



UEPB

UNIVERSIDADE ESTADUAL DA PARAÍBA

CAMPUS I - CAMPINA GRANDE

PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA

PROGRAMA DE PÓS-GRADUAÇÃO PROFISSIONAL EM CIÊNCIA E

TECNOLOGIA EM SAÚDE

**CURSO DE MESTRADO PROFISSIONAL EM CIÊNCIA E TECNOLOGIA EM
SAÚDE**

ADEILTON SOUSA

**DESENVOLVIMENTO DE SOFTWARE PARA AUXÍLIO A AVALIAÇÃO
HEURÍSTICA**

**CAMPINA GRANDE
2024**

ADEILTON SOUSA

**DESENVOLVIMENTO DE SOFTWARE PARA AUXÍLIO A AVALIAÇÃO
HEURÍSTICA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia em Saúde da Universidade Estadual da Paraíba como requisito para obtenção do título de Mestre em Ciência e Tecnologia em Saúde.

Área de concentração: Tecnologias em Saúde

Orientador: Prof. Dr. Daniel Scherer.

**CAMPINA GRANDE
2024**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

S725d Sousa, Adailton.
Desenvolvimento de software para auxílio a avaliação heurística [manuscrito] / Adailton Sousa. - 2024.
49 p. : il. colorido.

Digitado.

Dissertação (Mestrado Profissional em Ciência e Tecnologia em Saúde) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2024.

"Orientação : Prof. Dr. Daniel Scherer, Departamento de Computação - CCT. "

1. Avaliação. 2. Heurística. 3. Interface. I. Título

21. ed. CDD 372.35

ADEILTON SOUSA

DESENVOLVIMENTO DE SOFTWARE PARA AUXÍLIO A AVALIAÇÃO
HEURÍSTICA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência e Tecnologia em Saúde da Universidade Estadual da Paraíba como requisito para obtenção do título de Mestre em Ciência e Tecnologia em Saúde.

Área de concentração: Tecnologias em Saúde

Aprovada em 27/10/2023.

BANCA EXAMINADORA



Prof. Dr. Daniel Scherer
Universidade Estadual da Paraíba (UEPB)



Prof. Dr. Andrei Guilherme Lopes
Universidade Estadual da Paraíba (UEPB)

Prof. Dr. Ademar Virgolino da Silva Netto
Universidade Federal da Paraíba (UFPB)

RESUMO

Entre os métodos utilizados na avaliação de usabilidade de software, a Avaliação Heurística apresenta-se como um método eficiente e de baixo custo, quando comparado a outros métodos de avaliação. Como resultado da Avaliação Heurística, cada avaliador gera um relatório individual e posteriormente um relatório em conjunto com outros avaliadores. Este processo requer um esforço em utilização de softwares de edição de imagens e texto, que podem tirar o foco da avaliação e requerer bastante tempo. Neste trabalho, será proposto o desenvolvimento de uma aplicação para auxílio a atividade de Avaliação Heurística, oferecendo em uma única aplicação todos os recursos necessários para a realização do processo, de modo a colaborar para a agilidade e padronização do processo de avaliação. Para levantamento das características do software a ser desenvolvido, foi feita uma análise dos métodos e softwares comumente utilizados neste tipo de avaliação e o sistema proposto buscará auxiliar o avaliador tornando o processo de avaliar mais eficiente e prático, oferecendo ao especialista a oportunidade de centralizar todo o processo em apenas uma ferramenta, projetada levando em consideração as particularidades, características e necessidades deste método de avaliação.

Palavras-chave: avaliação, heurística; interface.

ABSTRACT

Among the methods used in software usability evaluation, Heuristic Evaluation presents itself as an efficient and low-cost method, when compared to other evaluation methods. As a result of Heuristic Evaluation, each evaluator generates an individual report and later a joint report with other evaluators. This process requires effort in using image and text editing software, which can distract from the evaluation and require a lot of time. In this work, the development of an application to assist the Heuristic Assessment activity will be proposed, offering in a single application all the resources necessary to carry out the process, in order to contribute to the agility and standardization of the assessment process. To survey the characteristics of the software to be developed, an analysis was made of the methods and software commonly used in this type of evaluation and the proposed system will seek to assist the evaluator by making the evaluation process more efficient and practical, offering the specialist the opportunity to centralize all the process in just one tool, designed taking into account the particularities, characteristics and needs of this evaluation method.

Keywords: evaluation, heuristics; interface.

SUMÁRIO

1	INTRODUÇÃO	6
1.1	OBJETIVO GERAL	7
1.2	OBJETIVOS ESPECÍFICOS	7
2	CONCEITOS GERAIS	9
2.1	USABILIDADE	9
2.2	FORMAS DE AVALIAÇÃO	10
2.3	AVALIAÇÃO HEURÍSTICA	12
2.3.1	ETAPAS PARA AVALIAÇÃO HEURÍSTICA	13
3	LEVANTAMENTO DE DADOS PARA DEFINIÇÃO DE REQUISITOS	16
4	TRABALHOS RELACIONADOS	17
4.1	FERRAMENTA MOCKUP DUE (AVALIAÇÃO HEURÍSTICA)	17
4.2	HEVA: UMA FERRAMENTA DE SUPORTE À AVALIAÇÃO HEURÍSTICA	17
4.3	THEM - TOOL FOR HEURISTIC EVALUATION METHODS	17
4.4	ANÁLISE COMPARATIVA A TRABALHOS RELACIONADOS	18
5	APRESENTAÇÃO DOS REQUISITOS	21
6	SOFTWARE DESENVOLVIDO	23
6.1	TECNOLOGIAS UTILIZADAS	28
6.2	CARACTERÍSTICAS DO SOFTWARE DESENVOLVIDO	31
7	CONCLUSÃO	45
7.1	TRABALHOS FUTUROS	46
	REFERÊNCIAS	48

1 INTRODUÇÃO

O uso de software nos mais variados dispositivos e telas é algo que cresce a cada dia. De modo geral, uma falha ou inconsistência no uso de tais softwares podem não trazer nada mais que insatisfação aos usuários. Mas existem sistemas críticos, em que erros podem trazer danos a integridade física ou mesmo a vida de uma pessoa, a exemplo dos sistemas em dispositivos médicos (MARAMBA; CHATTERJEE; NEWMAN, 2019).

Softwares de diagnósticos precisam receber e fornecer dados a pacientes e profissionais da saúde com exatidão, precisão e confiabilidade. Dado o aumento exponencial no número de softwares voltados para a área da saúde, torna-se imperativo que esses programas sejam submetidos a avaliações rigorosas, a fim de mitigar possíveis falhas e aprimorar sua segurança. Entre os métodos utilizados para realizar esta avaliação da usabilidade de software está a Avaliação Heurística (MARAMBA; CHATTERJEE; NEWMAN, 2019). Este método tem como objetivo analisar o software quanto à conformidade ou não, de um conjunto de heurísticas, que buscam tornar a experiência do utilizador mais simples, e ajudar na interação do homem com a máquina (NIELSEN; MOLICH, 1990).

Uma heurística é uma diretriz, princípio ou regra geral que descreve uma propriedade comum em interfaces com algum grau de usabilidade (WILSON, 2013). A Avaliação Heurística é um método eficiente e de baixo custo de execução, se comparada a outros métodos, e trás um bom retorno na detecção dos erros, conduzindo a retornos significativos na avaliação ou crítica a um produto. O termo Avaliação Heurística e os primeiros métodos de realização deste tipo de inspeção para detecção de problemas em uma interface, foram apresentados por Jakob Nielsen e Rolf Molich em 1990. Como resultado dos seus estudos, ambos apresentaram métodos de condução, e boas práticas que ajudam a atingir melhores níveis de eficiência na aplicação deste tipo de avaliação de interfaces (NIELSEN; MOLICH, 1990).

Para realizar a Avaliação Heurística os especialistas usam uma seleção de regras para medir a usabilidade das interfaces do usuário em análises independentes e identificar problemas. Comumente os avaliadores usam heurísticas estabelecidas e pré-selecionadas que revelam percepções que podem ajudar as equipes de design a aprimorar a usabilidade do produto desde o início do desenvolvimento (WILSON, 2013).

Ao trabalhar com este método de avaliação, a equipe pode identificar que em determinados momentos a avaliação pode sofrer interferência por conhecimentos preconcebidos dos avaliadores. Não podemos citar este fato como sendo um grande problema no método já que as heurísticas costumam ser baseadas nas experiências

de usuários reais com diversas interfaces (CYBIS; BETIOL; FAUST, 2017). Outra situação que foi identificada ao se analisar este método de avaliação, foi a falta de um software que auxilie na condução do processo com vários avaliadores, já que cada um poderá utilizar ferramentas e software que lhe for mais conveniente podendo gerar dificuldade em consolidar um relatório padronizado com os resultados dos avaliadores envolvidos.

Neste estudo será proposto o desenvolvimento de uma ferramenta de software com um conjunto de ferramentas pensadas para o processo de Avaliação Heurística, buscando mitigar os possíveis problemas deste método de avaliação.

1.1 OBJETIVO GERAL

Este estudo tem como objetivo analisar os métodos utilizados na realização da Avaliação Heurística, a fim de compreender os recursos empregados na execução desse método de avaliação. Para alcançar esse objetivo, o estudo busca identificar as necessidades dos profissionais de usabilidade na execução de uma Avaliação Heurística, catalogando os métodos, softwares e ferramentas utilizados para a realização das avaliações.

Os dados obtidos nesta pesquisa serão utilizados para propor o desenvolvimento de uma ferramenta que possa oferecer o máximo de recursos possíveis para auxiliar na execução da Avaliação Heurística. Essa solução de software visa reduzir ou eliminar problemas comuns nesse tipo de avaliação de software, como a subjetividade na detecção de falhas, e simplificar a consolidação do relatório final, que contém as observações e apontamentos da equipe de avaliação.

1.2 OBJETIVOS ESPECÍFICOS

Com o objetivo de atingir as metas propostas, o estudo deverá seguir as seguintes etapas:

- Identificar os softwares utilizados na realização da Avaliação Heurística;
- Identificar as principais características desses softwares;
- Mapear os métodos adotados por diferentes grupos de avaliadores;
- Gerar uma lista de características comuns presentes nos softwares;
- Identificar as formas de consolidação dos relatórios da Avaliação Heurística;

- Avaliar possíveis tecnologias que possam facilitar o processo de Avaliação Heurística de Software.

Essas etapas são cruciais para entender os recursos utilizados na execução da Avaliação Heurística e identificar as necessidades dos profissionais de usabilidade nesta área. Além disso, os dados coletados durante o estudo serão utilizados para propor o desenvolvimento de uma ferramenta que possa oferecer mais recursos e ajudar a simplificar o processo de avaliação, incluindo a consolidação do relatório final.

2 CONCEITOS GERAIS

Este capítulo aborda os conceitos gerais de usabilidade, incluindo sua definição, importância e métodos de avaliação.

2.1 USABILIDADE

Para a *International Organization for Standardization* na ISO 9241-11 a definição de Usabilidade está contida em três pilares: efetividade, eficiência e satisfação na utilização de uma ferramenta em seu contexto específico (CYBIS; BETIOL; FAUST, 2017). Jakob Nielsen diz que a usabilidade é “um atributo de qualidade que avalia como as interfaces de usuário são fáceis de usar”. Em seguida Nielsen cita que a usabilidade pode ser definida por meio de cinco qualidades: capacidade de aprendizado; eficiência; capacidade de memorização; erros e satisfação (NIELSEN, 1994).

“Sistemas difíceis de usar implicam em erros e perda de tempo, fatores que se multiplicam com a frequência das tarefas e o número de usuários. A perda de dados e informações pode implicar na perda de clientes e de oportunidades. Acontecimentos deste tipo causam desde uma resistência ao uso do sistema até a sua subutilização e abandono completo.” (CYBIS; BETIOL; FAUST, 2017).

Para garantir uma boa usabilidade de um software, é necessário que o usuário seja colocado no centro do processo, considerando suas necessidades para que as tarefas possam ser realizadas de maneira fácil e eficiente. A falta de usabilidade pode resultar em custos substanciais para os desenvolvedores, pois a insatisfação do cliente, baixa produtividade dos usuários e um maior número de solicitações de suporte são apenas alguns exemplos de problemas que podem surgir.

Por outro lado, um software com boa usabilidade pode resultar em maior satisfação do usuário, aprendizado mais rápido dos recursos e redução dos custos envolvidos em treinamento e suporte. A *International Organization for Standardization (ISO)* apresenta em sua norma *ISO 9241-11:2018(en) Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts*, cinco características que os sistemas com boa usabilidade geralmente possuem:

- facilidade para aprender: o sistema requer pouco esforço para que o usuário aprenda a utilizar;
- fácil memorização: os usuários devem lembrar da interface com facilidade mesmo após algum tempo de uso;

- maximizar a produtividade: a interface do software deve permitir aos usuários serem rápidos e eficientes em suas tarefas;
- mitigar a taxa de erros: o usuário deverá ser informado dos erros, e serem capaz de corrigi-lo de modo fácil;
- maximizar a satisfação do usuário: a interface deve transmitir confiança e segurança.

As avaliações de usabilidade são importantes para melhorar a aceitação do usuário e contribuir para a maximização da usabilidade. Aumentar a usabilidade é um dos fatores que podem contribuir para uma maior aceitação do usuário. (NIELSEN, 1994).

2.2 FORMAS DE AVALIAÇÃO

Dentro do estudo de avaliação de usabilidade, existem alguns métodos que são comumente utilizados em conjunto com a Avaliação Heurística. Podemos citar: Percurso Cognitivo, Mensuração de Desempenho e Questionários.

Percurso Cognitivo: este tipo de avaliação, envolve um avaliador ou um grupo que examinam um sistema, se colocando no papel do usuário, passando por um conjunto de tarefas e avaliando sua compreensão e facilidade de aprendizado. A interface poderá ser apresentada por meio de *mockups* em papel ou mesmo o sistema totalmente desenvolvido. Esta técnica é melhor utilizada na fase de desenvolvimento do projeto, podendo também ser aplicada durante os estágios de codificação, teste e implantação.

Para realizar uma avaliação por percurso cognitivo, é comum se buscar definir (N. SOKEN et al.):

- quem será o usuário do sistema?
- quais tarefas serão analisadas?
- qual é a sequência de ação correta para cada tarefa?
- como é definida a interface?

À medida que a análise começa, os avaliadores fazem as seguintes quatro perguntas (N. SOKEN et al.) :

- os usuários tentaram alcançar os objetivos corretos?
- o usuário notará que a ação correta está disponível?
- o usuário conseguirá associar a ação correta com o efeito a ser alcançado?
- se a ação correta for realizada, o usuário verá que o progresso está sendo feito em direção à solução da tarefa?

Mensuração de Desempenho: este método é utilizado para obter dados quantitativos e qualitativos sobre o software ou objeto que está sendo submetido a avaliação de usabilidade. Recomenda-se que seja conduzido em um laboratório de usabilidade para que os dados possam ser coletados com precisão e com o menor número de interferência (CYBIS; BETIOL; FAUST, 2017). Quanto ao número de participantes, utiliza-se pelo menos 5 usuários, podendo haver uma grande variação, a depender de o que está sendo avaliado (podendo chegar a 20 usuários). Pode-se utilizar este método nos seguintes estágios de desenvolvimento: código (avaliação de baixa fidelidade¹), teste (avaliação de baixa ou média fidelidade²) e implantação, porém preferencialmente é melhor aproveitado para produtos em fase de protótipo final ou acabados (avaliação de alta fidelidade³).

A avaliação se dá por procurar identificar as características do software, como por exemplo: facilidade de aprendizagem, eficiência, poucos erros e pouca subjetividade. Para gerar os dados sobre cada utilizador, os avaliadores buscam quantificar os problemas encontrados utilizando critérios como (N. SOKEN et al.) :

- o tempo que os usuários tomam para completar uma tarefa específica;
- o número de tarefas que podem ser completadas dentro de um limite de tempo;
- a relação entre interações bem-sucedidas e erros;
- o tempo gasto na recuperação de erros;
- o número de erros cometidos pelos usuários;
- o número de comandos ou outros recursos que nunca foram utilizados pelo usuário;
- o número de características do sistema que o usuário pode lembrar após a avaliação.

Embora esta técnica tenha como objetivo coletar dados quantitativos, deve-se notar que é muito importante coletar dados qualitativos para descobrir o processo mental do usuário e outras informações por trás dos dados quantitativos e levá-los em conta ao tirar as conclusões.

Questionários: Esta técnica utiliza perguntas que podem ser de formato aberto ou fechado, para tentar obter respostas objetivas e claras dos entrevistados. Geralmente este método é utilizado em um grupo grande de pessoas que possam estar distantes geograficamente (ROGERS; SHARP; PREECE, 2012).

Neste método é necessário muito cuidado para que as perguntas sejam elaboradas de forma correta, afim de fornecer os dados de modo eficiente. Alguns

1 Baixa fidelidade significa produto em fases preliminares (prototipação em papel) e ambiente de simulação distante do real do usuário.

2 Média fidelidade significa produto em fases mais avançadas (alfa teste) e ambiente de simulação com alguma proximidade com o real do usuário.

3 Alta fidelidade significa produto em fase final (beta teste, produto acabado) e ambiente de simulação com fidedignidade com o ambiente real do usuário.

pontos a serem observados na elaboração de uma avaliação por questionário (ROGERS; SHARP; PREECE, 2012):

- o sentido de uma pergunta pode ser influenciado pelo momento em que é feita;
- questionários ajustado a diferentes perfis de usuários;
- informações claras, boa redação e tipografia;
- tamanho adequado do questionário, para não inibir o preenchimento.

2.3 AVALIAÇÃO HEURÍSTICA

O termo Avaliação Heurística foi introduzido por Nielsen e Molich em 1990 com um artigo intitulado *Heuristic Evaluation of User Interfaces* em seguida publicado como parte de um livro (NIELSEN, 1990). As heurísticas são um conjunto de diretrizes ou princípios, utilizados para guiar os avaliadores para descobrirem problemas que um usuário provavelmente encontrará ao utilizar o sistema avaliado. Alguns especialistas, entre eles Jakob Nielsen (NIELSEN; MOLICH, 1990) , Ben Shneiderman (UNIVERSITY OF MARYLAND), Dominique Scapin e Christian Bastien, propuseram listas de heurísticas criadas para diversas áreas de aplicação (CYBIS; BETIOL; FAUST, 2017).

A Avaliação Heurística pode ser aplicada em qualquer etapa do projeto, seja ele um software ou não. Para realizar uma Avaliação Heurística comumente se adota os seguintes passos (NIELSEN; MOLICH, 1990):

- definir um conjunto de heurísticas a serem aplicadas;
- selecionar um time de 3 ou 5 avaliadores. É comum que se use especialistas com experiência de usabilidade e outros com domínio relacionado ao projeto;
- definir um sistema de escala de gravidade e certificar-se de que os avaliadores possam usá-lo de forma consistente;
- buscar identificar problemas com base no conjunto de heurísticas selecionadas;
- apontar onde o problema se encontra (a página/tela, localização na página), o nível de gravidade e sua relação com a heurística atingida;
- apresentar sugestões para correção do problema.

A entrega destes dados é feita habitualmente por meio de relatórios contendo as informações de diagnóstico dos especialistas envolvidos na avaliação (NIELSEN, 1994). No relatório deverá conter a descrição do erro ou mesmo um registro em imagem, exibindo o software ou uma representação do mesmo para melhor apontamento do diagnóstico. Os dados dos relatórios de cada participante devem

ser agrupados e discutidos em grupo, para se analisar as subjetividades e debater a consolidação dos pontos prioritários, para que sejam corrigidos no projeto .

A Avaliação Heurística é considerada uma maneira rápida e de baixo custo para medir e melhorar a usabilidade de um produto antes de realizar outros métodos de avaliação de usabilidade (SANTOS et al.). Deve ser uma parte do processo de desenvolvimento e comumente é utilizada em conjunto com outros métodos de avaliação de usabilidade, em especial o teste com usuários. Na seleção dos especialistas que estarão envolvidos na Avaliação Heurística, é recomendável que se esteja incluído um especialista duplo, ou seja, um especialista em usabilidade que domina o processo de avaliação e também possua conhecimento na área de aplicação do software a ser avaliado, podendo este ser capaz de identificar necessidades específicas para usuários reais (PAZ et al., 2015). Por outro lado, um grupo de pessoas diferentes, tendem a analisar uma interface de diferentes ângulos e, como resultado, são mais propensos a identificar um conjunto mais amplo de aspectos para melhoria (BARBOSA; SILVA, 2010).

Quanto à eficácia do método, os estudos realizados por Nielsen mostram que entre 65% e 75% dos problemas de usabilidade são identificados quando comparados com outros métodos de avaliação de usabilidade. Este número poderá ser diferente dependendo da quantidade, perfil e experiência da equipe que realiza a avaliação (CYBIS; BETIOL; FAUST, 2017).

2.3.1 ETAPAS PARA AVALIAÇÃO HEURÍSTICA

Planejamento: Uma vez que a Avaliação Heurística é uma técnica de avaliação da usabilidade, deve-se ter um objetivo claro do que se espera alcançar com a avaliação. Para isto deve-se levar em consideração o contexto, o usuário e a tarefa. Desta forma, será necessário definir os objetivos antes de realizar a inspeção, e espera-se que os avaliadores entendam o que exatamente precisa ser avaliado, certificando-se que os especialistas que estão envolvidos estarão informados sobre as especificações do software e entendem seus objetivos. Os avaliadores precisam compreender quem são seus usuários, qual o contexto de utilização, e quais as tarefas que o software foi projetado para realizar. É também nesta fase de planejamento que se define o conjunto de heurísticas que será utilizada e o número de avaliadores que estarão envolvidos (BARBOSA; SILVA, 2010).

Nesta fase preliminar de planejamento, será necessário selecionar (MORAES; ROSA) :

- Avaliadores: Personagens que podem ser especialistas apenas em

usabilidade, e conhecer as heurísticas, e métodos de avaliação;

- Especialistas: Além do conhecimento em usabilidade, possuem conhecimentos específicos da área do produto a ser testado;
- Todos os envolvidos devem buscar conhecimento sobre as regras de negócio do software.

É necessário que os avaliadores estejam alinhados quanto aos objetivos da avaliação e que possuam o mesmo checklist de tarefas a serem realizadas na avaliação do software. Entre as informações que serão passadas aos avaliadores, deve haver a relação de heurísticas que foram definidas e que se mostrem adequadas ao software que será avaliado. É possível que especialistas envolvidos no processo de avaliação, combinem heurísticas conhecidas, e amplamente utilizadas, como as dez heurísticas de Nielsen e Molich e as oito regras de ouro de Ben Shneiderman (MORAES; ROSA). Sendo também possível adicionar ao checklist, outras heurísticas desenvolvidas pela própria equipe, para que consigam atingir as particularidades do Software avaliado.

Para que os avaliadores possam compreender os requisitos a serem avaliados, uma etapa crucial no planejamento da avaliação é determinar o perfil dos usuários da aplicação. O levantamento do perfil do usuário deverá permitir que seja definido quais os conhecimentos, anseios, necessidades e capacidades dos futuros usuários do sistema. Esta análise dos perfis deve ser capaz de fornecer dados suficientes para guiar os avaliadores e projetistas do sistema a fazerem ajustes para que o software esteja de acordo com as expectativas e necessidades de cada perfil de usuário (MORAES; ROSA).

Na descrição do perfil do usuário deve conter seu conhecimento, capacidade, desejo e cultura. Na fase de planejamento será realizada também a análise da tarefa, que visa determinar as atividades que cada um dos papéis definidos irá realizar. Esta informação ajuda os avaliadores a simularem no momento da avaliação, as atividades que cada perfil provavelmente irá realizar. Ter em mente o contexto de uso, os perfis de usuários e as tarefas, dará aos avaliadores a oportunidade de simular com maior precisão as atividades mais relevantes a serem avaliadas para aquele software específico.

Execução: Uma vez que os objetivos estão claros, um conjunto de heurísticas definidas e uma equipe de avaliadores pronta, o próximo passo é a fase de execução. Os avaliadores irão percorrer os fluxos e as respectivas interfaces do software de forma individual, realizando uma análise com base nos princípios definidos e sempre que se deparar com um problema ou uma área de melhoria, deverão tomar nota, e detalhar junto ao problema encontrado, o maior número possível de detalhes relevantes informando qual o princípio ou heurística foi violada,

onde eles encontraram o problema, porque isso é um problema e, possivelmente, também sugerir maneiras de corrigi-lo.

Revisão: Depois que as avaliações são concluídas, os avaliadores irão apresentar as suas conclusões para eliminar duplicidades, e criar uma lista de problemas de usabilidade que devem ser abordados. Estas questões também devem ser separadas quanto a sua gravidade, utilizando algum critério pré-definido de apontamento destas gravidades. Os resultados são geralmente apresentados sob a forma de um relatório, descrevendo o processo de avaliação e uma descrição das recomendações para correção dos problemas encontrados. A conclusão do trabalho se dá por gerar um relatório consolidando as informações de todos os membros da equipe. Sendo recomendado que por meio de uma reunião, todos os avaliadores possam debater as suas análises e verificar possíveis pontos semelhantes que podem ter sido observados por dois ou mais especialistas.

3 LEVANTAMENTO DE DADOS PARA DEFINIÇÃO DE REQUISITOS

O objetivo desta pesquisa é oferecer uma ferramenta de apoio ao método de Avaliação Heurística, contribuindo para a redução dos problemas comumente encontrados por profissionais ao realizarem este método.

Para que fosse possível levantar os requisitos e características propostas para o projeto, foram realizadas pesquisas na literatura científica afim de identificar a metodologia adotado por profissionais que realizam a Avaliação Heurística. As principais fontes de pesquisa foram os indexadores de artigos: IEEE⁴ e Science Direct⁵.

Nas buscas foram utilizadas as seguintes strings: *usability, heuristic, software, heuristic evaluation, usability test*. Para cada um dos indexadores, foram utilizadas estruturas diferentes de acordo com as configurações oferecidas pelos sistemas de busca.

IEEE – A busca no indexador IEEE, seguiu os parâmetros: *usability AND software AND heuristic*, e o resultado foi refinado para artigos publicados entre os anos de 2017 a 2020. Foram encontrados na base de dados o total de 35 artigos, que foram considerados e selecionados por relevância, tendo sido considerados importantes os artigos que apresentavam estudos de caso, sobre Avaliação Heurística.

Science Direct – este indexador mostrou um número mais significativo de artigos. Os parâmetros de strings buscados foram os seguintes: *heuristic evaluation AND Usability Test*, e o resultado foi refinado para artigos publicados entre os anos de 2017 a 2020. Foram encontrados 315 artigos, e para maior refinamento, uma nova busca com outros parâmetros de strings foi realizada: *Heuristic* (no título). Desta forma o indexador retornou 89 artigos.

Dentro do universo de artigos localizados, foi feita uma seleção por relevância do tema, utilizando os seguintes critérios:

- a) avaliação de Heurística feita exclusivamente em software;
- b) abordagens sobre os processos adotados na realização da Avaliação Heurística.

As principais palavras chaves detectadas nos artigos foram: *Usability, Heuristics; Usability Engineering; Heuristic Evaluation; Usability Test; Heuristics Checklist; Human-Computer Interaction (HCI); Usability Study*.

4 <http://ieeexplore.ieee.org>

5 www.sciencedirect.com

4 TRABALHOS RELACIONADOS

4.1 FERRAMENTA MOCKUP DUE (AVALIAÇÃO HEURÍSTICA)

Uma Ferramenta de apoio ao processo de inspeção de usabilidade de *mockups* de aplicações web. Esta ferramenta tem como proposta reduzir o esforço cognitivo dos inspetores ao executarem a avaliação de usabilidade, por fornecer meios para organização e um método de preparar os *mockups* que serão avaliados. Entre os recursos disponíveis, a ferramenta oferece meios de simular a interação da aplicação testada por meio de *links* que fazem a transição entre *mockups*, e gerar relatório dos erros encontrados (RIVERO; VIANA; CONTE).

Disponível de forma gratuita, a Ferramenta Mockup Due, apesar de possuir recursos que visam auxiliar e aprimorar os métodos adotados hoje para Avaliação Heurística, não apresenta todos os recursos identificados neste estudo como exibido no quadro 1.

4.2 HEVA: UMA FERRAMENTA DE SUPORTE À AVALIAÇÃO HEURÍSTICA

A ferramenta HEVA foi criada com o objetivo de servir de ferramenta de auxílio a Avaliação Heurística. É uma ferramenta em forma de *plugin*, para o navegador Firefox. A forma como a HEVA foi desenvolvida, foi encarada por especialistas em avaliação de software como problemática, por limitar a avaliação para aplicações construídas para apenas uma versão de navegador (OEIRAS; BENTOLILA; FIGUEIREDO, 2008).

A HEVA possui ferramentas de auxílio a avaliação que diminuem o esforço manual e utilização de diversas ferramentas para que seja gerado o relatório de Avaliação Heurística, conforme apontado por Oeiras (2008). A HEVA difere da ferramenta proposta neste estudo principalmente pela ausência de recursos para agrupamento das falhas apontados por cada avaliador, a impossibilidade de gerar dados estatísticos após agrupamento das falhas e falta de flexibilidade ao manipular as imagens capturadas para apontamento das heurísticas violadas. Desta forma, mesmo tendo sido desenvolvida para este fim, não possui todas as características e recursos levantados neste estudo para uma ferramenta de auxílio a Avaliação Heurística, conforme exhibe o Quadro 1 (OEIRAS; BENTOLILA; FIGUEIREDO, 2008).

4.3 THEM - TOOL FOR HEURISTIC EVALUATION METHODS

A ferramenta THEM, se apresenta como solução para organização de métodos de avaliação de interface, tendo como principal foco a Avaliação Heurística.

Apresentando recursos para geração do relatório final, e do trabalho colaborativo entre avaliadores. A THEM conta com gerenciamento de usuários, formulários para apontamentos das heurísticas cadastradas, definir os níveis de gravidade da heurística violada, seguindo os padrões adotados por Nielsen (2003). A aplicação limita-se a oferecer os recursos por meio de formulários eletrônicos, onde são inseridas as informações das falhas e heurísticas foram violadas, permitindo anexar uma imagem, quando disponível (SANTOS et al.).

Por não possuir recursos de anotações e marcações nas imagens capturadas do software a ser avaliado, recurso este amplamente utilizado por avaliadores ao realizarem a Avaliação Heurística, os avaliadores precisam utilizar softwares auxiliares para execução do processo. Outros pontos observados no comparativo com o software proposto neste estudo encontram-se no quadro 1. Toda a análise foi feita com base em um artigo, a ferramenta não está disponível para uso ou compra na internet (SANTOS et al.).

4.4 ANÁLISE COMPARATIVA A TRABALHOS RELACIONADOS

Segue análise comparativa entre a ferramenta proposta neste estudo, e as ferramentas propostas em outros estudos:

Quadro 1 — Trabalhos Relacionados (continua)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM
Possibilitar ao avaliador realizar apontamentos e grifos por meio de capturas de tela;	Não Atende. A ferramenta não disponibiliza ferramentas de edição de imagens.	Atende parcialmente. Faz apenas retângulos, e os mesmos não podem ser editados após a marcação.	Não Atende. A ferramenta não disponibiliza ferramentas de edição de imagens.
Definir as heurísticas que serão utilizadas na avaliação do projeto;	Não atende. O Software possui apenas as 10 Heurísticas do Nielsen, e não permite adição de novas.	Atende parcialmente. As heurísticas podem ser cadastradas durante a avaliação, mas não antes de iniciar.	Atende. As heurísticas podem ser cadastradas na aplicação.
Possibilitar ao avaliador informar a heurística violada em cada falha	Atende Parcialmente. O Software permite utilizar apenas as 10	Atende. O sistema é eficaz neste ponto e	Atende. O sistema é eficaz neste ponto e

Quadro 1 — Trabalhos Relacionados (continuação)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM
detectada;	Heurísticas do Nielsen, previamente cadastradas e não permite adição de novas.	também possui uma breve Descrição/ comentário sobre cada heurística.	também possui uma breve Descrição/ comentário sobre cada heurística.
Registrar descrição do erro;	Atende. Permite Adicionar comentários para cada erro apontado.	Atende. É exibido um campo obrigatório logo após o apontamento de um erro.	Atende. Permite adicionar comentários para cada erro apontado.
Indicar a gravidade do erro, na escala de 0 a 4 (NIELSEN; LORANGER, 2007);	Não atende. Não possui um campo específico.	Atende. Para cada erro um campo é exibido, para apontamento do grau. OBS.: usa-se no HEVA os parâmetros de 1 a 5.	Atende. Para cada erro um campo é exibido, com opções de problema cosmético, pequeno ou catástrofe.
Descrever a solução para que a falha possa ser corrigida;	Não atende. Poderá ser feito no campo comentário, mas não possui um campo específico.	Não atende. Não foi identificado no software além do campo de descrição de erro, um campo específico para este fim.	Não atende. Não foi identificado no software além do campo de descrição de erro, um campo específico para este fim.
Possuir o recurso de agrupamento das falhas similares encontradas no mesmo projeto;	Não atende.	Não Atende.	Não Atende.
Possuir a capacidade de gerar dados estatísticos sobre as heurísticas violadas e gravidade dos erros encontrados;	Não atende. Por possuir apenas um campo de descrição, impossibilita que haja qualquer filtro das heurísticas e gravidade dos erros encontrados.	Não Atende. Mesmo possuindo, os apontamentos necessários, não existe a possibilidade de gerar os relatórios.	Atende Parcialmente. Não ficou claro quais recursos de relatórios estão disponíveis.
Suporte a Wireframes, Rabiscoframes ou	Atende. Permite importar	Atende parcialmente. Por ser uma	Atende. Permite importar

Quadro 1 — Trabalhos Relacionados (conclusão)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM
Protótipos;	arquivos com PNG. E possui um recurso de link dinâmico para simular a interação do Protótipo ou Muckups.	aplicação web via browser e faz capturas da área do mesmo, não dá suporte a importação de imagens externas.	arquivos com de imagem.
Entregar o Projeto em formato padrão para impressão.	Atende. Possibilita gerar um arquivo PDF com os dados de falhas e as telas utilizadas para avaliação.	Atende. Gera o relatório em PDF ou RTF, sendo RTF o padrão.	Atende. Possibilita gerar um arquivo PDF com os dados de falhas e as telas utilizadas para avaliação.

Fonte: Elaborado pelo autor (2021).

Todos os requisitos levantados no Quadro 1 para comparação entre as ferramentas analisadas no quadro, levam em consideração o que especialistas em usabilidade buscam durante o processo de Avaliação Heurística (NIELSEN, 1994). Conforme levantamento exposto neste trabalho, a ferramenta proposta irá preencher toda esta lacuna identificada como essencial para a melhora no processo de Avaliação por meio de Heurística.

5 APRESENTAÇÃO DOS REQUISITOS

Com base no levantamento realizado, conforme descrito na seção 3, foi identificado um conjunto de características relevantes que foram utilizadas para a definição dos requisitos do software proposto. O objetivo deste estudo é propor o desenvolvimento de uma ferramenta de software que possa auxiliar o processo de avaliação de usabilidade por meio de heurísticas.

Após análise dos métodos utilizados por especialistas em usabilidade na realização da Avaliação Heurística, foram julgados como requisitos e características relevantes para o software:

- Possibilitar ao avaliador realizar apontamentos e grifos por meio de capturas de tela (OEIRAS; BENTOLILA; FIGUEIREDO, 2008; NIELSEN; MOLICH, 1990)
- Definir as heurísticas que serão utilizadas na avaliação do projeto (OEIRAS; BENTOLILA; FIGUEIREDO, 2008; SANTOS et al.)
- Possibilitar ao avaliador informar a heurística violada em cada falha detectada (NIELSEN; MOLICH, 1990)(OEIRAS; BENTOLILA; FIGUEIREDO, 2008; NIELSEN; MOLICH, 1990) .
- Realizar descrição do erro (NIELSEN, 1994; OEIRAS; BENTOLILA; FIGUEIREDO, 2008; RIVERO; VIANA; CONTE)
- Indicar a gravidade do erro utilizando uma escala (NIELSEN; MOLICH, 1990; PAZ et al., 2015);
- Descrever uma possível solução para a falha detectada (OEIRAS; BENTOLILA; FIGUEIREDO, 2008; RIVERO; VIANA; CONTE)
- Possuir o recurso de agrupamento das falhas similares encontradas no mesmo projeto (OEIRAS; BENTOLILA; FIGUEIREDO, 2008; RIVERO; VIANA; CONTE);
- Possuir a capacidade de gerar dados estatísticos sobre as heurísticas violadas e gravidade dos erros encontrados (SANTOS et al.; NIELSEN, 2020)
- Suporte a Wireframes, Rabiscoframes ou Protótipos (HVANNBERG; LAW; LÉRUSDÓTTIR, 2007);
- Entregar o Projeto em formato padrão para impressão (OEIRAS; BENTOLILA; FIGUEIREDO, 2008; N. SOKEN et al.).

A meta consiste em criar um software que aprimore significativamente a experiência dos especialistas em usabilidade, simplificando e tornando mais eficiente o processo. O objetivo é alcançar agilidade, clareza, organização e padronização nos relatórios de avaliação, eliminando desafios apontados pelos avaliadores, como a potencial subjetividade na identificação de erros.

O desenvolvimento de uma ferramenta com os recursos citados, trará a

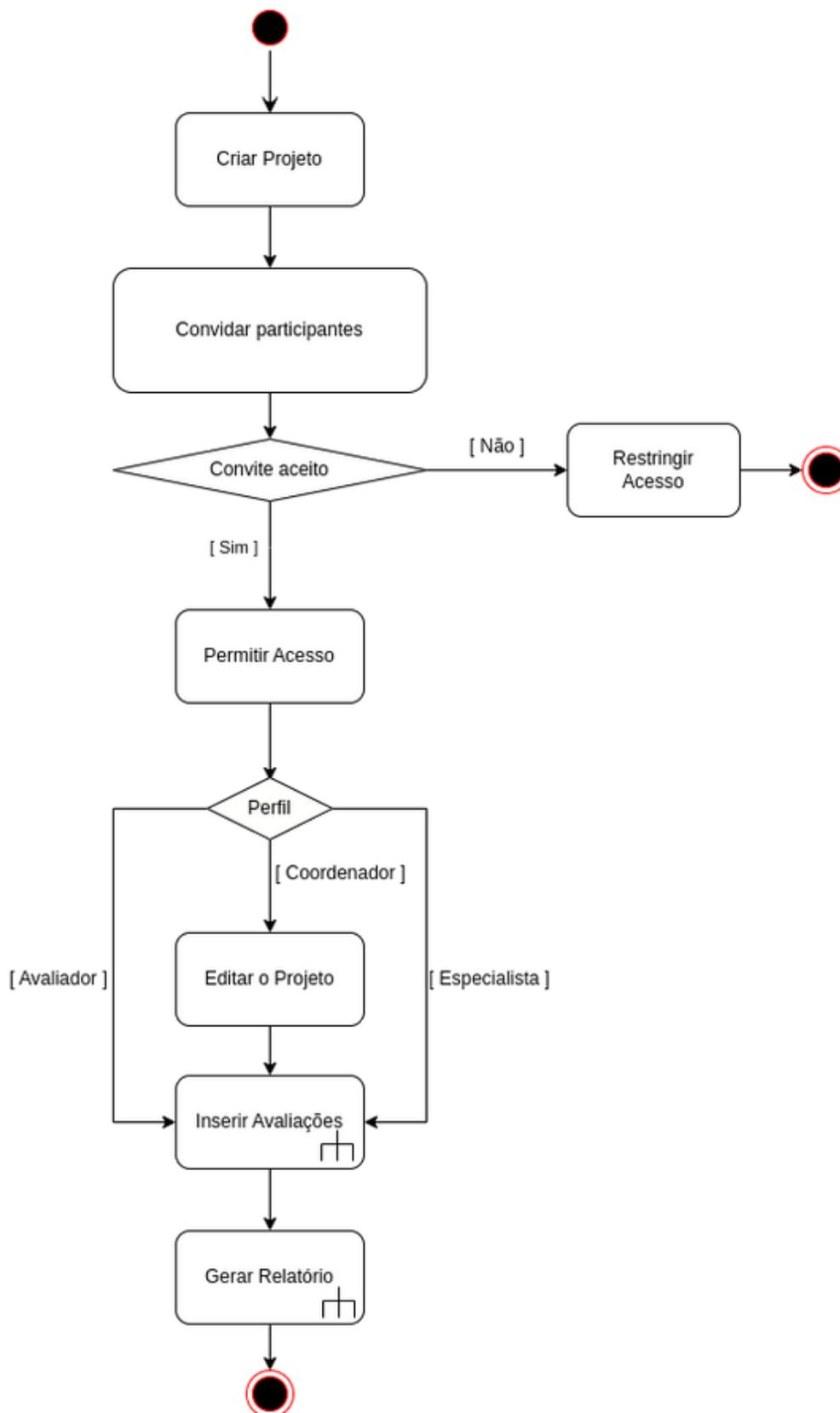
possibilidade de centralizar todo o processo de avaliação para uma aplicação específica. Desta forma o uso de ferramentas proprietárias, como editores de texto e editores de imagens, não seria mais necessário. Trazendo além da economia na compra de licenças, a possibilidade de ter recursos pensados e desenvolvidos especificamente voltados para este fim. Evitar o uso de vários aplicativos no processo de avaliação, trará também mais agilidade para a tarefa, sem a necessidade de navegar por vários aplicativos os avaliadores poderão focar sua atenção no processo mais importante que é identificar os problemas de usabilidade detectados.

6 SOFTWARE DESENVOLVIDO

O diagrama de atividades é uma ferramenta poderosa para a modelagem de processos de negócio. No contexto desta pesquisa, foi desenvolvido um software de avaliação heurística que permite aos usuários criar projetos de avaliação de usabilidade. O processo de criação de projetos é fundamental para a utilização do software, e por isso foi representado em um diagrama de atividades.

O diagrama de atividades de criação de projetos no sistema de avaliação heurística é composto por uma sequência de atividades que devem ser executadas pelos usuários. Inicialmente, o usuário deve acessar a funcionalidade de criação de projetos na interface do software, como mostra a Figura 1. Em seguida, o usuário deve inserir os dados do projeto, como o nome do projeto, a descrição, as heurísticas que serão utilizadas na avaliação, datas para início e fim do projeto e algumas informações relevantes sobre o cliente e o produto avaliado, como mostra a Figura 2. O usuário também deve definir os avaliadores que serão responsáveis pela avaliação dos softwares selecionados.

Figura 1 — Diagrama de Criar Projetos



Fonte: Elaborado pelo autor (2023).

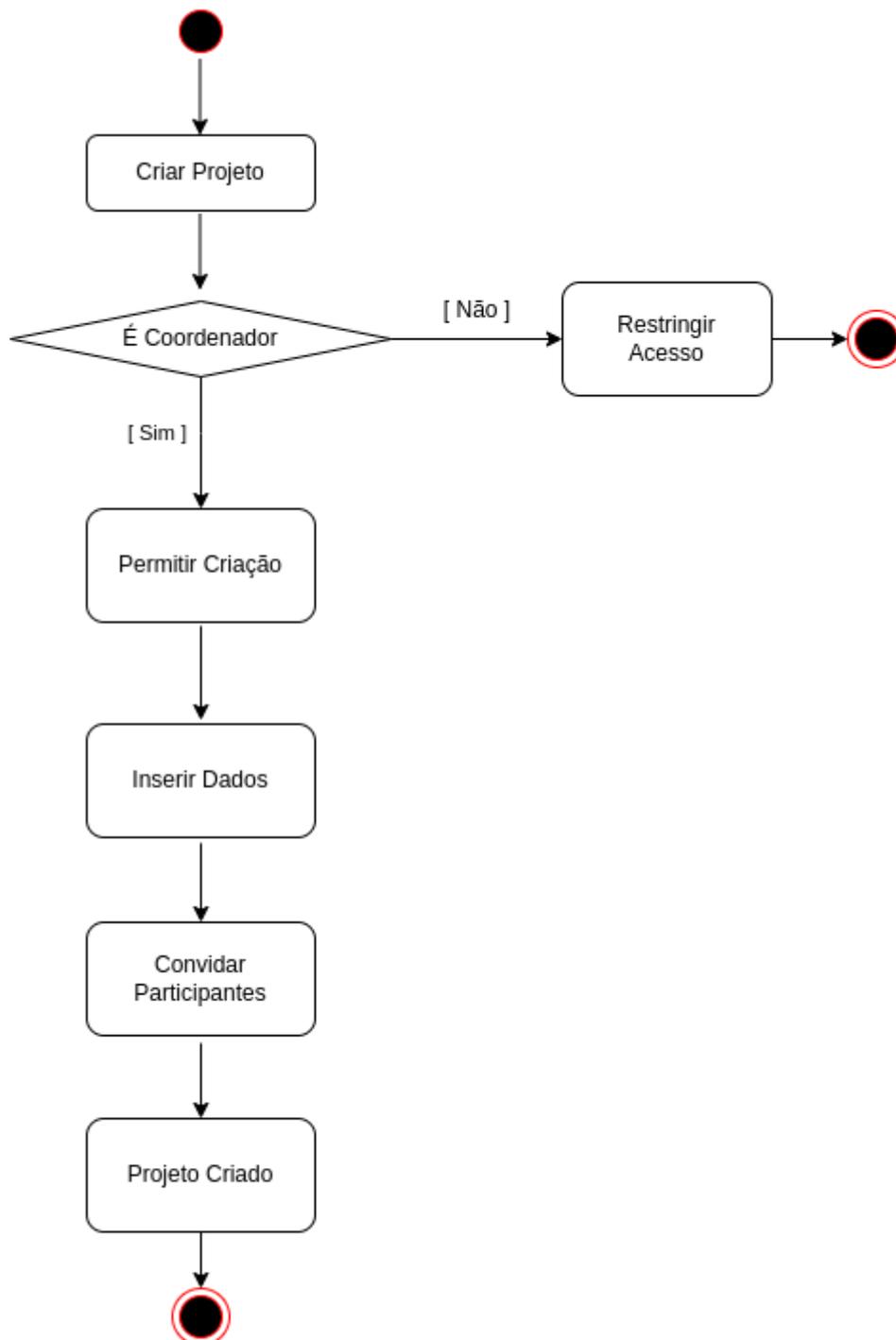
Além do diagrama de atividades para a criação de projetos, outro diagrama

importante desenvolvido no sistema de avaliação heurística é o de inserção de avaliações. Esse diagrama tem como objetivo demonstrar o fluxo de inserção das avaliações realizadas pelos avaliadores nos projetos criados.

No diagrama, é possível observar as atividades necessárias para inserir uma avaliação. Esta etapa do processo é descrita de forma clara e objetiva, com os possíveis desvios e exceções sendo também considerados.

Com a criação desse diagrama, os usuários do sistema de avaliação heurística podem entender melhor como inserir as avaliações realizadas, tornando o processo mais eficiente e fácil de ser realizado. Além disso, o diagrama serve como um registro visual das atividades envolvidas na inserção de avaliações, possibilitando uma melhor análise e melhoria do processo no futuro.

Figura 2 — Diagrama de Inserir Avaliações



Fonte: Elaborado pelo autor (2023).

Outro diagrama importante desenvolvido neste projeto é o da geração do relatório consolidado de um projeto de avaliação heurística. Este diagrama tem como objetivo mostrar como é gerado um relatório com as informações coletadas durante a avaliação, de forma organizada e clara para os usuários.

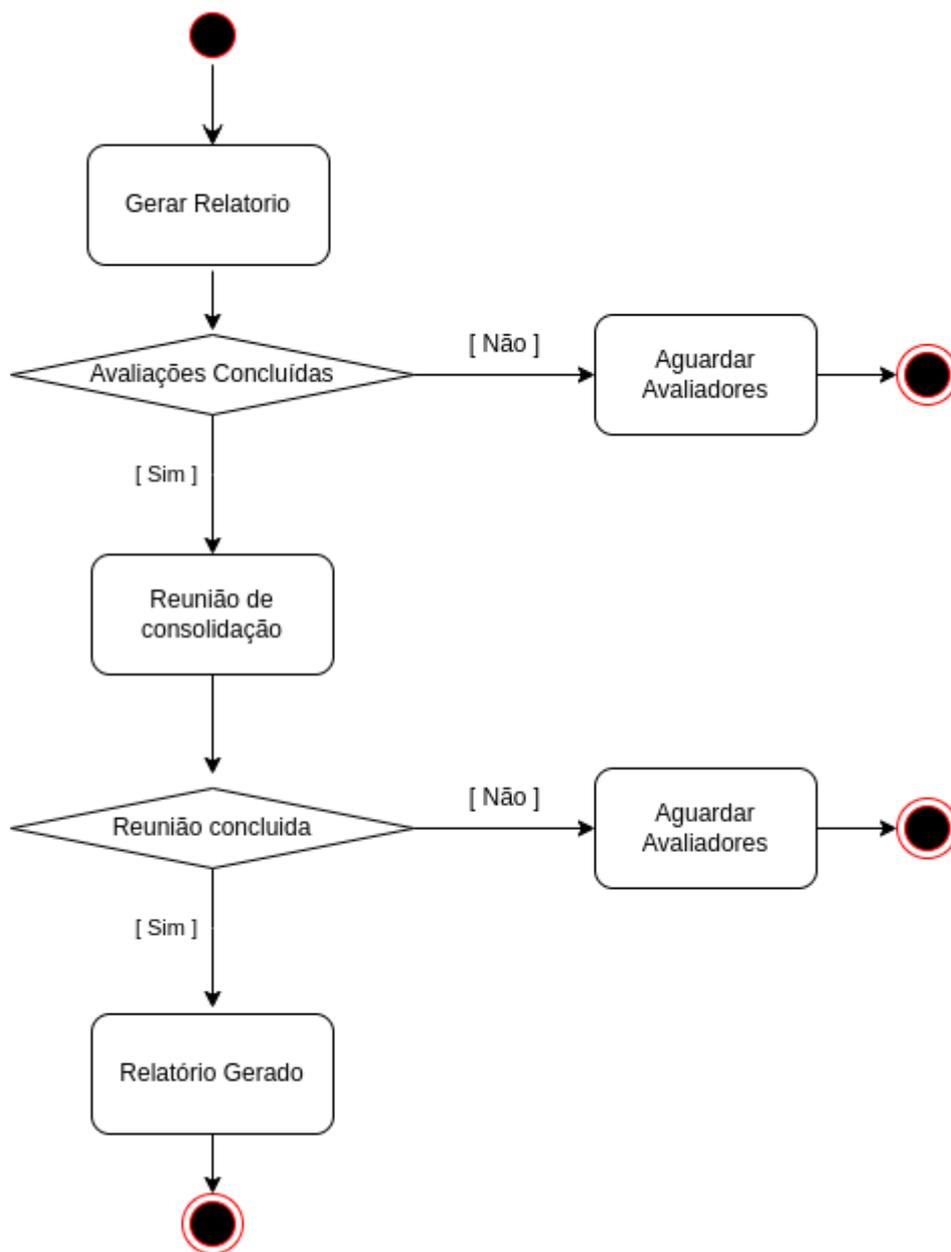
O processo de geração do relatório começa com a coleta de dados das avaliações realizadas em cada interface do software, realizada individualmente por

cada avaliador participante. Em seguida, esses dados são compilados e organizados em um formato padrão, que inclui informações como as heurísticas violadas, o grau de gravidade de cada problema encontrado, sugestões de melhorias, entre outras.

Depois disso, o sistema gera um relatório consolidado que é disponibilizado aos usuários, podendo ser visualizado no próprio sistema ou exportado para outros formatos, como PDF, como mostra a Figura 3. Esse relatório é uma importante ferramenta para os desenvolvedores, pois apresenta de forma clara e objetiva as áreas que precisam ser melhoradas no software, permitindo que os problemas sejam corrigidos de maneira eficiente.

Com o uso desse software, espera-se que a geração do relatório de uma avaliação heurística seja simplificada e padronizada, permitindo que os desenvolvedores possam corrigir os problemas de usabilidade de forma mais rápida e eficiente.

Figura 3 — Diagrama de Geração de Relatório



Fonte: Elaborado pelo autor (2023).

6.1 TECNOLOGIAS UTILIZADAS

No âmbito do desenvolvimento do software, a seleção das tecnologias adotadas inclui:

- Frontend: JavaScript, React
- API: Node
- Banco de Dados: MongoDB

A escolha destas tecnologias obedeceu a uma análise baseada nos seguintes critérios:

Documentação Disponível: todas as tecnologias escolhidas dispõem de documentação completa e atualizada, simplificando tanto o processo de desenvolvimento quanto de aprendizagem.

Licença de Uso Livre: cada uma das tecnologias selecionadas é caracterizada por possuir licença de uso livre, o que contribui para a redução dos custos de desenvolvimento.

Adequação para Aplicações Web: todas as tecnologias selecionadas demonstram ser apropriadas para o desenvolvimento de aplicações web, atendendo, assim, aos requisitos inerentes ao projeto.

Conhecimento Prévio e Curva de Aprendizado: vale destacar que o autor deste projeto já possui domínio prévio de todas as tecnologias selecionadas, fator que encurta a curva de aprendizado e acelera o processo de desenvolvimento.

A seguir, realizam-se avaliações mais detalhadas das tecnologias eleitas:

JavaScript: Trata-se de uma linguagem de programação de propósito geral, amplamente empregada no desenvolvimento web, favorecida pela disponibilidade de uma vasta gama de bibliotecas auxiliares que simplificam o desenvolvimento de funcionalidades complexas.

React: Este *framework* JavaScript foi escolhido para a construção de interfaces de usuário dinâmicas, uma exigência central no projeto. Além disso, o React destaca-se por possuir documentação completa e ser amplamente adotado no desenvolvimento de aplicações web.

Node: Esta tecnologia, como um ambiente de execução JavaScript no servidor, proporciona uma flexibilidade notável no desenvolvimento de APIs, adaptando-se de maneira eficaz aos requisitos específicos do projeto em consideração. Em um sistema que faz uso dessa tecnologia, o Node desempenha um papel crucial, viabilizando a execução eficiente de operações no servidor, o que é fundamental para o funcionamento adequado do sistema como um todo.

MongoDB: Este banco de dados NoSQL armazena informações em formato de documentos JSON, possui licença de uso livre e apresenta excelente desempenho no tratamento de imagens.

A seleção das tecnologias mencionadas pautou-se em uma abordagem criteriosa, visando a satisfazer as demandas do projeto. É importante notar que, além das tecnologias escolhidas, outras opções poderiam ter sido consideradas para o desenvolvimento do software. Algumas delas incluem:

Frontend:

Angular: Outro framework JavaScript destinado à construção de interfaces de usuário, mais adequado para projetos de maior porte ou que demandam uma curva de aprendizado mais suave.

Vue.js: Um framework JavaScript leve e flexível, particularmente indicado para projetos que requerem alto desempenho ou compatibilidade com dispositivos móveis.

API:

Django: Um framework em Python voltado para o desenvolvimento de APIs, sobretudo adequado para projetos que exigem alta performance ou operam em ambientes de produção.

Flask: Trata-se de um microframework Python para o desenvolvimento de APIs, sendo uma escolha apropriada para projetos de menor porte ou ambientes de desenvolvimento.

Banco de Dados:

MySQL: Um banco de dados relacional popular, ideal para projetos que demandam alta performance ou necessitam de integração com sistemas legados.

PostgreSQL: Um banco de dados relacional de código aberto, recomendado para projetos que requerem elevados níveis de segurança ou que lidam com grandes volumes de dados.

A determinação da tecnologia mais adequada envolve uma análise comparativa abrangente, levando em consideração os requisitos do projeto, as preferências do desenvolvedor e os custos associados. É crucial realizar essa análise a fim de assegurar a escolha mais apropriada para o contexto específico do projeto.

Vantagens e Desvantagens de Cada Opção

Ademais aos critérios supracitados, é crucial contemplar as vantagens e desvantagens inerentes a cada tecnologia selecionada. Por exemplo, o Angular

revela-se uma opção sólida para projetos robustos, devido à sua considerável comunidade de desenvolvedores e à riqueza de recursos disponíveis. Entretanto, é importante notar que o Angular pode apresentar complexidades e representar um desafio no processo de aprendizagem.

O Django, por sua vez, destaca-se em projetos que requerem desempenho elevado, visto que é um framework otimizado para tal. Contudo, sua curva de aprendizado pode ser mais íngreme quando comparada à do Flask.

O MySQL é uma escolha acertada para projetos que necessitam de integração com sistemas legados, em virtude de sua ampla adoção. Entretanto, em termos de segurança, pode ser considerado menos robusto que o PostgreSQL.

6.2 CARACTERÍSTICAS DO SOFTWARE DESENVOLVIDO

O software gerado por meio desta pesquisa apresenta uma série de características essenciais que visam aprimorar o processo de Avaliação Heurística.

Está representada na Figura 4 a tela principal de inserção das avaliações heurísticas.

Figura 4 — Tela Principal

LOGO AQUI

Home Criar Projeto Em Andamento Finalizados

Heurística Violada: A

Grau do erro: B

Baixo Médio Alto

Descreva aqui o erro: C

Sugestões de melhoria: D

Imagem (envie uma imagem do erro para mais detalhes): E

Clique aqui para enviar uma imagem

Opções

G

Projeto: Teste nome

Produto: Teste nome

Data inicial: 01/05/2023 Data final: 01/06/2023

H

Salvar

Outros reports: F

1 2 3

Fonte: Elaborado pelo autor (2023).

- A. Apontamento de heurística violada
- B. Apontar gravidade do erro identificado
- C. Campo para uma breve descrição do erro identificado
- D. Campo para apresentar uma sugestão de ajuste
- E. Inserção da Imagem apresentando o erro
- F. Seletor para a lista de erros já apontados
- G. Ferramentas para edição da imagem
- H. Informações sobre o projeto

A seguir, será destacado cada uma dessas características desenvolvidas nesta versão, ressaltando a importância e a contribuição de cada uma delas para o

sucesso da ferramenta:

1. Realização de Apontamentos e Grifos por Meio de Capturas de Tela

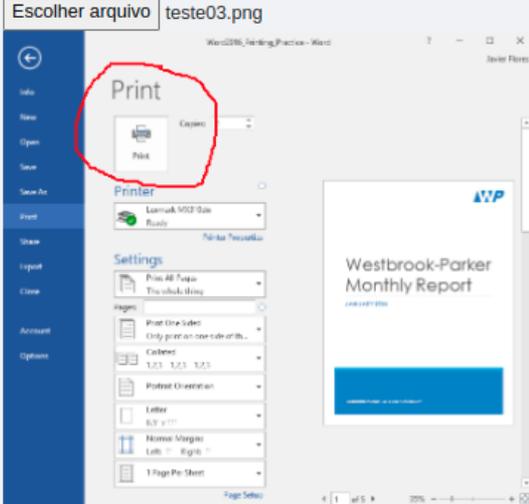
A funcionalidade que permite aos avaliadores realizar apontamentos e grifos diretamente nas capturas de tela representa uma maneira eficaz de documentar e comunicar problemas de usabilidade. Essa característica facilita a identificação e descrição de falhas de maneira visual e contextual, tornando o processo de avaliação mais preciso e detalhado. Figura 5.

Esta funcionalidade foi adicionado com a combinação de implementações na API, conforme o Quadro 2e na aplicação de front-end, conforme o Quadro 3.

Figura 5 — Página de registro de Falhas

Home Criar Projeto Andamento Finalizados Admin Adeilton Sousa Sair ↓

Escolher arquivo teste03.png



Print

Printer: LaserJet M1020 Ready

Settings

Print All Pages

Print One Page

Print Range

Print Orientation

Letter

Normal Margins

1 Page Per Sheet

Westbrook-Parker Monthly Report

Cor: Largura da linha:

Modo Carimbo de Seta

Heurística Violada:

Descrição do erro:

Gravidade do Erro: Baixa Média Alta

Sugestão de Ajuste:

Salvar Novo Item

1

Cliente: Nielsen
Projeto: Nutes 002
Produto: Proj Do Daniel Portal do aluno 002
Data Inicial: 27/09/2023, 13:57:40
Data Final: 21/09/2023, 13:57:51

Fonte: Elaborado pelo autor (2023).

Quadro 2 — API - Apontamento de Erros

```
async addItemEvaluation(request: Request, response: Response) {
  const { id } = request.body;
  const newItem = {
    idItem: short(),
    violatedHeuristic: "",
    errorDescription: "",
    errorSeverity: "1",
    adjustmentSuggestion: "",
    imageItem: "",
  };
  try {
    if (id) {
      await Project.updateOne(
        { _id: id },
        { $addToSet: { evaluations: newItem } }
      );
    }
    return response.status(200).send(newItem.idItem);
  } catch (error) {
    return response.status(500).send({
      error: "Id passado não foi encontrado",
      message: "Projeto não encontrado",
    });
  }
}
```

Fonte: Elaborado pelo autor (2023).

Quadro 3 — Editor de Imagens

```
const CanvasEditor = ({ imageSrc }) => {
  const canvasRef = useRef(null);
  const [drawing, setDrawing] = useState(false);
  const [context, setContext] = useState(null);

  useEffect(() => {
    const canvas = canvasRef.current;
    const ctx = canvas.getContext('2d');
    setContext(ctx);
    const image = new Image();
    image.src = imageSrc;

    image.onload = () => {
      ctx.drawImage(image, 0, 0, canvas.width, canvas.height);
    };
  }, [imageSrc]);

  const startDrawing = (e) => {
    setDrawing(true);
    context.strokeStyle = 'black';
    context.lineWidth = 2;
    context.lineCap = 'round';
    context.beginPath();
    const { offsetX, offsetY } = e.nativeEvent;
    context.moveTo(offsetX, offsetY);
  };

  const endDrawing = () => {
    setDrawing(false);
    context.closePath();
  };

  const draw = (e) => {
    if (!drawing) return;

    const { offsetX, offsetY } = e.nativeEvent;
    context.lineTo(offsetX, offsetY);
    context.stroke();
  };

  return (
    <div>
      <canvas
        ref={canvasRef}
        width={600}
        height={400}
        onMouseDown={startDrawing}
        onMouseUp={endDrawing}
        onMouseMove={draw}
      />
    </div>
  );
};

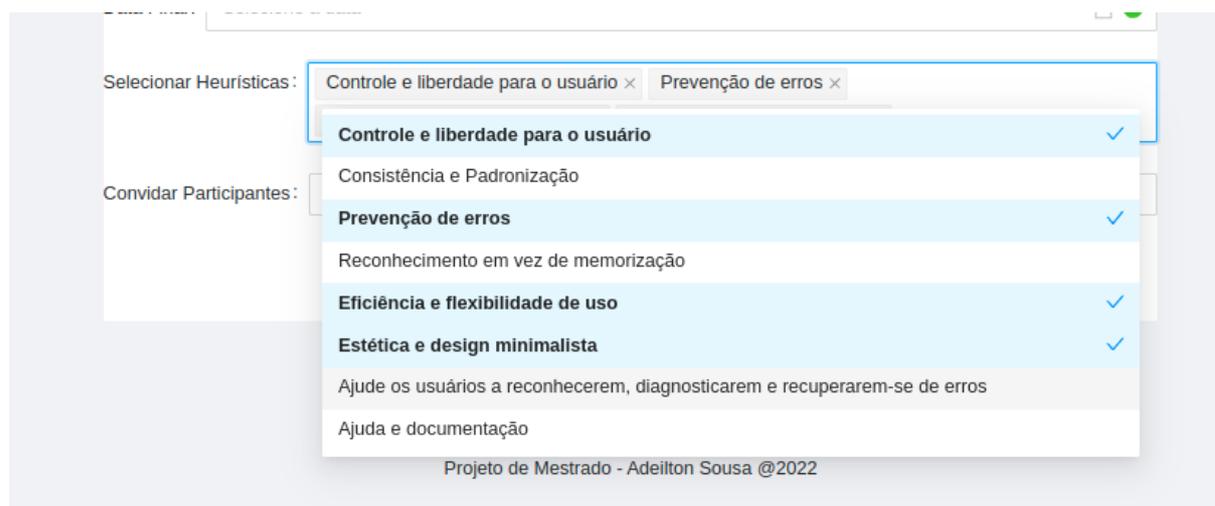
export default CanvasEditor;
```

Fonte: Elaborado pelo autor (2023).

2. Definição das Heurísticas Utilizadas na Avaliação do Projeto

A capacidade de definir as heurísticas a serem utilizadas na avaliação é fundamental para adaptar a ferramenta a diferentes contextos e necessidades de projetos, conforme Figura 6. Essa característica oferece flexibilidade aos avaliadores, permitindo que eles selecionem conjuntos de heurísticas específicos de acordo com os critérios de avaliação estabelecidos.

Figura 6 — Selecionando Heurísticas para o projeto



Fonte: Elaborado pelo autor (2023).

3. Informação da Heurística Violada em Cada Falha Detectada

A possibilidade de informar a heurística violada em cada falha detectada simplifica o processo de classificação e priorização de problemas de usabilidade. Essa característica auxilia os avaliadores a identificar de forma clara quais heurísticas estão relacionadas a cada erro, fornecendo uma base sólida para futuras correções. Figura 7.

Figura 7 — Apontar Heurística Violada

Heurística Violada: **Visibilidade do Status do Sistema** ▼

Descrição do erro: **Visibilidade do Status do Sistema**

Gravidade do Erro: **Compatibilidade entre o sistema e o mundo real**

Sugestão de Ajuste: **Compatibilidade entre o sistema e o mundo real**

Compatibilidade entre o sistema e o mundo real

Consistência e Padronização

Prevenção de erros

Reconhecimento em vez de memorização

Eficiência e flexibilidade de uso

Estética e design minimalista

Salvar Novo Item

Fonte: Elaborado pelo autor (2023).

4. Descrição Detalhada dos Erros

A inclusão de uma descrição detalhada dos erros identificados é crucial para uma comunicação eficaz com as equipes de desenvolvimento. Essa característica permite que os avaliadores forneçam informações específicas sobre cada problema, facilitando o entendimento e a resolução por parte dos desenvolvedores.

5. Indicação da Gravidade dos Erros

A capacidade de indicar a gravidade dos erros em uma escala de 0 a 2 fornece uma maneira objetiva de avaliar a importância de cada problema de usabilidade. Essa característica auxilia na priorização das correções, garantindo que os problemas mais críticos sejam abordados prioritariamente. Figura 8.

Figura 8 — Indicação de Gravidade do erro

Gravidade do Erro: Baixa Média Alta

Fonte: Elaborado pelo autor (2023).

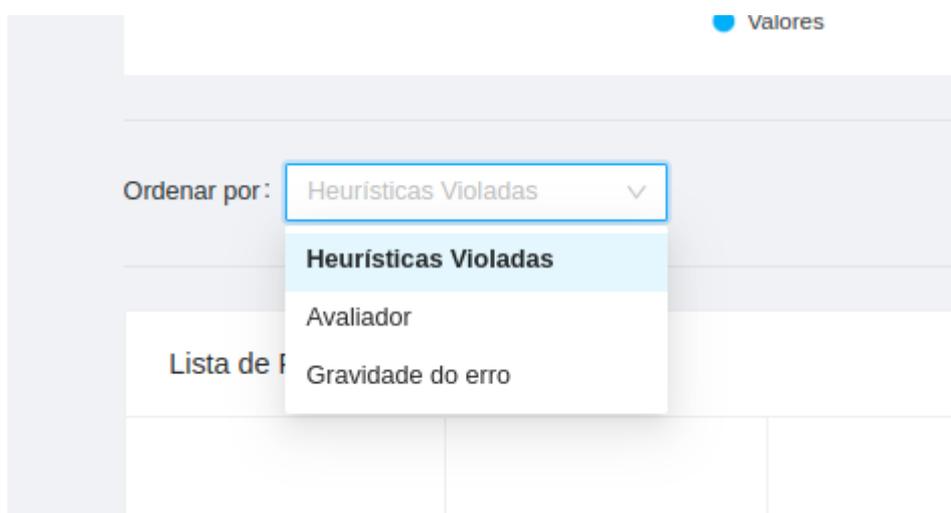
6. Sugestão de Possíveis Soluções para as Falhas Detectadas

A inclusão de sugestões de possíveis soluções para as falhas detectadas representa uma valiosa contribuição para o processo de Avaliação Heurística. Essa característica fornece orientações aos desenvolvedores sobre como corrigir os problemas identificados, agilizando o processo de melhoria da usabilidade.

7. Agrupamento de Falhas Similares no Mesmo Projeto

O recurso de agrupamento de falhas similares encontradas no mesmo projeto simplifica a análise e a comunicação de problemas recorrentes. Essa característica permite que os avaliadores identifiquem tendências e padrões de usabilidade, facilitando a tomada de decisões informadas para aprimorar o software. Figura 9.

Figura 9 — Ordenando Avaliações



Fonte: Elaborado pelo autor (2023).

8. Geração de Dados Estatísticos sobre Heurísticas Violadas e Gravidade dos Erros

A capacidade de gerar dados estatísticos sobre as heurísticas violadas e a gravidade dos erros encontrados fornece uma visão global do desempenho do produto testado em termos de usabilidade, conforme Figura 10e Quadro 4. Essa característica permite que as equipes de desenvolvimento acompanhem o progresso das avaliações e identifiquem áreas de melhoria.

Figura 10 — Gráficos Estatísticos



Fonte: Elaborado pelo autor (2023).

Quadro 4 — Gerando gráficos

```
function BarChart() {
  useEffect(() => {
    const data = [
      { name: 'Gravidade Baixa', y: 8 },
      { name: 'Gravidade Média', y: 5 },
      { name: 'Gravidade Alta', y: 3 },
    ]

    const options = {
      chart: {
        type: 'bar',
      },
      title: {
        text: 'Estáticas de Gravidades dos erros',
      },
      xAxis: {
        categories: data.map((item) => item.name),
      },
      yAxis: {
        title: {
          text: 'Número de Avaliações',
        },
      },
      series: [
        {
          name: 'Valores',
          data: data.map((item) => item.y),
        },
      ],
    }

    Highcharts.chart('bar-chart-container', options)
  }, [])

  return (
    <div id="bar-chart-container" style={{ width: '100%', height: '400px' }}>
      </div>
  )
}

export default BarChart
```

Fonte: Elaborado pelo autor (2023).

9. Entrega do Projeto em Formato Padrão para Impressão

A entrega do projeto em formato padrão para impressão garante que os resultados da Avaliação Heurística sejam apresentados de maneira clara e profissional. Essa característica facilita a comunicação dos resultados com

stakeholders e equipes de desenvolvimento, promovendo a compreensão e ação eficaz. Quadro 5.

Quadro 5 — Comparativo - Trabalhos Relacionados (continua)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM	FERRAMENTA DESTE PROJETO
Possibilitar ao avaliador realizar apontamentos e grifos por meio de capturas de tela;	Não Atende. A ferramenta não disponibiliza ferramentas de edição de imagens.	Atende parcialmente. Faz apenas retângulos, e os mesmos não podem ser editados após a marcação.	Não Atende. A ferramenta não disponibiliza ferramentas de edição de imagens.	Atende. A ferramenta possui o recurso de fazer grifos em cada imagem.
Definir as heurísticas que serão utilizadas na avaliação do projeto;	Não atende. O Software possui apenas as 10 Heurísticas do Nielsen, e não permite adição de novas.	Atende parcialmente. As heurísticas podem ser cadastradas durante a avaliação, mas não antes de iniciar.	Atende. As heurísticas podem ser cadastradas na aplicação.	Atende. As heurísticas podem ser cadastradas na aplicação.
Possibilitar ao avaliador informar a heurística violada em cada falha detectada;	Atende Parcialmente. O Software permite utilizar apenas as 10 Heurísticas do Nielsen, previamente cadastradas e não permite adição de novas.	Atende. O sistema é eficaz neste ponto e também possui uma breve Descrição/ comentário sobre cada heurística.	Atende. O sistema é eficaz neste ponto e também possui uma breve Descrição/ comentário sobre cada heurística.	Atende. O sistema é eficaz neste ponto e também possui uma breve Descrição/ comentário sobre cada heurística.
Registrar descrição do erro;	Atende. Permite Adicionar comentários para cada erro	Atende. É exibido um campo obrigatório logo após o	Atende. Permite adicionar comentários para cada erro	Atende. Permite Adicionar comentários para cada erro

Quadro 5 — Comparativo - Trabalhos Relacionados (continuação)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM	FERRAMENTA DESTE PROJETO
	apontado.	apontamento de um erro.	apontado.	apontado.
Indicar a gravidade do erro, na escala de 0 a 4 (NIELSEN; LORANGER, 2007);	Não atende. Não possui um campo específico.	Atende. Para cada erro um campo é exibido, para apontamento do grau. OBS.: usa-se no HEVA os parâmetros de 1 a 5.	Atende. Para cada erro um campo é exibido, com opções de problema cosmético, pequeno ou catástrofe.	Atende. Para cada erro um campo é exibido, para apontamento do grau. Ficou definido como: Baixa, Média e Alta
Descrever a solução para que a falha possa ser corrigida;	Não atende. Poderá ser feito no campo comentário, mas não possui um campo específico.	Não atende. Não foi identificado no software além do campo de descrição de erro, um campo específico para este fim.	Não atende. Não foi identificado no software além do campo de descrição de erro, um campo específico para este fim.	Atende. Para cada erro um campo é exibido, para apontamento de uma sugestão de solução para a falha.
Possuir o recurso de agrupamento das falhas similares encontradas no mesmo projeto;	Não atende.	Não Atende.	Não Atende.	Atende. No relatório final, possui filtros que podem ser usados para agrupar, por heurística, por grau de severidade e por avaliador.
Possuir a capacidade de gerar dados estatísticos sobre as heurísticas violadas e gravidade dos erros encontrados;	Não atende. Por possuir apenas um campo de descrição, impossibilita que haja qualquer filtro das heurísticas e	Não Atende. Mesmo possuindo, os apontamentos necessários, não existe a possibilidade gerar os relatórios.	Atende Parcialmente. Não ficou claro quais recursos de relatórios estão disponíveis.	Atende. Possui gráficos que mostram quais as Heurísticas mais violadas, e outro com estatísticas de gravidade dos erros

Quadro 5 — Comparativo - Trabalhos Relacionados (conclusão)

CARACTERÍSTICAS DESEJÁVEIS	FERRAMENTA MOCKUP DUE	FERRAMENTA HEVA	FERRAMENTA THEM	FERRAMENTA DESTE PROJETO
	gravidade dos erros encontrados.			encontrados;
Suporte a Wireframes, Rabiscoframes ou Protótipos;	Atende. Permite importar arquivos com PNG. E possui um recurso de link dinâmico para simular a interação do Protótipo ou Muckups.	Atende parcialmente. Por ser uma aplicação web via browser e faz capturas da área do mesmo, não dá suporte a importação de imagens externas.	Atende. Permite importar arquivos com de imagem.	Atende. Permite importar arquivos com de imagem.
Entregar o Projeto em formato padrão para impressão.	Atende. Possibilita gerar um arquivo PDF com os dados de falhas e as telas utilizadas para avaliação.	Atende. Gera o relatório em PDF ou RTF, sendo RTF o padrão.	Atende. Possibilita gerar um arquivo PDF com os dados de falhas e as telas utilizadas para avaliação.	Atende. Possibilita gerar um arquivo PDF com os dados de falhas e as telas utilizadas para avaliação.

Fonte: Elaborado pelo autor (2023).

As características citadas acima, que foram previamente identificadas nesta pesquisa estão integralmente incorporadas ao software desenvolvido, demonstrando o empenho em otimizar o processo de Avaliação Heurística, conferindo-lhe maior eficiência, precisão e adaptabilidade para atender às variadas demandas inerentes aos projetos de Avaliação Heurística.

7 CONCLUSÃO

Este estudo explorou a Avaliação Heurística como um método eficiente e de baixo custo, concebido para analisar a conformidade dos softwares com um conjunto de heurísticas destinadas a simplificar a experiência do usuário e melhorar a interação homem-máquina, como proposto por Jakob Nielsen e Rolf Molich em 1990. Durante a pesquisa, identificamos desafios associados a esse método, incluindo a possibilidade de viés por parte dos avaliadores e a falta de ferramentas padronizadas para auxiliar avaliações com múltiplos avaliadores.

Com base nas necessidades identificadas, este estudo se propôs a desenvolver uma ferramenta de software composta por um conjunto de recursos destinados à Avaliação Heurística. Os objetivos gerais e específicos delineados neste trabalho buscaram compreender as necessidades dos profissionais de usabilidade, mapear métodos, softwares e ferramentas utilizados nas avaliações e identificar maneiras de simplificar a consolidação dos relatórios finais.

Ao longo da pesquisa, foi alcançado com sucesso uma série de objetivos específicos que contribuíram para a melhoria do processo de Avaliação Heurística:

- Implementamos a funcionalidade que permite a realização de apontamentos e grifos diretamente nas capturas de tela, tornando a documentação de problemas de usabilidade mais eficaz e contextual.
- Incorporamos a capacidade de definir heurísticas específicas para cada projeto, oferecendo flexibilidade aos avaliadores para adaptar a ferramenta a diferentes contextos.
- Possibilitamos a informação da heurística violada em cada falha detectada, simplificando a classificação e priorização de problemas de usabilidade.
- Incluímos a função de fornecer descrições detalhadas dos erros identificados, facilitando a comunicação eficaz com as equipes de desenvolvimento.
- Implementamos a capacidade de indicar a gravidade dos erros em uma escala de 0 a 2, auxiliando na priorização das correções.
- Oferecemos sugestões de possíveis soluções para as falhas detectadas, agilizando o processo de melhoria da usabilidade.
- Introduzimos o recurso de agrupamento de falhas similares encontradas no mesmo projeto, simplificando a análise e comunicação de problemas recorrentes.
- Desenvolvemos a capacidade de gerar dados estatísticos sobre heurísticas violadas e a gravidade dos erros, fornecendo uma visão global do desempenho do produto em termos de usabilidade.

- Agora é possível entregar os resultados da Avaliação Heurística em um formato padrão para impressão, promovendo uma apresentação clara e profissional dos resultados aos stakeholders e equipes de desenvolvimento. Porém não houve o tempo hábil para avaliar os resultados obtidos, bem como, percebe-se que funcionalidades precisam ser aprimoradas, tais como:
 - Realizar testes com usuários para avaliar a usabilidade da ferramenta.
 - Avaliar a eficácia da ferramenta em diferentes contextos de uso.
 - Incorporar novos recursos e funcionalidades para atender a necessidades específicas de usuários.

7.1 TRABALHOS FUTUROS

A pesquisa e desenvolvimento deste projeto trouxeram contribuições significativas para apoiar a Avaliação Heurística. No entanto, há várias áreas que podem ser exploradas para futuras melhorias e expansões. Abaixo, são apresentadas algumas sugestões para trabalhos futuros:

- Perfis de Coordenador e Avaliador Convidado: A implementação de diferentes perfis de usuários, como coordenadores de avaliação e avaliadores convidados, representa um avanço significativo na gestão das avaliações de usabilidade. Isso permitiria a personalização de permissões específicas para cada perfil, possibilitando um acompanhamento mais eficaz do progresso das avaliações. Os coordenadores teriam acesso a ferramentas de gestão e coordenação de avaliações, enquanto os avaliadores convidados poderiam contribuir de maneira específica para avaliações designadas. Essa diferenciação de papéis promoveria uma colaboração mais eficiente entre os membros da equipe de avaliação.
- Sistema de Convites e Notificações: A introdução de um sistema de convites e notificações por e-mail representa uma maneira eficaz de simplificar a colaboração entre os participantes da avaliação. Essa funcionalidade permitiria aos coordenadores convidar avaliadores para participar de avaliações específicas e notificá-los sobre novas atribuições ou atualizações. Além disso, os avaliadores seriam notificados sobre prazos e ações relevantes, aprimorando a comunicação e a coordenação dentro da equipe.
- Conexão com Banco de Heurísticas: A integração da ferramenta com um banco de heurísticas expandiria a flexibilidade e a adaptabilidade do processo

de avaliação. Isso permitiria que os avaliadores escolhessem entre uma variedade de conjuntos de heurísticas já estabelecidos ou personalizassem suas próprias heurísticas de acordo com as necessidades específicas do projeto. Essa conexão simplificaria a seleção e a aplicação das heurísticas, tornando o processo de avaliação mais ágil e preciso.

- **Formatos Alternativos de Entrega de Resultados:** Além do formato padrão para impressão, considerar a entrega de resultados em outros formatos, como relatórios interativos na web, gráficos dinâmicos ou integrações com ferramentas de gestão de projetos, poderia aprimorar ainda mais a comunicação dos resultados da Avaliação Heurística. Essa diversificação de formatos permitiria que os resultados fossem apresentados de maneira mais acessível e interativa, atendendo às preferências e necessidades dos stakeholders.
- **Aprimoramento na Interface do Usuário:** O constante aprimoramento da interface do usuário é essencial para garantir a usabilidade da ferramenta. Trabalhos futuros podem focar na simplificação do uso, na intuitividade e na experiência geral do usuário.

REFERÊNCIAS

- ALVES, João Marcus; SAVARIS, Alexandre; WANGENHEIM, Christiane Gresse von. Criando e Validando um Checklist de Avaliação Heurística para Sistemas de Laboratórios Clínicos. *In: COMPUTER ON THE BEACH*. 2015, Flopianópolis, 2015, p. 318-327.
- BARBOSA, Simone; SILVA, Bruno. **Interação Humano-Computador**. Elsevier Brasil, v. 2, f. 204, 2010. 408 p.
- CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e Usabilidade: Conhecimentos, Métodos e Aplicações**. Novatec Editora, v. 3, f. 248, 2017. 496 p.
- HVANNBERG, Ebba Thora; LAW, Effie Lai-Chong; LÉRUSDÓTTIR, Marta Kristín. Heuristic evaluation: Comparing ways of finding and reporting usability problems. **Interacting with Computers**, v. 19, p. 225-240, Março 2007.
- MARAMBA, Inocencio; CHATTERJEE, Arunangsu; NEWMAN, Craig. Methods of usability testing in the development of eHealth applications: A scoping review. **International Journal of Medical Informatics**, p. 95-104, Jun, ano 2019, 1 jun. 2019.
- MORAES, Anamaria De; ROSA, Jose Guilherme Santa. **Avaliação E Projeto No Design De Interfaces**, f. 112. 224 p.
- N. SOKEN, *et al.* Methods for Evaluating Usability (Section 5B), Honeywell, Dec. 1993. *In: .*
- NIELSEN, Jakob. **10 Usability Heuristics for User Interface Design**. 2020. Disponível em: <https://www.nngroup.com/articles/ten-usability-heuristics/>. Acesso em: 3 set. 2021.
- NIELSEN, Jakob. **10 Usability Heuristics for User Interface Design**. Nielsen Norman Group. 2012. Disponível em: www.nngroup.com/articles/usability-101-introduction-to-usability. Acesso em: 2 set. 2021.
- NIELSEN, Jakob. **Usability Engineering**. Morgan Kaufmann, v. 3, f. 181, 1994. 362 p.
- NIELSEN, Jakob; MOLICH, Rolf. Heuristic evaluation of user interfaces. *In: PROCEEDINGS OF THE SIGCHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS: EMPOWERING PEOPLE*. 1990, p. 249-256.
- OEIRAS, Janne Yukiko Y.; BENTOLILA, David Leonardo M.; FIGUEIREDO, Mayara Costa. Heva: uma ferramenta de suporte à avaliação heurística para sistemas web. *In: IHC '08: PROCEEDINGS OF THE VIII BRAZILIAN SYMPOSIUM ON HUMAN FACTORS IN COMPUTING SYSTEMS*. 2008, Porto Alegre, 2008. 136–145 p.
- PAZ, Freddy *et al.* Heuristic Evaluation as a Complement to Usability Testing: A Case

Study in Web Domain. *In: 2015 12TH INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY - NEW GENERATIONS*. 2015, Las Vegas, 2015, p. 546-551.

RIVERO, Luis; VIANA, Davi; CONTE, Tayana. Mockup DUE: Uma Ferramenta de Apoio ao Processo de Inspeção de Usabilidade de Mockups de Aplicações Web. *In: IX WORKSHOP ANUAL DO MPS*. 2013.

ROGERS, Yvonne; SHARP, Helen; PREECE, Jennifer. **Design de Interação - 3ed.** Bookman Editora, f. 299, 2012. 597 p.

SANTOS, Francis *et al.* THEM: Ferramenta colaborativa para suporte a avaliações de interfaces baseadas na Avaliação Heurística. *In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO*, n. XXXVII. 2017.

SHNEIDERMAN, Ben; PLAISANT, Catherine. **Designing the User Interface: Strategies for Effective Human-computer-interaction**. Addison Wesley, f. 320, 1997. 639 p.

TORRENTE, Carmen Suarez M *et al.* Sirius: A heuristic-based framework for measuring web usability adapted to the type of website. **Journal of Systems and Software**, p. 649-663, Janeiro, ano 2012, 1 jan. 2012.

UNIVERSITY OF MARYLAND. **The Eight Golden Rules of Interface Design**. University of Maryland. Disponível em: www.cs.umd.edu/users/ben/goldenrules.html. Acesso em: 11 out. 2019.

WILSON, Chauncey. **User Interface Inspection Methods: A User-Centered Design Method**. Morgan Kaufmann, f. 64, 2013. 128 p.