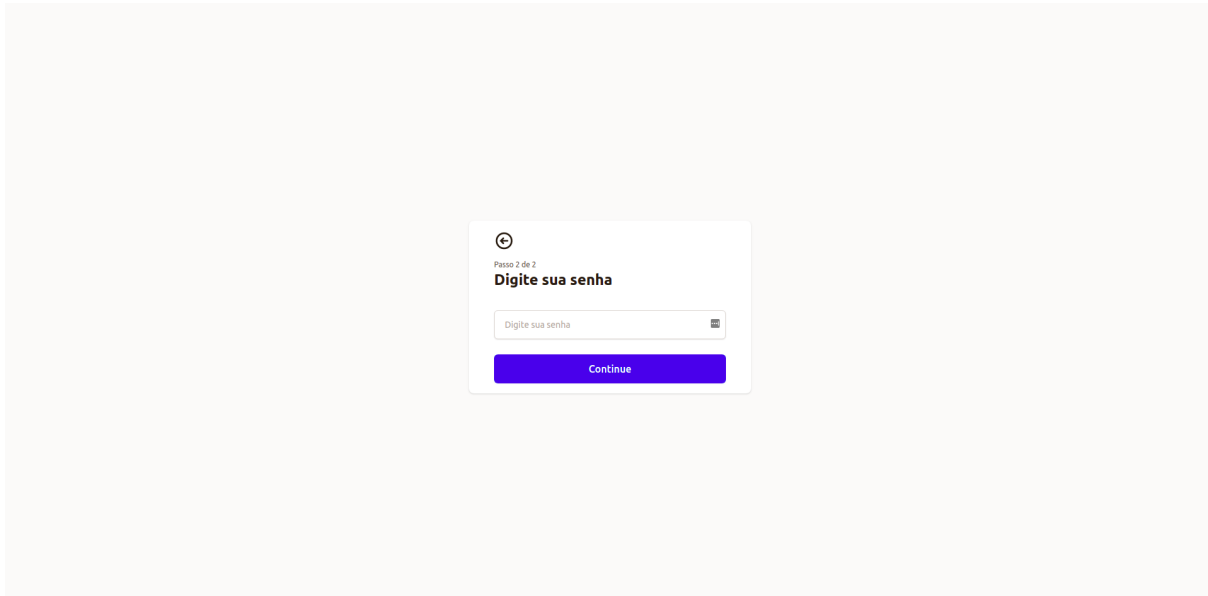
Fonte: autoria própria

Figura 47 – Tela de Cadastro no *Backlog Maker*



The screenshot shows a registration form titled "Cadastre-se" and "Preencha suas informações". It contains three input fields: the first contains "backlogmaker@gmail.com", the second contains "Backlog Maker", and the third is a password field with masked characters. A blue "Continue" button is located at the bottom.

Fonte: autoria própria

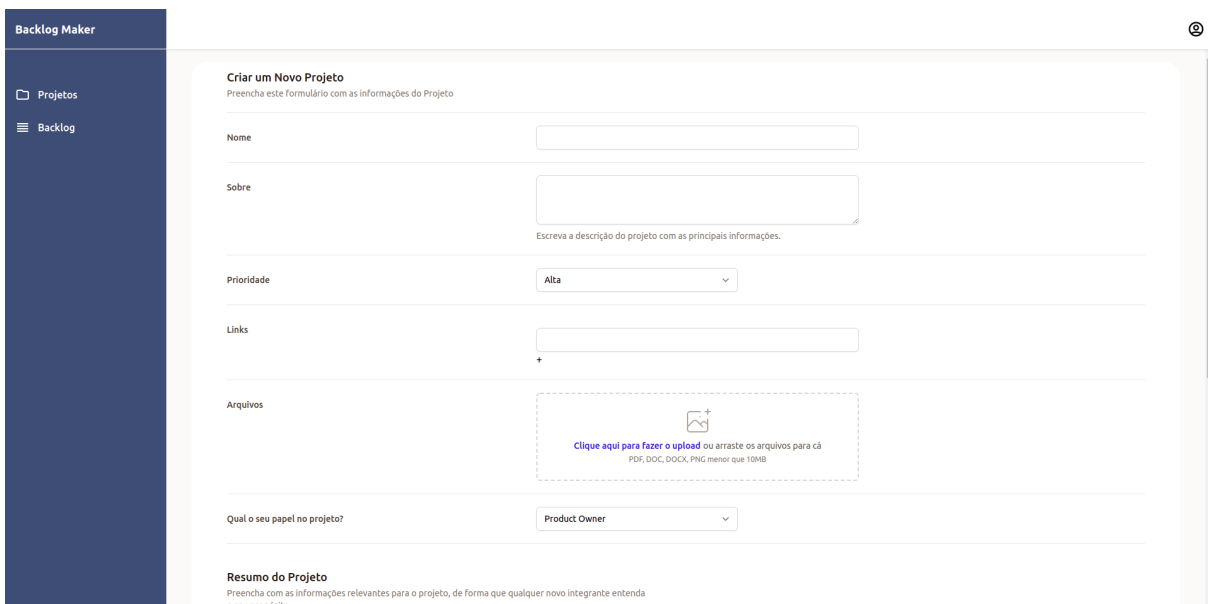
Figura 48 – Tela do Passo 2 do Login no *Backlog Maker*

Passo 2 de 2  
**Digite sua senha**

Digite sua senha

Continue

Fonte: autoria própria

Figura 49 – Tela de Criar Projeto no *Backlog Maker*

Backlog Maker

Projetos

Backlog

**Criar um Novo Projeto**  
Preencha este formulário com as informações do Projeto

Nome

Sobre

Escreva a descrição do projeto com as principais informações.

Prioridade

Alta

Links

Arquivos

Clique aqui para fazer o upload ou arraste os arquivos para cá  
PDF, DOC, DOCX, PNG menor que 10MB

Qual o seu papel no projeto?

Product Owner

**Resumo do Projeto**  
Preencha com as informações relevantes para o projeto, de forma que qualquer novo integrante entenda o caso envolvido.

Fonte: autoria própria

Figura 50 – Tela de Criar Resumo do Projeto no *Backlog Maker*

The screenshot shows the 'Criar Resumo do Projeto' (Create Project Summary) screen in the Backlog Maker application. The interface features a dark blue sidebar on the left with 'Backlog Maker' at the top and navigation options for 'Projetos' and 'Backlog'. The main content area is white and contains the following elements:

- A dashed box at the top with the text: 'Clique aqui para fazer o upload ou arraste os arquivos para cá' and 'PDF, DOC, DOCX, PNG menor que 10MB'.
- A dropdown menu for 'Qual o seu papel no projeto?' with 'Product Owner' selected.
- A section titled 'Resumo do Projeto' with the instruction: 'Preencha com as informações relevantes para o projeto, de forma que qualquer novo integrante entenda o seu propósito.'
- Four text input fields for: 'O que é este projeto?', 'Qual o objetivo deste projeto?', 'De onde surgiu este projeto?', and 'Qual o público-alvo deste projeto?'.
- 'Cancelar' and 'Salvar' buttons at the bottom right.

Fonte: autoria própria

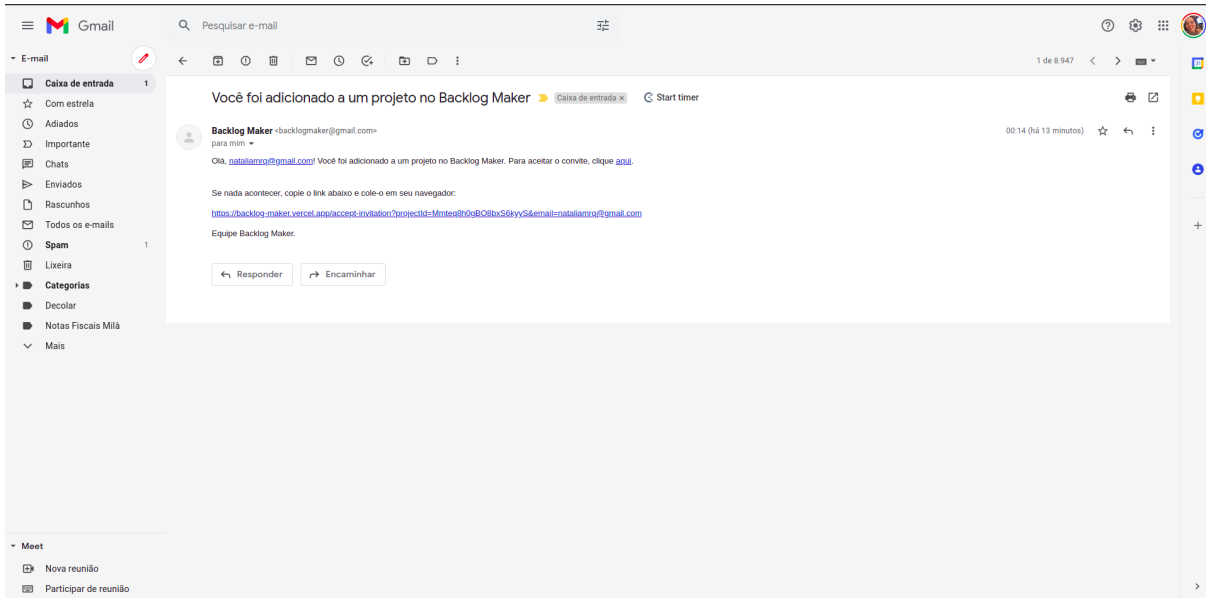
Figura 51 – Tela de Convidar Membro para Participar do Projeto no *Backlog Maker*

The screenshot shows the 'Convidar Membro para Participar do Projeto' (Invite Member to Participate in the Project) screen in the Backlog Maker application. The interface features a dark grey sidebar on the left with 'Backlog Maker' at the top and navigation options for 'Projetos' and 'Backlog'. The main content area is light grey and contains the following elements:

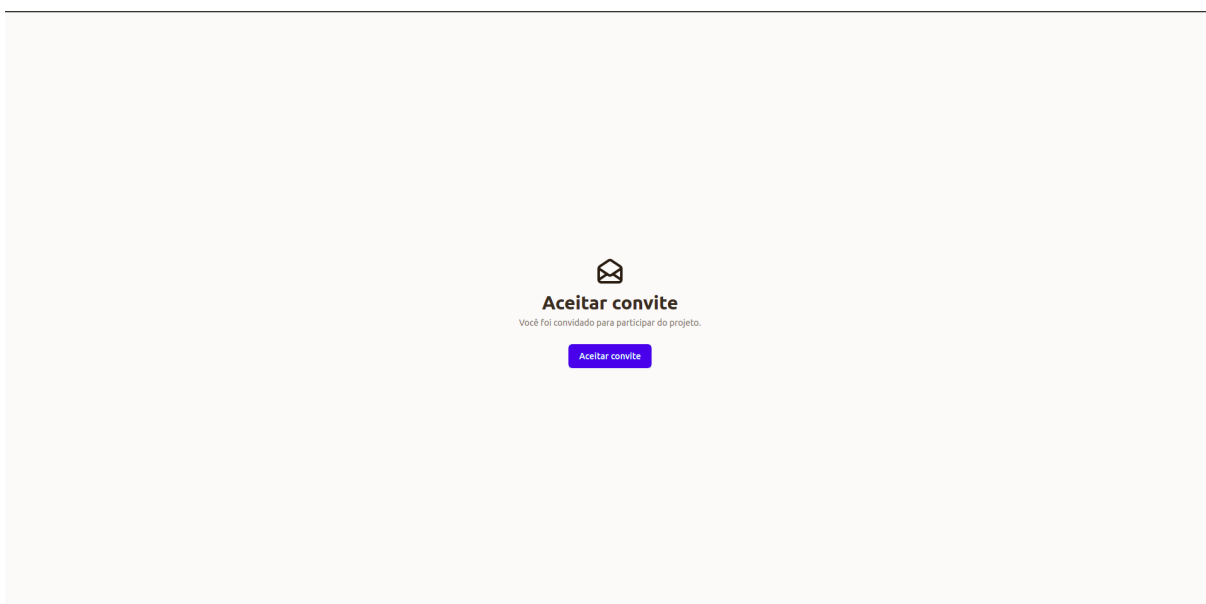
- A 'Novo Projeto' button in the top right corner.
- A list of projects under the heading 'PROJETOS', showing 'Projeto 1' with '0% concluído' and 'Prioridade Alta'.
- A modal dialog box titled 'Convidar usuário' (Invite user) with the following fields:
  - 'E-mail' input field with placeholder text 'Insira o e-mail'.
  - 'Papel' dropdown menu with 'Scrum Master' selected.
  - 'Enviar convite' button.

Fonte: autoria própria

Figura 52 – E-mail Enviado Pela Ferramenta Ao Membro Convidado

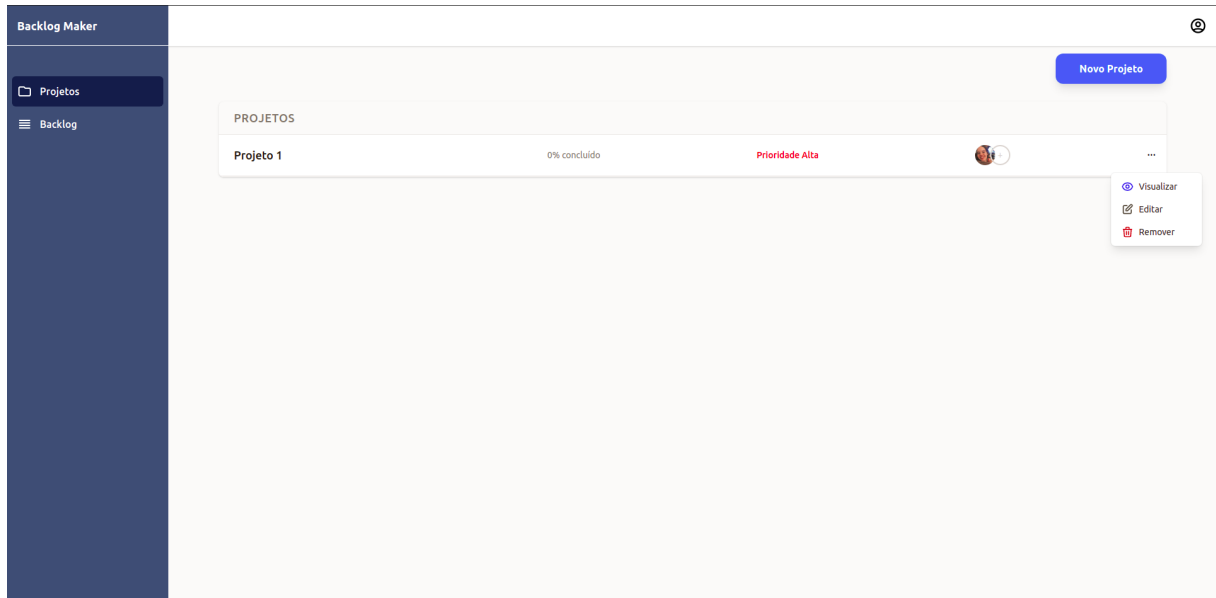


Fonte: autoria própria

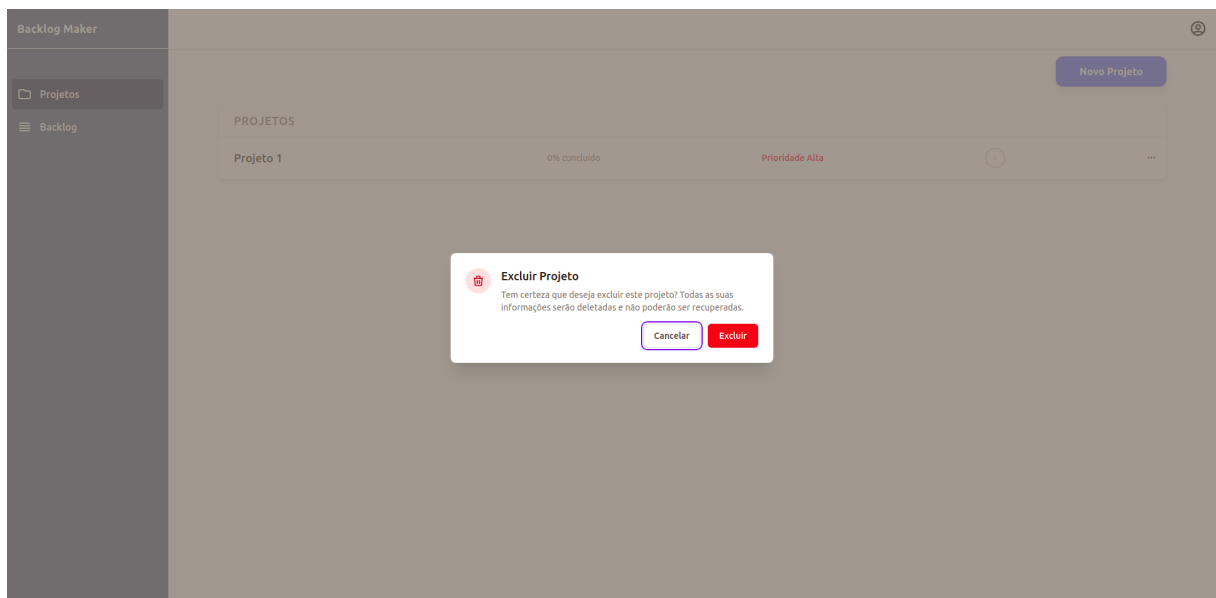
Figura 53 – Tela de Aceitar Convite para Participar do Projeto no *Backlog Maker*

Fonte: autoria própria



Figura 54 – Tela de Visualizar Projetos no *Backlog Maker*

Fonte: autoria própria

Figura 55 – Tela de Excluir Projeto no *Backlog Maker*

Fonte: autoria própria

Figura 56 – Tela de Editar Projeto no *Backlog Maker*

**Editar Projeto**  
Preencha este formulário com as informações do Projeto

Nome: Projeto 1

Sobre:   
Escreva a descrição do projeto com as principais informações.

Prioridade: Alta

Links:

Arquivos:   
Clique aqui para fazer o upload ou arraste os arquivos para cá  
PDF, DOC, DOCX, PNG menor que 10MB

**Resumo do Projeto**  
Preencha com as informações relevantes para o projeto, de forma que qualquer novo integrante entenda o seu propósito.

O que é este projeto?

Fonte: autoria própria

Figura 57 – Tela de Visualizar Projeto no *Backlog Maker*


**Visualizar Projeto**  
Você pode visualizar as informações de Projeto 1

Nome: Projeto 1

Sobre: Projeto Teste

Prioridade: Alta

Links: <https://www.google.com/>

Arquivos:  205895.jpg

**Resumo do Projeto**  
Aqui você encontra as informações relevantes do projeto, para que entenda o seu propósito.

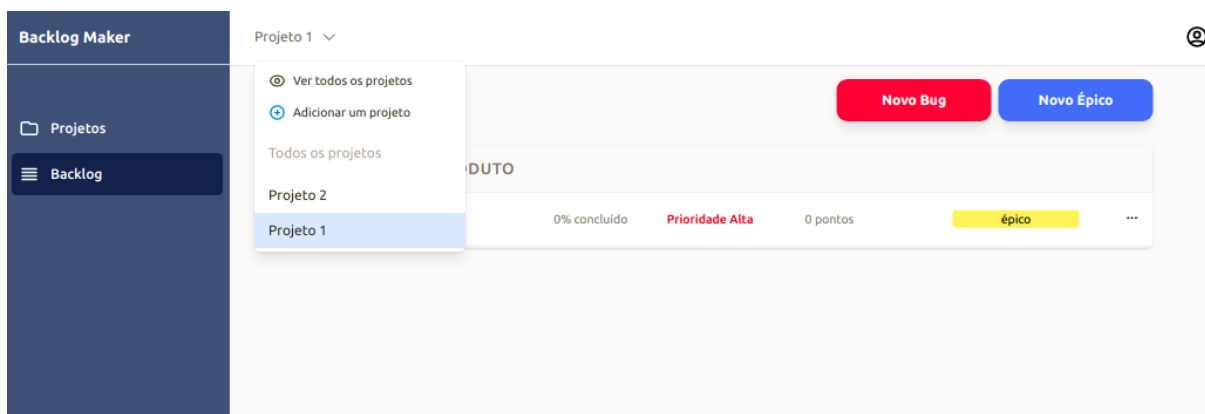
O que é este projeto? Um projeto teste.

Qual o objetivo deste projeto? Testar a ferramenta.

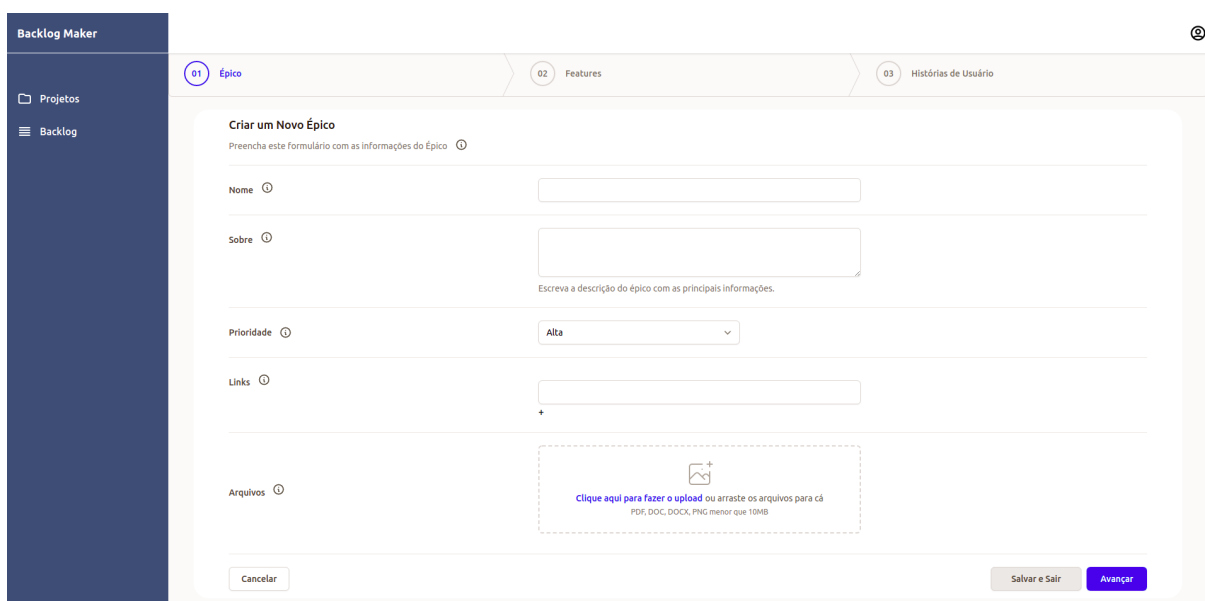
De onde surgiu este projeto? Do TCC.

Qual o público-alvo deste projeto? As autoras do projeto.

Fonte: autoria própria

Figura 58 – Tela de Selecionar Projeto no *Backlog Maker*

Fonte: autoria própria

Figura 59 – Tela de Criar Épico no *Backlog Maker*

Fonte: autoria própria

Figura 60 – Tela de Editar Épico no *Backlog Maker*

**Backlog Maker**

01 Épico | 02 Features | 03 Histórias de Usuário

### Editar Épico

Épico é uma descrição textual de um acúmulo de trabalho abrangente e relacionado, que pode ser realizado em dias, semanas ou meses. Um épico é dividido em Features (próximo passo). Exemplo: Autenticação em um software.

Preencha este formulário com as informações do Épico

Nome:

Sobre:   
Escreva a descrição do épico com as principais informações.

Prioridade:

Links:   
+

Arquivos:   
Clique aqui para fazer o upload ou arraste os arquivos para cá  
PDF, DOC, DOCX, PNG menor que 10MB

Cancelar | Salvar e Sair | Avançar

### Comentários

Sem comentários cadastrados

Digite o seu comentário aqui

Comentar

Fonte: autoria própria

Figura 61 – Tela de Visualizar Épico no *Backlog Maker*

**Backlog Maker**

Projetos  
Backlog

**Visualizar Épico** [Histórico de Edições](#)

Você pode visualizar as informações de Épico 3

**Nome** Épico 3

**Sobre** Sem informação cadastrada

**Prioridade** Alta

**Links** Sem informação cadastrada

**Arquivos** Sem arquivos cadastrados

**Features**

Feature 1	0%	Alta	5		
-----------	----	------	---	--	--

**Comentários**

Malu  
25/10/2021(18:12)  
Comentário

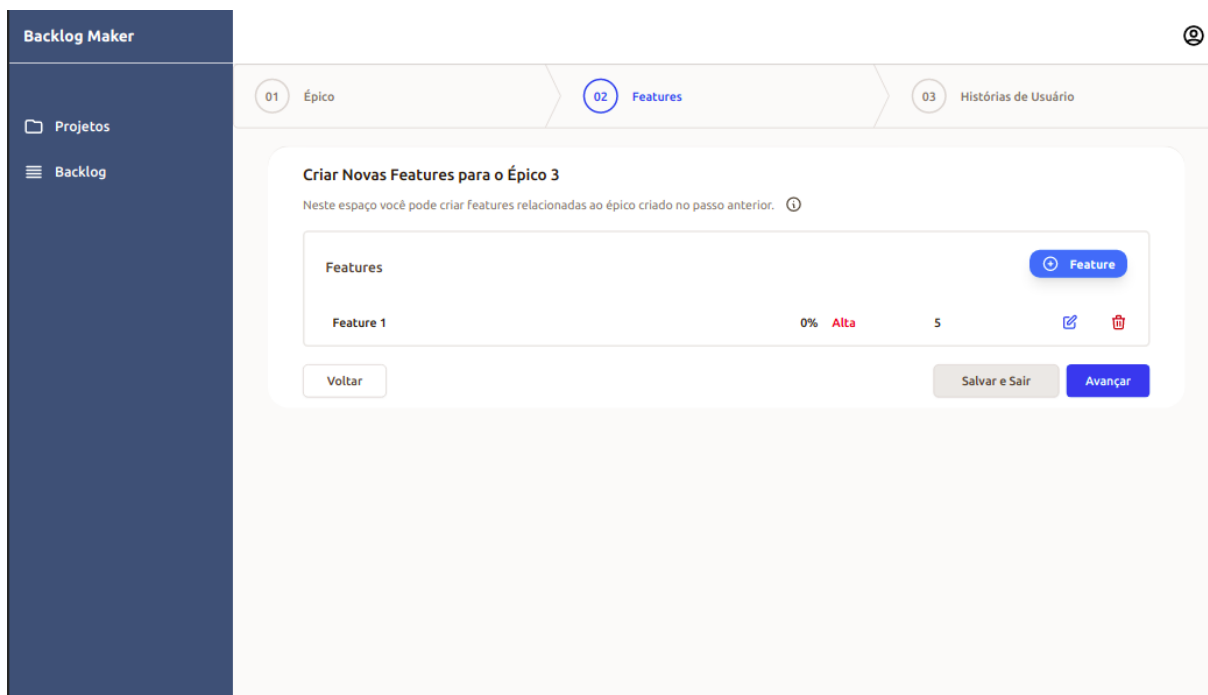
Maria Luiza Almeida (editado)  
25/10/2021(17:56)  
Comentário 1.

Maria Luiza Ferreira (editado)  
25/10/2021(17:59)  
Comentando aqui

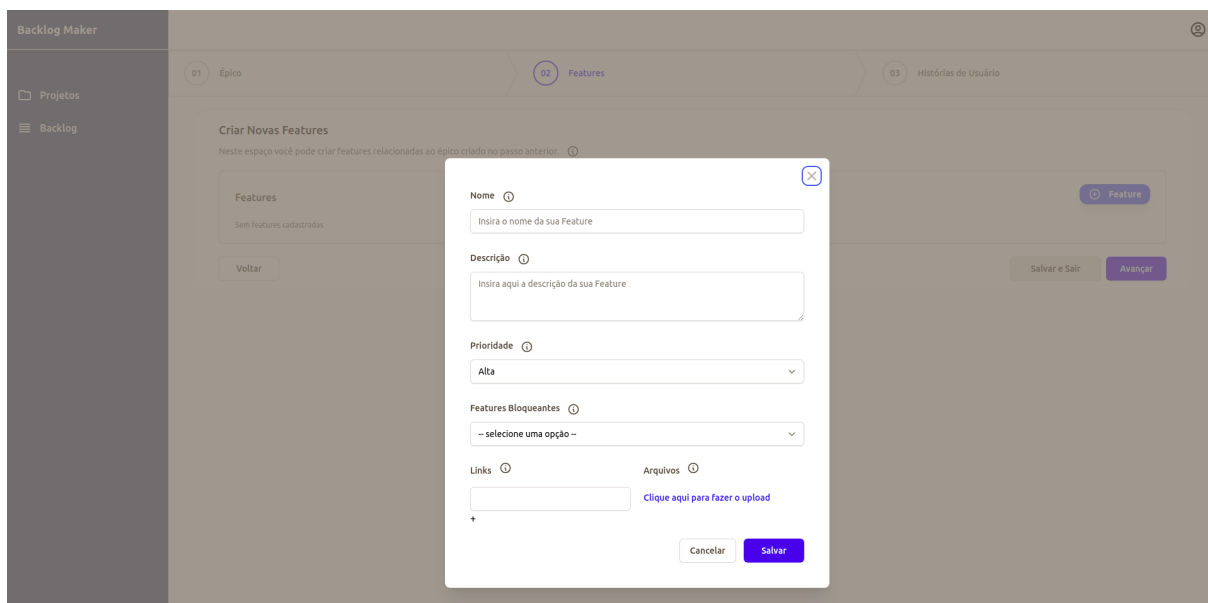
Digite o seu comentário aqui

**Comentar**

Fonte: autoria própria

Figura 62 – Tela de *Features* no Guia do *Backlog Maker*

Fonte: autoria própria

Figura 63 – Tela de Criar *Feature* no *Backlog Maker*

Fonte: autoria própria

Figura 64 – Tela de Editar *Feature* no *Backlog Maker*

Backlog Maker

01 Épico 02 Features 03 Histórias de Usuário

Projetos  
Backlog

### Criar Novas Features para o Épico 1

Neste espaço você pode criar Features relacionadas ao épico criado no passo anterior.

Features	Progresso	Prioridade	Quantidade	Ações
Feature 1	0%	Alta	0	<a href="#">✎</a> <a href="#">🗑️</a>

[Voltar](#) [Salvar e Sair](#) [Avançar](#)

[Histórico de Edições](#) ✕

Nome

Descrição

Prioridade

Features Bloqueantes

Links

Arquivos [Clique aqui para fazer o upload](#)

Captura de tela de 2021-06-09 12:19:44.png [📄](#) [🗑️](#)

[Cancelar](#) [Salvar](#)

#### Comentários

Sem comentários cadastrados

[Comentar](#)

Fonte: autoria própria

Figura 65 – Tela de *Features* com Ajuda no Guia do *Backlog Maker*

The screenshot shows the 'Backlog Maker' interface. On the left is a dark blue sidebar with 'Backlog Maker' at the top and 'Projetos' and 'Backlog' below. The main area has a light gray header with three steps: '01 Épico', '02 Features' (highlighted), and '03 Histórias de Usuário'. A help tooltip is open, explaining that a feature is a textual description of a user action or interaction with the product, typically representing a few days of work, and is composed of user stories. Below the tooltip, there is a section titled 'Criar Novas Features para o Épico 3'. It contains a table with one row for 'Feature 1' showing 0% progress, 'Alta' priority, and a value of 5. There are buttons for 'Voltar', 'Salvar e Sair', and 'Avançar'.

01 Épico

02 Features

03 Histórias de Usuário

**Criar Novas Features para o Épico 3**

Feature é a descrição textual de uma ação ou uma interação do usuário com o produto. Uma feature, geralmente representa um trabalho de alguns dias, menos que o épico. Ela é composta por Histórias de Usuário. Exemplo: Realizar cadastro em um software.

Neste espaço você pode criar features relacionadas ao épico criado no passo anterior.

Features				
Feature 1	0%	Alta	5	

Voltar

Salvar e Sair

Avançar

Fonte: autoria própria



Figura 66 – Tela de Visualizar *Feature* no *Backlog Maker*

Backlog Maker

Projeto

Projetos

Backlog

Visualizar Feature

Histórico de Edições

Você pode visualizar as informações de Feature 1

Nome Feature 1

Sobre Descrição Feature 1

Prioridade Alta

Links <http://localhost:3000/backlog/>

Arquivos

Histórias de Usuário

Eu como, usuário, desejo me cadastrar na ferramenta para que eu possa... Alta 1

Comentários

Sem comentários cadastrados

Digite o seu comentário aqui

Comentar

Fonte: autoria própria

Figura 67 – Tela de Selecionar *Feature* no Guia do *Backlog Maker*

Backlog Maker

01 Épico

02 Features

03 Histórias de Usuário

Projetos

Backlog

Criar Novas Features

Neste espaço você pode criar Features relacionadas ao épico criado no passo anterior.

Features

Feature 1

Voltar

0% Alta 0

Salvar e Sair Avançar

Para criar histórias é necessário que você selecione uma Feature específica

Features

Feature 1

Criar Histórias

Fonte: autoria própria

Figura 68 – Tela de Histórias no Guia do *Backlog Maker*

Backlog Maker

01 Épico 02 Features 03 Histórias de Usuário

Aqui você pode trocar de Feature --selecione uma opção--

**Criar Novas Histórias de Usuário para Feature 1**

ⓘ História de Usuário é uma descrição textual de uma pequena parte da funcionalidade desejada por um usuário final. Uma história, geralmente representa um trabalho de horas ou poucos dias. Exemplo: Como um Usuário não cadastrado, eu gostaria de me cadastrar utilizando o Facebook. ✕

Neste espaço você pode criar histórias de usuário relacionadas a feature criada no passo anterior. ⓘ

Histórias de Usuário			+ História	
Eu como, teste, desejo teste editado 2 para que eu possa teste	Baixa	3	✎	🗑️
Eu como, teste, desejo me cadastrar na ferramenta para que eu possa...	Alta	2	✎	🗑️

Voltar Finalizar

Fonte: autoria própria

Figura 69 – Tela de Criar História no *Backlog Maker*

Backlog Maker

01 Épico 02 Features 03 Histórias de Usuário

Aqui você pode trocar de Feature -- selecione uma opção --

**Descrição** ⓘ Membros

Eu, como  , desejo  para que eu possa

**Observações** ⓘ

**Prioridade** ⓘ

**Estimativa** ⓘ

**Histórias Bloqueantes** ⓘ

**Revisores** ⓘ [Clique aqui para adicionar revisores](#)

**Links** ⓘ

**Arquivos** ⓘ [Clique aqui para fazer o upload](#)

**Critérios de Aceitação** ⓘ Critério

Inserir o critério de aceitação ✕

**Tarefas Técnicas** ⓘ Tarefa

Inserir tarefas técnicas ✕

Cancelar Salvar

Fonte: autoria própria

Figura 70 – Tela de Editar História no *Backlog Maker*

selecione uma opção

Histórico de Edições

Fechar História

Membros

Descrição ⓘ

Eu, como  , desejo  para que eu possa

Observações ⓘ

Descrição da História aqui

Prioridade ⓘ

Alta

Estimativa ⓘ

2

Histórias Bloqueantes ⓘ

-- selecione uma opção --

Revisores ⓘ




[Clique aqui para adicionar revisores](#)

Links ⓘ


Link 1

Arquivos ⓘ

[Clique aqui para fazer o upload](#)


+  Captura de tela de 2021-06-09 18-10-51.png  

Critérios de Aceitação ⓘ

Critério 1 

[+ Critério](#)

Tarefas Técnicas ⓘ

Tarefa 1 

[+ Tarefa](#)

Cancelar

Salvar

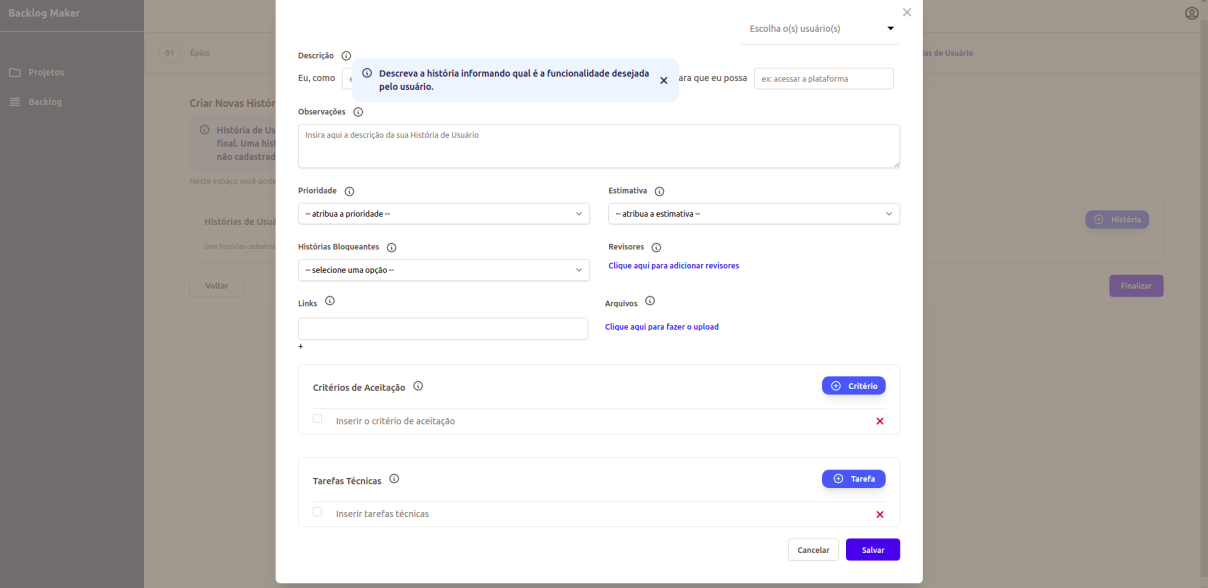
Comentários

Sem comentários cadastrados

Digite o seu comentário aqui

[Comentar](#)

Fonte: autoria própria

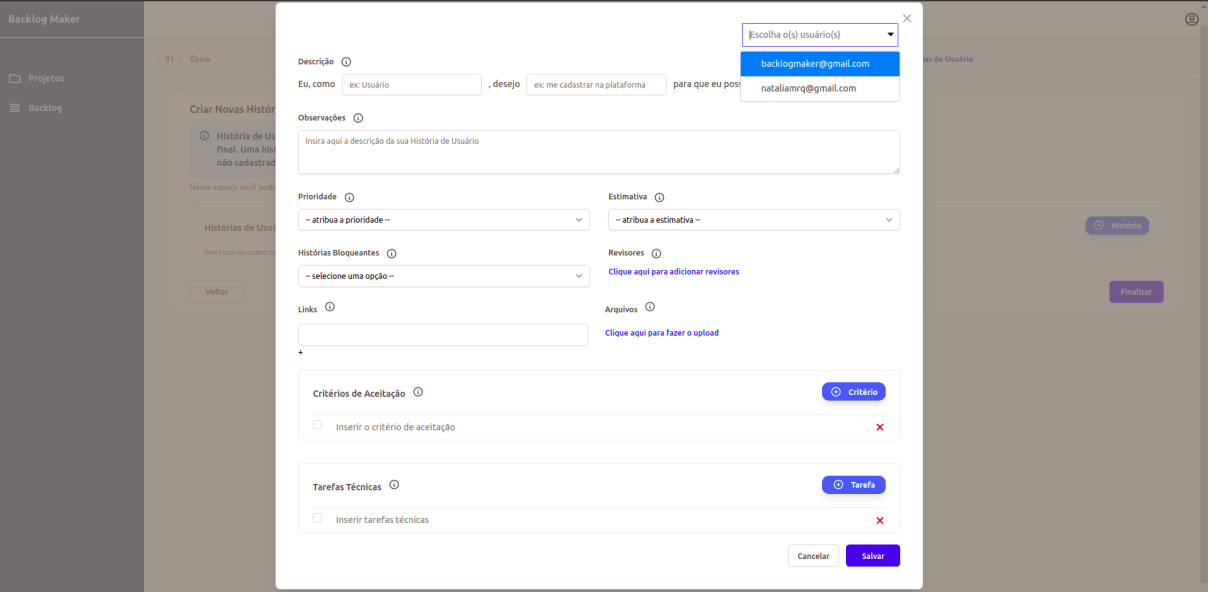
Figura 71 – Tela de Criar História com Ajuda no *Backlog Maker*

The screenshot shows the 'Criar História' (Create Story) form in the Backlog Maker application. The form is displayed in a modal window over a blurred background of the application interface. The form includes the following sections:

- Descrição:** A text input field with a tooltip that reads: "Descreva a história informando qual é a funcionalidade desejada pelo usuário." (Describe the story by informing what is the desired functionality for the user.)
- Observações:** A text area for additional notes.
- Prioridade:** A dropdown menu with the option "-- atribua a prioridade --".
- Estimativa:** A dropdown menu with the option "-- atribua a estimativa --".
- Histórias Bloqueantes:** A dropdown menu with the option "-- selecione uma opção --".
- Revisores:** A section with a link "Clique aqui para adicionar revisores" (Click here to add reviewers).
- Arquivos:** A section with a link "Clique aqui para fazer o upload" (Click here to upload).
- Links:** A text input field for adding links.
- Critérios de Aceitação:** A section with a "Critério" button and a checkbox "Inserir o critério de aceitação" (Insert acceptance criterion).
- Tarefas Técnicas:** A section with a "Tarefa" button and a checkbox "Inserir tarefas técnicas" (Insert technical tasks).

At the bottom of the form are "Cancelar" and "Salvar" (Save) buttons.

Fonte: autoria própria

Figura 72 – Tela de Adicionar Membros à História no *Backlog Maker*

This screenshot is similar to Figure 71, showing the 'Criar História' form. The key difference is in the 'Descrição' field, which now contains a dropdown menu for selecting users. The dropdown is open, showing a list of email addresses:

- backlogmaker@gmail.com
- nataliamrq@gmail.com

The form structure and other sections (Observações, Prioridade, Estimativa, etc.) are identical to the previous figure.

Fonte: autoria própria

Figura 73 – Tela de Visualizar História no *Backlog Maker*

**Backlog Maker**

Projetos

Backlog

**Visualizar História de Usuário** [Histórico de Edições](#)

Você pode visualizar as informações desta História de Usuário

História	Eu como, <b>usuário</b> , desejo <b>me cadastrar na ferramenta</b> para que eu possa <b>acessar a plataforma</b>
Observações	Descrição
Prioridade	Alta
Estimativa	1
Revisores	
Links	Link 1
Arquivos	Sem arquivos cadastrados
Critérios de Aceitação	<ul style="list-style-type: none"><li>[A fazer] Critério 1</li></ul>
Tarefas Técnicas	<ul style="list-style-type: none"><li>[A fazer] Tarefa 1</li></ul>

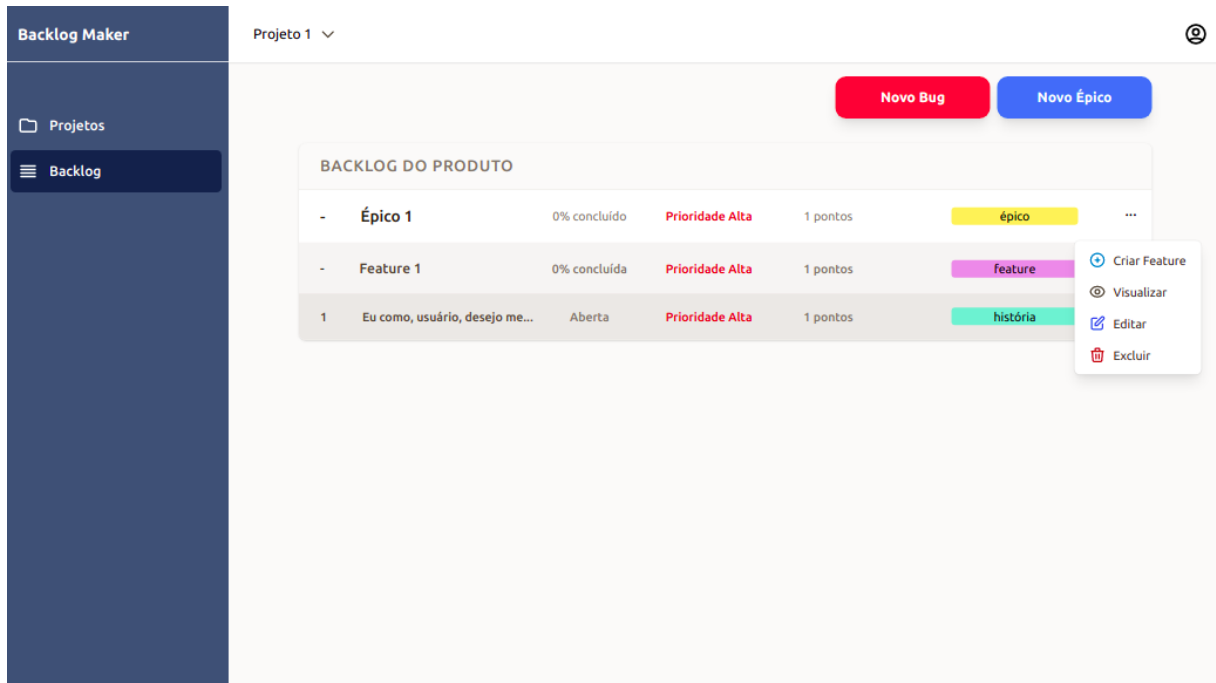
**Comentários**

Sem comentários cadastrados

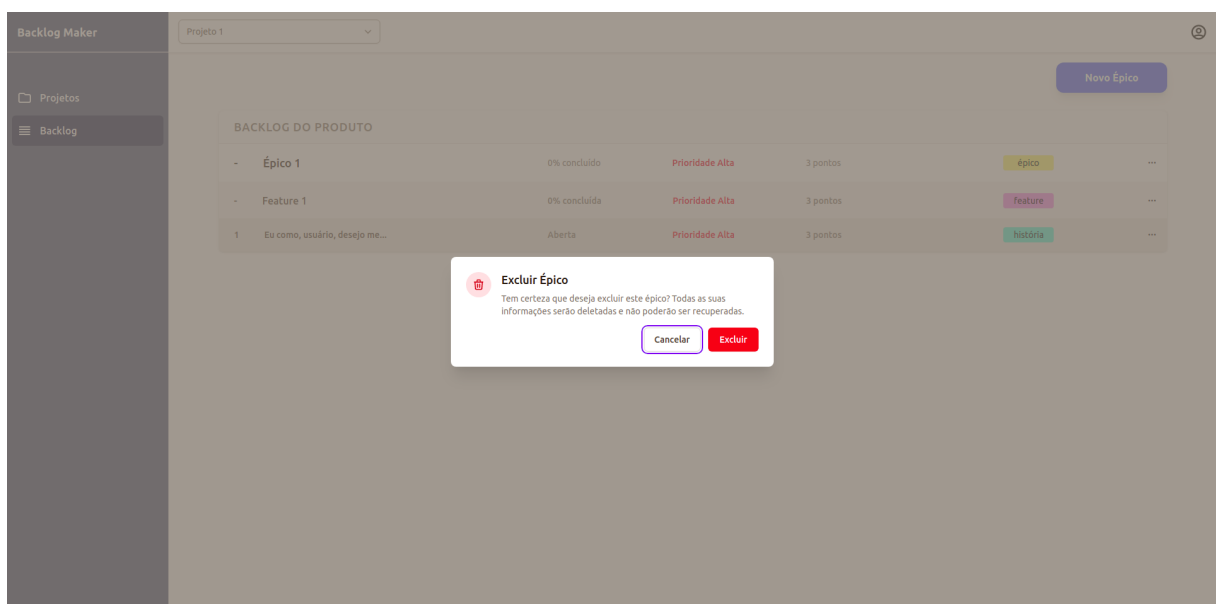
Digite o seu comentário aqui

Comentar

Fonte: autoria própria

Figura 74 – Tela de Visualizar o *Backlog* do Produto no *Backlog Maker*

Fonte: autoria própria

Figura 75 – Tela de Excluir Item do *Backlog* do Produto no *Backlog Maker*

Fonte: autoria própria

Figura 76 – Tela de Criar *Bugs* com Ajuda no *Backlog Maker*

**Backlog Maker**

Projetos

Backlog

### Criar um Novo Bug

📘 Bug é um erro ou falha em um software, o qual produz um resultado incorreto ou inesperado. ✕

Preencha este formulário com as informações do Bug 📘

Nome 📘

Resultado esperado 📘   
Escreva o resultado esperado do bug com as principais informações.

Resultado obtido 📘   
Escreva o resultado obtido do bug com as principais informações.

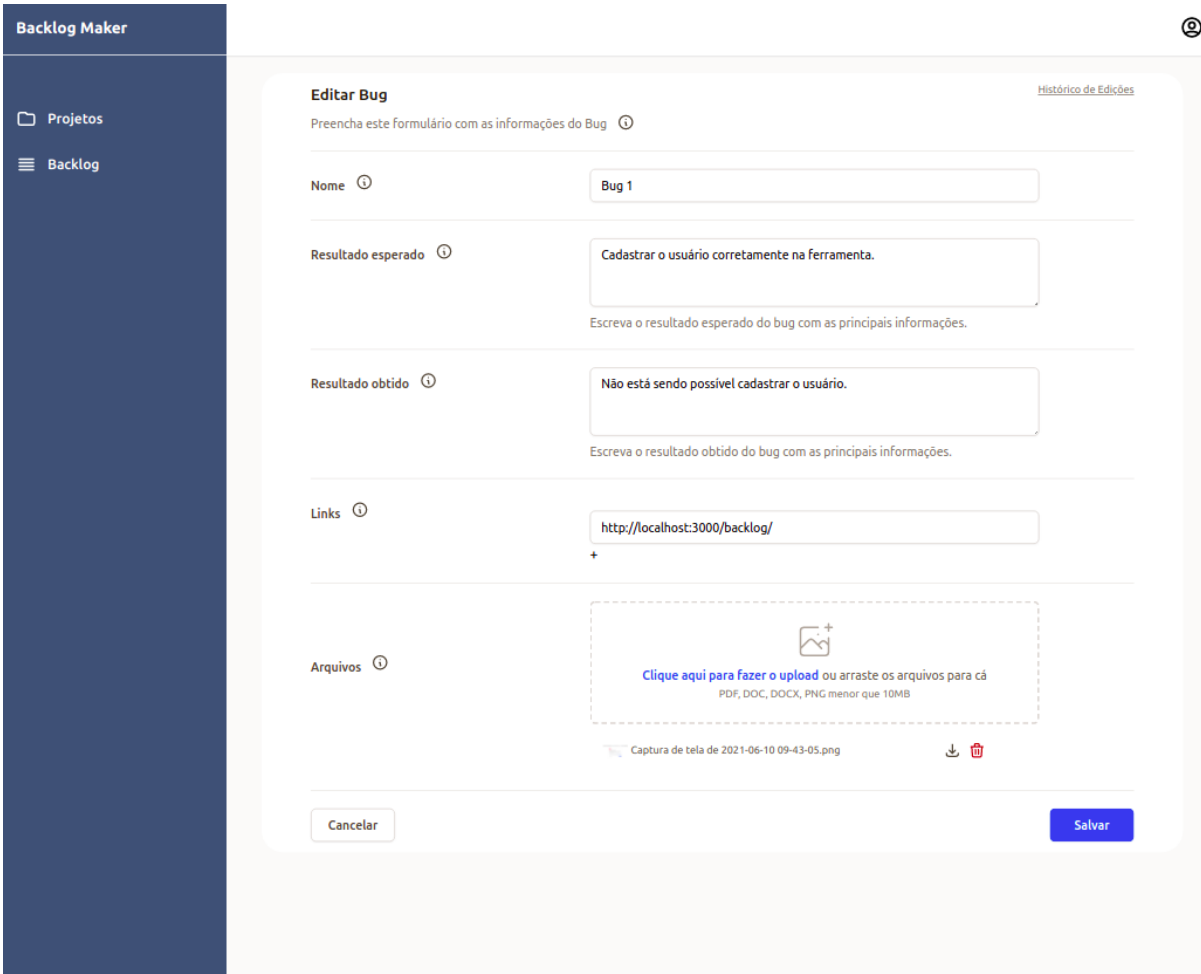
Links 📘   
+

Arquivos 📘   
Clique aqui para fazer o upload ou arraste os arquivos para cá  
PDF, DOC, DOCX, PNG menor que 10MB

Cancelar Salvar

Fonte: autoria própria



Figura 77 – Tela de Editar *Bugs* no *Backlog Maker*

**Backlog Maker**

Projeto

Backlog

### Editar Bug

[Histórico de Edições](#)

Preencha este formulário com as informações do Bug

Nome: Bug 1

Resultado esperado: Cadastrar o usuário corretamente na ferramenta.  
Escreva o resultado esperado do bug com as principais informações.

Resultado obtido: Não está sendo possível cadastrar o usuário.  
Escreva o resultado obtido do bug com as principais informações.

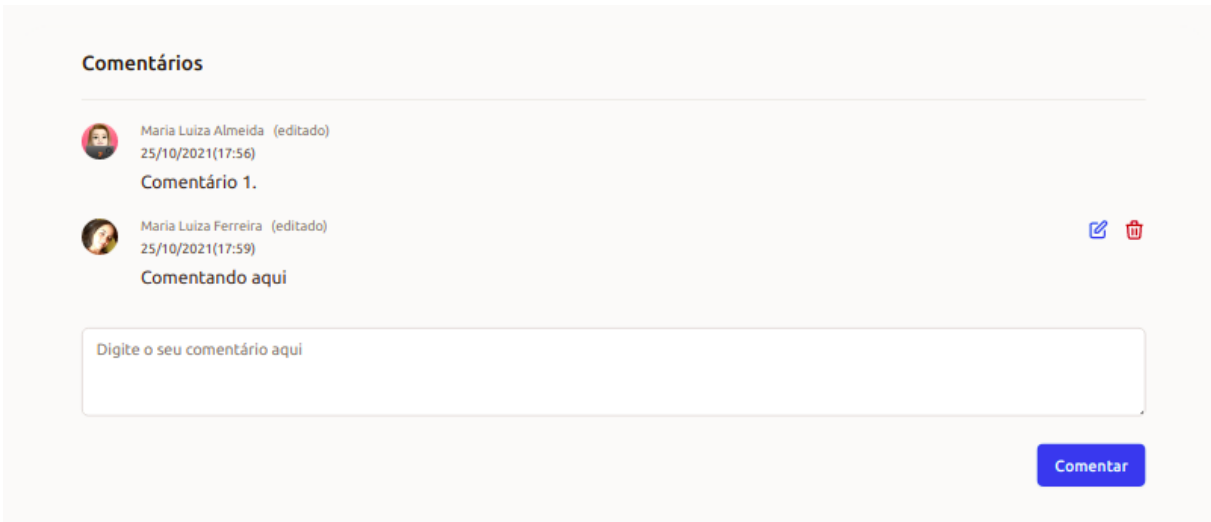
Links: <http://localhost:3000/backlog/>

Arquivos: [Clique aqui para fazer o upload](#) ou arraste os arquivos para cá  
PDF, DOC, DOCX, PNG menor que 10MB

Captura de tela de 2021-06-10 09-43-05.png

Cancelar Salvar

Fonte: autoria própria

Figura 78 – Tela de Comentários dos Itens do *Backlog* do Produto no *Backlog Maker*

### Comentários

Maria Luiza Almeida (editado)  
25/10/2021(17:56)  
Comentário 1.

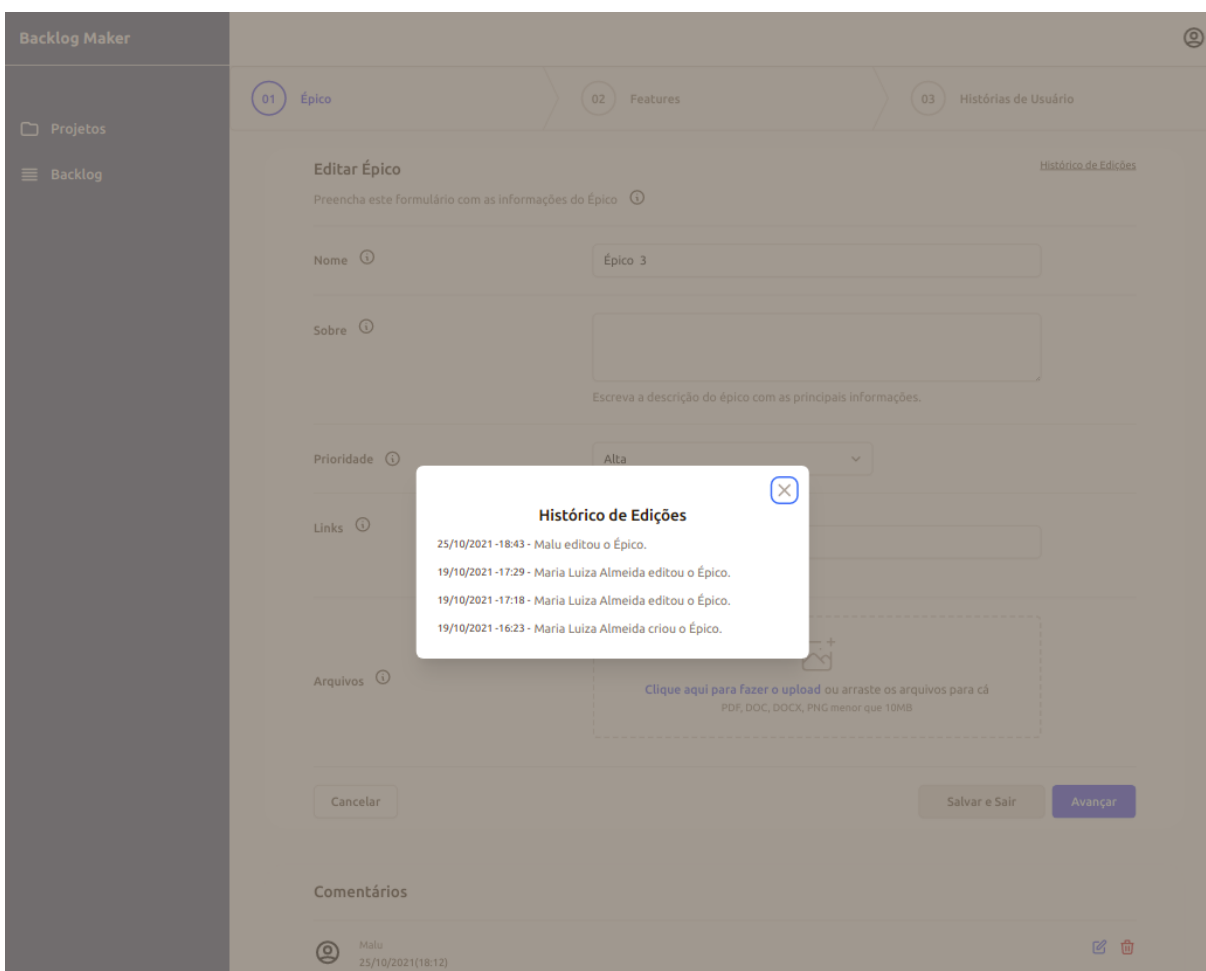
Maria Luiza Ferreira (editado)  
25/10/2021(17:59)  
Comentando aqui

Digite o seu comentário aqui

Comentar

Fonte: autoria própria

Figura 79 – Tela de Histórico de Edições dos Itens do *Backlog* do Produto no *Backlog Maker*



Fonte: autoria própria

Figura 80 – Tela de Visualizar *Backlog* do Produto com Itens Concluídos no *Backlog Maker*

The screenshot displays the 'Backlog Maker' interface for 'Projeto 2'. The left sidebar contains 'Projetos' and 'Backlog'. The main area shows a table titled 'BACKLOG DO PRODUTO' with the following items:

Ítem	Status	Prioridade	Pontos	Ação
- Épico 3	100% concluído	Prioridade Alta	5 pontos	épico concluído
- Feature 1	100% concluída	Prioridade Alta	5 pontos	feature concluída
1 Eu como, teste, desejo teste...	Concluída	Prioridade Baixa	3 pontos	história concluída
2 Eu como, teste, desejo me...	Concluída	Prioridade Alta	2 pontos	história concluída
+ Épico 1 editado	0% concluído	Prioridade Alta	0 pontos	épico
+ Épico 2	0% concluído	Prioridade Alta	0 pontos	épico
Bug 1				bug

Fonte: autoria própria