

Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de *Software*

***UserX Story+*: Incorporando Elementos de UX e Usabilidade a Histórias de Usuário**

Autor: Matheus Salles Blanco
Orientador: Profa. Dra. Elaine Venson

Brasília, DF
2023



Matheus Salles Blanco

***UserX Story+*: Incorporando Elementos de UX e Usabilidade a Histórias de Usuário**

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. Elaine Venson

Brasília, DF

2023

Matheus Salles Blanco

UserX Story+: Incorporando Elementos de UX e Usabilidade a Histórias de Usuário/ Matheus Salles Blanco. – Brasília, DF, 2023-
103 p. : il. (algumas color.) ; 30 cm.

Orientador: Profa. Dra. Elaine Venson

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2023.

1. Usabilidade. 2. Requisitos. I. Profa. Dra. Elaine Venson. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. *UserX Story+*: Incorporando Elementos de UX e Usabilidade a Histórias de Usuário

CDU 02:141:005.6

Este trabalho é dedicado a todos que acreditam em mim.

Agradecimentos

Agradeço primeiramente à minha família, por sempre acreditar em mim e me motivar a me desenvolver mais e mais.

Agradeço também aos meus amigos e colegas que estiveram acompanhando de perto minha jornada no curso, à minha empresa júnior que me proporcionou muitos aprendizados profissionais e ao meu grupo escoteiro, que me proporcionou aprendizados pessoais.

Por último agradeço à minha orientadora, que acatou minhas ideias e fez o melhor possível para me guiar no caminho certo.

Obrigado.

“I do all I can“.
(Sharon Van Etten)

Resumo

A elicitação de requisitos é uma área da Engenharia de Software de grande importância para o ciclo de vida de construção do mesmo e determina sua qualidade e potencial. Para que um software possa ser bem avaliado e utilizado pelo seu público-alvo é importante que este seja de fácil utilização e que o usuário tenha uma experiência agradável. Alguns estudos procuraram realizar adições de elementos de usabilidade em requisitos especificados com histórias de usuário. Este trabalho busca evoluir um destes estudos, em que os autores geraram uma nova gramática para uma das técnicas da engenharia de requisitos - a história de usuário. O estudo em questão indica a necessidade de se determinar o nível de detalhamento dos critérios de aceitação gerados com a nova técnica. Utilizando-se dos achados no estudo mencionado e dos entendimentos decorridos da leitura do *framework* Os Cinco Planos, o qual visa classificar a usabilidade de um sistema a partir de estruturas de abstração e concretude, desenvolveu-se e testou-se uma evolução da gramática do estudo foco deste trabalho. Para se verificar a eficácia da técnica, testes de aceitação foram realizados com dois públicos diferentes, desenvolvedores de software e escoteiros, estes últimos sendo o público-alvo de uma aplicação em que as histórias de usuário, geradas com a técnica, foram implementadas. A partir destes testes, este trabalho discute a eficácia da técnica desenvolvida.

Palavras-chave: usabilidade; UX; requisitos; histórias de usuário; critérios de aceitação.

Abstract

Requirements elicitation is an area of Software Engineering of great importance for the software construction life cycle and determines its quality and potential. For a software to be well evaluated and used by its target audience, it is important that it is easy to use and that the user has a pleasant experience. Some studies sought to add usability elements to requirements specified with user stories. This work seeks to evolve one of these studies, in which the authors generated a new grammar for one of the requirements engineering techniques - the user story. The study in question indicates the need to determine the level of detail of the acceptance criteria generated with the new technique. Using the findings of the aforementioned study and the understandings derived from reading the *framework* Os Cinco Planos, which aims to classify the usability of a system based on structures of abstraction and concreteness, a test was developed and tested. evolution of the grammar of the study focus of this work. To verify the effectiveness of the technique, acceptance tests were carried out with two different audiences, software developers and scouts, the latter being the target audience of an application in which user stories, generated with the technique, were implemented. Based on these tests, this paper discusses the effectiveness of the developed technique.

Key-words: usability; UX; requirements; user stories; acceptance criteria.

Lista de ilustrações

Figura 1 – <i>Framework</i> dos Cinco Planos, Adaptada de: Garrett (2010), (GARRETT, 2010)	35
Figura 2 – <i>UserX Story</i> , Fonte: Autor, (CHOMA; ZAINA; BERALDO, 2016)	39
Figura 3 – TCC: Fluxos de trabalho, Fonte: autor	43
Figura 4 – Fluxo de criação da técnica, Fonte: autor	46
Figura 5 – Fluxo de análise da técnica, Fonte: autor	47
Figura 6 – Fluxo geral de avaliação da técnica adaptada, Fonte: autor	48
Figura 7 – Proposta original da User Story, Fonte: Choma, Zaina e Beraldo (2016)	52
Figura 8 – Nova proposta de evolução de User Story, Fonte: autor	53
Figura 9 – Nova proposta de evolução de User Story, UserX Story, Fonte: autor	53
Figura 10 – Nova proposta de evolução de User Story, Camada de Superfície, Adaptado de: Garrett (2010)	54
Figura 11 – Nova proposta de evolução de User Story, Camada de Esqueleto, Adaptado de: Garrett (2010)	55
Figura 12 – Nova proposta de evolução de User Story, Camada de Estrutura, Adaptado de: Garrett (2010)	56
Figura 13 – Exemplo de história de usuário criada com a evolução da técnica, Fonte: autor	57
Figura 14 – US1, Fonte: desenvolvedores participantes do <i>Workshop</i>	84
Figura 15 – US2, Fonte: desenvolvedores participantes do <i>Workshop</i>	85
Figura 16 – US3, Fonte: desenvolvedores participantes do <i>Workshop</i>	86
Figura 17 – US4, Fonte: desenvolvedores participantes do <i>Workshop</i>	86
Figura 18 – US5, Fonte: desenvolvedores participantes do <i>Workshop</i>	87
Figura 19 – US6, Fonte: desenvolvedores participantes do <i>Workshop</i>	88
Figura 20 – US7, Fonte: desenvolvedores participantes do <i>Workshop</i>	89
Figura 21 – US8, Fonte: desenvolvedores participantes do <i>Workshop</i>	90
Figura 22 – US9, Fonte: desenvolvedores participantes do <i>Workshop</i>	91
Figura 23 – US10, Fonte: desenvolvedores participantes do <i>Workshop</i>	92
Figura 24 – US11, Fonte: desenvolvedores participantes do <i>Workshop</i>	93
Figura 25 – US12, Fonte: desenvolvedores participantes do <i>Workshop</i>	94
Figura 26 – US13, Fonte: desenvolvedores participantes do <i>Workshop</i>	95
Figura 27 – US14, Fonte: desenvolvedores participantes do <i>Workshop</i>	96
Figura 28 – Tela de organizador, Fonte: autor	97
Figura 29 – Tela de organizador em partida antes de escolher participante, Fonte: autor	97

Figura 30 – Tela de organizador em partida depois de escolher participante, Fonte: autor	98
Figura 31 – Tela de campista, Fonte: autor	98
Figura 32 – Tela de campista em partida antes de poder responder, Fonte: autor	99
Figura 33 – Tela de campista em partida com opções de escolha, Fonte: autor	99
Figura 34 – Tela de organizador, Fonte: autor	101
Figura 35 – Tela de organizador ao criar partida, Fonte: autor	101
Figura 36 – Tela de organizador criando partida com as opções, Fonte: autor	102
Figura 37 – Tela de organizador em partida antes de escolher participante, Fonte: autor	102
Figura 38 – Tela de campista em partida antes de ser escolhido, Fonte: autor	103
Figura 39 – Tela de campista em partida com as opções de resposta, Fonte: autor	103

Lista de tabelas

Tabela 1 – Tabela Exemplo de História de Usuário com Critérios de Aceitação . . .	34
Tabela 2 – Requisitos de novas funcionalidades	60
Tabela 3 – Requisitos de problemas encontrados com os testes	61
Tabela 4 – Tabela de médias das perguntas, versões 1 e 2	67
Tabela 5 – Tabela de notas gerais das versões 1 e 2	67

Lista de abreviaturas e siglas

UX	Experiência de Usuário
IHC	Interação Humano-Computador
ER	Engenharia de Requisitos
US	História de Usuário
AC	Critério de Aceitação
TCC1	Trabalho de Conclusão de Curso 1
TCC2	Trabalho de Conclusão de Curso 2

Sumário

1	INTRODUÇÃO	21
1.1	Contextualização	21
1.2	Problema	23
1.3	Objetivo	24
1.3.1	Objetivo Geral	24
1.3.2	Objetivos Específicos	26
1.3.3	Metodologia	26
1.4	Organização do Documento	26
2	REFERENCIAL TEÓRICO	29
2.1	Usabilidade	29
2.1.1	Usabilidade em produtos de <i>software</i>	30
2.2	Engenharia de Requisitos (ER)	30
2.2.1	Requisitos Não-Funcionais (NFR)	31
2.2.2	Requisitos Funcionais (RF)	32
2.2.3	Técnicas de elicitação de requisitos focadas neste trabalho	33
2.2.3.1	Histórias de usuário	33
2.2.3.2	Critérios de aceitação	33
2.3	Os Cinco Planos	34
2.3.1	Camada Superfície	35
2.3.2	Camada Esqueleto	35
2.3.3	Camada Estrutura	36
2.3.4	Camada Escopo	36
2.3.5	Camada Estratégia	36
2.3.6	Produto como Funcionalidade e Produto como Informação	36
2.4	As Heurísticas de Nielsen	37
2.4.1	Diálogo simples e natural	37
2.4.2	Falar a língua do usuário	37
2.4.3	Minimizar a carga de memória dos usuários	37
2.4.4	Consistência	37
2.4.5	<i>Feedback</i>	38
2.4.6	Saídas claras	38
2.4.7	Atalhos	38
2.4.8	Boas mensagens de erro	38
2.4.9	Prever erros	38

2.4.10	Ajuda e documentação	38
2.5	<i>UserX Story</i>	38
2.6	UX no desenvolvimento de <i>software</i>	39
2.7	Testes A/B	39
2.8	<i>Net Promoter Score</i>	40
3	METODOLOGIA	41
3.1	Detalhamento do plano metodológico	42
3.2	Planejamento da Pesquisa	43
3.3	Revisão Bibliográfica	43
3.3.1	Proposta de Solução	45
3.3.1.1	Metodologia de Desenvolvimento da técnica	46
3.4	Coleta de Dados	47
3.4.1	Metodologia de Análise da Técnica	47
3.4.2	Análise e interpretação dos dados	48
3.4.2.1	Sobre a coleta de percepções iniciais	48
3.4.2.2	Sobre o <i>Workshop</i>	49
3.4.2.3	Sobre a validação da técnica	49
3.4.2.4	Análise e avaliação de resultados	50
3.4.3	Divulgação dos resultados	50
4	A ABORDAGEM <i>USERX STORY+</i>	51
4.1	Definição da <i>UserX Story+</i>	51
4.2	Sobre a técnica	52
4.2.1	<i>UserX Story</i>	52
4.2.2	Os Cinco Planos	54
4.2.2.1	Critérios de aceitação da superfície	54
4.2.2.2	Critérios de aceitação do esqueleto	55
4.2.2.3	Critérios de aceitação da estrutura	55
4.2.3	Exemplo de história de usuário criada com a técnica	56
5	COLETA DE DADOS	59
5.1	Atividades pré-Workshop	59
5.1.1	Seleção de aplicação base	59
5.1.2	Testes de usabilidade com a aplicação original	59
5.1.3	Seleção e agregação de desenvolvedores para o <i>Workshop</i>	60
5.2	Atividades durante o workshop	61
5.3	Atividades pós-Workshop	62
5.3.1	Implementação das histórias de usuário	62
5.3.2	Nova coleta de <i>feedbacks</i>	62

6	ANÁLISE E INTERPRETAÇÃO DOS DADOS	65
6.1	Implementação das histórias de usuário na aplicação base	65
6.1.1	Metodologia de análise	65
6.1.2	Análise e resultados	67
6.2	Utilização da <i>UserX Story+</i> para criação de histórias de usuário . .	69
6.2.1	Metodologia de análise	69
6.2.2	Análise	70
7	RESULTADOS	73
8	CONCLUSÃO	75
	REFERÊNCIAS	77
	ANEXOS	81
	ANEXO A – HISTÓRIAS DE USUÁRIO CRIADAS NO <i>WORKSHOP</i>	83
	ANEXO B – APLICAÇÃO BASE ANTES DAS MELHORIAS . . .	97
	ANEXO C – APLICAÇÃO BASE DEPOIS DAS MELHORIAS . . .	101

1 Introdução

Este trabalho é uma monografia na categoria de Trabalho de Conclusão de Curso (TCC) a ser submetido para conclusão de graduação do aluno no curso de Engenharia de Software da Universidade de Brasília. O mesmo foi estruturado de acordo com os padrões e espera facilitar a leitura e entendimento do leitor sobre o assunto, apresentando aspectos-chaves para o entendimento geral do trabalho aqui realizado. Este capítulo terá textos explicativos da contextualização, o problema identificado na contextualização, objetivos do trabalho para atingir o problema e organização da monografia.

1.1 Contextualização

A elicitação de requisitos é uma fase importante para a idealização e projeção de um software. [Tiwari e Rathore \(2017\)](#) menciona que a qualidade do produto de software depende da qualidade dos requisitos elicitados e que esses definem o sucesso ou a falha de um projeto, sendo que a consideração desses requisitos deve ser constante e os requisitos podem e devem ser revisitados em etapas mais avançadas do projeto, de maneira que se garanta que o planejamento está sendo seguido ([JURISTO; MORENO; SANCHEZ-SEGURA, 2007](#)) ([SOUZA; CONTE; ZAINA, 2020](#)).

Os requisitos podem ser classificados em duas interpretações diferentes. Os funcionais, relacionados a funcionalidades e ações do sistema, e não-funcionais, relacionados a aspectos de qualidade ([ECKHARDT; VOGELSANG; FERNÁNDEZ, 2016](#)) ([JURISTO; MORENO; SANCHEZ-SEGURA, 2007](#)). A elicitação de requisitos não-funcionais não é uma tarefa simples, pelo fato dos mesmos serem comumente desassociados com funcionalidades específicas do sistema (e.g.: cadastro, ver lista de dados etc), e sim relacionados com questões como segurança, confiabilidade, facilidade de entendimento ([SILVA et al., 2017](#)). Esses requisitos se tornam mais abstratos de se entender, planejar e implementar ([SILVA et al., 2017](#)).

A correta adequação e planejamento desses requisitos é capaz de transformar um software em um produto de grande valor para seus usuários e para o(s) criador(es) e dono(s). E o contrário também se demonstra verdade, onde a negligência de se elicitar e planejar corretamente tais requisitos pode trazer sérias consequências para um projeto de software que já está em andamento ([SILVA et al., 2017](#)).

Elementos de usabilidade normalmente são trazidos para o contexto de um projeto de software como requisitos não-funcionais (NFR) ([SILVA et al., 2017](#)). A 11^a, décima-primeira, parte da ISO 9241-11:2018 ([STANDARDIZATION, 2018](#)) define usabilidade

como:

"Extensão em que um sistema, produto ou serviço pode ser usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico."

E define *User Experience* (UX), como:

"Percepções e respostas do usuário que resultam do uso e/ou uso antecipado de um sistema, produto ou serviço."

Pode-se considerar a usabilidade como uma parte de UX, sendo uma das áreas de estudo da mesma que atinge diretamente a facilidade de uso de um sistema ou aplicação, a partir da percepção do usuário, e pode ser vinculada a processos e melhorias de *design* (BEVAN, 2009 apud TRAYNOR, 2022).

A compreensão de todas as tarefas necessárias para que um produto se torne *user-friendly*, e garanta um *feedback* positivo do usuário abrange o entendimento de características que tragam ao usuário emoções de agrado e satisfação, é essencial e complexa. Isso ocorre por dependerem de fatores que não estão diretamente relacionados com a funcionalidade básica da aplicação, mas sim com as sensações e respostas dos usuários frente à utilização de dito produto (GARCEZ et al., 2021a).

A usabilidade possui uma forma de elicitação mais tradicional e rigorosa, o que contrasta e conflita com a elicitação em metodologia ágil de requisitos de software, causando dificuldades em se realizar a integração das duas atividades (ANANJEVA; PERSSON; BRUUN, 2020). Deve ser considerada e planejada com cautela e estudo, devendo ser feita nas fases iniciais da construção de um produto de software, pois no caso de haver a necessidade de inserir um elemento de usabilidade em etapas mais avançadas, o re-trabalho causaria grande estresse e custo para a equipe responsável (JURISTO; MORENO; SANCHEZ-SEGURA, 2007). Para que esse tipo de situação possa ser evitada e o software seja elicitado com qualidade, a inserção e compreensão de elementos de usabilidade durante a elicitação de requisitos se demonstra fundamental (SOUZA; CONTE; ZAINA, 2020) (ANANJEVA; PERSSON; BRUUN, 2020) (CHOMA; ZAINA; BERALDO, 2016).

Os conhecimentos de usabilidade não estão presentes e disponíveis para todos os times de projeto (NIELSEN, 2005). Juristo, Moreno e Sanchez-Segura (2007) informa que há a dificuldade em se entender os requisitos de usabilidade existentes pelo fato dos mesmos não conterem informações o suficiente para que desenvolvedor e *designer* possam inseri-la na aplicação. Há a necessidade de se simplificar a aplicação de elementos de

usabilidade e de facilitar o entendimento desses elementos a partir de uma especificação mais informativa.

Além da elicitação e especificação de requisitos de funcionalidade, deve-se pensar em como essas funcionalidades serão percebidas pelo utilizador. Aspectos de qualidade devem ser incorporados para que não se crie a impressão no usuário que o software não possui valor como produto ou não é confiável (SILVA et al., 2016). Tal impressão acaba por definir se o usuário irá continuar a usar o produto, ou não.

1.2 Problema

Há uma dificuldade em se elicitar qualidade em software. Várias áreas de conhecimento são necessárias para que todos os aspectos sejam pensados e apropriadamente planejados (SILVA et al., 2016), e caso haja uma tentativa de se elicitar tais requisitos sem o conhecimento necessário pode-se gerar requisitos superficiais (SILVA et al., 2016).

Assim é o caso de elementos de usabilidade que, como são entendidos como requisitos não-funcionais (SILVA et al., 2017), requerem um conhecimento e experiência específicos para a área em questão. Além do entendimento base de usabilidade e de sua importância, por vezes não há um planejamento eficaz para a sua inserção no projeto, ou tal planejamento é negligenciado em favor de outros aspectos (CHOMA; ZAINA; BERHALDO, 2016).

Outro problema se mostra na abstração criada a partir da geração de requisitos de usabilidade. A escrita dada por vários exemplos de artigos e leituras sobre requisitos de usabilidade mostram que os mesmos são esquematizados de uma maneira abstrata e genérica. Um exemplo tirado do artigo de Juristo, Moreno e Sanchez-Segura (2007) que fala sobre permitir que o usuário cancele ou desfaça ações na aplicação é escrita da seguinte maneira:

"O sistema deve prover aos usuários a possibilidade de cancelar ações."

Tal requisito é capaz de dizer o que deve ser feito, mas não como deve ser feito, de maneira que a implementação não saberá quais os requisitos específicos de cada cancelamento, quais ações devem ser desfeitas etc. (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

Além do entendimento de elementos de usabilidade por parte do time responsável pela especificação, deve-se permitir também que esses requisitos já sejam inseridos durante a elicitação dos requisitos funcionais. A partir do *design* da funcionalidade, as caracterís-

ticas de usabilidade das mesmas já devem estar pensadas e planejadas, de maneira que o desenvolvedor seja capaz de adicionar tais elementos durante a implementação.

A partir dessas considerações, apresenta-se a problemática desta monografia: como colaborar com o processo de inserção de elementos de usabilidade nos requisitos elicitados pelo time de gerenciamento e desenvolvimento de um projeto.

Isso será feito a partir da evolução de um estudo anteriormente realizado por [Choma, Zaina e Beraldo \(2016\)](#). Neste estudo, os autores desenvolveram uma técnica chamada *UserX Story* a qual traz elementos de UX para o contexto da história de usuário, com técnicas como *Personas*, *Cenários*, *feedbacks* e avaliação pelas Heurísticas de Nielsen. Seus resultados mostraram que houve confusão em relação ao nível de granularidade e detalhamento dos critérios de aceitação da técnica criada.

Pretende-se, neste trabalho, corrigir esta situação, a partir da utilização de uma técnica utilizada no trabalho de [Souza, Conte e Zaina \(2020\)](#), o *framework* Os Cinco Planos ([GARRETT, 2010](#)). O *framework* foi analisado e seções que poderiam contribuir com o maior detalhamento dos critérios de aceitação, mantendo a usabilidade, foram inseridos no formato da técnica de *UserX Story*.

Sendo assim a questão de pesquisa desta monografia é:

Como adicionar elementos de usabilidade durante a elicitação de requisitos em um projeto de software a partir da evolução da técnica de histórias de usuário, *UserX Story*?

1.3 Objetivo

1.3.1 Objetivo Geral

Há o desejo de se desenvolver aplicações que sejam melhor avaliadas por usuários e ao mesmo tempo sejam capazes de resolver problemas mais facilmente. [Juristo, Moreno e Sanchez-Segura \(2007\)](#) expõem que a melhoria da usabilidade proporciona melhor eficácia e eficiência para o usuário, e [Nielsen \(2005\)](#) afirma que, mesmo em 2005, já existiam demasiado situações onde uma aplicação ou interface sofria pela falta de usabilidade, de maneira que as grandes quantidades de *websites* e interfaces de telefones celulares não estavam necessariamente suprindo a necessidade de seu cliente e usuário, pelo fato de não conseguirem realizar aplicações de usabilidade em suas linhas de produção.

Uma maneira de se realizar tal tarefa ocorre na elicitação de requisitos. O planejamento de elemento de usabilidade no começo de um projeto de software pode ser positivo de maneira a ajudar o time e os desenvolvedores a entenderem o que deve ser feito e como deve ser feito ([JURISTO; MORENO; SANCHEZ-SEGURA, 2007](#)). O começo de um projeto de software se traduz na elicitação de requisitos, funcionais e não-funcionais, sendo

os requisitos funcionais mais fáceis de serem elicitados do que requisitos não-funcionais (SILVA et al., 2017).

Como apresentado anteriormente, os requisitos não-funcionais devem ser considerados tão importantes quanto os requisitos funcionais, apesar de sua dificuldade de especificação. No caso do requisito não-funcional de usabilidade, existem estudos que tentam apresentar uma abordagem diferente para a adição e avaliação de usabilidade na elicitação de requisitos funcionais.

Dois desses estudos são os trabalhos de Choma, Zaina e Beraldo (2016) e Souza, Conte e Zaina (2020). Ambos estudos trabalharam com usabilidade em histórias de usuário, entretanto possuíam abordagens diferenciadas.

No caso do estudo de Choma, Zaina e Beraldo (2016), os mesmos propuseram uma nova abordagem para escrita de histórias de usuário. Ao adicionar elementos da técnica *Persona*, um contexto para a interação, um espaço para possível *feedback* e um critério de aceitação que utilizasse as Heurísticas de Nielsen (NIELSEN, 1993) para avaliar a presença de usabilidade na história de usuário, poderiam chegar a um resultado expressivo e facilitar a criação de USs com um *template* mais direcionado.

No caso do estudo de Souza, Conte e Zaina (2020), um foco maior foi dado à inserção de experiência de usuário em critérios de aceitação juntamente das histórias de usuário. O estudo reuniu um grupo de desenvolvedores em um *Workshop* para a criação de USs e critérios de aceitação, tentando inserir elementos de usabilidade nos mesmos e classificando-os a partir do *framework* de Garrett (2010), o *The Five Planes*. O resultado final foi que 63,3% dos critérios de aceitação não possuíam relação com usabilidade e, os que possuíam, estavam nos planos mais abstratos do *framework* de Garrett (2010). Não houve tentativa de inovação neste estudo, entretanto um *framework* para classificação de usabilidade foi utilizado, Os Cinco Planos, tal qual um entendimento do quão abstrata era o elemento de usabilidade especificada nos critérios de aceitação.

Choma, Zaina e Beraldo (2016) e Souza, Conte e Zaina (2020) apresentam abordagens pautadas na utilização da técnica histórias de usuário para entender como será a interação do usuário com o sistema, tirando a partir destas elementos e outras funcionalidades que irão abranger os requisitos e necessidades de usabilidade para que se crie o valor para o produto.

Esta monografia irá trabalhar em cima destes dois estudos e seu objetivo geral será o de evoluir uma solução relacionada à adição de usabilidade no processo de elicitação e especificação de requisitos com a técnica criada no estudo de Choma, Zaina e Beraldo (2016), a *UserX Story*, utilizando como elementos da evolução, artefatos e resultados utilizados no trabalho de Souza, Conte e Zaina (2020).

1.3.2 Objetivos Específicos

Este trabalho tem como objetivos específicos:

- Propor uma solução capaz de atingir positivamente os pontos de melhoria da pesquisa de [Choma, Zaina e Beraldo \(2016\)](#) com os resultados e técnicas usados por [Souza, Conte e Zaina \(2020\)](#).
- Aplicar os resultados com um grupo de desenvolvedores em um *Workshop* com o objetivo de:
 1. Avaliar a eficiência da solução ser utilizada pelo desenvolvedor para a elicitação de um projeto específico e;
 2. Avaliar quais foram os resultados criados e se os mesmos são capazes de melhorar a usabilidade de dito projeto, a partir da implementação destes requisitos em uma aplicação pré-existente ¹.

1.3.3 Metodologia

Com o estudo dos artigos escolhidos, será planejada e desenhada uma solução, como evolução da técnica *UserX Story* que vise trabalhar em cima dos pontos de melhoria do trabalho de [Choma, Zaina e Beraldo \(2016\)](#). A partir da junção da ideia de se utilizar o *framework* de [Garrett \(2010\)](#), utilizado por [Souza, Conte e Zaina \(2020\)](#), propôs-se ao engenheiro de requisito e de software uma nova maneira de fazer a adição de usabilidade nos requisitos.

Um grupo de desenvolvedores foi reunido em um *Workshop* e contextualizado para utilização da mesma e criação de histórias de usuário com critérios de aceitação condizentes com elementos de usabilidade para uma aplicação específica, um *website* desenvolvido pelo autor². Os requisitos foram então implementados nesta aplicação apresentada e a mesma passou por um processo de testes de aceitação e satisfação.

1.4 Organização do Documento

Este trabalho está organizado por capítulos, sendo o primeiro capítulo esta introdução, que será seguida por:

- **Capítulo 2 - Referencial Teórico:** este capítulo apresenta os conceitos básicos que serviram de base para a realização deste trabalho, informando ao leitor quais preceitos foram pesquisados;

¹ <https://aeromosquito-quiz.netlify.app/>

² <https://aeromosquito-quiz-v2.netlify.app/>

- **Capítulo 3 - Metodologia:** este capítulo especifica a metodologia escolhida para a pesquisa e desenvolvimento deste trabalho;
- **Capítulo 4 - A abordagem *UserX Story+*:** este capítulo apresenta a proposta principal do trabalho, como a pesquisa foi realizada, quais resultados serão esperados de serem encontrados.
- **Capítulo 5 - Coleta de dados:** este capítulo apresenta a maneira como os dados foram coletados durante o desenvolvimento prático do trabalho.
- **Capítulo 6 - Análise e interpretação dos dados:** este capítulo apresenta a maneira como os dados foram analisados e quais foram os resultados do trabalho.
- **Capítulo 7 - Resultados:** este capítulo apresenta os resultados do trabalho.

2 Referencial Teórico

Neste capítulo apresentam-se conceitos que fundamentam a concepção do trabalho a ser desenvolvido. Pelo fato da solução proposta ser baseada na facilitação de adição de usabilidade em histórias de usuário, serão introduzidos os conceitos básicos, extraídos de referências bibliográficas utilizadas no decorrer do trabalho, para demonstrar com o que será trabalhado e como os conceitos impactam a engenharia de *software*.

Na seção 2.1 será apresentado o conceito de usabilidade. Posteriormente será inserida uma apresentação de engenharia de requisitos e suas sub-áreas. A diferença entre requisitos não-funcionais e requisitos funcionais e onde a usabilidade se encaixa, demonstrando as dificuldades em ser especificada, seguido da apresentação de duas técnicas de elicitação e especificação de requisitos as quais serão o enfoque deste trabalho, as histórias de usuário e os critérios de aceitação.

Os conceitos aqui apresentados são importantes para situar o leitor em relação a quais áreas do campo serão trabalhados.

2.1 Usabilidade

Todo *software* serve de um propósito, seja para tarefas simples, como realizar uma conta, seja para tarefas mais complexas, como armazenar dados de processos jurídicos e permitir que um cidadão se comunique com seu advogado. O *software* é uma ferramenta que permite ao seu usuário a realização de tarefas, complexas e demoradas, simples e rápidas, de maneira que o mesmo não precise se preocupar com os menores detalhes de tal tarefa. O *software* deve ser capaz de ajudar o usuário, e não se tornar um obstáculo em seu objetivo (NIELSEN, 1993).

Para que um programa possa exercer sua tarefa eficientemente, é necessário que o mesmo possa ser usado pelo usuário de maneira eficiente e eficaz, ao mesmo tempo que procura introduzir sensações de satisfação no mesmo, com o objetivo de incentivá-lo a voltar para a plataforma num período posterior (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

Tais fatores devem ser o objetivo da inserção de elementos de usabilidade. Usabilidade é um atributo de qualidade, e um requisito não-funcional (NFR), relacionado à utilização de uma ferramenta por seus usuários. É a extensão pela qual usuários específicos utilizam um produto e obtêm satisfação nessa utilização, procurando maximizar a dita eficiência com a qual tal utilização será capaz de auxiliar o usuário em atingir seus objetivos, além de trazer mais valor ao produto para que os seus outros *stakeholders* pos-

sam se beneficiar com os futuros lucros e garantias de melhoramento (NIELSEN, 1993) (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

2.1.1 Usabilidade em produtos de *software*

No contexto de um produto de *software*, isso se entende a partir da apresentação e qualidade da informação que é dada ao usuário (JURISTO; MORENO; SANCHEZ-SEGURA, 2007), de uma maneira que seja fácil de entender e de interagir com o mesmo a partir de estratégias certas incorporadas ao seu funcionamento (JURISTO; MORENO; SANCHEZ-SEGURA, 2007) (GARCEZ et al., 2021b), ou a partir da permissão que o usuário interaja com o sistema de uma certa maneira, desfazendo uma ação feita de maneira incorreta ou mostrando de maneira destacada e fácil um botão ou maneira de interação (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

Essa melhoria de usabilidade em um *software* se demonstra demasiado importante para o usuário, pois o mesmo não gostaria de interagir com um produto que fosse complicado ou confuso de se utilizar. É importante que uma solução possa ser utilizada por várias pessoas de vários contextos diferentes, e nem todos os usuários serão capazes de entender imediatamente como um sistema funciona (JAKKAEW; HONGTHONG, 2017) (NIELSEN, 1993).

Por ser uma parte importante no ciclo de vida de um produto de *software*, a usabilidade é um elemento que deve ser considerado e planejado com cautela e estudo, devendo ser feita nas fases iniciais da construção de um produto de software, pois no caso de houver a necessidade de inserir um elemento de usabilidade em etapas mais avançadas, o re-trabalho necessário para tal causaria grande estresse e custo para a equipe responsável por tal (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

Não só focada em *design*, a aplicação de usabilidade deve se estender de maneira a entender como o usuário interage com a plataforma em questão, tentando minimizar a quantidade de erros (JAKKAEW; HONGTHONG, 2017) que podem ser causados por um usuário não treinado, além de permitir que o usuário seja capaz de realizar uma tarefa de várias maneiras diferentes, obter respostas rápidas do sistema et al (SILVA et al., 2017).

2.2 Engenharia de Requisitos (ER)

Para se realizar um projeto de *software*, vários fatores devem ser levados em consideração. Desde a concepção da ideia inicial até a implementação e *release* do sistema em si, a equipe responsável pelo programa deve ter uma base sólida sobre o que, como e quando devem ser feitas as funcionalidades e atributos pensados para tal.

Sendo uma das fases mais importantes desse processo, também uma das fases

iniciais, a Engenharia de Requisitos é o processo de elicitar, especificar e validar com os vários envolvidos no processo de software (ODEH, 2017), de maneira a se planejar e moldar inicialmente as ideias angariadas anteriormente.

Para se realizar a RE, uma série de técnicas podem ser empregadas pelo especialista, o engenheiro de requisitos (ODEH, 2017). Essas técnicas são empregadas com o objetivo de poder transmitir os requisitos para os *stakeholders* da maneira mais simples, completa e proponente de entendimento possível (ODEH, 2017). A partir do momento em que os requisitos são todos (ou parcialmente) validados e aprovados pelos *stakeholders*, podem se iniciar as implementações e construções do *software* em questão.

Na fase de elicitação de requisitos, dois aspectos devem ser considerados. O que o sistema faz, e o quão bem ele faz o que foi proposto a fazer (NURBOJATMIKO; BUDIARDJO; WIBOWO, 2018).

2.2.1 Requisitos Não-Funcionais (NFR)

Não basta para um *software* apenas fazer algo. É necessário que o mesmo seja capaz de fazer esse algo da maneira apropriada. O programa criado deve ser capaz de lidar com situações adversas, proteger os dados, ser de fácil instalação e operacionalização, e permitir que o usuário seja capaz de realizar sua tarefa por menor que seja sua experiência com tecnologias contemporâneas (SILVA et al., 2017) (NURBOJATMIKO; BUDIARDJO; WIBOWO, 2018) (JAKKAEW; HONGTHONG, 2017).

Requisitos não funcionais são requisitos atrelados à qualidade com a qual o *software* realiza uma tarefa (NURBOJATMIKO; BUDIARDJO; WIBOWO, 2018), de maneira que o usuário não se sinta atrapalhado ou inseguro com o software. Os mesmos ditam como o programa deve gerenciar suas funcionalidades e como devem ser devolvidas as respostas para o usuário, de maneira que o mesmo sinta que está sendo atendido (ECKHARDT; VOGELANG; FERNÁNDEZ, 2016).

Entretanto, a tarefa de se elicitar requisitos não-funcionais comumente se prova um desafio para o time do projeto. Pelo fato de um NFR não descrever especificamente o que um software deve fazer, suas especificações são um desafio de entendimento e escrita, causados pela sua abstração e falta de controle direto (ECKHARDT; VOGELANG; FERNÁNDEZ, 2016). Por exemplo, em seu estudo sobre quais as diferenças práticas entre um requisito não-funcional e um requisito funcional, Eckhardt (ECKHARDT; VOGELANG; FERNÁNDEZ, 2016) dá um exemplo de requisito funcional e um requisito funcional. São eles:

- FR: O usuário deve poder remover artigos de sua cesta de compras;
- NFR: O sistema deve responder em menos de 10 milissegundos a cada entrada feita.

Pode-se notar que o requisito funcional fala que o uma ação deve poder ser realizada pelo usuário, no caso de tirar itens da cesta de compras, denotando que haverá uma funcionalidade especificamente criada para tal tarefa. Enquanto isso, o requisito não-funcional denota que a resposta do sistema deve ser rápida. Essa velocidade do sistema é algo que é possível ser feito em termos de código, mas é algo que não especificamente está atrelado a uma funcionalidade específica, mas sim algo que garantirá uma melhor qualidade do funcionamento do sistema. Requisitos não-funcionais variam em nível de detalhamento e abstração (ECKHARDT; VOGELSANG; FERNÁNDEZ, 2016), e alguns, como por exemplo "A acessibilidade não deve ser menos que [x]%" (ECKHARDT; VOGELSANG; FERNÁNDEZ, 2016), dependem de mais fatores do que apenas a codificação que ocorre durante a criação do sistema (tal requisito por exemplo depende da capacidade dos servidores onde o mesmo está hospedado). "Requisitos não-funcionais são requisitos funcionais sub-especificados" (POHL, 2010 apud ECKHARDT; VOGELSANG; FERNÁNDEZ, 2016).

A usabilidade é um aspecto de software comumente escrito no formato de requisitos não-funcionais, no sentido que tratam da eficiência e eficácia com as quais o usuário deve realizar as atividades, bem como a satisfação que será gerada pós-utilização (JURISTO; MORENO; SANCHEZ-SEGURA, 2007).

2.2.2 Requisitos Funcionais (RF)

A tarefa de elicitação de requisitos deve abranger todo o escopo da identificação de dados e modelos que serão utilizados para a criação do *software*. Para que os *stakeholders* possam entender por completo o que o programa deverá fazer, quais dados serão manipulados e quais as respostas esperadas, deve-se transmitir de maneira objetiva e simplificada todos estes requisitos (ODEH, 2017). Um dos resultados dessa elicitação, e que cobre especificamente quais tarefas e funções o sistema deverá ser capaz de realizar, os requisitos funcionais são de vital importância (GLINZ, 2007). São os requisitos mais trabalhados pelo time de *software*, pelo fato de serem os requisitos com menor grau de dificuldade em se especificar, diferentemente dos requisitos não-funcionais que não atingem uma área específica do sistema (GLINZ, 2007).

Os requisitos funcionais definem num programa seus variados comportamentos, o que o *software* pode, deve, consegue fazer (GLINZ, 2007), quais bases de dados devem ser acessadas para quais tarefas, onde o sistema deve retornar um dado específico, quais condições ativam determinado comportamento etc. São o produto base da engenharia de requisitos e servem para guiar os *stakeholders*, no rumo que o projeto irá tomar e quais são os objetivos principais de cada área do sistema, e para isso devem ser de fácil entendimento tanto para o time de desenvolvedores quanto para os outros envolvidos no processo (ODEH, 2017).

2.2.3 Técnicas de elicitação de requisitos focadas neste trabalho

Existem várias técnicas que podem ser utilizadas por um time de *software* para a elicitação e especificação de requisitos de um projeto. Esta monografia irá focar em duas, especificamente. São elas:

2.2.3.1 Histórias de usuário

Histórias de usuário, ou US (*user story*), como é mais popularmente conhecido, é um artefato gerado durante a elicitação de requisitos de um projeto de software. Nas metodologias ágeis, a US é um dos artefatos gerados mais populares para comunicação e especificação de requisitos (CHOMA; ZAINA; BERALDO, 2016). A história de usuário é criada com o objetivo de se simular qual fluxo de interação que o usuário terá com a aplicação em questão, de maneira que pode-se retirar da mesma possíveis funcionalidades e como as mesmas devem ser acessadas (DRUMOND; ALVES, 2010).

Tipicamente, as histórias de usuário são escritas de acordo com o ponto de vista do usuário. Isso permite que o time de planejamento e desenvolvimento entenda quais são as funcionalidades que apresentarão maior utilidade funcional para o usuário (DRUMOND; ALVES, 2010), permitindo-os assim transformar o *backlog* do produto, artefato que junta as histórias de usuário, em um documento orientador para quais funcionalidades devem ser priorizadas de acordo com seu nível de necessidade (DRUMOND; ALVES, 2010) (DAVIES; SEDLEY, 2009). (DAVIES; SEDLEY, 2009) define que o formato de escrita de uma história de usuário é:

- "Eu como [usuário] preciso de [tarefa]". Ex.:
 - "Eu como aluno preciso enviar meu TCC para avaliação da professora orientadora."

Esse formato permite que se obtenha um melhor entendimento de quais são os objetivos do usuário ao realizar uma tarefa (DRUMOND; ALVES, 2010). A aplicação de outras técnicas de especificação de requisitos, de elementos de usabilidade (como a técnica *Persona*) também facilita o planejamento desses requisitos.

2.2.3.2 Critérios de aceitação

Critérios de aceitação, ou AC (*acceptance criteria*), são elementos listados em uma história de usuário (SOUZA; CONTE; ZAINA, 2020) ou para descrever testes de aceitação em um sistema (VEIZAGA et al., 2020). São derivados dos requisitos que foram elicitados anteriormente e servem principalmente para demonstrar para o desenvolvedor ou testador a maneira que o sistema deve funcionar e sob quais condições, mostrando pontos críticos

Tabela 1 – Tabela Exemplo de História de Usuário com Critérios de Aceitação

US:	Eu como aluno preciso enviar meu TCC para avaliação da professora orientadora
AC 1	Só deve aceitar arquivos PDF
AC 2	Apresenta um alerta verde de sucesso em caso de sucesso
AC 3	Apresenta um alerta vermelho de erro, e a mensagem deve conter qual erro específico ocorreu, em caso de erro

que devem estar funcionando para que a funcionalidade possa ser considerada como correta (SOUZA; CONTE; ZAINA, 2020) (VEIZAGA et al., 2020).

Um exemplo de critério de aceitação em uma história de usuário, utilizando a US acima como base pode ser visto na Tabela 1.

Muitas vezes esses critérios são utilizados de maneira a serem as necessidades mínimas para que uma funcionalidade possa ser aceita para ser utilizada pelo usuário. É interessante entender que os mesmos podem ser utilizados para especificar elementos de usabilidade no requisito, pelo fato de que cada funcionalidade pode ter um fluxo de utilização que necessita que o *design* e a interação sejam adaptados para o seu contexto. Os critérios de aceitação serão uma parte essencial na formulação da solução proposta neste trabalho.

2.3 Os Cinco Planos

(GARRETT, 2010) descreve em seu livro (*The Elements of User Experience: User-Centered Design for the Web and Beyond*) sobre um *framework*, chamado de *The Five Planes* (Os Cinco Planos), o qual informa que um elemento deve passar por cinco planos de entendimento. Em sua concepção, o *framework* funciona de maneira vertical, onde os planos mais abaixo em sua estrutura contemplam conceitos mais abstratos sobre determinado assunto. Quando vai-se subindo nos planos, essas abstrações e ideias vão sendo substituídas por conceitos mais concretos e formulados 2.

Essa técnica é comumente utilizada no *design* de usabilidade por permitir que elementos de UX possam ser incorporados de maneira gradual, passando por revisões e alterações. Caso um elemento seja alterado em seu estado mais abstrato, essa alteração deve ser refletida nos planos mais concretos, onde identidades visuais já são conceitualizadas. Caso uma ideia já em estado concreto indique que houve alguma falha na abstração de outra ideia, pode-se usar esse entendimento para refazer o que for necessário. O *framework* funciona bem para que alterações e revisões possam ser feitas em todos os diferentes planos, de maneira uma afete a outra.

Este *framework* será utilizado no subsequente trabalho como uma nova proposta

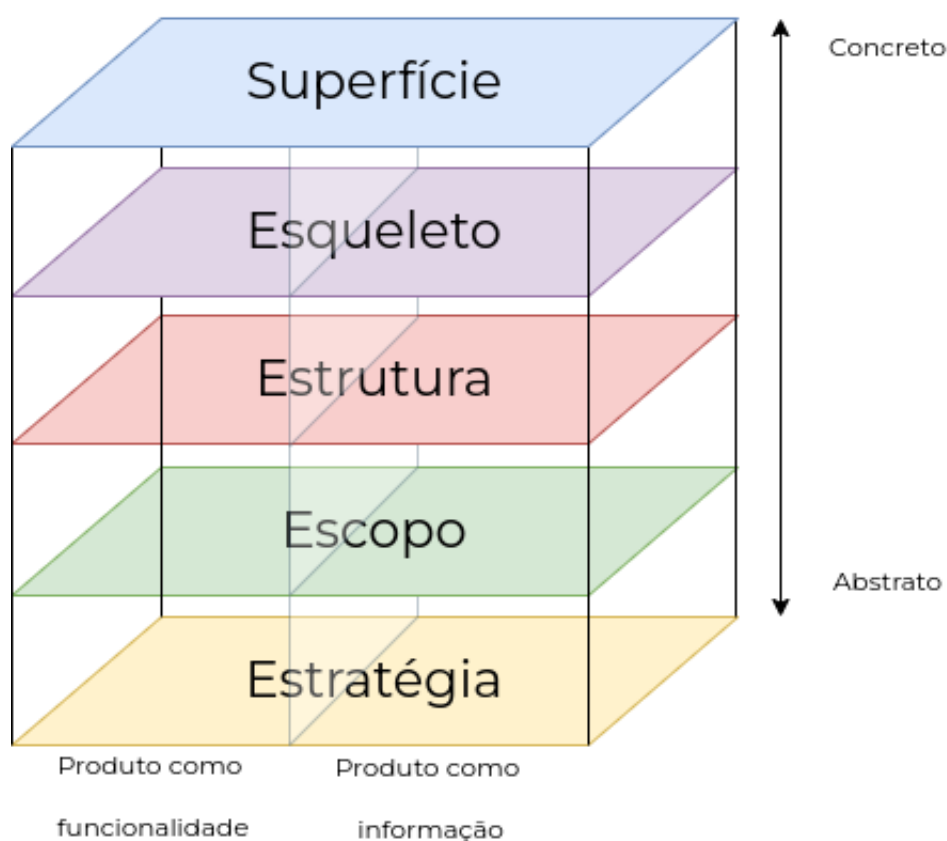


Figura 1 – *Framework* dos Cinco Planos, Adaptada de: Garrett (2010), (GARRETT, 2010)

de gramática. Os utilizadores da técnica proposta deverão formular pontos que devem ser considerados para a história a partir da especificação dos elementos da mesma com o *framework*.

2.3.1 Camada Superfície

A camada mais acima na hierarquia do *framework*, também a camada que deve representar que a solução se encontra mais concreta. A mesma apenas é formada após a especificação das outras camadas e nela são considerados elementos da experiência sensorial com o usuário, ou seja, elementos visuais de destaque, cores combinadas e que deem a sensação de boa usabilidade (GARRETT, 2010).

2.3.2 Camada Esqueleto

A camada esqueleto trata sobre a especificação de três elementos:

- **Design da informação**, que é a maneira como os dados do sistema devem ser apresentados para o usuário de maneira entendível e simples de aproveitamento

(GARRETT, 2010);

- **Design de interface**, onde o arranjo de elementos do sistema deve levar em conta as funcionalidades mais importantes e mais propensas a serem utilizadas, de maneira a criar uma interação mais direta (GARRETT, 2010);
- **Design de navegação**, que demonstra os elementos na tela que permitem ao usuário acessar as informações definidas, em qual ordem (GARRETT, 2010).

2.3.3 Camada Estrutura

A camada estrutura trata sobre a especificação de:

- **Design da interação**, onde são definidas as possíveis respostas do sistema de acordo com a interação específica do usuário (GARRETT, 2010);
- **Arquitetura da informação**, onde se define quais informações o usuário terá acesso em quais páginas, pensado em maneira de se tentar facilitar a interação do mesmo com os dados relevantes (GARRETT, 2010).

2.3.4 Camada Escopo

A camada escopo é considerada pelo autor (SOUZA; CONTE; ZAINA, 2020) uma das duas camadas mais abstratas, onde se trata dos objetivos gerais do sistema, quais funcionalidades o mesmo terá e quais dados principais serão passados para o usuário (GARRETT, 2010).

2.3.5 Camada Estratégia

A camada estratégia, a camada mais abstrata do *framework*, também é a camada que irá tratar de pensar em como resolver os objetivos do usuário, definindo assim os objetivos do produto. Essa camada serve como o início da definição do produto, pois trás à tona os planos e estratégias de mercado para que o produto resolva as necessidades de seus utilizadores (GARRETT, 2010).

2.3.6 Produto como Funcionalidade e Produto como Informação

Além de se separar o *framework* em cinco camadas horizontais, (GARRETT, 2010) também explicita que há a necessidade de se separar cada camada em mais duas camadas verticais. Essas tratam do produto como funcionalidade e como informação. São essas diferenças que trazem à tona a necessidade de alguns designs e arquiteturas serem especificados em alguns dos cinco planos .

O lado da funcionalidade considera as ações e passos envolvidos no processo de uma funcionalidade. O produto é visto como uma ferramenta para que o usuário possa completar ações. Enquanto que o lado da informação é visto pelo usuário como uma experiência onde o mesmo deve ser capaz de absorver e entender a informação que lhe é dada (GARRETT, 2010).

2.4 As Heurísticas de Nielsen

As Heurísticas de Nielsen são um conjunto de princípios que sugerem maneiras de se melhorar a usabilidade em um contexto de *software* (NIELSEN, 1993). São chamadas de heurísticas por não serem partes de um guia específico a ser utilizado para a construção de um *software*, mas sim como uma maneira de possivelmente identificar problemas de usabilidade existentes.

As heurísticas desenvolvidas por Nielsen fazem parte de um conjunto de princípios famosos por serem capazes de ajudar no desenvolvimento e UX a partir da avaliação do sistema em questão. Estas heurísticas serão utilizadas neste trabalho como uma maneira de identificar possíveis erros e acertos nos requisitos criados pelos participantes do *Workshop*.

2.4.1 Diálogo simples e natural

(NIELSEN, 1993) fala que diálogos devem ser relevantes e visíveis. Informações desnecessárias não devem ser mostradas pois a visibilidade das informações importantes acabam diminuídas.

2.4.2 Falar a língua do usuário

(NIELSEN, 1993) informa que a comunicação com o usuário deve ser simples e em palavras e frases que o usuário entenda.

2.4.3 Minimizar a carga de memória dos usuários

Fala que o usuário não deve ser obrigado a decorar grandes quantidades de informações, e que as mesmas devem ser facilmente acessáveis no sistema. (NIELSEN, 1993).

2.4.4 Consistência

Fala sobre criar e manter uma similaridade nos dados e *designs* que estão sendo criados, para que quando o usuário seja apresentado a uma nova tarefa, o mesmo saiba onde interagir e o que cada elemento significa (NIELSEN, 1993).

2.4.5 Feedback

(NIELSEN, 1993) informa que o sistema deve sempre mostrar para o usuário quais são os resultados e respostas de quaisquer ação que o mesmo tomar, de maneira que o usuário saiba o que está fazendo e o que está acontecendo no momento.

2.4.6 Saídas claras

Direcionado à possibilidade de o usuário cometer algum erro, (NIELSEN, 1993) especifica que deve haver alguma maneira do mesmo desfazer o erro ou sair facilmente da situação onde se encontra.

2.4.7 Atalhos

Maneiras de ajudar o usuário a realizar ações mais rapidamente, de maneira que o mesmo consiga acumular experiência sobre determinada tarefa e como realizá-la posteriormente (NIELSEN, 1993).

2.4.8 Boas mensagens de erro

De acordo com (NIELSEN, 1993), boas mensagens de erro devem ser fáceis de entender e informar o usuário com possíveis sugestões de correção.

2.4.9 Prever erros

(NIELSEN, 1993) também fala que a melhor situação seria o sistema prever e impedir erros de acontecerem.

2.4.10 Ajuda e documentação

O sistema pode ser demasiado complexo para certos usuários. Dessa maneira (NIELSEN, 1993) informa que é importante que uma documentação de como se usar o mesmo seja provida ao usuário.

2.5 UserX Story

UserX Story é uma variação da técnica de Histórias de Usuário, a qual propõe adicionar elementos de UX e usabilidade em sua gramática para ajudar na elicitação de requisitos a partir da técnica de Histórias de Usuário (CHOMA; ZAINA; BERALDO, 2016). A técnica foi idealizada no artigo de (CHOMA; ZAINA; BERALDO, 2016), e inclui em seu formato a adição de *Personas* para o entendimento da perspectiva do usuário,

juntamente de cenário e *feedback*, para que o autor da história possa compreender por quê o usuário está realizando tal tarefa e qual seu objetivo com a mesma.

A técnica também incluiu validação dos critérios de aceitação a partir das heurísticas de Nielsen (NIELSEN, 1993), onde deveria ser especificado qual problema a história estaria cobrindo.

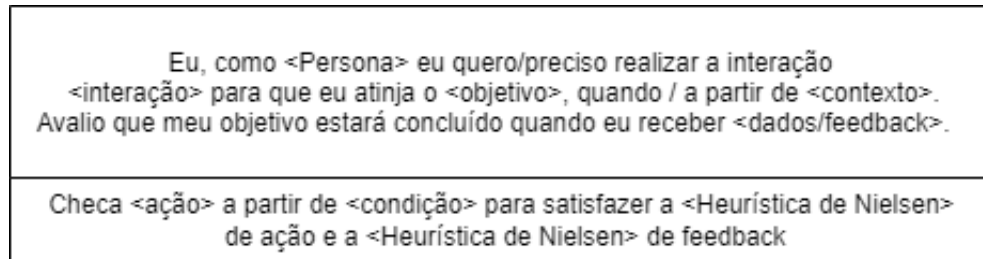


Figura 2 – *UserX Story*, Fonte: Autor, (CHOMA; ZAINA; BERALDO, 2016)

2.6 UX no desenvolvimento de *software*

Experiência de Usuário, UX, é um conceito que vem ganhando força e espaço no desenvolvimento de *software* nos últimos anos (CHOMA; ZAINA; BERALDO, 2016). O conceito em si é relacionado à melhoria de interação do usuário com o sistema, onde diferentes percepções e respostas podem impactar a visão que o mesmo tem do sistema (STANDARDIZATION, 2018).

Entretanto, UX é um conceito amplo e que abrange vários outros conceitos menores, sendo que esses conceitos menores são capazes de afetar diferentemente a maneira como um produto de *software* funciona ou é percebido (MEINERS et al., 2021), podendo envolver elementos de *design* e estilos e até segurança, facilidade de uso e sensações de confiabilidade. Essas podem ser atingidas através de várias abordagens diferenciadas.

2.7 Testes A/B

Um teste A/B é caracterizado pela testagem de duas versões de um mesmo sistema (KHARITONOV; DRUTSA; SERDYUKOV, 2017). Um versão em produção, denominada A, é comparada com uma versão experimental do mesmo sistema, denominada B. A versão A é enviada para um grupo de pessoas, grupo controle, para ser utilizado e testado, com objetivos de coletar métricas de uso, enquanto a versão B é enviada para um grupo denominado grupo de tratamento, com os mesmos objetivos.

Ao final dos testes, as métricas de interesse e aceitação são avaliadas entre as duas versões do sistema (KHARITONOV; DRUTSA; SERDYUKOV, 2017).

2.8 *Net Promoter Score*

O conceito de NPS, *Net Promoter Score*, é derivado de respostas para um tipo de pergunta que se baseia no quanto alguém recomendaria um produto, serviço ou companhia para alguém (LEE, 2018).

O cálculo de NPS é baseado em notas que derivam de 0 a 10. Dentre essas notas, existem três separações que influenciam o NPS final, sendo que notas dadas entre os valores de 0 a 6 são consideradas detratoras, que não recomendam o produto, notas 7 e 8 são notas neutras em relação à recomendação, e notas 9 e 10 são notas promotoras, que efetivamente recomendam o produto.

A partir da obtenção destas notas, um cálculo é realizado, que resulta em uma nota que varia entre -100 (cem negativo, pior nota) e 100 (cem positivo, melhor nota). NPS acima de 0 é considerado bom, acima de 50 é excelente e acima de 70 é considerado o melhor na área (LEE, 2018).

3 Metodologia

A metodologia planejada para este trabalho foi classificada quanto à sua natureza, abordagem, objetivos e aos procedimentos técnicos. Este capítulo apresenta os aspectos mencionados e como irão impactar a construção desta monografia, posteriormente demonstrando a metodologia de trabalho das duas principais etapas da monografia, juntamente da metodologia da concretização do produto-solução.

A metodologia possui as seguintes características:

- Natureza: Aplicada;
- Abordagem: Qualitativa;
- Objetivo: Descritiva;
- Procedimentos técnicos:
 1. Pesquisa Bibliográfica;
 2. Avaliação em pequena escala.

O produto final da pesquisa, uma proposta para inserção de elementos de usabilidade em artefatos de USs a partir da evolução da técnica criada no trabalho de [Choma, Zaina e Beraldo \(2016\)](#), visa suprir a necessidade de times de desenvolvimento em elicitarem tais requisitos de maneira orgânica, de acordo com uma gramática e linguagem de escrita direta e detalhista. Dessa maneira, a pesquisa possui uma natureza aplicada, visto que o produto se destina a resolver problemas encontrados na técnica da qual foi baseada.

Quanto a abordagem, a mesma se define como qualitativa. Visto que a proposta deste trabalho se baseia na aplicação de uma evolução de técnica com um grupo específico de pessoas, deseja-se obter das mesmas como a técnica pode ser aceita e utilizada para a especificação de requisitos de usabilidade, com os resultados da implementação dos requisitos gerados a partir desta evolução em aplicação prática. Foi realizado um *workshop* com desenvolvedores para a aplicação da técnica e geração de requisitos e, juntamente da agregação de dados estatísticos sobre sua aceitação do produto, o foco principal foi o de avaliar os resultados gerados pelo grupo do *workshop* no âmbito de compreender o quanto a solução foi capaz de facilitar a adição de requisitos de usabilidade a partir da quantidade de USs que continham o requisito citado. De acordo com [Moresi e Pós-Graduação \(2003\)](#), essa captação de dados define a pesquisa como qualitativa.

O objetivo da pesquisa é o de torná-la descritiva. [Moresi e Pós-Graduação \(2003\)](#) constata que uma pesquisa descritiva pode ser utilizada para definir correlações entre

variáveis e definir sua natureza. Visto que espera-se definir uma evolução de formato para criação de histórias de usuário a partir da técnica *UserX Story*, com integração dos dois modelos supracitados, definiu-se a pesquisa como descritiva.

Por fim, definem-se os procedimentos técnicos da pesquisa. Foi realizada uma pesquisa bibliográfica do assunto para que se possa entender quais referências já trataram do assunto, a mesma contribuiu para obter informações sobre a situação atual do tema e identificar opiniões e relatos similares a respeito do tema (MORESI; PÓS-GRADUAÇÃO, 2003) (SALVADOR, 1978).

No caso deste trabalho, a pesquisa bibliográfica é centrada principalmente no estudo e análise dos resultados encontrados nos artigos de Choma, Zaina e Beraldo (2016) e Souza, Conte e Zaina (2020). O objetivo será o de encontrar pontos onde o trabalho de Souza, Conte e Zaina (2020) possa complementar o trabalho de Choma, Zaina e Beraldo (2016) para criar uma evolução da solução que atenda os pontos de melhoria encontrados.

Foi utilizada também a técnica de avaliação em pequena escala. A mesma é definida como uma pesquisa feita por um único pesquisador ou um pequeno time em um curto período de tempo, onde o mesmo irá realizar a demonstração ou estudo de valor de algum empreendimento (ROBSON, 2017 apud WOHLIN; RAINER, 2022). A evolução é uma tentativa de evoluir e avaliar a veracidade de uma técnica já existente, com o auxílio de desenvolvedores e do público-alvo de determinada aplicação num⁰.

A avaliação em pequena escala foi realizada com um pequeno grupo de desenvolvedores. Esses foram escolhidos entre alunos da Universidade de Brasília, Campus Gama, Curso de Engenharia de *Software* e que já tenham passado pelas matérias de Requisitos de *Software*, Interação Humano-Computador e Engenharia de Produto de *Software*.

3.1 Detalhamento do plano metodológico

As etapas de desenvolvimento foram separadas de acordo com as entregas principais do TCC. São elas:

1. Planejamento da Pesquisa;
2. Revisão Bibliográfica;
3. Proposta de Solução.
4. Coleta de Dados;
5. Análise e Interpretação dos Dados;
6. Divulgação dos Resultados.

⁰ <https://aeromosquito-quiz.netlify.app/>

3.2 Planejamento da Pesquisa

A pesquisa teve duas ações principais, as quais serão melhor exemplificadas abaixo. O plano metodológico do TCC seguirá a seguinte estrutura, mostrada na Figura 3:

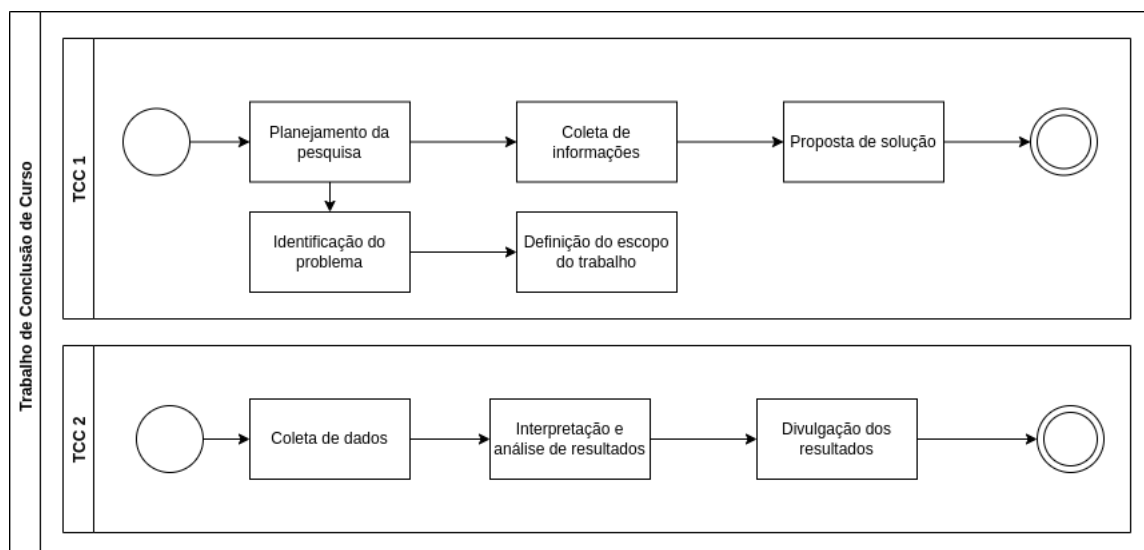


Figura 3 – TCC: Fluxos de trabalho, Fonte: autor

A primeira parte do projeto, a qual foi realizada juntamente da escrita da primeira grande entrega deste trabalho, é focado na definição das atividades e objetivos a serem realizados e buscados.

Posterior ao TCC1 foi iniciado o plano que compreenderá o escopo da segunda parte deste trabalho, o TCC2, o qual focou em se utilizar das informações coletadas para desenvolver, aplicar e validar a solução proposta no TCC1. Sendo assim, o plano metodológico do trabalho de TCC2 tem a seguinte estrutura:

- Avaliação em pequena escala a partir dos dados coletados;
- Análise e interpretação dos resultados;
- Divulgação dos resultados.

3.3 Revisão Bibliográfica

A revisão bibliográfica é um aspecto deste trabalho presente desde o seu início. A partir da revisão bibliográfica pode-se refinar a ambição que abrangerá o trabalho aqui realizado, de maneira que se pode entender quais são possíveis problemas encontrados na área de especificação de requisitos de usabilidade.

A partir da análise dos artigos anteriormente referenciados pôde-se entender que existe uma possibilidade e oportunidade de estudo na área, e assim, definiu-se um escopo com base nesse entendimento. Os dados encontrados serão melhor apresentados no capítulo onde se apresentará a proposta de solução.

Com o objetivo de se garantir as justificativas deste trabalho, em relação aos artigos escolhidos para servirem de fundamentação, foi realizada uma revisão bibliográfica da literatura existente, onde seus objetivos seriam o de encontrar, outros artigos e trabalhos que estivessem envolvidos com a área de estudo deste, requisitos de usabilidade juntamente de histórias de usuário.

A *string* de busca utilizada na base *ACM Digital Library* para encontrar tais resultados foram: *UX AND "user stories"AND usability*. Os resultados da *strings* trouxe uma variedade de artigos, os quais tratavam tanto de UX, usabilidade e a técnica de histórias de usuário em variados contextos.

Para selecionar os principais artigos lidos para este trabalho, primeiro foi realizado um filtro por título, onde os artigos que davam a entender que tratavam de histórias de usuário e interação humano-computador eram escolhidos. Posteriormente liam-se os resumos de cada artigo, tentando entender se houve uma aplicação ou estudo especificamente na área de usabilidade e histórias de usuário. Artigos que continham títulos relativos a UX e outras maneiras de eliciação de requisitos (*storytelling*, critérios de aceitação) também foram escolhidos.

Dentre os artigos que foram lidos com maior cautela, os que passaram pelo filtro mencionado acima, foram escolhidos seis leituras, por trazerem estudos relacionados usabilidade e sua correlação com, ou aplicação em, requisitos de software:

- [Choma, Zaina e Beraldo \(2016\)](#), propõem uma gramática variante da técnica de histórias de usuário, mais focada na persona e nas possíveis respostas que a funcionalidade dará. Também adicionou validação com as heurísticas de Nielsen ([NIELSEN, 1993](#)) nos critérios de aceitação, entretanto, em seus testes, teve problemas em relação a qual seria a granularidade e nível de detalhamento dos critérios de aceitação;
- [Souza, Conte e Zaina \(2020\)](#) realizaram um estudo para verificar qual seria a taxa de inserção e consideração de elementos de usabilidade na eliciação de histórias de usuário, mas com foco principal em critérios de aceitação. Utilizou o *framework* de [Garrett \(2010\)](#), Os Cinco Planos, para fazer uma classificação da existência da usabilidade e o quão concreta ela era;
- [Juristo, Moreno e Sanchez-Segura \(2007\)](#), demonstram uma metodologia de eliciação de requisitos funcionais com aspectos de usabilidade. O trabalho definia padrões

e instanciava requisitos funcionais de acordo com esses padrões de usabilidade, gerando elementos generalizados e que poderiam ser inseridos no contexto de várias atividades e funcionalidades do software;

- [Oliveira, Ferreira e Marques \(2020\)](#), propõem uma técnica baseada em personas e histórias de usuário, onde a construção da mesma passa pelas etapas de criar personas, extrair requisitos iniciais a partir de suas necessidades, escrever as histórias de usuário e enriquecê-las a partir da utilização de *Usability Elicitation Patterns*, definidos por [Juristo, Moreno e Sanchez-Segura \(2007\)](#);
- [Lopes et al. \(2017\)](#) e [Lopes et al. \(2018\)](#), trabalhos de mesmos autores, fizeram um estudo de criação de histórias de usuário por desenvolvedores, juntamente de uma explicação da área de usabilidade. Nestes trabalhos, foram analisadas quais as técnicas de entendimento de problemas mais seriam utilizadas nas histórias de usuário (e.g.: *Persona*, *Cenário*) para justificar a escrita da técnica e a existência do elemento de usabilidade;
- [Hudson \(2013\)](#), faz uma crítica a como as histórias de usuário são prescritivas, ao invés de descritivas, sobre as necessidades do cliente. O mesmo sugere um novo modelo, chamado de *Person Stories*.

Todos os artigos citados acima trazem resultados e propostas relevantes para o objetivo deste trabalho, dos quais se destacam os trabalhos de [Choma, Zaina e Beraldo \(2016\)](#) e [Souza, Conte e Zaina \(2020\)](#). O primeiro pelo fato de que faz uma padronização direta das histórias de usuário para inserir elementos de usabilidade, buscando facilitar sua escrita e entendimento. O segundo por se utilizar principalmente de critérios de aceitação para definir usabilidade, ponto que foi apontado no primeiro trabalho escolhido como possível melhoria.

O *framework The Five Planes* ([GARRETT, 2010](#)), utilizado no segundo trabalho também demonstra potencial para servir como um avaliador da concretização da usabilidade criada, onde o mesmo serve para mostrar o quão abstrato é o elemento de usabilidade.

Sendo assim, os dois servirão de base para o trabalho a ser realizado neste TCC, sendo que o foco principal será o de propor uma alteração no trabalho de [Choma, Zaina e Beraldo \(2016\)](#) a partir das técnicas utilizadas no trabalho de [Souza, Conte e Zaina \(2020\)](#)

3.3.1 Proposta de Solução

A partir das informações coletadas, escolheram-se dois artigos para serem a fundação da solução a ser proposta. A partir da análise mais profunda destes artigos resultou-se

uma ideia onde uma solução de melhoria e combinação é apresentada. A proposta de solução deste trabalho, também melhor explicada no capítulo de proposta, é a de se idealizar uma evolução de um trabalho já existentes usando resultados do outro trabalho.

3.3.1.1 Metodologia de Desenvolvimento da técnica

Para o desenvolvimento da técnica, uma série de etapas deverão ser compreendidas para que a maior eficiência seja atingida em sua mentalização.

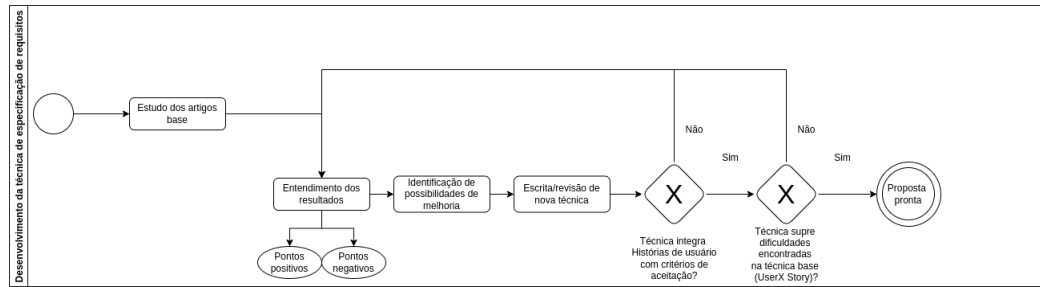


Figura 4 – Fluxo de criação da técnica, Fonte: autor

A Figura 4 demonstra em resumo as atividades que compreenderão esta fase.

- **Estudo dos artigos base:** Nesse subprocesso, foram analisados e revisados os dois artigos base (CHOMA; ZAINA; BERALDO, 2016) (SOUZA; CONTE; ZAINA, 2020), de maneira a entender como a técnica, *UserX Story*, foi criada, quais seus propósitos e quais são os pontos de melhoria. Foi feito o trabalho de se entender, também, como o trabalho de Souza, Conte e Zaina (2020) colabora a solucionar as melhorias necessárias na técnica de Choma, Zaina e Beraldo (2016);
- **Entendimento dos resultados:** Esse subprocesso avaliou quais foram os resultados de cada artigo, o que foi criado, o que foi avaliado e o que pode ser melhorado. Deve-se entender quais foram seus pontos onde melhorias podem ser aplicadas e seus pontos de sucesso, para que se possa planejar as melhorias e aproveitamentos possíveis.
- **Identificação de possibilidades de melhoria:** Nesse subprocesso, trabalhou-se em prol da criação da nova evolução da técnica, identificando onde a técnica *UserX Story* trouxe resultados positivos e como o artigo de Souza, Conte e Zaina (2020) pode melhorá-la;
- **Escrita/Revisão da nova técnica:** O subprocesso final da criação da técnica, a mesma foi escrita e esquematizada com a inclusão de histórias de usuário e critérios de aceitação. A mesma deverá também atingir diretamente os pontos de melhoria identificados na *UserX Story* e utilizar alguns dos resultados de Souza, Conte e Zaina (2020), de maneira que espera-se que um resultado positivo seja alcançado.

Com a técnica pronta para ser testada e aplicada, a mesma foi utilizada com o grupo do *workshop* para que se possa entender melhor como a mesma impacta os resultados da elicitación e especificação de requisitos que contenham elementos apropriados de usabilidade.

Diferentemente da maneira que o estudo de Souza, Conte e Zaina (2020) se utilizou do *framework* dos cinco planos, neste trabalho, foi utilizado o mesmo para que se possa direcionar o autor do requisito a especificar os elementos da funcionalidade de acordo com uma interpretação baseada nos 3 planos mais concretos do *framework*.

3.4 Coleta de Dados

Aplicação do procedimento técnico da avaliação em pequena escala. O objetivo desta aplicação é a de realizar uma intervenção no contexto do problema descrito (MORESI; PÓS-GRADUAÇÃO, 2003). No contexto deste trabalho, a avaliação em pequena escala segue um fluxo de estudo das técnicas escolhidas como base, seguido da tentativa de formulação de uma nova técnica baseadas nas duas.

Posteriormente, foi aplicada com um grupo de desenvolvedores, que usaram a técnica para gerar requisitos que serão implementados na aplicação base.

3.4.1 Metodologia de Análise da Técnica

Após a criação da técnica, iniciou-se a etapa de análise da eficiência da mesma em proporcionar uma elicitación e especificação mais rápida de elementos de usabilidade. A Figura 5 mostra o fluxo de como os dados dessa análise serão coletados e como serão avaliados.

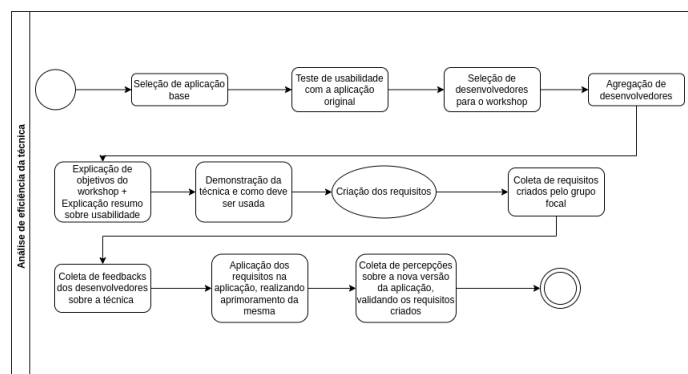


Figura 5 – Fluxo de análise da técnica, Fonte: autor

O objetivo geral dessa análise foi de juntar um grupo de desenvolvedores participantes em um *workshop* onde os mesmos se utilizaram da técnica criada para elicitar e

especificar requisitos com usabilidade em um software pré-existente num⁰, com o objetivo de melhorar sua usabilidade e experiência de usuário.

Para que se possa realizar o *workshop*, algumas tarefas ocorreram antes, durante e depois do mesmo. Estas estão descritas a seguir.

3.4.2 Análise e interpretação dos dados

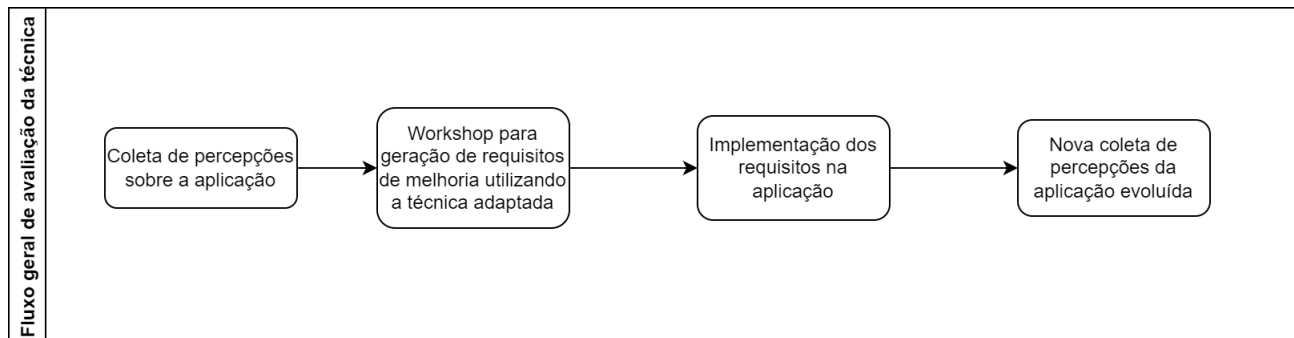


Figura 6 – Fluxo geral de avaliação da técnica adaptada, Fonte: autor

A avaliação da evolução de história de usuário foi realizada em uma série de etapas. Para verificar a validação da técnica em gerar requisitos com usabilidade positiva, uma aplicação já existente, e desenvolvida pelo autor deste trabalho num⁰, foi utilizada como objeto de experimento.

As etapas as listadas a seguir e podem ser identificadas também no diagrama de fluxo da Figura 6:

1. Coleta de percepções sobre a aplicação;
2. *Workshop* para geração de requisitos de melhoria utilizando a técnica adaptada;
3. Implementação dos requisitos na aplicação;
4. Nova coleta de percepções da aplicação evoluída.

Cada etapa será melhor exemplificada nas seções subsequentes, bem como uma breve explicação sobre a aplicação em questão.

3.4.2.1 Sobre a coleta de percepções iniciais

A aplicação existente foi inicialmente avaliada pelo seu público-alvo para responder perguntas em relação as suas percepções quanto a facilidade de uso a partir de um teste de usabilidade.

⁰ <https://aeromosquito-quiz.netlify.app/>

⁰ <https://aeromosquito-quiz.netlify.app/>

Foram coletadas percepções gerais sobre a aplicação, de maneira a permitirem a criação de novos requisitos que atendam os problemas especificados.

3.4.2.2 Sobre o *Workshop*

Com as percepções coletadas, foi planejado e realizado o *Workshop* para geração de requisitos com a técnica adaptada. Para tal, além das percepções, os participantes foram providos com o contexto da aplicação, um exemplo de funcionamento em tempo real, um resumo sobre as *Personas* que serão o público-alvo da aplicação e seus contextos de uso (jovens e chefes escoteiros), bem como o exemplo de história de usuário criada com a técnica.

O trabalho desenvolvido neste *workshop* será realizado com os participantes de acordo com as características especificadas na definição de proposta e separado nas seguintes etapas:

1. Apresentação do contexto do trabalho e objetivos do *workshop*;
2. Apresentação da aplicação que servirá como objeto de experimento, usos, objetivos e exemplos;
3. Apresentação da técnica adaptada, explicação sobre a mesma e exemplo;
4. Apresentação de dados como *Personas* e percepções coletadas sobre a aplicação;
5. Momento para que os participantes criem os requisitos de acordo com as *Personas* e as percepções;
6. Momento para tirar dúvidas e coletar *feedbacks* sobre o *workshop* e o trabalho desenvolvido.

Com o *workshop* realizado e finalizado, os requisitos criados serão coletados para a próxima etapa.

3.4.2.3 Sobre a validação da técnica

Após a realização do *workshop* e a coleta dos requisitos, o foco do trabalho foi o de aplicar os mesmos na aplicação já existente. Os requisitos, que poderão conter melhorias de funcionalidade, *design* ou novas funcionalidades, serão implementados em uma nova *branch* do repositório do projeto no *GitHub*. Isso foi feito para que uma nova implantação seja realizada de maneira a se manter duas versões, a original e a evoluída com os novos requisitos.

Posterior à implementação e implantação, novos testes de usabilidade foram realizados com os jovens chefes escoteiros e desenvolvedores, em um momentos separados

para tal. As mesmas perguntas foram feitas para todos os participantes, de maneira que tentem responder em relação à evolução das funcionalidades e sua aceitação geral.

3.4.2.4 Análise e avaliação de resultados

Com os novos resultados colhidos e catalogados, se realizou o processo de análise e avaliação. A princípio foram analisadas as respostas das percepções antes e depois da evolução para melhor entender em um contexto geral se a aplicação demonstrou melhora em sua usabilidade e facilidade de uso. Um teste A/B, com questões baseadas nas heurísticas de Nielsen, foi realizado para verificar a evolução do sistema em atingir pontos explicados nas mesmas.

Cada funcionalidade foi analisada com o olhar de procurar possíveis soluções para as situações existentes nas heurísticas, na versão original da aplicação e na versão evoluída da aplicação. Isso serve para demonstrar quais foram as evoluções da aplicação após a implementação dos requisitos criados.

Percepções sobre a técnica foram coletadas com os desenvolvedores participantes do *workshop*, de maneira a melhor entender as visões gerais sobre a mesma, sentimentos de aceitação, facilidade e eficiência de uso. Uma métrica de NPS foi coletada a partir de pergunta baseada na recomendação da técnica pelos desenvolvedores.

Caso os resultados de novas percepções e análise das heurísticas de Nielsen se mostrem positivos, pode-se entender que a técnica é capaz de colaborar com os desenvolvedores na criação e especificação de requisitos com usabilidade mais definida.

3.4.3 Divulgação dos resultados

Após a análise dos resultados, os mesmos foram estruturados e apresentados, de maneira que a efetividade da técnica possa ser apresentada.

4 A abordagem *UserX Story+*

Neste capítulo serão apresentadas as definições da proposta, com o objetivo de melhor exemplificar quais ações foram tomadas para que a solução proposta seja construída e avaliada. Serão também apresentados pontos como o cronograma prévio de atividades e suas respectivas entregas, e prazos para entrega das mesmas, sujeito a alterações, os resultados esperados e como os serão avaliados.

4.1 Definição da *UserX Story+*

Como apresentado anteriormente, o objetivo geral desta monografia é o de

Como adicionar elementos de usabilidade durante a elicitação de requisitos em um projeto de software a partir da evolução da técnica de histórias de usuário, *UserX Story+*?

Para se atingir esse objetivo foi idealizada uma evolução da técnica de especificação de requisitos, *UserX Story*, a qual se mostra como uma variação da técnica de Histórias de Usuário com foco no elemento da *Persona* (CHOMA; ZAINA; BERBALDO, 2016). Essa evolução traz como base a gramática para História de Usuário da técnica mencionada com apoio da estrutura de classificação existente no *framework* de Garrett (2010), Os Cinco Planos, o qual foi utilizado no trabalho de Souza, Conte e Zaina (2020) para avaliar os elementos de usabilidade descritos nos critérios de aceitação.

A técnica não pretende suprir em completude as necessidades e dificuldades da inserção ágil de usabilidade na elicitação e especificação de requisitos, mas sim tentar solucionar o problema de granularidade apontado no artigo de Choma, Zaina e Beraldo (2016), onde os critérios de aceitação foram planejados sem se saber qual o nível de detalhamento dos mesmos, causando confusão na avaliação e criação.

A técnica possuía critérios de aceitação que cobravam avaliações das heurísticas de Nielsen, mas há espaço para uma escrita mais detalhada e minuciosa dos critérios de aceitação. A evolução da técnica traz uma releitura do *framework* de Garrett (2010) utilizado por Souza, Conte e Zaina (2020) nos critérios de aceitação, de maneira que durante a escrita da história de usuário o autor também pensa em como será transcrita a disposição de elementos e quais informações poderão ser passadas para o usuário, bem como as interações do mesmo com os elementos da tela e quais seriam suas respostas.

Com a evolução idealizada e passada para um artefato, foi organizado um *workshop* para aplicação desta.

4.2 Sobre a técnica

A proposta original da *UserX Story*, como definido por [Choma, Zaina e Beraldo \(2016\)](#) é tal qual a figura 7:

<p>Eu, como <Persona> quero/preciso realizar a interação <interação> para que eu atinja o objetivo <objetivo>, quando / a partir de <contexto>. Avalio que meu objetivo estará concluído quando eu receber <feedback>.</p>
<p>Critérios de aceitação:</p> <p>Checa <ação> a partir de <condições> para satisfazer <Heurística de Nielsen> da ação e <heurística de Nielsen> do feedback</p>

Figura 7 – Proposta original da User Story, Fonte: [Choma, Zaina e Beraldo \(2016\)](#)

A técnica traz elementos de usabilidade para a elicitação de requisitos a partir do foco em *Personas* e seu olhar de utilização do software para o qual a *UserX Story* foi criada. Uma situação encontrada por seus autores e que dificultou sua utilização e validação foi a falta de especificação e detalhamento dos critérios de aceitação. Como reportado pelos seus testadores, não se entendeu o quão detalhado devem ser os critérios de aceitação, sendo que os usuários da técnica deram opiniões diferentes.

A evolução da técnica foi escrita de maneira a manter a gramática de *UserX Stories*, seu objetivo principal de inserir elementos de usabilidade, e muda a maneira como os critérios de aceitação são escritos na mesma, de maneira que ao invés dos mesmos serem criados de maneira a garantir que as heurísticas de Nielsen sejam atendidas, os elementos, suas posições e possibilidades de interação deverão ser individualmente especificados de acordo com a estrutura do *framework* Os Cinco planos, como mostrado na Figura 8.

Isso foi feito de maneira a definir um maior grau de detalhamento dos critérios de aceitação, para que as histórias possam ser lidas pelos desenvolvedores e os mesmos terem uma visão mais completa sobre a funcionalidade a ser descrita.

Nas seguintes seções são explicados os elementos da história de usuário e os motivos para escolha dos mesmos.

4.2.1 *UserX Story*

Como pode se ver na Figura 9, a gramática da primeira metade da história de usuário, a qual possui o objetivo principal da história, é a gramática da técnica UserX Story, pela qual seus criadores, [Choma, Zaina e Beraldo \(2016\)](#), idealizaram sua técnica.

Esta gramática foi mantida pelo motivo de permitir que o seu usuário seja capaz de fazer o *design* do requisito em questão diretamente da perspectiva de seu cliente, a

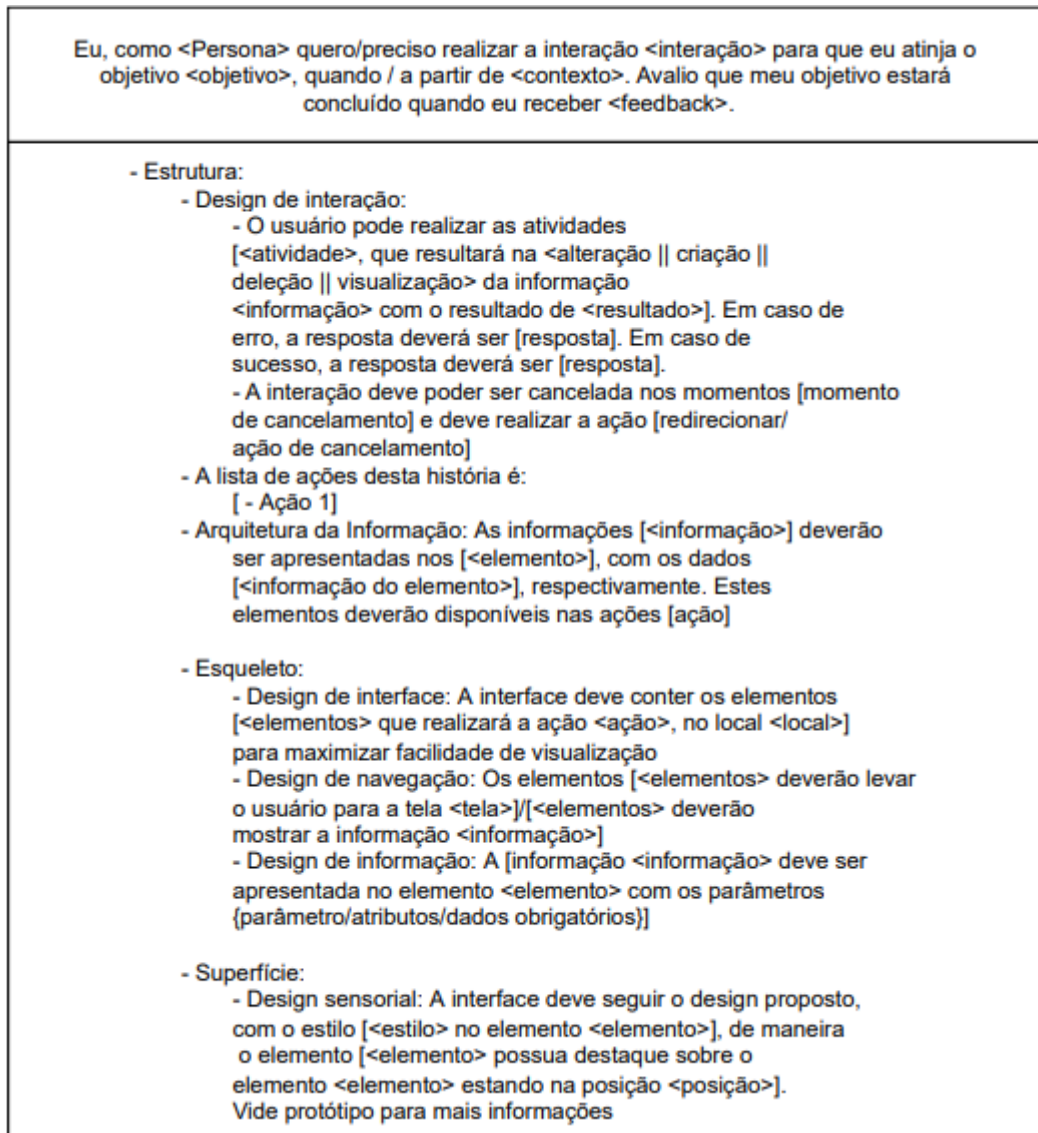


Figura 8 – Nova proposta de evolução de User Story, Fonte: autor

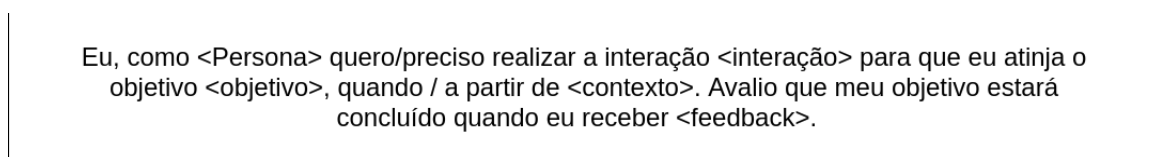


Figura 9 – Nova proposta de evolução de User Story, UserX Story, Fonte: autor

partir da técnica Persona.

Entendendo suas necessidades, o engenheiro usuário da técnica consegue imaginar com mais facilidade que tipo de interação o usuário deverá passar para que alcance seu objetivo, quais passos deverão ser seguidos e em qual situação específica tal interação acontecerá, de maneira que isso deve ser especificado no texto da história. Isso pode ajudar o engenheiro a escolher elementos mais precisos e de fácil acesso para que a ação em questão seja mais facilmente visualizada e iniciada pelo seu usuário (CHOMA; ZAINA;

BERALDO, 2016).

Também traz a necessidade de se especificar o *feedback* final da interação com a funcionalidade, para que o usuário tenha ciência do ocorrido e de qual possível resultado pode ter sido criado.

4.2.2 Os Cinco Planos

Na estrutura da técnica, onde são especificados os critérios de aceitação, assim como feito no trabalho de Souza, Conte e Zaina (2020), foram escolhidos e incluídos os elementos que compõem três dos cinco planos do *framework* em questão.

Dois desses planos não foram incluídos pelo fato de trabalharem com questões de objetivos gerais e escopo de aplicação, enquanto os outros três (os três planos que representam os aspectos mais concretos no *framework*) possuem elementos que ajudam a especificar elementos e interações específicas de funcionalidade. Sendo assim, os critérios de aceitação possuirão três camadas principais.

Os critérios de aceitação foram redigidos de acordo com o entendimento e tradução de cada um dos elementos dos planos, de maneira que a especificação dos elementos e dos requisitos obrigatórios para que a história de usuário seja aceita já incluirão em sua escrita as especificações que deverão melhorar sua usabilidade.

4.2.2.1 Critérios de aceitação da superfície

- Superfície:
 - Design sensorial: A interface deve seguir o design proposto, com o estilo [<estilo> no elemento <elemento>], de maneira que o elemento [<elemento>] possua destaque sobre o elemento <elemento> estando na posição <posição>].
Vide protótipo para mais informações

Figura 10 – Nova proposta de evolução de User Story, Camada de Superfície, Adaptado de: Garrett (2010)

A primeira camada representa a experiência final do usuário com o sistema. Como mostrado na Figura 10, o critério de aceitação exigirá de seu autor que especifique o *design* do elemento que está sendo cobrado na funcionalidade, bem como sua posição na tela. Isso para que os elementos mais importantes tenham seu destaque na tela de maneira que o usuário possa facilmente encontrá-los (GARRETT, 2010).

O estilo dos elementos também será especificado para manter a consistência de *design* de elementos na tela (GARRETT, 2010).

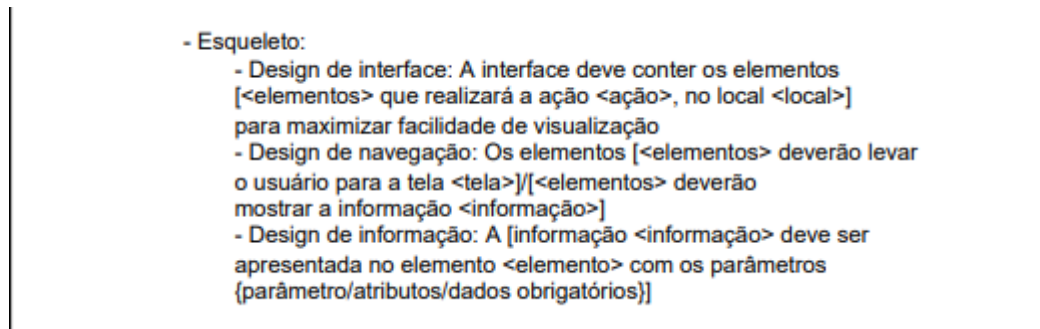


Figura 11 – Nova proposta de evolução de User Story, Camada de Esqueleto, Adaptado de: [Garrett \(2010\)](#)

4.2.2.2 Critérios de aceitação do esqueleto

A segunda camada representa a composição e escolha de elementos que irão compor a nova funcionalidade. [Garrett \(2010\)](#) menciona que três componentes devem ser levados em consideração:

- O *design* de interface define quais elementos serão inseridos na funcionalidade e sua posição de encontro e visualização para o usuário;
- O *design* de navegação serve para que sejam definidos os pontos e elementos onde a informação será mostrada e a partir de qual interação. Também pode ser usado para definir o fluxo de navegação entre telas;
- O *design* de informação define onde cada pedaço de dados e informações relevantes serão apresentados, de maneira a facilitar sua visualização por parte do usuário.

4.2.2.3 Critérios de aceitação da estrutura

A terceira camada, e mais abstrata das três camadas escolhidas para esta evolução de história de usuário, define e exige que sejam especificados quais informações o usuário deverá ter acesso na interação com a funcionalidade, e que tipo de alterações poderá realizar nelas. [Garrett \(2010\)](#) define para este plano os componentes:

- *Design* de interação, que define como o sistema irá responder de acordo com a interação do usuário, neste caso será utilizado para definir qual informação sofrerá qual alteração (criação, atualização, deleção);
- Lista de ações, que lista todas as ações que o usuário terá acesso na funcionalidade;
- Arquitetura de informação, que define qual informação será permitida a leitura pelo usuário. Isso pode conter também quais dados específicos serão disponibilizados para maximizar a eficiência.

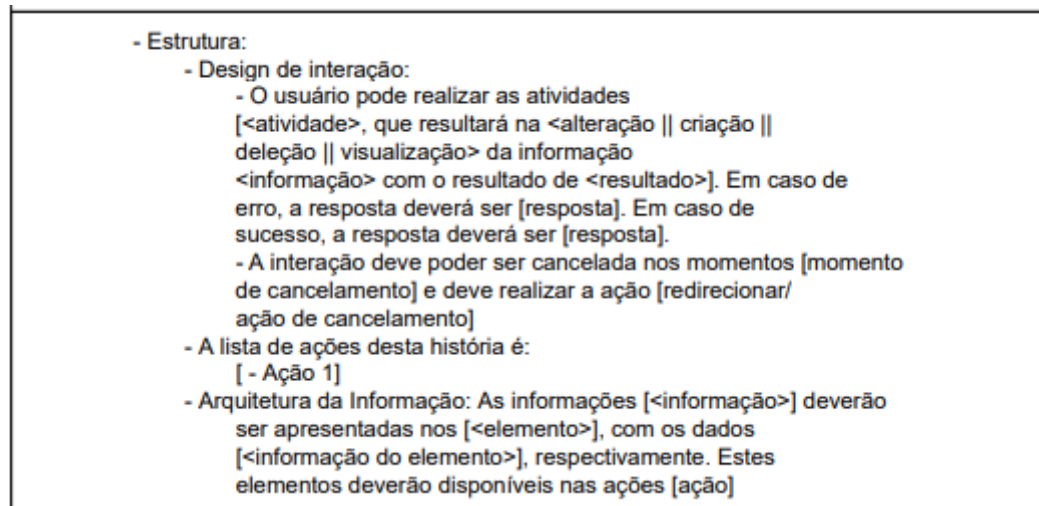


Figura 12 – Nova proposta de evolução de User Story, Camada de Estrutura, Adaptado de: [Garrett \(2010\)](#)

4.2.3 Exemplo de história de usuário criada com a técnica

Um exemplo de história de usuário foi criado utilizando a gramática da técnica, para melhor mostrar como a mesma pode ser estruturada. O exemplo se encontra na Figura 13.

<p>Eu, como escoteiro da tropa fênix quero/preciso realizar a interação selecionar o tipo de perguntas de uma partida para que eu atinja o objetivo de participar de um quiz temático, quando estiver em uma atividade voltada a tal tema. Avalio que meu objetivo estará concluído quando eu receber todas as perguntas voltadas apenas àquela temática.</p>
<ul style="list-style-type: none"> - Estrutura: <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<escolher tema>, que resultará na <alteração> da informação <perguntas da partida> com o resultado de <listar apenas perguntas do tema>]. Em caso de erro, a resposta deverá ser ['Não foi possível criar a partida com essa temática. Confira a mesma exista e tente novamente]. Em caso de sucesso, a resposta deverá ser ['Partida do tema <temática> criada com sucesso! Bom divertimento]. - A interação deve poder ser cancelada nos momentos [de escolher o tema] e deve realizar a ação [limpar a escolha de temas e voltar o sistema para a tela de escolha de temas] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Selecionar criação de partida - Selecionar o tema desejado - Clicar em criar partida - Receber feedback sobre a escolha do tema - Visualizar tema na tela da partida] - Arquitetura da Informação: As informações [lista de temas] deverá ser apresentadas nos [select de temas], com os dados [labels dos temas], respectivamente. Estes elementos deverão estar disponíveis nas ações [Selecionar o tema desejado] - Esqueleto: <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<select de temas> que realizará a ação <Selecionar o tema desejado>, no local <abaixo da label de seleção de temas e logo acima do botão de criar partidas>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<botão criar partida> deverá levar o usuário para a tela <de partida do tema>], [<select de temas> deverão mostrar a informação <lista de temas>] - Design de informação: A [informação <tema escolhido> deve ser apresentada no elemento <tela da partida> com os parâmetros {nome do tema, pergunta do turno relativa ao tema escolhido}] - Superfície: <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<botão com a cor verde escuro e fonte bold branca> no elemento <criar partida>, select verde escuro e opções em verde claro , fonte bold branca no elemento select de temas, janela com header bold branco com o nome do tema, ...demais informações da partida], de maneira que o elemento [select de temas] possua destaque sobre o elemento botão criar partidas estando na posição acima deste, o header com o nome do tema esteja acima de todos os outros elementos da partida, destacando-o dos demais]. Vide protótipo para mais informações

Figura 13 – Exemplo de história de usuário criada com a evolução da técnica, Fonte: autor

5 Coleta de dados

A fase de coleta de resultados seguiu o fluxo, com alterações esporádicas, nas tarefas insinuadas na 5. Como informado, o objetivo da coleta e da análise dos dados coletados foi o de avaliar se a *UserX Story+* seria capaz de trazer uma experiência de especificação de histórias com usabilidade mais completa do que a *UserX Story*, a partir do maior detalhamento dos critérios de aceitação a partir do *framework* os Cinco Planos. Este capítulo irá apresentar as atividades realizadas, em três etapas distintas, e a maneira como os os dados relativos ao resultado final foram coletados pelo autor. Estas três etapas foram nomeadas em torno do *Workshop* de criação de histórias de usuário com a *UserX Story+*.

5.1 Atividades pré-Workshop

Anteriormente ao *Workshop*, foram realizadas atividades de preparação para a realização do mesmo. As atividades listadas nesta seção foram realizadas com o objetivo de gerar insumos que poderiam ser utilizados pelos participantes do *Workshop* para realizar a criação e escrita das histórias de usuário.

5.1.1 Seleção de aplicação base

Foi escolhida uma aplicação para servir de base para as histórias de usuário. Esta aplicação, o Quiz Aeromoscito *Show*¹, foi desenvolvida pelo autor para o contexto de um acampamento escoteiro realizado no mês de agosto de 2022. A aplicação foi escolhida para servir de objeto de estudo para a técnica pelos motivos de ser uma aplicação pronta e testada pelo seu público-alvo, pela afinidade que o autor tem com o público-alvo, o que facilitaria seu engajamento para sua construção e refatoração. Pelo fato de ser uma aplicação criada sem uma metodologia ou planejamento, protótipo ou requisitos, mostrando potencial para evolução a partir de um planejamento melhor construído.

5.1.2 Testes de usabilidade com a aplicação original

Após a escolha da aplicação base para evolução, foi necessário entender o que da aplicação base poderia ser desenvolvido nas histórias de usuário. Para isso, procurou-se entender as maiores falhas do sistema aos olhos de seu público-alvo.

Foram reunidos 12 jovens escoteiros, na faixa etária de 11 a 17 anos, para que pudessem ser realizados testes de aceitação e usabilidade. Os jovens foram separados em

¹ <https://aeromosquito-quiz.netlify.app/>

Tabela 2 – Requisitos de novas funcionalidades

Novas funcionalidades
A plataforma deve controlar o tempo que um participante pode responder à pergunta dada
A plataforma deve possuir várias imagens de acerto ou erro em perguntas e deve apresentá-los aleatoriamente nos momentos corretos
O organizador da partida deve ser capaz de mudar a paleta de cores do sistema para aquela partida em específico, de acordo com uma seleção específica de cores
O organizador deve ser capaz de criar uma partida com base em um tema, onde as perguntas e respostas serão baseadas nesse tema
O sistema deve trazer animações no lugar de imagens estáticas para quando uma resposta for concedida, seja a mesma positiva ou negativa
O sistema deve ser capaz de permitir a criação tanto de grupos quanto de participantes individuais
O sistema deve bloquear que um grupo compita com um participante individual, apenas com outro grupo, e vice-e-versa

trios e os testes foram feitos em horários e dias diferentes. Para cada trio, o teste ocorreu de maneira a realizar um ciclo completo por todas as atividades do sistema, isto é, a criação e realização de partida de *quiz*.

A partir da realização das atividades, foram anotados a maneira como cada jovem interagia com a atividade específica e os possíveis problemas encontrados durante a realização da mesma. Ao todo 11 (onze) problemas distintos foram identificados pelo público-alvo, os quais variavam entre questões de *design* e quais informações estavam ou não disponíveis em cada etapa do ciclo de atividades, e o quanto estas eram importantes para o melhor entendimento do sistema e seus funcionamentos.

Além dos problemas, foram coletadas 7 (sete) sugestões de novas funcionalidades, relacionadas ao que os jovens gostariam de ver numa futura versão da aplicação. Os problemas e as novas funcionalidades foram escritos em formato de requisitos e encontram-se nas tabelas 2 e 3.

Nem todos os requisitos foram escolhidos para serem transformados em histórias de usuário. Alguns foram considerados muito simples ou abstratos para serem aplicados na técnica.

5.1.3 Seleção e agregação de desenvolvedores para o Workshop

Durante a realização dos testes de usabilidade, os convites para participação do *Workshop* foram enviados para 9 desenvolvedores. Estes, como explicado no capítulo de metodologia, são alunos ou recentes egressos do Curso de Engenharia de software da Universidade de Brasília, Campus Gama, e que já teriam experiência com a análise e escrita de requisitos.

Entretanto, diferentemente do que foi constatado anteriormente, nem todos os

Tabela 3 – Requisitos de problemas encontrados com os testes

Problemas
As perguntas devem aparecer em todas as telas do sistema
A aplicação deve mostrar responsividade em sua versão mobile, com letras entendíveis e botões clicáveis
O sistema deve contar com uma interface mais moderna do que a atual, com fontes mais modernas
O sistema deve contar com uma página mais personalizada e menos simples do que se vê atualmente
O sistema deve mostrar a pontuação para os participantes, bem como para o organizador, a todo momento
O sistema deve mudar a maneira como mostra as pontuações, para um design mais identificável (atualmente mostra-se um número com a quantidade de pontos)
O sistema deve possuir uma navegação com botões de redirecionamento em todas as telas possíveis
O sistema deve deixar especificado que é necessário que uma atividade externa seja realizada para que o organizador escolha quem o primeiro participante a responder
O sistema não deve possuir em seu design áreas não clicáveis em grandes tamanhos (retirar as áreas inutilizadas do design)
O sistema deve ter instruções mais claras sobre como escrever o nome do grupo/participante
O sistema deve ser capaz de identificar quando um nome já foi cadastrado em sua base de dados, e impedir a criação de outro grupo com o nome igual

desenvolvedores já haviam realizado a disciplina de Engenharia de Produto de software, mais precisamente dois deles. Foi decidido continuar com estes dois desenvolvedores nos testes pela confiança existente em suas habilidades com projetos de software e a área de requisitos.

Com todos os convites enviados e os participantes confirmados, foram marcadas 3 rodadas, de aproximadamente duas horas, do *Workshop*, de maneira que todos pudessem gerar ao menos uma história de usuário com a técnica *UserX Story+*.

5.2 Atividades durante o workshop

A realização do *Workshop* foi separada em 3 momentos.

No primeiro momento, os desenvolvedores receberam uma apresentação geral da aplicação base, com quais intuítos foi criada, o que seu público-alvo percebia sobre a mesma. Posteriormente, foi realizada uma apresentação breve sobre os objetivos do trabalho que seria realizado durante o *Workshop*, que os resultados seriam a criação de uma série de histórias de usuário no formato da técnica, a serem utilizadas para a evolução da

aplicação apresentada.

Em um segundo momento, uma breve apresentação sobre a técnica foi realizada. Nesta apresentação, mostrou-se a técnica original, a *UserX Story*, por quais motivos esta evolução estava sendo realizada e como a mesma deveria ser utilizada. Foi apresentado também um exemplo criado com a técnica sobre um dos requisitos coletados nos testes de aceitação.

No terceiro e último momento, os participantes foram liberados para criarem as histórias de usuário utilizando a técnica como base e o exemplo como apoio. Os requisitos foram separados pelos participantes, de maneira que cada um criou pelo menos uma história de usuário.

As histórias geradas se encontram no anexo (A).

5.3 Atividades pós-Workshop

Após a realização das rodadas de *Workshop*, um formulário de percepções foi passado para que os participantes pudessem responder. Esse formulário é relativo à maneira como os mesmos perceberam a utilização da técnica para a geração das histórias de usuário, casos de dificuldade e facilidade, dúvidas e entendimentos sobre seu funcionamento geral. Os resultados serão explorados em capítulos posteriores.

5.3.1 Implementação das histórias de usuário

As histórias de usuário criadas foram coletadas e listadas em um documento, disponível no anexo (A), e foram utilizadas como guia para implementação das novas funcionalidades e das correções de bugs. Nem todas as histórias de usuário foram implementadas, sendo que uma delas não foi implementada pelo fato de se encontrar muito superficial em termos de novas implementações, e outras duas por falta de tempo, sendo que essas foram deixadas por último por gerarem menor impacto na experiência do público-alvo com a aplicação.

O código gerado com esse momento de implementação foi salvo em uma *branch* separada do projeto, no *Github*, e posta ao ar em um servidor separado da versão original². Isso foi realizado de maneira a permitir que as duas aplicações pudessem ser comparadas enquanto estivessem em seu pleno funcionamento.

5.3.2 Nova coleta de *feedbacks*

Para que se pudesse avaliar a eficácia da técnica de gerar histórias de usuário mais completas a partir do detalhamento dos critérios de aceitação, foi realizado um novo teste

² <https://aeromosquito-quiz-v2.netlify.app/>

de usabilidade. Entretanto, neste momento o objetivo do teste não era o de encontrar falhas ou sugestões de melhoria, mas sim o de se comparar as duas versões da aplicação e entender qual recebeu maior aceitação em seu uso.

Para melhor comprovar a real aceitação das duas versões da aplicação, foi realizado um teste similar ao teste A/B, em que se realiza a comparação a partir do uso de duas versões distintas da aplicação. Diferentemente de testes A/B tradicionais, os grupos controle e tratamento serão compostos pelas mesmas pessoas, escoteiros, chefes e desenvolvedores de software que não tiveram contato com nenhuma das versões do sistema.

Juntamente do teste A/B, foram criadas perguntas que buscariam responder diretamente algumas das situações listadas nas Heurísticas de Nielsen (NIELSEN; MOLICH, 1990).

Para cada pergunta, o público-alvo deveria responder suas sensações ao realizar uma determinada tarefa da aplicação e responder a pergunta com base em uma escala de 1 a 10. Por exemplo, a primeira pergunta se escrevia como: 'Entendi o que precisa ser feito e como deve ser feito', em relação à funcionalidade da aplicação especificada no cabeçalho do formulário e o participante deveria responder na escala escolhida, sendo 1 relativo a não entender o que deve ser feito e como deve ser feito, e 10 a entender facilmente o que deve ser feito na funcionalidade. A média simples tirada em cada uma das funcionalidades, de cada versão, permitirá entender se a aplicação da técnica no sistema foi positiva ou não.

As perguntas foram repetidas uma determinada quantidade de vezes, para todas as possíveis atividades das duas versões da aplicação. Ao se realizar a análise dos resultados, espera-se entender se a segunda versão foi melhor recebida, o quão melhor recebida, se a nota dada às perguntas das heurísticas melhorou e se isso indica que a técnica foi capaz de atingir o que propunha.

Este teste foi realizado com um total de 11 (onze) pessoas, sendo que nem todas fazem parte do movimento escoteiro e não são o público-alvo principal da aplicação. Isso devido ao fato de que as atividades do movimento escoteiro estão em período de férias no momento da realização deste trabalho, e houve uma certa dificuldade em reunir os participantes.

Para cobrir a ausência de uma boa quantidade de participantes escoteiros, 6 desenvolvedores, que estão cursando ou se formaram no curso que o autor estuda, foram chamados para colaborarem com a coleta de *feedbacks*. Apesar de não serem escoteiros, a participação destes desenvolvedores se mostrou interessante pois trouxe um olhar mais crítico em relação às duas versões da aplicação, apontando erros e melhorias que não foram identificados pelo público-alvo principal da aplicação.

6 Análise e interpretação dos dados

Nesta seção serão apresentados os resultados e as metodologias de avaliação da técnica em dois âmbitos separados. O primeiro será relativo à aplicação da técnica no sistema base ¹, causando a geração da nova versão do sistema ². O segundo será relativo à utilização da técnica pelos desenvolvedores, no momento da criação das histórias de usuário e os *feedbacks* coletados a partir dessa utilização.

6.1 Implementação das histórias de usuário na aplicação base

Esta seção irá analisar quais foram os *feedbacks* e resultados relativos à implementação das histórias de usuário geradas nos *Workshops*.

6.1.1 Metodologia de análise

Para melhor analisar a veracidade da técnica em cumprir seu papel, de permitir a especificação de requisitos com usabilidade a partir do melhoramento da técnica original (CHOMA; ZAINA; BERALDO, 2016) a partir do detalhamento de critérios de aceitação, foi realizado com o público-alvo e com pessoas não pertencentes ao público-alvo, alunos de engenharia de software e engenheiros de software formados, um teste de usabilidade.

Diferentemente do teste de usabilidade realizado inicialmente com o público-alvo, este teste seguiu o formato de Teste A/B, onde se pretende realizar a comparação de duas plataformas ou duas versões diferentes da mesma plataforma, com fins de entender se houve evolução positiva na nova versão.

Diferentemente de testes A/B tradicionais, os grupos controle e tratamento foram compostos pelas mesmas pessoas, escoteiros, chefes e desenvolvedores de software que não tiveram contato com nenhuma das versões do sistema.

O roteiro de teste foi pensado de maneira que os participantes passassem por um ciclo completo de atividades da primeira versão da aplicação, por um novo ciclo completo de atividades na segunda versão, e posteriormente deveriam responder dois formulários, quase idênticos, sobre as duas versões.

Estes formulários foram pensados de maneira a coletar insumos sobre facilidade de uso e entendimento das tarefas da aplicação. Foram criadas 6 (seis) perguntas que deveriam ser respondidas, sendo que essas perguntas estariam correlacionadas a algumas heurísticas de Nielsen (NIELSEN; MOLICH, 1990). Isso foi feito para que se pudesse

¹ <https://aeromosquito-quiz.netlify.app/>

² <https://aeromosquito-quiz-v2.netlify.app/>

entender melhor quais notas de completude as interações teriam relativas às heurísticas e se seriam entendidas como fáceis de se utilizar pelos participantes do teste.

As perguntas definidas, e as respectivas heurísticas envolvidas, são:

- Pergunta 1: Entendi o que precisa ser feito e como deve ser feito:
 1. Diálogo simples e natural, avaliado a partir da simplicidade e da visibilidade de textos e informações relevantes;
 2. Falar a língua do usuário, avaliado a partir da simplicidade de entendimento dos comandos textos;
 3. Minimizar a carga de memória dos usuários, avaliado a partir de ajudas e posicionamento estratégico de dados que deveriam ajudar o usuário a realizar a ação;
 4. Ajuda e documentação, avaliado a partir da existência das ajudas mencionadas acima.
- Pergunta 2: Entendi qual foi o resultado da ação realizada:
 1. *Feedback*, avaliado a partir da mostra do resultado da ação que o usuário acabou de realizar.
- Pergunta 3: Consegui navegar para a página da atividade e para fora dela sem dificuldades:
 1. Consistência, parcialmente, avaliado a partir de manter a consistência de alguns elementos do sistema que ajudariam o usuário a navegar pelas telas do sistema.
- Pergunta 4: Consegui realizar a atividade de maneira rápida e simples:
 1. Atalhos, avaliado a partir da velocidade e facilidade que o sistema permitiu que o usuário realizasse uma atividade.
- Pergunta 5: Entendi qual foi o erro e como resolver tal erro:
 1. Boas mensagens de erro, avaliado a partir da eficiência da mensagem de erro de comunicar o que houve;
 2. Saídas claras, avaliado a partir da maneira como o usuário poderia facilmente resolver possíveis problemas.
- Pergunta 6: Entendi como prever um possível erro:
 1. Prever erros, avaliado a partir de como o sistema informa o usuário em como prever um erro.

Tabela 4 – Tabela de médias das perguntas, versões 1 e 2

	Médias V1	Médias V2
Pergunta 1	8,69	9,09
Pergunta 2	8,96	9,25
Pergunta 3	8,99	9,26
Pergunta 4	9,01	9,55
Pergunta 5	7,7	8,95
Pergunta 6	7,4	8,9

Tabela 5 – Tabela de notas gerais das versões 1 e 2

Nota Final V1	Nota Final V2
8,73	9,22

Para cada uma dessas perguntas, uma escala de 1 a 10 foi dada ao participante para que o mesmo pudesse responder de maneira direta suas percepções sobre a pergunta. A partir dessas respostas, foram tiradas médias das respostas de cada pergunta, uma média da pergunta em relação a todas as atividades do sistema. Essa média será usada para avaliar qual foi a evolução do sistema em cumprir as exigências das heurísticas, entre as duas versões.

Também foram tiradas notas gerais de cada uma das versões, essas notas sendo a média aritmética do sistema responder cada uma das perguntas em todas as atividades do mesmo.

6.1.2 Análise e resultados

Como demonstrado na tabela 4, as notas das médias indicam que houve uma melhora entre a versão 1 e a versão 2. Em alguns pontos a melhora foi mais discreta, como a questão de mostrar ao usuário o resultado da ação e a navegação entre páginas. Em outras questões, como a mostragem de erros ao usuário, a melhora se demonstrou mais forte, talvez inspirado pelo fato de que a técnica gerou histórias que explicitamente falavam quais seriam os erros que deveriam ser mostrados aos usuários, elementos que não existiam na primeira versão do sistema. As notas relativas às perguntas 1 e 3 mostram que o sistema já era fácil de ser utilizado, mas a implementação das história de usuário melhorou a experiência geral de seus usuários.

A nota geral de cada uma das versões 5 também demonstra uma melhora de quase 5%. O entendimento dessa diferença entre notas demonstra que as histórias de usuário geradas com a proposta de *UserX Story+* tiveram um impacto positivo na experiência geral do usuário com a aplicação. Entretanto, ao analisar os dados coletados dos formulários, notou-se que em alguns pontos específicos houve uma piora na experiência. As menores notas recebidas entre todos os formulários foram aquelas dadas no formulário relativo à

segunda versão do sistema, e principalmente em relação à navegabilidade.

Foi um ponto que não foi corretamente abordado na proposta deste trabalho, sendo que o formato da técnica demonstrava quais botões ou ações deveriam realizar algum tipo de navegação, entretanto não especificava que todos os casos e possíveis botões deveriam ter alguma interação assim, apenas definia que os elementos principais deveriam ter alguma interação.

Questões de *design* também não foram bem contextualizadas no formato da história de usuário. Apesar da tentativa de detalhar o máximo de dados possível, ao receber as histórias dos desenvolvedores participantes do *Workshop*, percebeu-se que o *design* a ser implementado mencionava apenas os elementos principais, botões que seriam adicionados e algumas informações sobre posicionamento. Entretanto, não conseguiu trazer em detalhes todos os dados necessários para que o trabalho do desenvolvedor fosse facilitado, fato que gerou uma certa liberdade artística em alguns pontos específicos da aplicação, por não estarem mencionados nas histórias.

Ainda em relação ao *design*, um dos *feedbacks* dados pelos desenvolvedores pós-*Workshop* foi o de que apesar da proposta ser interessante, ainda seria melhor ter a presença de um protótipo de alta fidelidade, e tais detalhes deveriam estar todos lá.

Durante a implementação das histórias, por parte do autor, algumas percepções sobre a utilização das mesmas, para implementar as funcionalidades, foram anotadas. As principais foram:

- A estrutura da técnica ficou muito grande, o que causava exaustão na hora de ler e entender a funcionalidade;
- Alguns pontos como os critérios de aceitação de superfície da *UserX Story+*, e seu *design* sensorial que é a área de planejamento do formato que a técnica da funcionalidade, deveriam estar mais acima na ordem de leitura, sendo que muitas vezes era o ponto mais importante e o último a ser lido;
- Em algumas histórias, pode-se observar informações repetidas ou que não eram relativas à funcionalidade. Isso pode ter sido causado pelo entendimento de seus autores que todos os campos deveriam ser preenchidos. Como alguns requisitos não necessitavam que todos os campos fossem preenchidos, pois necessitavam apenas de uma correção de design por exemplo, os autores acabaram preenchendo com informações genéricas sobre a aplicação, como seu objetivo geral;
- Ainda sobre o ponto acima, deveria estar explícito que nem todos os elementos da história devem ser preenchidos. Não foi o caso, e causou confusão durante sua escrita.

6.2 Utilização da *UserX Story+* para criação de histórias de usuário

Esta seção irá analisar quais foram os feedbacks e resultados relativos à utilização da *UserX Story+* para criação de histórias de usuário, pelos desenvolvedores, durante o *Workshop*.

6.2.1 Metodologia de análise

A metodologia para análise dos resultados da utilização da técnica foi mais simples, apenas a partir de visões durante os *Workshops* e estudo dos *feedbacks* passados pelos desenvolvedores.

Em relação a estes *feedbacks*, foi passado um formulário de coleta de informações, que possuía três perguntas principais, espaço para comentários após cada uma delas e um espaço para sugestões de melhoria. As perguntas novamente foram baseadas em escala de 1 a 10, entretanto neste caso também foram com foco em obter respostas relativas à recomendações de uso. São elas:

- Numa escala de 1 a 10, o quão fácil você identifica o uso da técnica?
- A partir da utilização e entendimento do funcionamento da *UserX Story +*, em uma escala de 1 a 10, como você avalia a capacidade da técnica de facilitar a especificação de usabilidade em histórias de usuário?
- Numa escala de 1 a 10, o quanto você recomendaria o uso da mesma num projeto de software próprio ou de terceiros?

A primeira pergunta obteve resultados positivos. A menor nota foi 6, a maior foi 9, e a média foi 7,55. Entende-se que isso significa que a técnica não foi difícil de ser utilizada, mas houveram percepções de que a mesma não estava ainda extremamente fácil de ser utilizada. Esperava-se uma nota acima de nove para que a mesma pudesse ser considerada realmente fácil.

A segunda pergunta obteve resultados parecidos, a menos nota foi 6 e a maior foi 10, e a média se manteve em 7,89. Entende-se que há uma aceitação maior em a técnica ser capaz de eliciar requisitos que facilitem a implementação de usabilidade. Mas novamente esperava-se que uma nota acima de nove, o que demonstra necessidade de mais evoluções.

A terceira pergunta obteve resultados parecidos com a segunda, média de 7,89, menor nota de 5 e maior nota de 10. Entretanto a menor nota ser a menor nota em todos os *feedbacks* mostra que a técnica pode ser considerada mas não está pronta para ser recomendada.

Os dados coletados na última pergunta também foram levados a uma calculadora de NPS, onde se obteria o *Net Promoter Score* geral da técnica, de maneira a entender-se a satisfação dos desenvolvedores com a técnica e seu uso. O NPS encontrado foi o de 10 pontos.

6.2.2 Análise

Ao se analisar a primeira pergunta, entende-se que a técnica foi reconhecida como não tão complexa, mas não foi considerada fácil de ser utilizada. Ao analisar o contexto de uso e os *feedbacks* escritos, entende-se ser pelo motivo da técnica ser muito extensa de ser escrita e alguns pontos, como os critérios de aceitação de superfície, serem abstratos em relação ao que deve ser escrito e criado para o design da funcionalidade.

A segunda pergunta mostra que a técnica ajudou a escrever pontos que devem ser levados em conta quando o usuário estiver utilizando o software, mas como alguns pontos se mantiveram abstratos mesmo com o uso da técnica e de suas especificações, a mesma deve sofrer evoluções futuras para garantir que os pontos mencionados serão trabalhados.

A última teve um NPS positivo de 10 pontos. Isso significa que de acordo com a avaliação geral de NPS, a técnica teve um resultado bom e que permitiria sua promoção em certos casos. Mas como visto nas respostas, alguns *feedbacks* informavam que apenas usariam a técnica em casos onde não houvesse uma pessoa especializada em UX.

Alguns dos principais *feedbacks* construtivos passados pelos participantes do *Workshop* foram:

- A técnica está muito grande;
- As informações escritas acabaram se repetindo muitas vezes;
- Não substitui um protótipo de alta fidelidade;
- Poderia ser utilizada junto do protótipo;
- A técnica faltou com a presença de seções melhores relativas a consistência de *design*, prevenção de erros e padrões;
- A técnica, por ser muito grande e detalhada, não seria bem recebida por organizações que não tem uma boa cultura de planejamento;
- Indicar que não precisa preencher todos os campos.

E alguns dos principais *feedbacks* positivos foram:

- A técnica ajuda a detalhar questões que ficam abstratas em histórias de usuário;

- Apesar de grande, a técnica é fácil de ser utilizada por aqueles que têm prática e costume em escrever histórias de usuário;
- O grau de detalhamento foi bem recebido;
- A técnica deixa as camadas, a serem abordadas durante um desenvolvimento de *feature*, claras.

7 Resultados

Após a análise dos dados e dos resultados obtidos, entende-se que houve duas situações distintas.

Por um lado, a técnica se mostrou positiva em melhorar a usabilidade da aplicação de uma maneira geral. Isso demonstra que a mesma cumpriu seu objetivo de procurar trazer elementos de usabilidade quando se especifica e cria histórias de usuário para o desenvolvimento de um produto de *software*.

Por outro lado, ao se utilizar a técnica para escrever histórias de usuário e, posteriormente, utilizar as histórias escritas para se desenvolver as funcionalidades criadas, percebe-se que a técnica possui falhas de *design* que impactam a experiência do escritor e do desenvolvedor, em sua utilização. Ao mesmo tempo que foi elogiada por ter um nível de detalhamento minucioso, ponto que foi objeto de *feedback* na versão original da *UserX Story* de [Choma, Zaina e Beraldo \(2016\)](#).

Isso demonstra que a técnica cumpriu o objetivo de trazer usabilidade, mas não cumpriu completamente o objetivo de ser uma alternativa eficaz e detalhística que analistas de requisitos e desenvolvedores pudessem utilizar para sempre gerarem histórias de usuário com elementos de usabilidade.

A técnica possui espaço para melhorias, principalmente relativo ao texto dos critérios de aceitação, que podem ter uma revisão do que se espera ser escrito e o que é obrigatório ou opcional, ser demasiado complexa para algumas funcionalidades mais simples ou *bugs* menores, e não trazer uma abordagem melhor sobre pontos essenciais de *design*, como manutenção de consistência e padrões de identidade.

O NPS obtido a partir das respostas dos desenvolvedores, NPS 10, mostra que a técnica ainda deve ser amadurecida para que possa ser utilizada eficientemente. Estudos futuros devem ser realizados para que a mesma seja melhor desenvolvida de acordo com os pontos apresentados pelos *feedbacks*.

No geral a técnica foi positiva, mas com espaço para muitas melhorias que amadureceriam ainda mais seu escopo e formato, além de facilitar e agilizar sua escrita. Pode-se entender que técnica não possui uma boa usabilidade para se desenvolver histórias de usuário com elementos de usabilidade, e que apesar de ter trazido esses elementos para a aplicação onde seria testada, há uma dificuldade em ser utilizada em situações corriqueiras ou casuais no planejamento de um *software*.

8 Conclusão

Neste trabalho, fez-se um estudo sobre a evolução de uma técnica de escrita de histórias de usuário, relativo aos pontos de crítica construtiva apresentados em seu artigo de origem. A partir destes pontos, uma proposta de mudança foi desenvolvida e formalizada, sendo que a mesma tem o objetivo de resolver a situação apresentada no artigo de sua versão original.

A evolução desta técnica se mostra relevante a partir da necessidade de se detalhar e concretizar conceitos de usabilidade durante o planejamento de um software. Esses elementos muitas vezes são deixados de lado ou permanecem abstratos durante o planejamento, sendo vistos com mais clareza apenas na prototipação ou na testagem de dito sistema com usuários reais.

Com a utilização da técnica, espera-se suprir esta necessidade e aproximar o planejamento de funcionalidades com a especificação de elementos de usabilidade.

Para que a comprovação de sua eficácia pudesse ser realizada, a técnica foi apresentada para um grupo de desenvolvedores, os quais deveriam utilizar a técnica para criar histórias de usuário a partir de uma lista de requisitos para evolução de uma aplicação de teste. Os desenvolvedores realizaram a escrita destas histórias e as mesmas foram posteriormente utilizadas para se implementar ditas melhorias na aplicação.

Com as funcionalidades implementadas, novos testes de usabilidade foram feitos, para se comparar a versão anterior com a versão atualizada do sistema. Isso foi feito com a utilização de um teste A/B e validação de cada funcionalidade a partir de perguntas, em um formulário, moldadas a partir das heurísticas de Nielsen.

Os resultados destes testes se mostraram positivos, pois elevaram a nota de aceitação e facilidade de uso em alguns pontos. Entretanto, os resultados gerados a partir da utilização dos desenvolvedores da técnica e os *feedbacks* dados mostram que a técnica se demonstra difícil e longa de ser usada. A mesma pode sofrer mais alterações para que atinja os pontos listados pelos desenvolvedores, como tentativa de torná-la detalhada mas sem a necessidade de preencher todos os campos, e a adição de pontos essenciais, como consistência e padrões de *design* na hora de escrever a história de usuário.

Por fim, pode-se concluir que os resultados obtidos foram positivos com ressalvas e que a evolução da *UserX Story* proposta atinge seu objetivo principal com a aplicação teste, mas gera outros problemas relativos à sua própria utilização. Novos estudos e evoluções devem ser realizados para que se desenvolva ainda mais.

Referências

ANANJEVA, A.; PERSSON, J. S.; BRUUN, A. Integrating UX work with agile development through user stories: An action research study in a small software company. *Journal of Systems and Software*, v. 170, p. 110785, dez. 2020. ISSN 0164-1212. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0164121220301953>>. Citado na página 22.

BEVAN, N. International Standards for Usability Should Be More Widely Used. *J. Usability Studies*, v. 4, n. 3, p. 106–113, maio 2009. Place: Bloomingdale, IL Publisher: Usability Professionals' Association. Citado na página 22.

CHOMA, J.; ZAINA, L. A. M.; BERALDO, D. UserX Story: Incorporating UX Aspects into User Stories Elaboration. In: KUROSU, M. (Ed.). *Human-Computer Interaction. Theory, Design, Development and Practice*. Cham: Springer International Publishing, 2016. (Lecture Notes in Computer Science), p. 131–140. ISBN 978-3-319-39510-4. Citado 19 vezes nas páginas 11, 22, 23, 24, 25, 26, 33, 38, 39, 41, 42, 44, 45, 46, 51, 52, 54, 65 e 73.

DAVIES, R.; SEDLEY, L. *Agile Coaching*. Pragmatic Bookshelf, 2009. (Pragmatic Bookshelf Series). ISBN 978-1-934356-43-2. Disponível em: <<https://books.google.com.br/books?id=1JVWPgAACAAJ>>. Citado na página 33.

DRUMOND, K.; ALVES, L. Uso de User Story Mapping Para Planejamento de Produtos Interativos. In: *Proceedings of the IX Symposium on Human Factors in Computing Systems*. Porto Alegre, BRA: Brazilian Computer Society, 2010. (IHC '10), p. 243–248. Event-place: Belo Horizonte, Minas Gerais, Brazil. Citado na página 33.

ECKHARDT, J.; VOGELSANG, A.; FERNÁNDEZ, D. M. Are "Non-Functional" Requirements Really Non-Functional? An Investigation of Non-Functional Requirements in Practice. In: *Proceedings of the 38th International Conference on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2016. (ICSE '16), p. 832–842. ISBN 978-1-4503-3900-1. Event-place: Austin, Texas. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/2884781.2884788>>. Citado 3 vezes nas páginas 21, 31 e 32.

GARCEZ, L. V. M. et al. UX e Design Inovação: evoluções metodológicas no processo de construção de apps. *Design e Tecnologia*, v. 11, n. 23, p. 116–126, dez. 2021. ISSN 2178-1974. Disponível em: <<https://www.ufrgs.br/det/>>. Citado na página 22.

GARCEZ, L. V. M. et al. UX e Design Inovação: evoluções metodológicas no processo de construção de apps. *Design e Tecnologia*, v. 11, n. 23, p. 116–126, dez. 2021. ISSN 2178-1974. Disponível em: <<https://www.ufrgs.br/det/>>. Citado na página 30.

GARRETT, J. J. *The Elements of User Experience: User-Centered Design for the Web and Beyond*. 2nd. ed. USA: New Riders Publishing, 2010. ISBN 978-0-321-68368-7. Citado 14 vezes nas páginas 11, 24, 25, 26, 34, 35, 36, 37, 44, 45, 51, 54, 55 e 56.

GLINZ, M. On Non-Functional Requirements. In: *15th IEEE International Requirements Engineering Conference (RE 2007)*. [S.l.: s.n.], 2007. p. 21–26. ISSN: 2332-6441. Citado na página 32.

HUDSON, W. User Stories Don't Help Users: Introducing Persona Stories. *Interactions*, v. 20, n. 6, p. 50–53, nov. 2013. ISSN 1072-5520. Place: New York, NY, USA Publisher: Association for Computing Machinery. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/2517668>>. Citado na página 45.

JAKKAEW, P.; HONGTHONG, T. Requirements elicitation to develop mobile application for elderly. In: *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*. [S.l.: s.n.], 2017. p. 464–467. Citado 2 vezes nas páginas 30 e 31.

JURISTO, N.; MORENO, A.; SANCHEZ-SEGURA, M.-I. Guidelines for Eliciting Usability Functionalities. *IEEE Transactions on Software Engineering*, v. 33, n. 11, p. 744–758, nov. 2007. ISSN 0098-5589. Disponível em: <<http://ieeexplore.ieee.org/document/4339231/>>. Citado 9 vezes nas páginas 21, 22, 23, 24, 29, 30, 32, 44 e 45.

KHARITONOV, E.; DRUTSA, A.; SERDYUKOV, P. Learning Sensitive Combinations of A/B Test Metrics. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. Cambridge United Kingdom: ACM, 2017. p. 651–659. ISBN 978-1-4503-4675-7. Disponível em: <<https://dl.acm.org/doi/10.1145/3018661.3018708>>. Citado na página 39.

LEE, S. Net Promoter Score: Using NPS to Measure IT Customer Support Satisfaction. In: *Proceedings of the 2018 ACM SIGUCCS Annual Conference*. Orlando Florida USA: ACM, 2018. p. 63–64. ISBN 978-1-4503-5582-7. Disponível em: <<https://dl.acm.org/doi/10.1145/3235715.3235752>>. Citado na página 40.

LOPES, L. A. et al. Using UxD Artefacts to Support the Writing of User Stories: Findings of an Empirical Study with Agile Developers. In: *Proceedings of the 19th International Conference on Agile Software Development: Companion*. New York, NY, USA: Association for Computing Machinery, 2018. (XP '18). ISBN 978-1-4503-6422-5. Event-place: Porto, Portugal. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3234152.3234158>>. Citado na página 45.

LOPES, L. A. et al. Adding Human Interaction Aspects in the Writing of User Stories: A Perspective of Software Developers. In: *Proceedings of the 31st Brazilian Symposium on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2017. (SBES'17), p. 194–203. ISBN 978-1-4503-5326-7. Event-place: Fortaleza, CE, Brazil. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3131151.3131154>>. Citado na página 45.

MEINERS, A.-L. et al. Which UX Aspects Are Important for a Software Product? Importance Ratings of UX Aspects for Software Products for Measurement with the UEQ+. In: *Mensch Und Computer 2021*. New York, NY, USA: Association for Computing Machinery, 2021. (MuC '21), p. 136–139. ISBN 978-1-4503-8645-6. Event-place: Ingolstadt, Germany. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3473856.3473997>>. Citado na página 39.

MORESI, E.; PÓS-GRADUAÇÃO, P.-R. d. UNIVERSIDADE CATÓLICA DE BRASÍLIA – UCB. p. 108, 2003. Citado 3 vezes nas páginas 41, 42 e 47.

NIELSEN, J. *Usability Engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 0-12-518405-0. Citado 7 vezes nas páginas 25, 29, 30, 37, 38, 39 e 44.

NIELSEN, J. Usability for the Masses. *J. Usability Studies*, v. 1, n. 1, p. 2–3, nov. 2005. Place: Bloomington, IL Publisher: Usability Professionals' Association. Citado 2 vezes nas páginas 22 e 24.

NIELSEN, J.; MOLICH, R. Heuristic Evaluation of User Interfaces. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 1990. (CHI '90), p. 249–256. ISBN 0-201-50932-6. Event-place: Seattle, Washington, USA. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/97243.97281>>. Citado 2 vezes nas páginas 63 e 65.

NURBOJATMIKO; BUDIARDJO, E. K.; WIBOWO, W. C. Slr on Identification & Classification of Non-Functional Requirements Attributes, and Its Representation in Functional Requirements. In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*. New York, NY, USA: Association for Computing Machinery, 2018. (CSAI '18), p. 151–157. ISBN 978-1-4503-6606-9. Disponível em: <<http://doi.org/10.1145/3297156.3297200>>. Citado na página 31.

ODEH, Y. BPMN in Engineering Software Requirements: An Introductory Brief Guide. In: *Proceedings of the 9th International Conference on Information Management and Engineering*. New York, NY, USA: Association for Computing Machinery, 2017. (ICIME 2017), p. 11–16. ISBN 978-1-4503-5337-3. Event-place: Barcelona, Spain. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3149572.3149584>>. Citado 2 vezes nas páginas 31 e 32.

OLIVEIRA, G. F. de; FERREIRA, B.; MARQUES, A. B. USARP Method: Eliciting and Describing USAbility Requirements with Personas and User Stories. In: *Proceedings of the 34th Brazilian Symposium on Software Engineering*. New York, NY, USA: Association for Computing Machinery, 2020. (SBES '20), p. 437–446. ISBN 978-1-4503-8753-8. Event-place: Natal, Brazil. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3422392.3422435>>. Citado na página 45.

POHL, K. *Requirements Engineering: Fundamentals, Principles, and Techniques*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. ISBN 3-642-12577-8. Citado na página 32.

ROBSON, C. *Small-Scale Evaluation: Principles and Practice*. Second edition. 55 City Road, London: [s.n.], 2017. Disponível em: <<https://methods.sagepub.com/book/small-scale-evaluation-2e>>. Citado na página 42.

SALVADOR, D. *MÉTODOS E TÉCNICAS DE PESQUISA BIBLIOGRÁFICA*. [S.l.]: Sulina, 1978. Citado na página 42.

SILVA, A. et al. A Process for Creating the Elicitation Guide of Non-functional Requirements. In: . [S.l.: s.n.], 2016. p. 293–302. ISBN 9783319336206. Citado na página 23.

SILVA, A. et al. Evaluation of an approach to define elicitation guides of non-functional requirements. *IET Software*, v. 11, n. 5, p. 221–228, 2017. ISSN 1751-8814. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-sen.2016.0302>>. Citado 5 vezes nas páginas 21, 23, 25, 30 e 31.

SOUZA, J. H. J.; CONTE, T. U.; ZAINA, L. A. M. Descrevendo requisitos de User eXperience em Critérios de Aceitação de User Stories. In: . São José dos Campos – SP, Brazil: [s.n.], 2020. p. 14. Citado 15 vezes nas páginas 21, 22, 24, 25, 26, 33, 34, 36, 42, 44, 45, 46, 47, 51 e 54.

STANDARDIZATION, I. the International Organization for. *ISO 9241-11:2018(en), Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*. 2018. Disponível em: <<https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>>. Citado 2 vezes nas páginas 21 e 39.

TIWARI, S.; RATHORE, S. S. *A Methodology for the Selection of Requirement Elicitation Techniques*. arXiv, 2017. ArXiv:1709.08481 [cs]. Disponível em: <<http://arxiv.org/abs/1709.08481>>. Citado na página 21.

TRAYNOR, B. UX Standards and UX Maturity. *J. Usability Studies*, v. 17, n. 2, p. 31–40, fev. 2022. Place: Bloomingdale, IL Publisher: Usability Professionals' Association. Citado na página 22.

VEIZAGA, A. et al. Leveraging Natural-Language Requirements for Deriving Better Acceptance Criteria from Models. In: *Proceedings of the 23rd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems*. New York, NY, USA: Association for Computing Machinery, 2020. (MODELS '20), p. 218–228. ISBN 978-1-4503-7019-6. Event-place: Virtual Event, Canada. Disponível em: <<https://doi-org.ez54.periodicos.capes.gov.br/10.1145/3365438.3410953>>. Citado 2 vezes nas páginas 33 e 34.

WOHLIN, C.; RAINER, A. Is it a case study?—A critical analysis and guidance. *Journal of Systems and Software*, v. 192, p. 111395, 2022. ISSN 0164-1212. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0164121222001121>>. Citado na página 42.

Anexos

ANEXO A – Histórias de Usuário criadas no *Workshop*

Este anexo lista as histórias criadas pelos desenvolvedores durante o *Workshop*, se utilizando da técnica *UserX Story+*. Algumas alterações de gramática foram realizadas para melhor leitura, e as histórias que parecem incompletas estão assim pelo fato do autor não conseguir terminá-la completamente e a tempo.

<p>Eu, como <Organizador> quero/preciso realizar a interação <selecionar temas> para que eu atinja o objetivo <escolha de tema>, quando / a partir de <criação de perguntas>. Avalio que meu objetivo estará concluído quando eu <u>receber</u> <o tema que quero utilizar naquela pergunta>.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<escolher tema>, que resultará na <alteração> da informação <tema da pergunta> com o resultado de <a perguntar ficar vinculada ao tema selecionado>, <criar um tema>, que resultará na <criação> da informação <tema> com o resultado de <tema ser criado>]. Em caso de sucesso, a resposta deverá ser [1. Pergunta atribuída ao tema <tema> com sucesso. 2. Tema criado com sucesso]. Em caso de erro, a resposta deverá ser [1. Ocorreu um erro na criação da pergunta, tente novamente mais tarde. 2. Ocorreu um erro na criação do tema, tente novamente mais tarde.]. - A interação deve poder ser cancelada nos momentos [clicar no botão cancelar/fechar] e deve realizar a ação [redirecionar para a tela de listagem de perguntas] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Logar na aplicação como organizador - Ir para a página de criação de perguntas - Clicar no botão criar perguntas - Selecionar um tema para a perguntar - Escrever a pergunta - Escrever as respostas - Clicar em salvar] - Arquitetura da Informação: As informações [<lista de temas>] deverão ser apresentadas nos [<select com os temas criados>], com os dados [<lista de temas>], respectivamente. Estes elementos deverão disponíveis nas ações [Selecionar um tema para a perguntar] - Esqueleto: <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<select de temas> que realizará a ação <selecionar tema>, no local <tela de seleção de temas>, <botão de criação de tema> que realizará a ação <criar tema>, no local <tela de seleção de tema>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<select de tema, botão de criação de tema> deverão levar o usuário para a tela <criação de perguntas>] [<select de temas> deverão mostrar a informação <lista de temas>] - Superfície: <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<botão verde escuro com o texto branco> no elemento <botão de criação de tema>, <caixa de seleção com a lista de temas espaçados> no elemento <select de temas>], de maneira o elemento [<select de tema> possua destaque sobre o elemento <botão de criação de temas> estando na posição <tela de seleção de temas>]. Vide protótipo para mais informações

Figura 14 – US1, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como <Organizador> quero/preciso realizar a interação <Visualizar informação sobre dinâmica para escolha de grupo para responder a pergunta> para que eu atinja o objetivo <escolher o grupo que vai começar respondendo>, quando / a partir de <Selecionar time para responder a pergunta>. Avalio que meu objetivo estará concluído quando eu receber <escolher o grupo que vai responder primeiro>.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<visualizar informação sobre dinâmica para escolha de grupo>, que resultará na <visualização> da informação <como escolher um grupo para responder a pergunta> com o resultado de <ter ideias de dinâmicas>] - A interação deve poder ser cancelada nos momentos [fechar instruções] e deve realizar a ação [redirecionar a tela de seleção de grupo para responder a pergunta] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Logar na aplicação - Criar sala - Clicar no botão de informações sobre escolha de grupo] - Arquitetura da Informação: As informações [<informação sobre escolha de grupo>] deverão ser apresentadas nos [<modal>], com os dados [<informação sobre escolha do grupo>], respectivamente. Estes elementos deverão disponíveis nas ações [tela de seleção de grupos para responder a pergunta] <p>- Esqueleto:</p> <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<botão com interrogação> que realizará a ação <mostrar informação sobre escolha do grupo>, no local <ao lado do texto “Escolha o grupo que poderá responder à pergunta”>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<botão com interrogação> deverão levar o usuário para a tela <modal com informação sobre a escolha de grupo>[/<modal com informação sobre a escolha de grupo> deverão mostrar a informação <informação sobre a escolha de grupo>] <p>- Superfície:</p> <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<botão com ícone de interrogação branco com o fundo verde> no elemento <elemento>], de maneira o elemento [<botão com interrogação > possua destaque sobre o elemento <texto “Escolha o grupo que poderá responder à pergunta” > estando na posição <ao lado do texto>]. Vide protótipo para mais informações

Figura 15 – US2, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como participante (Fernanda/Pedro), quero realizar a interação de receber instruções claras para que eu atinja o objetivo de escrever o nome do grupo ou participante, quando for iniciar um novo jogo. Avalio que meu objetivo estará concluído quando eu receber a tela da primeira pergunta do jogo.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar a atividade de [<acessar um ícone de informação>, que resultará na <visualização> da informação <limitações quanto ao nome do grupo/participante> com o resultado de <permitir a inscrição correta e o seguimento do jogo>]. Em caso de sucesso, a resposta deverá ser [avançar para a próxima tela (primeira pergunta do jogo)]. - A lista de ações desta história é: <ul style="list-style-type: none"> [- Iniciar nova partida como campista] [- Acessar ícone de informação] [- Ler informação disposta] [- Escrever nome do grupo/participante] [- Inserir código da partida] [- Acessar botão de conectar com partida] [- Avançar para próxima tela] - Arquitetura da Informação: As informações [<limitações quanto ao nome do grupo/participante>] deverão ser apresentadas nas [<caixas de informação>], respectivamente. Estes elementos deverão estar disponíveis nas ações [<acessar um ícone de informação>] - Esqueleto: <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<ícone de informação> que realizará a ação <mostrar mensagem>, no local <de sobreposição do mouse>] para maximizar facilidade de visualização - Superfície: <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<ícone na cor preta e fundo transparente> no elemento <caixa verde escuro arredondada>], de maneira que o elemento [<ícone>] possua destaque sobre o elemento <caixa> estando na posição <à direita do título do campo>]. Vide protótipo para mais informações

Figura 16 – US3, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como participante (Fernanda/Pedro) preciso realizar a interação de cadastrar um nome único para mim ou meu grupo para que eu atinja o objetivo de participar de um novo jogo, quando na tela de início de jogo. Avalio que meu objetivo estará concluído quando eu receber a tela da primeira pergunta do jogo.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<inserir um nome de participante ou grupo>, que resultará na <verificação> da informação <nome inserido> com o resultado de <validação de nome único>]. Em caso de erro, a resposta deverá ser [uma mensagem avisando ao usuário que o nome já existe]. Em caso de sucesso, a resposta deverá ser [avançar para a próxima tela]. - A interação deve poder ser cancelada nos momentos [antes de clicar em "Conectar com partida"] e deve realizar a ação [de se manter na página, sem iniciar um novo jogo]

Figura 17 – US4, Fonte: desenvolvedores participantes do *Workshop*

Eu, como **Organizador**, quero/preciso realizar a interação **Alterar imagem de acerto e erro** de para que eu atinja o objetivo **Customizar o questionário**, quando / a partir de eu estiver aplicando o questionário em **algum evento temático**. Avalio que meu objetivo estará concluído quando **as imagens forem alteradas e todos consigam visualizar elas**

Design de Interação:

- O usuário pode realizar a atividade de **alteração de imagens de acerto e erro**, que resultará na **modificação das imagens** com o resultado de **customizar o questionário**.
- Em caso de erro ao fazer a modificação, a resposta deverá ser **NÃO FOI POSSÍVEL ALTERAR IMAGENS PADRÃO DE ACERTO E ERRO DE QUESTÕES**.
- Em caso de acerto, as imagens devem ser modificadas.
- A interação deverá ser cancelada no momento em que o **usuário escolher utilizar as imagens padrões do sistema**, ou **cancelar a criação de uma nova sala** e deve realizar a ação **Voltar para a tela de Criação de Salas / Perguntas no Menu do Organizador**.

Lista de ações:

- Ação 1: O usuário deverá acessar o Quiz, como Organizador.
- Ação 2: O usuário deverá clicar em Criar Sala
- Ação 3: Deverá ser exibido uma tela de opção para configuração da sala, antes de ser exibido às perguntas, para que ele consiga alterar a imagem de Acerto e Erro.
- Ação 4: O usuário, querendo alterar as imagens, deve ser redirecionado para um tela que permita o Upload de imagens para Acerto e para Erro.
- Ação 5: O usuário deverá fazer o upload das imagens desejadas.

Arquitetura da Informação:

- As informações **Imagens de Acerto e Erro**, deverão ser apresentadas na **tela após a resposta de uma pergunta**, com o **feedback para o usuário sobre a resolução da pergunta**. Este elemento deverá estar disponível nas ações de **Respostas dos Jogadores**.

Esqueleto:

- Design de Interface: A interface deverá conter **uma tela adicional** que realizará a ação de **Personalização das Salas**, no local **Após o usuário clicar em "Criar sala"** e **ao centro da tela** para maximizar a facilidade de visualização.
- Design de Navegação: Os elementos **tela de personalização** devem redirecionar o para a **tela de upload de imagens** onde o usuário poderá efetuar o upload de novas imagens de acerto e erro.
- Design de Informação: a **tela de upload de imagens** deverá mostrar ao usuário quais imagens foram feitas o upload, e para qual categoria **Acerto** ou **Erro** o upload de imagens está sendo feito.

Superfície:

- Design sensorial: A interface deverá seguir o design proposto, com o **estilo de elementos verdes conforme as outras delas**, a **tela de personalização** deverá ter um texto **"Você quer alterar as imagens que irão aparecer para os jogadores em caso de erro ou acerto das perguntas?"**. Seguindo por 2 botões **"Alterar imagens"** e **"Pular"**. Os elementos deverão estar alinhados verticalmente, o texto seguido de 2 botões.

Figura 18 – US5, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como Jogador, quero/preciso realizar a interação Visualizar a pergunta que está sendo exibida para o Organizador de para que eu atinja o objetivo Ver a pergunta, quando / a partir de eu estiver respondendo um questionário. Avalio que meu objetivo estará concluído quando conseguir ver a mesma pergunta que estiver sendo exibida para o Organizador.</p>
<p>Design de Interação:</p> <ul style="list-style-type: none">- O usuário pode realizar a atividade de visualizar as perguntas, que resultará na visualização de qual pergunta está sendo feita no momento.- Em caso de erro ao fazer a exibição das perguntas, um feedback indicando que não foi possível carregar a pergunta deverá ser exibido ao usuário.- Em caso de acerto, a pergunta deverá ser mostrada.- A interação deverá ser cancelada no momento em que o usuário sair do questionário, ou sair do aplicativo e deve realizar a ação voltar à tela inicial do aplicativo.
<p>Lista de ações:</p> <ul style="list-style-type: none">- Ação 1: O usuário deverá acessar o Quiz, como Campista.- Ação 2: O usuário deverá preencher o nome do grupo e o código da sala.- Ação 3: Deverá ser exibido a pergunta atual que estiver sendo feita pelo Organizador, independente se o grupo escolhido para resposta for o seu ou não.
<p>Arquitetura da Informação:</p> <ul style="list-style-type: none">- As informações Perguntas, deverão ser apresentadas na tela do questionário do campista, abaixo do título Questão X/10. Este elemento deverá estar disponível nas ações de Exibir questionário ao Campista.
<p>Esqueleto:</p> <ul style="list-style-type: none">- Design de Interface: A interface deverá conter um texto adicional que realizará a ação de Exibição de Pergunta, no local Questionário sendo mostrado para o Campista para maximizar a facilidade de visualização.- Design de Navegação: ---- Design de Informação: o texto da pergunta deverá mostrar ao usuário a pergunta sendo feita pelo organizador para todos os participantes do Quiz na tela de "Questão" independente se está respondendo ou não a pergunta..
<p>Superfície:</p> <ul style="list-style-type: none">- Design sensorial: A interface deverá seguir o design proposto, na tela já existente de Questão e também na tela de Alternativas para resposta. Como forma de texto seguindo o padrão de tamanho e fonte do aplicativo, para que seja de fácil visualização e entendimento do público

Figura 19 – US6, Fonte: desenvolvedores participantes do *Workshop*

Eu, como Escoteiro quero realizar a interação de **criar grupos ou partidas individuais** para que eu atinja o objetivo de **jogar**, quando **criar uma partida**. Avalio que meu objetivo estará concluído quando **a partida começar**.

- Estrutura:
 - Design de interação:
 - O usuário pode realizar as atividades **jogar como jogador individual ou criar/entrar em um grupo**, que resultará no **início da partida** com o resultado de **informar o tipo de partida**. Em caso de erro, a resposta deverá ser **de que não foi possível entrar na partida**. Em caso de sucesso, a resposta deverá ser **entrando na partida**.
 - A interação deve poder ser cancelada nos momentos **antes do início e durante a partida** e deve realizar a **ação saindo do jogo**.
 - A lista de ações desta história é:
 - **Escolher como participar do jogo (individual ou em grupo)**
 - **Em caso de partidas em grupo:**
 - **Criar um grupo ou entrar em um grupo existente**
 - Arquitetura da Informação: A informação **tipo de partida (individual ou grupo)** deverá ser apresentada na **tela de criação de partida ou na tela de entrar em uma partida**, respectivamente. Estes elementos deverão estar disponíveis nas ações **criar ou entrar em uma partida**.
- Esqueleto:
 - Design de interface:
 - A interface deve conter o elemento **botão de jogar como participante individual**, que realizará a **ação de entrar na partida**, no local de **participar de um jogo** para maximizar facilidade de visualização
 - A interface deve conter o elemento **botão criar um grupo**, no local de **participar de um jogo**
 - A interface deve conter o elemento **botão de entrar em um grupo existente**, no local de **participar de um jogo**
 - A interface deve conter o elemento **caixa de texto** no local de **criar um grupo**
 - A interface deve conter o elemento **caixa de texto** no local de **entrar em um grupo existente**
 - A interface deve conter o elemento **botão de entrar na partida** nos locais **criar um grupo, entrar em um grupo existente**
 - Design de navegação:
 - O elemento **botão de jogar como participante individual** deverá levar o usuário para a **tela início de partida**
 - O elemento **botão de criar um grupo** deverá levar o usuário para a **tela criar um grupo**
 - O elemento **botão de entrar em grupo existente** deverá levar o usuário para a **tela encontrar um grupo**
 - O elemento **botão de entrar na partida** deverá levar o usuário para a **tela início de partida**
 - Design de informação: A **nome do grupo** deve ser apresentada no elemento **caixa de texto** com os parâmetros **nome do grupo**
- Superfície:
 - Design sensorial: A interface deve seguir o design proposto:
 - com o estilo **botão** com a cor **verde escuro** e fonte **bold branca** nos elementos **botão de jogar como participante individual, botão criar um grupo, botão de entrar em um grupo existente, botão de entrar na partida**
 - com o estilo **sem bordas, preenchimento verde claro, linha escura** na parte inferior e fonte **bold branca** no elemento **caixa de texto** de maneira

Vide protótipo para mais informações

Figura 20 – US7, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como sistema preciso realizar a interação de bloquear que um grupo compita com um participante individual e vice-versa para que eu atinja o objetivo de não misturar os participantes, quando os participantes forem entrar na plataforma. Avalio que meu objetivo estará concluído quando participantes diferentes não compitam entre si.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<competir com participantes individuais ou se ele for de um grupo, competir com outros grupos>, que resultará na <visualização> da informação <perguntas da partida> com o resultado de <grupo ou individual>]. Em caso de erro, a resposta deverá ser [não é possível competir com um grupo sendo participante individual e vice-versa]. Em caso de sucesso, a resposta deverá ser [sucesso ao entrar na partida]. - A interação deve poder ser cancelada nos momentos [de escolher o tipo de partida (individual ou em grupo)] e deve realizar a ação [mostrar mensagem de erro] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Escolher como participar do jogo (individual ou em grupo) - Em caso de partidas em grupo (Criar um grupo ou entrar em um grupo existente) - Organizador ser bloqueado de colocar grupo para competir com participante individual] - Arquitetura da Informação: As informações [tipo de partida] deverão ser apresentadas nos [select de participação], com os dados [individual ou grupo], respectivamente. Estes elementos deverão estar disponíveis nas ações [Selecionar a participação desejada] <p>- Esqueleto:</p> <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<select de participação> que realizará a ação <escolher entre (grupo ou individual)>, no local <após a seleção de campista>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<botão de selecionar se é grupo ou individual> deverão levar o usuário para a tela <de jogar a partida>] deverão mostrar a informação <da partida>] - Design de informação: A [informação <nome do grupo ou nome individual> deve ser apresentada no elemento <caixa de texto> com os parâmetros {dados obrigatórios}] <p>- Superfície:</p> <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto: com o estilo botão com a cor verde escuro e fonte bold branca nos elementos botão de jogar como participante individual, botão criar um grupo, botão de entrar em um grupo existente, botão de entrar na partida com o estilo sem bordas, preenchimento verde claro, linha escura na parte inferior e font bold branca no elemento caixa de texto de maneira <p>Vide protótipo para mais informações</p>

Figura 21 – US8, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como Escoteiro Sênior, quero criar uma partida para que os demais escoteiros possam participar, no quiz da aplicação. Avalio que meu objetivo estará concluído quando eu receber um código de partida.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades de criar uma partida. Em caso de erro, a resposta deverá ser "Houve um erro no sistema ao tentar criar a partida, tente novamente mais tarde.". Em caso de sucesso, a resposta deverá ser exibir a primeira pergunta da partida e o código da partida. - A interação deve poder ser cancelada nos momentos em que o organizador clica no botão sair e deve realizar a ação retornar para a tela inicial. - A lista de ações desta história é: <ul style="list-style-type: none"> - Selecionar que o usuário é um organizador - Logar com o google - Criar partida - Arquitetura da Informação: O código da partida deverá ser apresentado nas telas de perguntas, com o valor do código da partida, respectivamente. Este elemento deverá estar disponível na ação "Criar partida". <p>- Esqueleto:</p> <ul style="list-style-type: none"> - Design de interface: - Design de navegação: Os elementos <ul style="list-style-type: none"> [Botão de Criação de perguntas que irá redirecionar o usuário para uma tela de gerenciamento de perguntas] [Botão de Desconectar que irá desconectar o usuário e redirecionar para a tela inicial] [Botão de Criação da partida que irá redirecionar o usuário para uma tela que possua o código da partida e número de usuários ativos] - Design de informação: O código da partida deve ser apresentado em uma nova página após clicar no Botão "Criar Partida", o código deve ser único e possuir 6 caracteres aleatórios. <p>- A informação "conectar em uma outra partida" deve ser apresentada no elemento botão "conectar a uma partida existente" com os parâmetros um campo de input para inserção do código e uma mensagem clara solicitando a inserção do código.</p> <p>- Superfície:</p> <ul style="list-style-type: none"> - Design sensorial: <ul style="list-style-type: none"> - A interface deve seguir o design proposto, com destaque no elemento botão de criar partida, de maneira o elemento botão de criar partida possua destaque sobre os outros elementos da tela estando na posição mais baixa da tela. Vide protótipo para mais informações - A interface deve seguir o design proposto, com o estilo de vermelho no elemento botão de desconectar, de maneira o elemento botão de desconectar possua destaque sobre os demais elementos da tela estando na posição ao lado do botão de criar partida. Vide protótipo para mais informações

Figura 22 – US9, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como Escoteiro Sênior quero/preciso realizar a interação de entrar em uma partida para que eu atinja o objetivo de participar de uma partida, quando / a partir da tela inicial da aplicação. Avalio que meu objetivo estará concluído quando eu puder responder as perguntas do quiz.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades de se conectar a uma partida, que resultará no redirecionamento do usuário para a tela de partida. Em caso de erro, a resposta deverá ser "Código <u>inexistente</u>. <u>Insira um código válido</u>". Em caso de sucesso, a resposta deverá ser o redirecionamento para a tela de partida. - A interação deve poder ser cancelada na tela de conectar com partida e deve realizar a ação de redirecionar para a tela inicial. - A lista de ações desta história é: <ul style="list-style-type: none"> - Selecionar que é campista - Inserir nome do grupo - Inserir código da partida - Conectar com a partida - Arquitetura da Informação: A informação "nome do grupo" deverá ser apresentada na tela da partida, com os dados "nome do grupo", respectivamente. Estes elementos deverão estar disponíveis nas ações de conectar com a partida <p>- Esqueleto:</p> <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos <ul style="list-style-type: none"> [Input de "Nome do Grupo" que realizará a ação de coletar o nome do grupo, na tela de conectar com partida] [Input de "Código da partida" que realizará a ação de coletar o código da partida que o usuário deseja ingressar, na tela de conectar com partida] para maximizar facilidade de visualização - Design de navegação: Os elementos <ul style="list-style-type: none"> [Botão "Conectar com partida" deverá levar o usuário para a tela da partida] [Botão "Voltar" deverá levar o usuário para a tela inicial] - Design de informação: A <ul style="list-style-type: none"> [informação "nome do grupo" deve ser apresentada na tela da partida com o próprio nome do grupo] <p>- Superfície:</p> <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo ícone de seta voltando no elemento botão de voltar, de maneira que o botão de voltar esteja no canto superior esquerdo da tela. Vide protótipo para mais informações

Figura 23 – US10, Fonte: desenvolvedores participantes do *Workshop*

Eu, como <SÊNIOR> quero/preciso realizar a interação < DE CONTROLAR O TEMPO QUE CADA PESSOA/GRUPO TEM PARA RESPONDER UMA QUESTÃO> para que eu atinja o objetivo <DE IGUALDADE NO TEMPO DE RESPOSTA>, quando / a partir de <O GRUPO/PESSOA RECEBER UMA PERGUNTA>. Avalio que meu objetivo estará concluído quando eu receber <QUANTO TEMPO RESTA PARA RESPONDER QUESTÕES>.

- Estrutura:

- Design de interação:
 - O usuário pode realizar as atividades [<DE ADICIONAR UM TEMPO PARA RESPONDER>, que resultará na <alteração || criação || deleção || visualização> da informação < TEMPO DE RESPOSTA> com o resultado de <APRESENTAR UMA CONTAGEM REGRESSIVA>]. Em caso de erro, a resposta deverá ser [‘NÃO FOI POSSÍVEL CRIAR A CONTAGEM REGRESSIVA NA PARTIDA’]. Em caso de sucesso, a resposta deverá ser [‘CONTAGEM REGRESSIVA ADICIONADA NA PARTIDA’].
 - A interação deve poder ser cancelada nos momentos [CRIAÇÃO DE TEMPO DE RESPOSTA] e deve realizar a ação [CRIAR UMA CONTAGEM PARA RESPOSTAS]
 - A lista de ações desta história é:
 - SELECIONAR UMA PARTIDA
 - ADICIONAR UM TEMPO DE RESPOSTA
 - EDITAR SE O USUÁRIO QUER CONTAGEM REGRESSIVA OU NÃO
 - ADICIONAR O TEMPO DE RESPOSTA
- Arquitetura da Informação: As informações [<ADICIONAR TEMPO PARA RESPOSTA>] deverão ser apresentadas nos [<QUESTIONÁRIOS>], com os dados [<DE TEMPO>], respectivamente. Estes elementos deverão disponíveis nas ações [RESPONDER UMA PERGUNTA]
- Esqueleto:
 - Design de interface: A interface deve conter os elementos [<DE TEMPO DE RESPOSTA> que realizará a ação <ADICIONAR TEMPO PARA RESPOSTA>, no local <ACIMA DAS OPÇÕES DE RESPOSTA DA QUESTÃO E ACIMA DA PERGUNTA>] para maximizar facilidade de visualização
 - Design de navegação: Os elementos [<BOTÃO CRIAR TEMPO DE RESPOSTA> deverão levar o usuário para a tela <DE ADICIONAR UM TEMPO PARA CADA QUESTIONÁRIO>> deverão mostrar a informação <DE TEMPO DE RESPOSTA>]
 - Design de informação: A [informação <TEMPO PARA RESPOSTA> deve ser apresentada no elemento <TELA DE PERGUNTAS> com os parâmetros {TEMPO RESTANTE PARA RESPOSTA}]
- Superfície:
 - Design sensorial: A interface deve seguir o design proposto, com o estilo [<botão com a cor verde escuro e fonte bold branca> no elemento <ADICIONAR TEMPO DE RESPOSTA>], de maneira o elemento [<TEMPO RESTANTE PARA RESPOSTA> possua destaque sobre o elemento <BOTÃO DE ADICIONAR TEMPO DE RESPOSTA> estando na posição <ADICIONAR TEMPO NO QUESTIONÁRIO>].
 - Vide protótipo para mais informações

Figura 24 – US11, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como pessoa organizadora da partida, quero/preciso realizar a mudança da paleta de cores de uma partida para que eu atinja o objetivo de personalizar a partida, quando / a partir de a partida começar e exibir cores personalizadas. Avalio que meu objetivo estará concluído quando eu o usuário puder visualizar cores da partida.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<mudar a paleta de cores da partida>, que resultará na <mudança de cores de fundo, botões e cards> da informação <partida> com o resultado de <ter partidas personalizadas>]. Em caso de erro, a resposta deverá ser ["Não foi possível alterar a paleta de cores da partida. Tente novamente."]. Em caso de sucesso, a resposta deverá ser ["Alteração da paleta de cores realizada com sucesso! "]. - A interação deve poder ser cancelada no momento [de personalizar a partida] e deve realizar a ação [limpar as escolhas personalizadas e voltar para a tela de painel de gerenciamento de organizador] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Selecionar opção Organizador - Fazer login na aplicação - Selecionar criação de partida - Selecionar opção de personalizar partida - Realizar alteração da paleta de cores - Clicar em criar partida - Receber feedback sobre a mudança da paleta - Visualizar cores na tela da partida] - Arquitetura da Informação: As informações [<paleta de cores>] deve ser apresentada na [<tela de personalização da partida>], com os dados [<paletas de cores pré-definidas pelo sistema>], respectivamente. Estes elementos deverão disponíveis nas ações [selecionar paleta de cores da partida] - Esqueleto: <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<select de paletas de cores pré-definidas pelo sistema> que realizará a ação <selecionar paleta desejada>, no local <modal de personalizar partida>] para maximizar facilidade de visualização. - Design de navegação: O elemento [<botão de personalizar partida>] deverá levar o usuário para a tela <de personalização da partida> deverá mostrar a informação de <seleção de paleta de cores>] - Design de informação: A [informação <paleta de cores>] deve ser apresentada no elemento <tela da partida> com os parâmetros {cores de fundo, botões e cards de acordo com a paleta selecionada}} - Superfície: <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<fundo branco com select da amostra da cor da paleta de cores> no elemento <personalizar partida>], de maneira que o elemento [<criar partida>] possua destaque sobre o elemento <selecionar paleta de cores> estando na posição <abaixo da seleção da paleta>]. Vide protótipo para mais informações

Figura 25 – US12, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como participante e pessoa organizadora da partida, quero/preciso visualizar a pontuação dos times para que eu atinja o objetivo de feedback imediato, quando / a partir do começo da partida. Ávalio que meu objetivo estará concluído quando eu visualizar a pontuação em tempo real.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<visualizar pontuação dos times>, que resultará na <exibição de pontos do time/pontos totais> da informação <partida do jogo> com o resultado de <ter feedback imediato>]. Em caso de erro, a resposta deverá ser ["Pontuação não disponível. Recarregue a página e tente novamente"]. Em caso de sucesso, a resposta deverá ser [Exibição da pontuação]. - A interação deve poder ser cancelada nos momentos [após a partida finalizar] e deve realizar a ação [redirecionar para tela inicial] - A lista de ações desta história é: <ul style="list-style-type: none"> [- Selecionar opção Organizador - Fazer login na aplicação - Selecionar criação de partida - Clicar em criar partida - Receber feedback sobre pontuações dos times] ou [- Selecionar opção Campista - Preencher nome do grupo e código da partida - Clicar em Conectar em partida - Aguardar escolha dos grupos - Receber feedback sobre pontuações dos times] - Arquitetura da Informação: As informações [<pontuação do time>] deverão ser apresentadas na [<partida do jogo>], com os dados [<pontos dos times e pontos totais da partida>], respectivamente. Estes elementos deverão disponíveis [dentro da partida do jogo] - Esqueleto: <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<pontos dos times e pontos totais da partida> que realizará a ação de <exibir pontuação dos times>, no local <tela da partida do jogo>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<criar partida/conectar em partida> deverão levar o usuário para a tela <partida do jogo>] deverão mostrar a informação <pontuação dos times>] - Design de informação: A [informação <pontuação dos times>] deve ser apresentada no elemento <partida do jogo> com os parâmetros {pontuação do time/pontuação total da partida} - Superfície: <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<progress bar exibindo nome do time e pontuação baseada na pontuação total da partida> no elemento <partida do jogo>], de maneira o elemento [<pergunta>] possua destaque sobre o elemento <pontuação> estando na posição <lateral da tela>]. Vide protótipo para mais informações

Figura 26 – US13, Fonte: desenvolvedores participantes do *Workshop*

<p>Eu, como <ESCOTEIRO> quero/preciso realizar a interação <DE VISUALIZAR AS PONTUAÇÕES> para que eu atinja o objetivo <DE SABER A PONTUAÇÃO DOS GRUPOS>, quando / a partir de <EU ACERTAR OU ERRAR UMA PÉRGUNTA>. Avalio que meu objetivo estará concluído quando eu receber <UM RANKING COM AS PONTUAÇÕES DE TODOS OS GRUPOS>.</p>
<p>- Estrutura:</p> <ul style="list-style-type: none"> - Design de interação: <ul style="list-style-type: none"> - O usuário pode realizar as atividades [<DE VISUALIZAÇÃO DA PONTUAÇÃO>, que resultará na <alteração criação deleção visualização> da informação <DA PONTUAÇÃO> com o resultado de <APRESENTAR A PONTUAÇÃO QUE CADA GRUPO POSSUI>]. Em caso de erro, a resposta deverá ser [VOCÊ ERROU A QUESTÃO, NÃO MARCOU PONTOS]. Em caso de sucesso, a resposta deverá ser [PARABÉNS, VOCÊ ACERTOU A RESPOSTA]. - A interação deve poder ser cancelada nos momentos [FINAL DA PARTIDA] e deve realizar a ação [REDIRECIONAR A TELA INICIAL] - A lista de ações desta história é: <ul style="list-style-type: none"> [- APRESENTAR O RANKING COM AS PONTUAÇÕES NA TELA DE QUESTÕES - APRESENTAR O RANKING DE PONTUAÇÃO NA TELA DE OPÇÕES DE RESPOSTA -] - Arquitetura da Informação: As informações [<RANKING DE PONTUAÇÃO>] deverão ser apresentadas nos [<TELAS DE PERGUNTA E TELA DE RESPOSTA>], com os dados [<DE QUAL TIME ESTÁ EM PRIMEIRO/SEGUNDO ETC E A PONTUAÇÃO OBTIDA>], respectivamente. Estes elementos deverão disponíveis nas ações [DE RESPOSTA DA QUESTÃO] <p>- Esqueleto:</p> <ul style="list-style-type: none"> - Design de interface: A interface deve conter os elementos [<RANKING> que realizará a ação <APRESENTAR A PONTUAÇÃO>, no local <ACIMA DA QUESTÃO A SER RESPONDIDA>] para maximizar facilidade de visualização - Design de navegação: Os elementos [<RANKING> deverão levar o usuário para a tela <DE PONTUAÇÃO>] deverão mostrar a informação <DE QUANTAS PERGUNTAS FORAM RESPONDIDAS CORRETAMENTE>] - Design de informação: A [informação <RANKING COM PONTUAÇÃO> deve ser apresentada no elemento <EM QUE O GRUPO RESPONDE UMA PERGUNTA> com os parâmetros {ÍCONE DE MEDALHA, NOME DO GRUPO, PONTUAÇÃO}} <p>- Superfície:</p> <ul style="list-style-type: none"> - Design sensorial: A interface deve seguir o design proposto, com o estilo [<ÍCONE DE MEDALHA DE OUTRO PARA O PRIMEIRO, PRATA PARA O SEGUNDO, BRONZE PARA O TERCEIRO NO RANKING> no elemento <PONTUAÇÃO>], de maneira o elemento [<PONTUAÇÃO> possua destaque sobre o elemento <QUESTÃO A SER RESPONDIDA> estando na posição <ACIMA DA PERGUNTA>]. <p>Vide protótipo para mais informações</p>

Figura 27 – US14, Fonte: desenvolvedores participantes do *Workshop*

ANEXO B – Aplicação base antes das melhorias

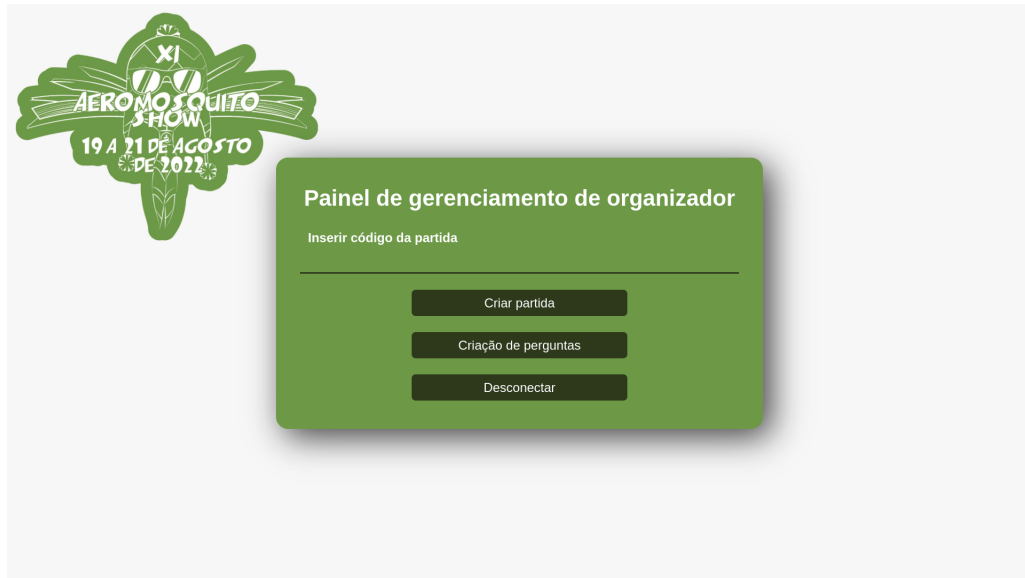


Figura 28 – Tela de organizador, Fonte: autor



Figura 29 – Tela de organizador em partida antes de escolher participante, Fonte: autor



Figura 30 – Tela de organizador em partida depois de escolher participante, Fonte: autor

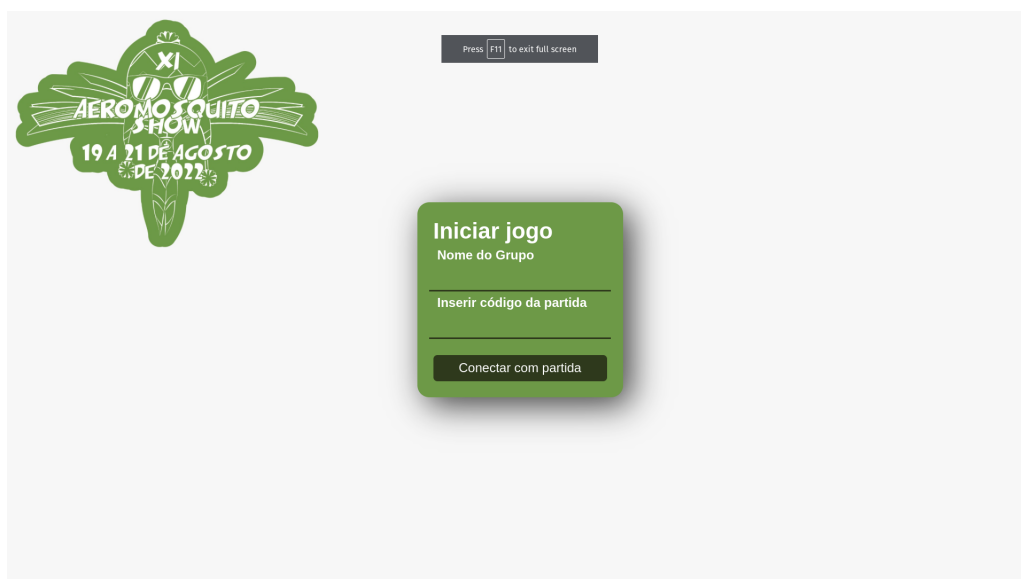


Figura 31 – Tela de campista, Fonte: autor

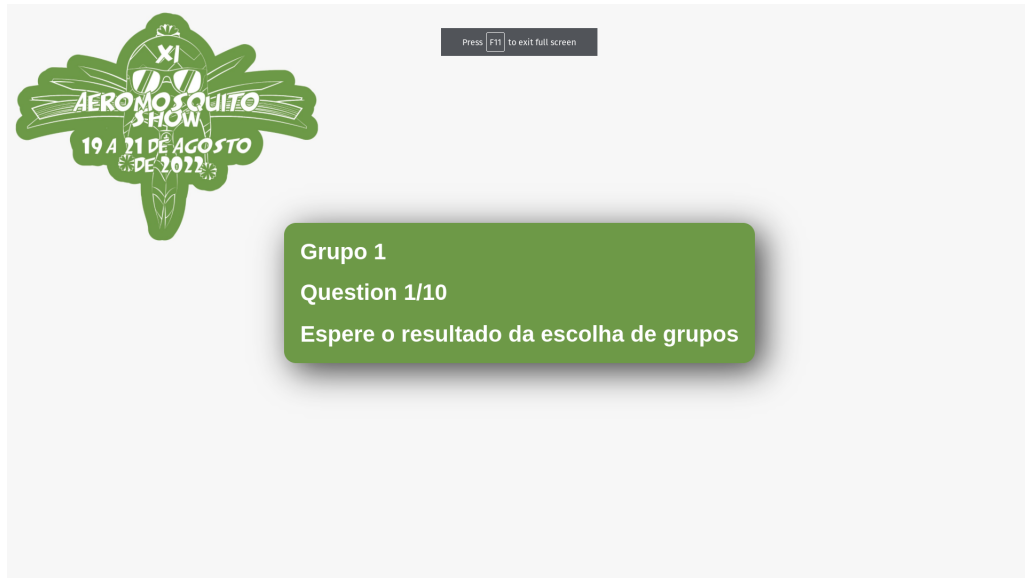


Figura 32 – Tela de campista em partida antes de poder responder, Fonte: autor



Figura 33 – Tela de campista em partida com opções de escolha, Fonte: autor

ANEXO C – Aplicação base depois das melhorias



Figura 34 – Tela de organizador, Fonte: autor

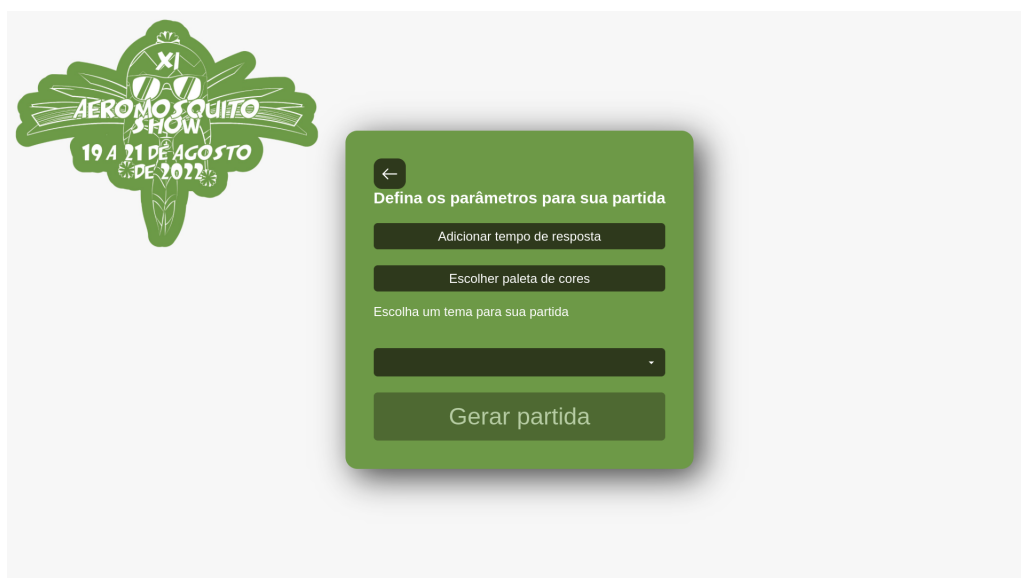


Figura 35 – Tela de organizador ao criar partida, Fonte: autor



Figura 36 – Tela de organizador criando partida com as opções, Fonte: autor



Figura 37 – Tela de organizador em partida antes de escolher participante, Fonte: autor



Figura 38 – Tela de campista em partida antes de ser escolhido, Fonte: autor

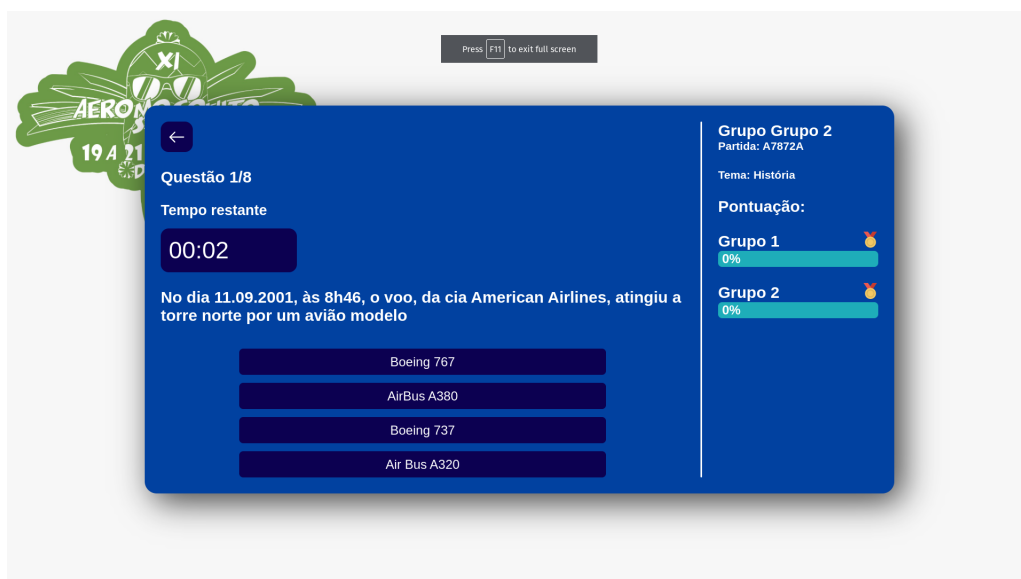


Figura 39 – Tela de campista em partida com as opções de resposta, Fonte: autor