Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

Uma Abordagem para Apoiar Avaliações de Usabilidade de Sistemas Web Remotamente

Matheus de Meneses Campanhã Souza

Número de Ordem PPGCC: M001

Teresina-PI, 12 de Agosto de 2016

Matheus de Meneses Campanhã Souza

Uma Abordagem para Apoiar Avaliações de Usabilidade de Sistemas Web Remotamente

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Pedro de Alcântara dos Santos Neto

Teresina-PI

12 de Agosto de 2016

Matheus de Meneses Campanhã Souza

Uma Abordagem para Apoiar Avaliações de Usabilidade de Sistemas Web Remotamente/ Matheus de Meneses Campanhã Souza. – Teresina-PI, 12 de Agosto de 2016-

110 p. : il. (algumas color.) ; 30 cm.

Orientador: Pedro de Alcântara dos Santos Neto

Dissertação (Mestrado) – Universidade Federal do Piauí – UFPI
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação, 12 de Agosto de 2016.

1. Usabilidade. 2. Web. 3. Teste de Software. I. Pedro de Alcântara dos Santos Neto. II. Universidade Federal do Piauí. III. Uma Abordagem para Apoiar Avaliações de Usabilidade de Sistemas Web Remotamente.

CDU 02:141:005.7

Uma Abordagem para Apoiar Avaliações de Usabilidade de Sistemas Web Remotamente

MATHEUS DE MENESES CAMPANHÃ SOUZA

Dissertação apresentada ao Programa de Pós–Graduação em Ciência da Computação do Centro de Ciências da Natureza da Universidade Federal do Piauí, como parte integrante dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Aprovado por:

Prof. Pedro de Alcântara dos Santos Neto

(Presidente da Banca)

Prof. Sérgio Castelo Branco Soares

(Examinador Externo)

Prof. Erick Baptista Passos

(Examinador Interno)

Prof. Raimundo Santos Moura

(Examinador Interno)

Teresina, 12 de agosto de 2016

"We build too many walls
and not enough bridges."
(Isaac Newton)

Resumo

Sistemas Web estão cada vez mais presentes no cotidiano das pessoas. No entanto, se partes desses sistemas não forem adequadas para uso, podem causar problemas na sua adoção e assim determinar seu sucesso ou fracasso. A adequação ao uso é algo chave e é também conhecida como usabilidade. Por conta da predominância de sistemas disponíveis na Internet, entender seu nível de usabilidade tornou-se um aspecto chave. Dentre os métodos tradicionais de avaliação de usabilidade, testes laboratoriais destacam-se por avaliar a interação de usuários com um software. Entretanto, a complexidade e os custos associados aos testes laboratoriais desencorajam a sua execução, uma vez que exigem uma equipe especializada e dedicada presencialmente durante a avaliação. Devido à crescente necessidade de realizar avaliações de usabilidade foi proposto neste trabalho uma abordagem de apoio a tais avaliações, objetivando torná-las mais simples e menos onerosa. A abordagem proposta possui uma ferramenta de apoio, denominada UseSkill, que se baseia na captura das interações dos usuários (*logs*) de forma remota e automática. A UseSkill permite a captura desses *logs* em contextos controlados, com a realização de tarefas pré-definidas, e em contextos de produção, onde o usuário utiliza livremente o sistema em seu dia a dia. Com base nos dados capturados em um contexto controlado, a ferramenta compara as ações realizadas por usuários "experientes" e "novatos" no sistema. A ferramenta calcula métricas associadas ao uso do sistema e, a partir da comparação das métricas dos dois grupos, aponta possíveis problemas de usabilidade. Um estudo experimental foi realizado para avaliar o comportamento da ferramenta em um contexto experimental e os resultados obtidos indicaram que ela pode ser um grande aliado na redução dos custos das avaliações de usabilidade. Embora a técnica desenvolvida tenha obtido um resultado satisfatório, o custo e complexidade logística envolvida nas avaliações dificultam a sua utilização. Objetivando amenizar tais problemas, foi proposta uma extensão da técnica para o ambiente real de produção de software, visando com isso permitir avaliações de usabilidade *on the fly*. Essa extensão permite que sejam realizadas avaliações periódicas de sistemas Web, sinalizando quais são as funcionalidades de mais baixo nível de usabilidade e auxiliando na identificação de problemas. Avaliações foram realizadas com apoio dessa extensão e em seguida comparadas com avaliações baseadas em um método de inspeção de usabilidade. Os resultados obtidos apresentam indícios de que a abordagem proposta é uma alternativa relevante para apoiar avaliações de usabilidade de sistemas Web em ambientes de produção.

Palavras-chaves: usabilidade, web, teste de software, *web usage mining*, *clustering*.

Abstract

Web systems are increasingly present in our daily activities. However, some systems are not suitable for use, causing problems in its adoption. Suitability for use is something known as usability. Understanding the usability level of a system has become a key issue to assure the success of these systems available on the Internet. Among traditional usability evaluation methods, laboratory tests stand out evaluating the interaction of users with a software. However, the complexity and costs associated with laboratory usability testing discourages its execution, since they require dedicated specialists during the evaluation. Due to the increasing need for usability evaluations, it is proposed in this work an approach to support these evaluations, aiming to make them simpler and less costly. The approach has a supporting tool, called UseSkill, which is based on remotely capture of user interactions (logs). UseSkill allows capturing these logs in controlled environments, with the completion of pre-defined tasks, and production environments, where users are free to use the system in their daily activities. Based on data captured in a controlled environment, this tool compares actions taken by "experienced" and "beginners" Web systems users. UseSkill calculates metrics related to system usage and points possible usability problems, suggested from differences between experienced and novice interactions. An experimental study was conducted to evaluate the results of this idea in an experimental context and the results indicate that it can be a great alternative in reducing the cost of usability evaluations. Although the developed technique has obtained a satisfactory outcome, the costs and logistical complexities involved in these evaluations in controlled environments hamper its use. Aiming to mitigate these problems, we proposed an extension to the production environment, called "on the fly" extension. This extension allows regular evaluations in web systems, indicating functionalities with lower usability indicators. Evaluations were carried out with support of this extension and then compared with evaluations based on a usability inspection method. The results show evidences that the proposed tool is a relevant alternative to support Web systems usability evaluations in production environments.

Keywords: usability, web, software testing, web usage mining, clustering.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AUET	*Automated Usability Evaluation Tool*
AA	Ações Alerta
AC	Ações Corretas
AO	Ações Obrigatórias
AP	Ações Problemáticas
API	*Application Programming Interface*
CSS	*Cascading Style Sheets*
CO	Caminho Obrigatório
CW	*Cognitive Walkthrough*
DAO	*Data Access Object*
DOM	*Document Object Model*
ES	Engenharia de Software
EC2	*Amazon Elastic Compute Cloud*
FSP	*Frequent Sequential Patterns*
GSR	Grupo de Sessões Referência
GDS	Grupo das Demais Sessões
GOMS	*Goals, Operators, Methods, Selection rules*
IHC	Interface Humano-Computador
KLOC	*Kilo Lines of Code*
MRP	*Maximal Repeating Patterns*
MUSiC	*Metrics for Usability Standards in Computing*
MVC	*Model-View-Controller*
OLAP	*Online Analytical Processing*

PACO	Percentual de Ações em relação ao Caminho Ótimo
PAO	Percentual de Ações Obrigatórias
PAGR	Percentual de Ações contidas no GSR
PARP	Percentual de Ações Redundantes ou Problemáticas
REST	*Representational State Transfer*
SQL	*Structured Query Language*
SUM	*Single, Standardized and Summated Usability Metric*
SGBD	Sistema de Gerenciamento de Banco de Dados
UEM	*Usability Evaluation Method*
UI	*User Interface*
URL	*Uniform Resource Locator*
USC	UseSkill *Control*
USOTF	UseSkill *OnTheFly*
UX	*User Experience*
WUM	*Web Usage Mining*
XPath	*XML Path Language*

Sumário

Introdução

Contexto

Aplicações Web fazem parte de diversas atividades cotidianas, como ler notícias, buscar informações, estudar e realizar interações. Com a crescente popularização de computadores, *smartphones* e do acesso à internet, há cada vez mais usuários de aplicações Web. O crescimento de dispositivos móveis fez com que não apenas aplicativos nativos sejam muito utilizados, mas com que a Web móvel comece a superar o uso da Web em *desktops* e *laptops* (QUINN et al., 2015).

Para se destacar dentre as diversas aplicações é imprescindível possuir uma boa experiência de usuário (*User eXperience*, UX). Apesar de possuir diversas definições, o principal requisito para alcançar uma UX exemplar é atender às necessidades do cliente (LAW et al., 2008). Para isso, deve haver uma fusão de serviços de várias áreas, incluindo *marketing*, engenharia e *design* de interfaces (NIELSEN; NORMAN, 2015).

Dentre os conceitos presentes em engenharia para UX, usabilidade é um dos destaques, sendo um atributo que mede a qualidade de interfaces de usuário (UI), verificando, por exemplo, se o sistema é fácil de aprender e eficiente ao utilizar. Em geral, UX é um conceito ainda mais amplo e que contempla usabilidade (NIELSEN; NORMAN, 2015).

Para construir interfaces com boa aceitação é necessário levar em consideração os aspectos de usabilidade desde o princípio do desenvolvimento e por todo o ciclo de vida de um software (NIELSEN; LANDAUER, 1993). Segundo a norma ISO 9241-11 (ISO, 1998), usabilidade é "a capacidade de um produto ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso".

Eficácia, eficiência e satisfação são aspectos relativos à utilização do usuário, sendo os dois primeiros referentes à qualidade de uso, ou seja, se o sistema permite que os usuários desempenhem corretamente suas atividades e sem esforço demasiado. A satisfação refere-se a aspectos subjetivos, como o conforto que o usuário sente ao utilizar o sistema.

Existem diversos métodos para avaliar a usabilidade de aplicações Web, como inspeções realizadas por especialistas em usabilidade, simulações, aplicação de questionários, utilização de modelos analíticos e, por fim, a realização de testes de usabilidade (IVORY; HEARST, 2001). O teste de usabilidade se destaca por ser o único a avaliar a interação de usuários com o sistema real, possibilitando a detecção de quais as reais necessidades, as dificuldades dos usuários, seus padrões de interação e, consequentemente, a detecção de problemas de usabilidade existentes no sistema Web (VARGAS; WEFFERS; ROCHA, 2011a).

Durante os testes de usabilidade, alguns usuários são selecionados para utilizar um conjunto pré-definido de tarefas no sistema sob teste, enquanto são observados por um ou mais avaliadores, que registram informações sobre seu comportamento, visando identificar problemas de usabilidade na aplicação (NIELSEN; LANDAUER, 1993). Entretanto, avaliar a usabilidade de aplicações Web é uma atividade muitas vezes ignorada devido à complexidade e custo relacionados a ela (SANTANA; BARANAUSKAS, 2015).

Motivação

Devido ao público crescente e ao grande número de aplicações Web disponíveis, a facilidade de uso de uma aplicação é um fator decisivo para seu sucesso ou fracasso (CHEN; YEN; HWANG, 2012). Segundo Nielsen (NIELSEN, 2000), a usabilidade ganha ainda mais importância na Web, pois os usuários podem facilmente trocar de aplicação se não estiverem satisfeitos. No Brasil, o ambiente Web foi o que mais apresentou estudos sobre usabilidade dentre os anos de 2002 e 2012 (FRANCISCO; BENITTI, 2014).

Apesar da necessidade de preocupação com a usabilidade durante todo o ciclo de vida de um software, mesmo com a detecção e correção de problemas de usabilidade nas etapas iniciais de desenvolvimento, diversos problemas são comumente descobertos apenas quando os usuários começam a utilizar tais sistemas em seu dia a dia (CHILANA et al., 2011; REDISH, 2007).

Dada a importância de se avaliar a usabilidade, torna-se igualmente importante escolher a melhor abordagem de avaliação. Geralmente tal seleção é influenciada pelo tempo, custo, eficiência, eficácia e facilidade de aplicação das abordagens existentes para tal finalidade (SSEMUGABI; VILLIERS, 2007).

O teste de usabilidade, apesar de ser eficiente na descoberta de problemas e de avaliar diretamente a interação dos usuários com o sistema, possui um custo e uma complexidade consideráveis. Isso advém diretamente da necessidade de selecionar usuários adequados ao teste, preparar o ambiente de utilização do software, acompanhar fisicamente a execução do teste e ainda exigir uma avaliação posterior dos especialistas acerca do comportamento dos participantes (LÓPEZ; FAJARDO; ABASCAL, 2007; BAKER et al., 2008). Essa avaliação pós execução do teste é normalmente demorada e não trivial.

Devido ao alto custo e complexidade, testes de usabilidade muitas vezes são negligenciados e postergados durante o desenvolvimento de sistemas Web (SANTANA; BARANAUSKAS, 2015). Uma forma de reduzir os custos está associada à automação total ou de algumas de suas etapas (IVORY; HEARST, 2001).

Apesar de existirem métodos que automatizam parte do processo de avaliação de usabilidade, tendo assim um potencial de reduzir o tempo e o custo envolvido no processo,

a necessidade de novos métodos e ferramentas surge por conta da complexidade associada ao processo de utilização dos métodos existentes (PAGANELLI; PATERNÒ, 2002). Segundo Santana e Baranauskas (SANTANA; BARANAUSKAS, 2015), a usabilidade desses métodos também é essencial, pois eles não devem exigir grandes esforços por parte dos especialistas durante a criação de avaliações, bem como dos usuários finais que irão participar de uma avaliação.

Objetivos

O objetivo principal desta Dissertação é apresentar uma abordagem para reduzir o custo e a complexidade de uma avaliação de usabilidade. A abordagem se propõe a apoiar avaliações de usabilidade, automatizando algumas de suas etapas de forma remota.

Os objetivos secundários do trabalho são:

1. Realizar um levantamento bibliográfico sobre a área de avaliações de usabilidade, identificando métodos, ferramentas e lacunas existentes. Isso será feito via um mapeamento sistemático sobre ferramentas que apoiam de forma automática a realização de avaliações de usabilidade em aplicações Web, fornecendo assim uma visão geral em relação às abordagens que apoiam a avaliação de usabilidade de sistemas Web;

2. Desenvolver uma abordagem utilizável também no contexto Web *mobile*, que apesar da crescente popularidade de *smartphones* e *tablets*, são raramente contemplados pelos trabalhos na área (FERNANDEZ; INSFRAN; ABRAHÃO, 2011);

3. A criação de uma abordagem versátil que possa ser aplicada tanto em contextos formais, bem definidos e controlados, quanto em ambientes reais de uso, em que os usuários utilizam o sistema no seu dia-a-dia, também é um dos objetivos deste trabalho;

4. Criar uma ferramenta de apoio a abordagem que permita aplicar o método de forma consistente, apoiando a identificação de problemas e que possa ser amplamente usada pela indústria de software;

5. Realizar avaliações da abordagem em contextos controlados, bem como em contextos de produção, para emitir um relato da experiência de uso da tecnologia desenvolvida neste trabalho e assim disponibilizar um roteiro de uso desta tecnologia por parte da indústria de software.

Embora a automação completa de uma avaliação de usabilidade seja um desejo desta pesquisa, sabe-se que isso é algo difícil de ser alcançado. Assim, o intuito deste

trabalho é gerar avanços que possam facilitar as avaliações de usabilidade ou reduzir seu custo e complexidade.

Contribuições

A principal contribuição deste trabalho é a definição de uma abordagem para auxiliar a realização de avaliações de usabilidade, que reduz o custo e o tempo gasto nessa atividade. No entanto, outras contribuições podem ser listadas, conforme é detalhado a seguir:

1. Realização de um levantamento geral do estado da arte na área, na forma de uma extensão de um Mapeamento Sistemático sobre ferramentas que apoiam avaliações de sistemas Web;

2. Definição de um método de avaliação de usabilidade baseado na comparação de manuseios "adequados" e "inadequados" de um sistema, de forma a expor as principais diferenças entre esses grupos e assim direcionar à procura por problemas de usabilidade;

3. Criação de uma ferramenta para permitir a avaliação de usabilidade a partir da comparação de ações entre grupos de usuários experientes e iniciantes. A ferramenta, denominada UseSkill Control, permite que avaliações sejam realizadas por diversos usuários simultaneamente, de forma não intrusiva e simples, potencializando a adoção de testes de usabilidade em sistemas Web;

4. Avaliação do método e da ferramenta desenvolvida para avaliação da usabilidade a partir da comparação de grupos de usuários em um contexto controlado. Com base nessa avaliação foi possível inferir que o uso das tecnologias desenvolvidas pode reduzir o custo e o tempo gasto nas avaliações de usabilidade;

5. Extensão da ferramenta para suportar ambientes Web *mobile*, permitindo assim apoiar um contexto de aplicações que normalmente é negligenciado por boa parte dos trabalhos que foram identificados nesta pesquisa;

6. Definição de uma abordagem para a mineração de dados de uso de sistemas Web para identificar eventuais problemas de usabilidade, facilitando ainda mais a realização de avaliações de usabilidade, por meio da eliminação da necessidade de se convidar usuários para o teste, usando, ao invés disso, as sessões de uso reais, em um contexto de produção;

7. Criação de uma ferramenta para permitir a avaliação de usabilidade a partir da mineração de dados de uso de sistemas Web, denominada UseSkill *OnTheFly*, apoiada

por modelos consolidados na área e disponibilizada para aplicação por parte da indústria de software de forma simples;

8. Avaliação da abordagem e da ferramenta para mineração de dados de uso em um contexto de produção, em parceria com uma empresa de desenvolvimento de software local, descrevendo seus resultados e mostrando que seu uso pode ser um grande aliado para a descoberta de problemas de usabilidade.

Estrutura da Dissertação

Este trabalho encontra-se dividido em cinco capítulos e a introdução, que aborda o contexto do trabalho, sua motivação, objetivos geral e específicos, além do relato das principais contribuições e desta seção, com a estruturação do trabalho.

No Capítulo 1 são descritos alguns dos conceitos fundamentais para uma boa compreensão do trabalho. Inicialmente são expostos os conceitos sobre usabilidade e suas formas avaliação. Em seguida são descritas as principais técnicas para captura de interação dos usuários com um sistema. Posteriormente são apresentadas as principais abordagens para analisar *logs* de forma automática. Por fim, alguns conceitos sobre *Web Usage Mining* (WUM), mineração de padrões sequenciais e agrupamento de dados são descritos.

O Capítulo 2 apresenta os trabalhos relacionados na forma de um mapeamento sistemático sobre ferramentas semiautomáticas que apoiam avaliações de usabilidade de sistemas Web.

O Capítulo 3 apresenta os métodos propostos, englobando tanto o método baseado na comparação entre grupos de usuário, quanto o método baseado em mineração de dados de uso, detalhando as diretrizes utilizadas, além de um detalhamento das suas ferramentas de apoio.

O Capítulo 4 apresenta as avaliações dos métodos propostos, feitos a partir da aplicação de suas ferramentas em sistemas Web reais, utilizados por uma grande gama de usuários.

Em seguida, o Capítulo 5 apresenta as conclusões obtidas com o trabalho e as perspectivas para trabalhos futuros.

1 Fundamentação Teórica

Este capítulo apresenta os principais conceitos relacionados ao trabalho, seguindo uma ordem cronológica dos temas que foram estudados durante esta pesquisa. Primeiramente apresenta-se a definição de usabilidade utilizada pelo trabalho e uma visão geral sobre os métodos de avaliação usabilidade, visando identificar as principais características de cada método.

Em seguida são apresentados abordagens existentes para se realizar a captura da interação de usuários e meios de se avaliar a usabilidade de forma remota e automatizada. Por fim, são apresentados conceitos relacionados ao funcionamento do módulo UseSkill *OnTheFly*, como a contextualização sobre *Web Usage Mining*, *Sequential Pattern Mining* e algoritmos de Agrupamento, com foco no K-Means.

1.1 Usabilidade e seus Métodos de Avaliação

No campo da Engenharia de Software (ES), uma definição de usabilidade bastante aceita é a proposta pela norma ISO 9126-1: "usabilidade é a capacidade de um software ser compreendido, aprendido, usado, atraente para seu utilizador e estar em conformidade com as normas/orientações, quando utilizado sob condições especificadas" (ISO, 2000).

Na área de Interface Humano-Computador (IHC), o conceito de usabilidade mais amplamente aceito é o da norma ISO 9241-11: "a capacidade de um produto ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação, em um contexto específico de uso" (ISO, 1998). A definição dada pela norma ISO 9241-11 é a que mais se aproxima da perspectiva da interação humana (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), sendo assim, a definição de usabilidade adotada por este trabalho.

Após a definição de usabilidade é necessário definir como é feita sua avaliação. Segundo Hilbert e Redmiles (HILBERT; REDMILES, 2000), avaliação de usabilidade é o ato de mensurar (ou identificar potenciais problemas que afetam) o nível de uso de um software ou dispositivo, em relação a determinados usuários, executando determinadas tarefas e em um determinado contexto de uso.

Dentre os diversos métodos para se avaliar a usabilidade de software, Ivory e Hearst propuseram uma taxonomia para classificar os tipos de métodos existentes (IVORY; HEARST, 2001):

- **Inspeção**: envolve um avaliador experiente, utilizando um conjunto de critérios para identificar potenciais problemas de usabilidade (e.g., avaliações heurísticas,

revisões de *guidelines*, orientações cognitivas). Dentre os métodos de inspeção, os mais conhecidos e utilizados nessa categoria são: Avaliação Heurística e o *Cognitive Walkthrough* (CW) ou Passo a Passo Cognitivo (NIELSEN, 1994);

- **Modelagem Analítica**: apresenta uma abordagem de engenharia que permite aos avaliadores prever problemas de usabilidade, aplicando diferentes tipos de modelos (e.g., análise GOMS, análise cognitiva de tarefas). Os métodos baseados em modelos buscam estimar o desempenho e o esforço do usuário na utilização de uma aplicação, detalhando cada ação do usuário para realizar uma tarefa e estimando o tempo necessário para sua realização. Dentre os métodos de modelagem analítica, o GOMS (*Goals, Operators, Methods, Selection rules*) é um dos mais utilizados (PATERNÒ; SANTORO, 2008);

- **Simulação**: utiliza modelos de interface e de usuários para simular como seria a interação entre usuários e interfaces (e.g., Modelos de Redes de Petri, Modelos de Markov). A utilização de métodos de simulação permite ao avaliador testar diversos cenários de uso, sem a participação de usuários reais. Dentre os métodos de simulação, comumente são aplicadas Redes de Petri (PALANQUE et al., 2006);

- **Investigação**: apresenta um método que reúne entradas subjetivas a partir dos participantes, como suas preferências ou seus sentimentos (e.g., grupos focais, questionários, entrevistas). Esse grupo de métodos visa obter informações sobre os usuários, suas necessidades, percepções e objetivos em relação à interação com o software (NIELSEN; LANDAUER, 1993);

- **Teste**: trata de avaliadores observando como os participantes interagem com a interface de usuário, com o objetivo de determinar problemas de usabilidade (e.g., pensamento em voz alta, teste laboratorial, teste remoto via análise de *logs*). Os avaliadores buscam identificar as dificuldades que cada usuário teve durante a realização de tarefas na aplicação e que podem revelar problemas na interface (NIELSEN; LANDAUER, 1993).

As principais vantagens dos métodos de inspeção são: a possibilidade de serem executados em qualquer etapa do desenvolvimento de um software e o baixo custo, pois não necessitam da participação de usuários (NIELSEN, 1994; DIX, 2009; CONTE et al., 2007). Entretanto, por não haver participação direta de usuários, os métodos de inspeção baseiam-se apenas na experiência de seus avaliadores em prever como os usuários irão interagir com a interface.

Quanto aos métodos de modelagem analítica, são descritos os passos de uma tarefa de forma hierárquica, ou seja, qual é a sequência de ações que devem ser executadas para atingir determinado objetivo. Assim como os métodos de inspeção, a maior vantagem

dos métodos baseados em modelos é seu baixo custo, porém por também não utilizarem usuários reais, não levam em conta a real utilização do sistema sob avaliação.

Os métodos de simulação permitem obter dados que auxiliam a análise das interações dos usuários, porém de maneira artificial, sem a real utilização do sistema sob teste. A ideia é simular alterações de *design* em interfaces e obter projeções de como seria o comportamento dos usuários nesses diferentes cenários (IVORY; HEARST, 2001). Da mesma forma que os métodos descritos até o momento, a ausência de usuários reais reduz os custos da utilização da técnica, porém os seus resultados são baseados em simulações, que não permitem avaliar a usabilidade de aplicações em seu contexto de uso real.

Nos métodos de investigação, a maior vantagem é o conjunto *feedbacks* dos usuários, permitindo obter informações sobre percepções dos usuários reais do sistema. Entretanto, a obtenção de boas informações sobre a interação com o sistema depende da habilidade e boa vontade do usuário em reportar problemas. Apesar de *feedbacks* dos usuários reais, que permitem capturar características relacionados à satisfação, é inviável a captura de informações sobre a eficácia e eficiência dos usuários por meio apenas de técnicas de investigação.

Dentre os tipos de métodos para avaliar a usabilidade de sistemas, o teste de usabilidade é o foco deste trabalho, pois além de ser um dos mais utilizados, também consegue captar a eficácia, eficiência e satisfação dos usuários ao utilizarem a aplicação a ser testada. Teste de usabilidade com participantes reais é um método de avaliação de usabilidade que fornece informações diretas aos avaliadores sobre quais problemas os usuários enfrentam na interface que está sendo testada (NIELSEN; LANDAUER, 1993; SHNEIDERMAN, 1992).

Entretanto, os resultados de testes com usuários são estritamente relacionados a alguns aspectos, como o quão representativos são os usuários utilizados nos testes, o grau de experiência dos especialistas em usabilidade (avaliadores) envolvidos e o quão o contexto do teste se assemelha com o ambiente real de uso. Para sistemas Web fica ainda mais difícil avaliar todos os perfis de usuários desejados, dada a diversidade de seus utilizadores.

O teste tradicional é realizado em laboratório, com alguns participantes utilizando a aplicação a ser testada, enquanto são monitorados por especialistas em usabilidade e tem suas ações gravadas por meio de câmeras para análises posteriores. Entretanto os usuários tendem a agir de forma diferente do habitual por estarem sendo observados por avaliadores e por estarem em um ambiente não familiar, dificultando a obtenção de dados reais nos testes (CASTILLO; HARTSON; HIX, 1998).

Além dos possíveis problemas já citados, o teste laboratorial consome bastante tempo e é muito oneroso, devido à necessidade da preparação do ambiente, que exige

a utilização de equipamentos específicos e de especialistas, além do deslocamento dos participantes, e de não reproduzir as condições de uso da aplicação no ambiente real (MUELLER et al., 2009; DRAY; SIEGEL, 2004).

Os testes com usuários podem ser executados remotamente, de maneira que os usuários não necessitam se deslocar para um laboratório de testes. A principal diferença entre testes remotos e presenciais é a separação espacial entre os especialistas e os usuários. Dentre os testes remotos, eles podem ser síncronos ou assíncronos. Nesse caso, a separação temporal entre os especialistas e os usuários é a principal diferença, permitindo que os usuários executem os testes a qualquer momento, independente do acompanhamento de especialistas no momento da realização do teste (BRUUN et al., 2009).

A seguir são apresentadas algumas formas automáticas de captura de dados de usuários.

1.2 Captura de Dados de Interação

A forma mais simples de capturar dados de interação dos usuários se dá por meio de gravações de áudio e vídeo. No entanto, apesar dessa abordagem permitir que a captura ocorra de forma automática e remota, quando há vários usuários essa abordagem se torna demasiadamente cara, uma vez que todos os áudios e vídeos devem ser analisados para se chegar a uma afirmação sobre possíveis problemas de usabilidade, assemelhando-se a testes laboratoriais. Desse modo, para capturar interações de usuários em larga escala é necessário viabilizar a automatização do processo de avaliação. Automatizar esse processo requer que tanto a captura, e em especial, a análise dos dados, sejam automatizadas, ou pelo menos facilitada por meio do uso de ferramentas de apoio.

Profissionais de IHC categorizam as ferramentas para avaliação de usabilidade de sistemas Web remotamente em dois grupos principais: as baseadas em código fonte (conteúdo ou estrutura) e as que analisam dados de utilização contidos nos *logs* (SANTANA; BARANAUSKAS, 2015). Para a captura de dados de utilização de forma automática e remota, utilizam-se técnicas que permitem registrar as ações realizadas pelos usuários, na interface das aplicações, armazenando-as em *logs* de interação (IVORY; HEARST, 2001).

Com o uso de *logs* é possível capturar as interações enquanto os usuários utilizam a aplicação, de modo transparente, sem interferir na interação. Assim, é possível obter dados de interação de grandes quantidades de usuários simultaneamente (WEST; LEHMAN, 2006; BALBO et al., 2008). A utilização de *logs* permite a realização de avaliações de forma assíncrona e remota, mas *logs* não conseguem captar a satisfação dos usuários, focando assim a avaliação de usabilidade em quesitos relacionados à qualidade de uso (eficácia e eficiência) (BEVAN, 1995).

A captura de *logs* pode ser tanto em servidores (lado do servidor de aplicação), como em navegadores (lado do cliente) (SANTANA; BARANAUSKAS, 2015). Capturar *logs* em servidores é tecnicamente mais simples, mas os dados capturados revelam apenas informações relacionadas às páginas que o usuário visitou. Por outro lado, capturar *logs* nos navegadores é computacionalmente mais complexo, pois é necessário interceptar eventos que ocorrem no dispositivo do usuário, porém, as informações capturadas são mais detalhadas, apontando, além da página do sistema Web, o elemento e a ação associada.

Geralmente os *logs* em servidores são formados por um conjunto de ações representados por: endereço IP, tempo, URL (*Uniform Resource Locator*) da página desejada e URL da página de origem. *Logs* de servidores não podem capturar as interações dos usuários que ocorrem no navegador (*e.g.* cliques em objetos, preenchimento de formulários ou botão voltar). Além disso, esse tipo de *log* enfrenta alguns empecilhos que dificultam sua utilização para refletir a usabilidade de sistema, como por exemplo: cache entre servidores *proxy* e navegadores, interpretação complexa e difícil identificação de tarefas realizadas (BYRNE et al., 1999).

Registos de *logs* no lado do cliente capturam dados de uso mais precisos e abrangentes que os *logs* de servidor, pois permitem que praticamente todos os eventos do navegador sejam gravados. Tal exploração pode fornecer mais detalhes sobre a usabilidade. Entretanto, eles exigem o uso de navegadores modificados, servidores *proxy* ou que cada página do sistema Web seja modificada para capturar os dados de interação (HILBERT; REDMILES, 2000).

Além da captura dos dados ser automatizada, para reduzir complexidade e custo é desejado que boa parte da avaliação também seja realizada de forma automática. A seguir são apresentados as principais técnicas para automatizar a avaliação de usabilidade presentes na literatura.

1.3 Automação de Avaliações de Usabilidade

Durante testes de usabilidade, avaliadores usam os resultados das avaliações para determinar quão bem a interface suporta que usuários completem as tarefas, bem como outras medidas, como o número de erros e tempo para conclusão da tarefa (IVORY; HEARST, 2001). Para reduzir os custos e o esforço para avaliar os usuários por meio de testes, alguns métodos foram propostos para automatizá-los. Dentre eles, a captura dos dados de uso, a análise desses dados e até mesmo a sugestão de melhorias no sistema.

O *survey* realizado por Ivory e Hearst (IVORY; HEARST, 2001) identificou quatro características gerais de técnicas para a análise de *logs* durante avaliações de usabilidade:

- **Baseada em Métrica**: os métodos para análise de *logs* baseados em métricas

geram medidas quantitativas de performance. Essa abordagem tem sido eficaz para auxiliar o avaliador a compreender as condutas dos usuários, no entanto, as abordagens com base em métricas exigem que o avaliador realize análises detalhadas para determinar a origem dos problemas de usabilidade. Um exemplo dos métodos baseados nessa abordagem, o DRUM (MACLEOD; RENGGER, 1993) processa *logs* para calcular as métricas baseadas nas MUSiCs (*Metrics for Usability Standards in Computing*) (BEVAN, 1995) e sincronizar a ocorrência de eventos no *log* com a filmagem, auxiliando a análise de vídeos;

- **Correspondência de Padrões**: os métodos referentes à correspondência de padrões analisam se os comportamentos dos usuários capturados em *logs* correspondem a padrões pré-determinados. Dentre eles, o MRP (*Maximal Repeating Patterns*) detecta e gera relatórios sobre ações repetidas, por exemplo, chamadas consecutivas do mesmo comando, que podem indicar problemas de usabilidade. Estudos com o MRP mostraram que a técnica pode ser útil para detectar problemas com usuários experientes, mas uma pré-filtragem dos dados é necessária para a detecção com usuários novatos (SIOCHI; EHRICH, 1991);

- **Baseada em Tarefas**: métodos baseados em tarefas analisam discrepâncias entre modelos de tarefas realizados por *designers* e o que o usuário realmente realizou durante o uso do sistema. Uma das ferramentas pioneiras nesse método é a UsAGE (UEHLING; WOLF, 1995), que captura *log* de dados de tarefas e compara os *logs* de dois usuários, sendo um "experiente" e um "novato". A UsAGE gera um grafo contendo as ações dos dois usuários, porém a identificação de cada evento na interface é complexa. Abordagens baseadas em tarefa produzem bons *insights* de como aprimorar a usabilidade;

- **Inferencial**: análises inferenciais de *logs* focam em permitir a visualização dos dados capturados em diversas formas diferentes. Geralmente incluem técnicas estatísticas, como as baseadas em tráfego (*e.g.*, agrupamento de páginas mais acessadas por visitantes) e baseadas no tempo (como, durações dos usuários nas páginas). A *Starfield* (HOCHHEISER; SHNEIDERMAN, 2001), por exemplo, permite que avaliadores explorem de forma interativa os dados de *logs* de servidores Web, a fim de ganhar uma compreensão de questões de fatores humanos relacionados aos padrões de visitação;

Além de ferramentas que focam em apenas um dos métodos, há também ferramentas híbridas, que aproveitam as vantagens de cada uma das abordagens discutidas e converge em uma única solução. A WebRemUSINE é uma hibridização das técnicas "baseada em tarefas" com "correspondência de padrões" (PAGANELLI; PATERNÒ, 2002), provendo *insights* para aprimorar a usabilidade a partir de análises de tarefas. No entanto, ela

demanda a concepção de um modelo que será comparado com a utilização dos usuários, além de detectar apenas problemas pré-definidos, por conta da correspondência de padrões.

1.4 Métricas de Usabilidade

Existem diversos padrões de métricas que mensuram a usabilidade de sistemas Web. A seguir são descritos os dois mais utilizados na literatura e que foram considerados durante a definição de métricas neste trabalho.

A SUM (*Single, Standardized and Summated Usability Metric*) é uma proposta para padronizar métricas de usabilidade em um único núcleo, concatenando métricas de testes de usabilidade e questionários, permitindo a comparação quantitativa de usabilidade entre funcionalidades e sistemas distintos. Para o cálculo da eficácia é utilizada a taxa de completude e corretude (exatidão) das tarefas, a eficiência com base no tempo desprendido (esforço) e a satisfação a partir de questionários (SAURO; KINDLUND, 2005).

A MUSiC (*Metrics for Usability Standards in Computing*) mede a eficácia, que é calculada multiplicando a completude e a corretude dos *logs* em relação à tarefa. Ela também mede a eficiência, que é a eficácia em relação ao tempo de conclusão da tarefa. Por fim, é medido o período produtivo, que é a porção de tempo que o usuário não tem problemas e nem está lendo manuais ou ajudas para utilizar o sistema (BEVAN, 1995).

A seção a seguir apresenta os principais conceitos de *Web Usage Mining* utilizados neste trabalho.

1.5 *Web Usage Mining*

O volume de *logs* coletados diariamente por sistemas Web atinge grandes proporções. A análise desses dados pode ajudar organizações a determinar estratégias de *marketing* de produtos e serviços, avaliar a eficácia de campanhas, otimizar funcionalidades de sistemas Web, fornecer conteúdos personalizados para os visitantes e encontrar a estrutura lógica mais eficaz (MOBASHER, 2006). Este tipo de análise envolve a descoberta automática de padrões e relacionamentos a partir de uma grande coleção de dados semiestruturados, geralmente armazenados em *logs* de acesso de servidores Web e aplicações.

A utilização de mineração de dados associados à Web é denominado *Web Mining*, que pode ser dividido em três classes: *Web Content Mining*, que analisa os dados contidos nas páginas Web, geralmente baseado em textos e imagens; *Web Structure Mining*, que avalia a organização do conteúdo, como os *links* entre as páginas, geralmente representadas no formato de árvores; e *Web Usage Mining*, que avalia padrões de uso em páginas Web (SRIVASTAVA et al., 2000).

Web Usage Mining (WUM) é a aplicação de técnicas de mineração de dados para descobrir padrões de uso de dados da Web, a fim de entender e atender melhor as necessidades de aplicações (SRIVASTAVA et al., 2000). WUM consiste em três fases: pré-processamento, descoberta de padrões e análise de padrões.

1.5.1 Pré-processamento

A etapa de pré-processamento geralmente é a mais onerosa e demorada atividade de WUM, pois geralmente as informações disponíveis na Web são heterogêneas e desestruturadas (SRIVASTAVA et al., 2000; CHITRAA; DAVAMANI; SELVDOSS, 2010). Para pré-processar os dados, geralmente são realizadas as seguintes etapas:

- **Limpeza dos Dados**: processo de remoção de dados irrelevantes. Para *logs* capturados em servidores, dados como requisição de imagens ou arquivos geralmente são removidos. No caso de *logs* capturados no lado do cliente, são filtrados de acordo com ações relevantes (MOBASHER, 2006; CHITRAA; DAVAMANI; SELVDOSS, 2010);

- **Identificação de Usuários**: agrupar os *logs* de acordo com os usuários que realizaram tais ações. A identificação dos usuários de acordo com os endereços IP nem sempre funciona, pois a utilização de *Proxy* permite que o mesmo endereço IP seja compartilhado por diferentes usuários (CHITRAA; DAVAMANI; SELVDOSS, 2010). Uma alternativa é associar cada usuário a *cookies* específicos (MOBASHER, 2006);

- **Identificação de Sessões**: uma sessão de um usuário pode ser definida como o conjunto de ações realizadas por determinado usuário durante uma utilização do sistema Web. As principais heurísticas são: usar um limiar de tempo entre ações ou usar uma topologia de relações entre as páginas. A segunda heurística exige a construção de um grafo ligando todas as páginas do sistema, tornando-se inviável em sistemas mais complexos (CHITRAA; DAVAMANI; SELVDOSS, 2010);

- **Comprimento de Caminho**: essa etapa é necessária caso a captura de dados seja no servidor, pois há chance de perda de dados por conta de *cache* ou *proxy*, que podem causar "ausências" nos dados acessados. Uma das técnicas para inferir referências faltosas é basear-se na estrutura do site para identificar pontos faltosos entre uma ação e outra (MOBASHER, 2006; CHITRAA; DAVAMANI; SELVDOSS, 2010);

- **Integração de Dados**: essa etapa concatena dados que contribuam com informações para análise aos *logs* capturados. Por exemplo, dados demográficos, histórico dos usuários, pontuações, etc. (MOBASHER, 2006).

1.5.2 Descoberta de Padrões

Após a realização do pré-processamento dos dados, eles estão aptos a serem avaliados para buscar padrões de uso. Métodos e algoritmos de diversas áreas são aplicadas aos dados nessa etapa, como estatística, mineração de dados, aprendizagem de máquina e reconhecimento de padrões (CHITRAA; DAVAMANI; SELVDOSS, 2010).

As técnicas estatísticas são o método mais utilizado para extrair conhecimento sobre os visitantes do sistema Web. Ao analisar as sessões é possível realizar análises estatísticas descritivas (frequência, média, mediana, etc.), além do tempo e tamanho dos caminhos percorridos.

A geração de regras de associação também é muito utilizada, em especial para identificar páginas que mais são referenciadas juntas em uma única sessão no servidor. Essa técnica é muito utilizada para realizar sugestões para os usuários durante suas navegações.

Técnicas de agrupamento de dados são voltadas para agrupar usuários em conjuntos que possuem características semelhantes. Uma das principais finalidades desses agrupamentos é identificar usuários que percorrem caminhos semelhantes, visando oferecer conteúdo personalizado (CHITRAA; DAVAMANI; SELVDOSS, 2010).

A técnica de descoberta de padrões sequenciais visa identificar padrões entre sessões, nas quais um conjunto de itens é seguido de outro item, permitindo a predição de padrões futuros de navegação. Esse tipo de análise é muito utilizada para avaliar o *marketing* Web, como análise de tendências ou detecção de ponto de alteração (CHITRAA; DAVAMANI; SELVDOSS, 2010).

As técnicas de mineração de padrões sequenciais e de agrupamento de dados foram utilizadas na abordagem proposta. A seguir há uma breve descrição das técnicas utilizadas.

1.5.2.1 Mineração de Padrões Sequenciais

Mineração de Padrões Sequenciais é um subtópico de Mineração de Dados e é voltada para a descoberta de sequências frequentes em um banco de dados. Esses bancos armazenam eventos ordenados, com ou sem a definição concreta de tempo. Em geral, os problemas de mineração de padrões sequenciais podem ser classificados como: mineração de *string*, que são baseados em algoritmos de processamento de texto; e mineração de *itemsets*, que baseiam-se em regras de associação (MABROUKEH; EZEIFE, 2010). O foco deste trabalho é em problemas de *itemsets*.

Para entender um pouco melhor sobre tais algoritmos é necessário conhecer alguns termos. *Itemset* é um conjunto não vazio e não ordenado de itens. Uma sequência é uma lista ordenada de *itemsets*. No caso de WUM, por exemplo, cada item representa uma ação realizada por um usuário.

Tradicionalmente, a mineração de *itemset* é usada em aplicações relacionadas a *marketing* para, por exemplo, descobrir relações frequentes entre itens em grandes transações. A descoberta de *itemsets* busca regras como: "se o cliente comprar um sapato, ele é suscetível a comprar meias em seguida"; ou no contexto de análise de *logs* em WUM, "se o usuário clicar no link do produto e em seguida preencher o campo "quantidade" é provável que ele clique no botão "detalhes" dentro de 2 minutos".

A base de técnicas de mineração de *itemset* consiste em algoritmos de regras de associação, como o Apriori (AGRAWAL; SRIKANT, 1994). Dentre os algoritmos de mineração, destacam-se o GSP, FreeSpan, PrefixSpan, SPADE e CloSpan. Apesar de diferenças relacionadas ao funcionamento dos algoritmos, todos buscam o mesmo resultado: um conjunto de padrões sequenciais frequentes dentre os dados analisados. O CM-SPADE foi utilizado neste trabalho por ser uma versão alterada do SPADE com melhorias de performance (FOURNIER-VIGER et al., 2014; ZAKI, 2001).

O CM-SPADE utiliza os dados no formato de uma lista vertical de IDs, onde cada sequência é associada a uma lista de objetos. Em seguida, as sequências frequentes podem ser encontrados de forma eficiente usando cruzamentos na lista de IDs. O método reduz o número de verificações no conjunto de dados e, por conseguinte, reduz o tempo de execução do algoritmo (FOURNIER-VIGER et al., 2014).

Para utilizar o algoritmo CM-SPADE é necessário definir o parâmetro *minsup*, que corresponde a um limiar de corte para podar a árvore de possibilidades do Apriori. O *minsup* é o tamanho mínimo desejado para as sequências de *itemsets*.

Ao final da mineração, o CM-SPADE retorna o conjunto completo das subsequências frequentes que satisfazem o *minsup*. No caso de WUM, cada item das subsequências é uma ação realizada pelo usuário ao utilizar o sistema. Apesar das ações retornadas serem sequenciais, não necessariamente uma ação foi realizada logo após a outra, ou seja, podem haver outras ações entre duas ações das subsequências retornadas.

1.5.2.2 Agrupamento de Dados

Agrupamentos de dados ou *clustering* são técnicas para agrupar objetos de tal forma que ao considerar alguma característica, os objetos do mesmo grupo são mais semelhantes entre si do que com os objetos de outros grupos. Técnicas de agrupamento são bastante utilizadas e possuem aplicações em diversas áreas, tais como: mineração de dados, análises estatísticas, reconhecimento de padrões, aprendizagem de máquina, dentre outras.

Não há uma definição precisa e unânime de *cluster*, mas há um conceito presente em todas as definições: um grupo de objetos de dados. Existem diversas variações entre os algoritmos de clusterização, que vão desde a pertinência de um objeto a mais de um grupo simultaneamente, até se os objetos possuem hierarquias.

Apesar das diversas ramificações entre os algoritmos de agrupamento de dados, esse trabalho focou em um algoritmo onde um objeto pertença a apenas um grupo, não possua hierarquia e permita o agrupamento de um grande número de objetos. Dentre os diversos algoritmos disponíveis, o K-Means foi utilizado e detalhado a seguir (ARTHUR; VASSILVITSKII, 2007).

O algoritmo K-Means agrupa os dados tentando separar os n objetos em k grupos com amostras similares. Uma heurística bastante utilizada para medir a similaridade dos dados é a distância euclidiana. A distância euclidiana entre os pontos $P = (p_1, p_2, \ldots, p_n)$ e $Q = (q_1, q_2, \ldots, q_n)$ é calculada segundo a Formula 1.1.

$$\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{1.1}$$

Para executar o K-Means é necessário informar o número de *clusters* desejado. O algoritmo divide um conjunto de n amostras de objetos X em k *clusters* disjuntos C, cada um descrito pela média μ_j das amostras no *cluster*. As médias são comumente chamadas "centróides". Eles não são, em geral, pontos de X, embora estejam no mesmo espaço (MACQUEEN et al., 1967). O algoritmo K-Means mira na escolha dos centróides que minimizem a dissimilaridade entre os objetos de um mesmo *cluster*, como pode ser visto na Fórmula 1.2 (ARTHUR; VASSILVITSKII, 2007):

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_j - \mu_i||^2) \tag{1.2}$$

1.5.3 Análise de Padrões

A análise de padrões é a última etapa de *Web Usage Mining*. A principal motivação para essa etapa é filtrar regras ou padrões que não são interessantes e foram identificadas na etapa de descoberta de padrões. A forma mais comum de analisar os padrões é por meio de mecanismos de consulta de conhecimento, como SQL, ou a utilização de cubos de dados OLAP (SHAHABI et al., 1997). Técnicas de visualização, como padrões gráficos ou utilização de cores para os dados distintos, podem destacar padrões ou tendências nos dados (MOBASHER, 2006; CHITRAA; DAVAMANI; SELVDOSS, 2010; SRIVASTAVA et al., 2000).

1.6 Considerações Finais

Este capítulo apresentou os principais conceitos utilizados durante o trabalho. Além da definição de usabilidade adotada no trabalho, este capítulo discorre sobre os tipos de métodos de avaliação, seguindo a taxonomia proposta por (IVORY; HEARST, 2001).

Outros conceitos referentes à forma de capturar e analisar automaticamente dados de interação de usuários, discutidos na abordagem proposta, também são apresentados no capítulo.

2 Trabalhos Relacionados

Avaliação de usabilidade remota automática ou semiautomática é um importante instrumento para apoiar o desenvolvimento de aplicações Web modernas. A automatização desse tipo de avaliação reduz o seu custo, pois o tempo necessário para avaliar a usabilidade diminui significativamente e a necessidade de avaliadores também é reduzida ou mesmo eliminada (IVORY; HEARST, 2001). Este capítulo apresenta os trabalhos identificados por meio de um Mapeamento Sistemático, onde é identificado o estado da arte em ferramentas que apoiam de forma automática avaliações de sistemas Web.

2.1 Mapeamento Sistemático

Mapeamento sistemático é um método de pesquisa que permite identificar estudos relevantes em determinado tema de interesse, fornecendo uma visão geral de uma determinada área de pesquisa (PETERSEN et al., 2008).

O alto custo e a complexidade de testes de usabilidade fazem com que a técnica seja foco de propostas para a sua automação. De acordo com o mapeamento sistemático realizado por Fernandez et al. (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), dentre as classes de métodos de avaliação de usabilidade, o teste de usabilidade é a técnica que mais possui trabalhos voltados para a sua automação. Apesar do mapeamento realizado ter identificado quais UEMs (*Usability Evaluation Methods*) foram mais utilizados para avaliar a usabilidade de aplicações Web, não houve um levantamento de características específicas sobre ferramentas que se propõem a automatizar testes de usabilidade.

Para identificar e entender melhor as diferenças entre tais ferramentas, foi proposta a realização de uma extensão do mapeamento sistemático realizado por Fernandez et al. A extensão proposta foca em "ferramentas que apoiam de forma automática a realização de avaliações de usabilidade em aplicações Web" (*automated usability evaluation tools*, AUETs). Dentre as principais características, deve ser analisado como tais ferramentas são aplicadas no contexto Web e quais as características e restrições das ferramentas.

Uma revisão sistemática da literatura (KITCHENHAM; CHARTERS, 2007) foi considerada uma opção menos viável devido à amplitude da nossa questão geral de pesquisa: "Qual é o estado da arte na literatura relativos a Web AUETs?". Revisões são mais aprofundadas e visam mensurar a qualidade dos trabalhos identificados, enquanto o mapeamento é uma análise quantitativa da área.

Para a realização da extensão do mapeamento sistemático, foram utilizadas as *guidelines* propostas em (KITCHENHAM; CHARTERS, 2007) e o processo de mapeamento

sistemático proposto por Petersen (PETERSEN et al., 2008). A Figura 1 ilustra o processo adotado.

Apesar de ser uma extensão de um mapeamento sistemático, o processo é baseado no processo definido por Petersen, entretanto sem a etapa de *keywording*, pois todos os trabalhos são da mesma área de conhecimento (usabilidade). A definição do protocolo seguiu os seguintes estágios de definição: questão de pesquisa, estratégia de pesquisa, seleção dos estudos primários, extração dos dados e estratégia de síntese.

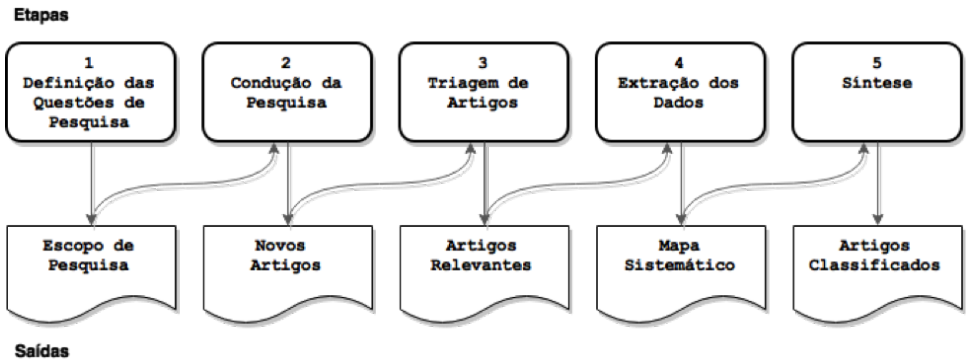


Figura 1 – Processo utilizado neste mapeamento sistemático, baseado no proposto por Pertersen (PETERSEN et al., 2008).

2.1.1 Objetivos

Os principais objetivos deste mapeamento são: identificar, quantificar e entender as principais contribuições do estado da arte de ferramentas existentes que apoiam avaliações de usabilidade de forma automática para a Web. Existem diversas formas de automatizar avaliações de usabilidade, que variam desde a captura dos dados de uso, a análise dos dados e até a sugestão de melhorias no sistema avaliado.

2.1.2 Metodologia de Pesquisa

2.1.2.1 Definição das Questões de Pesquisa

A questão de pesquisa deste estudo de mapeamento é: "Qual é o estado da arte na literatura relativo a Web AUETs?". Para responder esta questão, ela foi subdividida nas seguintes sub-questões:

- **Q1:** Que etapas das avaliações são automatizadas?
- **Q2:** Quais os métodos utilizados para automatizar o apoio a avaliações de usabilidade no domínio Web??
- **Q3:** Qual esforço de cada AUET?

- **Q4:** Qual o grau de intrusão das ferramentas?
- **Q5:** A ferramenta apresenta avaliação remota?
- **Q6:** Qual o contexto de avaliação *web*?

2.1.2.2 Condução da Pesquisa de Estudos Primários

Inicialmente foram identificados os estudos primários exercitando a *string* de busca em banco de dados científicos. A *string* de busca foi estruturada a partir da definição dos termos base da pesquisa. Em seguida foram identificados os sinônimos e agrupados com o operador lógico *OR*, enquanto os termos foram agrupados com operador lógico *AND*.

Os termos referentes à "Web", "Usabilidade" e "Avaliação" foram adaptados do mapeamento realizado por Fernandez (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), enquanto o termo "Automático" foi inserido para identificar apenas os trabalhos que apresentam métodos que automatizam o processo de avaliação. Para evitar que a *string* de busca fosse demasiadamente restritiva, o termo "ferramenta" foi concatenado como um sinônimo do termo "avaliação", pois alguns trabalhos que propõem ferramentas se auto declaram como técnica, abordagem, métodos, etc.

O espaço de busca utilizado foi entre 2009 (ano do mapeamento de Fernandez et al.) e 2015. De uma forma geral, a *string* final de busca foi baseada nos termos e sinônimos presentes na Tabela 1:

Tabela 1 – *String* de busca utilizada para busca dos trabalhos nos mapeamento.

Termo	**Sinônimos**
Automático	(automat* OR semiautomat* OR semi-automat*)
Web	(web OR website OR internet OR www)
Usabilidade	(usability OR usable)
Ferramenta/Avaliação	(evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach* OR tool*)

Para facilitar a replicação das buscas realizadas, as *strings* foram aplicadas em bases de buscas científicas específicas. Outra preocupação foi evitar que trabalhos analisados ficassem indisponíveis ou inacessíveis, restringindo a busca para as principais editoras na área de Ciência da Computação. As bases de buscas e as editoras utilizadas foram:

- Base de Buscas: Scopus, Engineering Village e Web of Science;
- Editoras: IEEE, Springer, Science Direct, Elsevier e ACM *journals*/*conferences*;

As *strings* de busca utilizadas especificamente para cada uma das bases selecionadas foram:

- **Scopus**: TITLE-ABS-KEY((automat* OR semiautomat* OR semi-automat*) AND (web OR website OR internet OR www) AND (usability OR usable) AND (evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach* OR tool*)) AND (LIMIT-TO(PUBYEAR, 2015) OR LIMIT-TO(PUBYEAR, 2014) OR LIMIT-TO(PUBYEAR, 2013) OR LIMIT-TO(PUBYEAR, 2012) OR LIMIT-TO(PUBYEAR, 2011) OR LIMIT-TO(PUBYEAR, 2010) OR LIMIT-TO(PUBYEAR, 2009));
- **Engineering Village**: (((automat* OR semiautomat* OR semi-automat*) AND (web OR website OR internet OR www) AND (usability OR usable) AND (evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach* OR tool*)) WN TI) OR (((automat* OR semiautomat* OR semi-automat*) AND (web OR website OR internet OR www) AND (usability OR usable) AND (evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach* OR tool*)) WN AB);
- **Web of Science**: TS=((automat* OR semiautomat* OR semi-automat*) AND (web OR website OR internet OR www) AND (usability OR usable) AND (evalu* OR assess* OR measur* OR experiment* OR stud* OR test* OR method* OR techni* OR approach* OR tool*)).

2.1.2.3 Triagem de Trabalhos Relevantes

Para realizar a triagem dos trabalhos identificados na etapa de buscas, foram definidos critérios para excluir trabalhos que não sejam relevantes para este mapeamento. Os critérios de exclusão também foram estendidos do mapeamento realizado por Fernandez et al. (FERNANDEZ; INSFRAN; ABRAHÃO, 2011) e adaptados para a questão de pesquisa deste trabalho. Os critérios de exclusão utilizam o operador lógico *OR*, ou seja, se ao menos um for verdadeiro, o trabalho é excluído. Os critérios utilizados foram:

- Não foca em usabilidade (não aborda usabilidade ou apenas relata uma avaliação de usabilidade);
- Não aborda domínio Web;
- Não apresenta uma ferramenta;
- Não foca em métodos automáticos ou semiautomáticos para o teste de usabilidade;
- Não apresenta um método/ferramenta que avalia a usabilidade;

- Trabalho introdutório e não conclusivo;
- Trabalho duplicado e publicado em fontes distintas;
- Não escrito em inglês;

O primeiro critério de exclusão, apesar de compacto, engloba diversos critérios que foram bem debatidos entre os participantes do mapeamento antes da etapa de seleção de estudos primários. Os critérios englobados pelo primeiro critério acima são:

- Apresentam apenas recomendações, *guidelines* ou princípios de Web *design*;
- Apresentam apenas atributos de usabilidade e suas métricas associadas;
- Apresentam apenas estudos sobre acessibilidade;
- Apresentam apenas técnicas de como melhorar métricas de usabilidade;
- Apresentam apenas processos de teste que focam em verificar aspectos funcionais;

Para a inclusão de trabalhos é necessário que nenhum dos critérios de exclusão seja verdadeiro. A triagem dos trabalhos segue o *workflow* descrito na Figura 2. Com o suporte da ferramenta TheEnd, que foi desenvolvida durante o mapeamento para auxiliar no processo de "Condução de Pesquisa" e "Triagem", os resultados das buscas foram mesclados em uma lista que passou por remoção de trabalhos duplicados, com problemas de metadados ou que não possuíam as palavras-chave buscadas.

Um problema comum durante a realização de mapeamentos sistemáticos é a presença de trabalhos duplicados. Visando reduzir esse esforço, a ferramenta sugeriu a exclusão de todos os artigos com título igual ou que possuíam a distância de Levenshtein (LEVENSHTEIN, 1966) abaixo de um limiar de quatro caracteres de diferença em relação aos demais trabalhos da lista mesclada. Cada sugestão foi analisada para confirmar ou não a exclusão, apresentando também o ano e os autores do trabalho.

Em seguida foi verificado se os artigos realmente apresentam as *strings* que foram utilizadas nas bases de dados. Alguns trabalhos retornados por essas bases de dados não possuem os assuntos buscados e isso gera fortes indícios de que tais trabalho serão excluídos. A ferramenta proposta analisa alguns metadados (título, *abstract* e palavras-chave), buscando pelas palavras presentes na *string* de busca. Se não houver referência para "web", "ferramenta/avaliação", "usabilidade" e "automático", a ferramenta marca como um trabalho que provavelmente será rejeitado e insere um comentário informando qual das strings de busca não foi encontrada no trabalho.

Após a limpeza dos artigos inválidos, os demais artigos são analisados um a um, com a leitura de seus metadados (titulo, *abstract* e palavras-chave). Se durante a análise

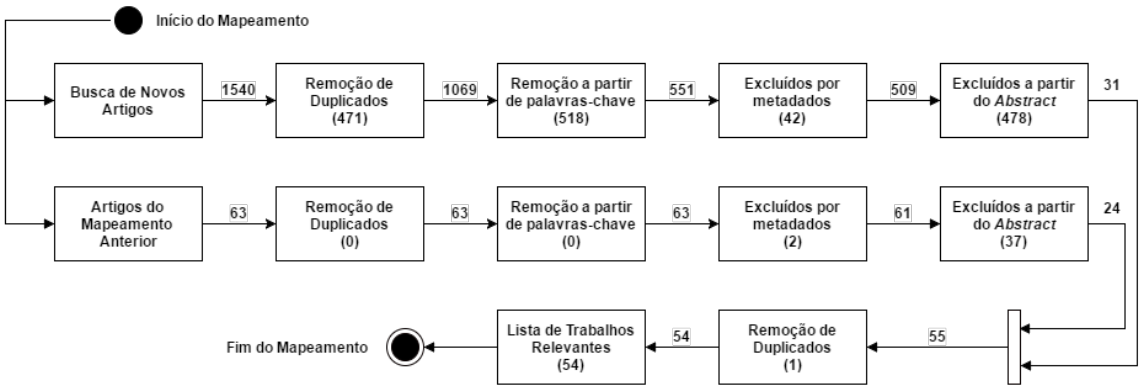


Figura 2 – Diagrama contendo o processo de seleção utilizado neste mapeamento sistemático. Os números dentro das caixas representam a quantidade de artigos impactados na etapa e os números acima das setas são os artigos restantes após cada etapa.

dos trabalhos for encontrada outra palavra-chave não utilizada inicialmente, a *string* de busca é aprimorada e reaplicada nas bases de dados. Os novos artigos passam por todas as etapas de seleção como todos os trabalhos selecionados.

Para assegurar a confiabilidade dos critérios de inclusão e exclusão, foi aplicado um teste de Kappa (FLEISS; COHEN, 1973). Três pesquisadores participaram da etapa de seleção dos estudos primários. Dois deles atuam diretamente com usabilidade e outro que atua em Engenharia de Software de uma forma geral. As divergências entre os pesquisadores que participaram da triagem foram resolvidas com base no bom senso. Foi realizada uma reunião com todos os participantes, onde todas as divergências foram analisadas e chegaram a um consenso. A ferramenta desenvolvida para auxiliar a realização do mapeamento filtrou apenas os novos trabalhos, enquanto todos os trabalhos previamente mapeados por Fernandez et al (FERNANDEZ; INSFRAN; ABRAHÃO, 2011) foram lidos integralmente.

2.1.2.4 Estratégia de Extração de Dados

Para cada estudo selecionado, foi aplicada uma estratégia que provê um conjunto de possíveis respostas para cada sub-questão definida abaixo. Todos os trabalhos foram submetidos à mesma estratégia, facilitando a classificação dos mesmos. Para cada trabalho foram extraídos os seguintes dados, inspirado por outros estudos semelhantes (DYBÅ; DINGSØYR, 2008; JORGENSEN; SHEPPERD, 2007):

- Título do artigo;
- Primeiro autor;
- Conferência/*Journal*;
- Ano da Publicação;

- Universidade ou Grupo de Pesquisa;
- País dos Pesquisadores;
- Resultados da Síntese;

Algumas sub-questões presentes no mapeamento realizado por Fernandez et al. foram relevantes para a questão de pesquisa deste mapeamento e também passaram pela etapa de refinamento. O esquema de classificação final foi composto por tais sub-questões, além da respectiva descrição:

- **Q1 - Abordagem para Automação** (BALBO et al., 2008; IVORY; HEARST, 2001; FERNANDEZ; INSFRAN; ABRAHÃO, 2011): **a) *Nonautomatic***, realizado apenas por especialistas em IHC; **b) Captura automática**, utilizar auxílio de softwares para gravar informações importantes sobre o usuário e o sistema, como dados visuais, falas, ações no teclado e *mouse*; **c) Análise automática**, permite identificar problemas de usabilidade automaticamente; e **d) Crítica automática**, não apenas aponta dificuldades, como sugere melhorias;
- **Q2 - Método** (IVORY; HEARST, 2001; FERNANDEZ; INSFRAN; ABRAHÃO, 2011): **a) Teste**, avaliador observa as interações dos usuários com a interface para determinar problemas de usabilidade; **b) Inspeção**, um avaliador usa um conjunto de critérios ou heurísticas para identificar potenciais problemas de usabilidade em uma interface; **c) Questionários**, usuários provêem *feedbacks* sobre interfaces a partir de entrevistas, *surveys* e semelhantes; **d) Modelagem Analítica**, utiliza-se modelos de usuários e interfaces para gerar predições de usabilidade; e **e) Simulação**, modelos de usuários e interfaces para imitar a interação de usuários e relatar os resultados dessa interação (por exemplo, atividades simuladas, erros e outras medidas quantitativas);
- **Q3 - Esforço Necessário** (IVORY; HEARST, 2001): **a) Esforço mínimo**, não requer uso de interface ou modelagem; **b) Desenvolvimento de modelo**, requer que o avaliador desenvolva um modelo de interface do usuário e/ou modelo de usuário, a fim de utilizar o método; **c) Uso informal (contexto de produção)**, requer a conclusão de tarefas livremente escolhidas, isto é, o uso irrestrito por um usuário ou avaliador; e **d) Uso formal (contexto controlado)**, requer a conclusão de tarefas determinadas selecionadas, ou seja, com uso limitado por um usuário ou avaliador.
- **Q4 - Grau de Intrusão**: **a) Código fonte**, para utilizar a ferramenta é necessário alterar o código fonte da aplicação a ser testada; **b) Infraestrutura**, para utilizar a ferramenta é necessário alterar configurações do servidor da aplicação ou alterações na infraestrutura, como *proxys*; **c) Instalação**, necessário que os participantes ou

avaliadores instalem programas para realizar os testes; e **d) Nenhuma**, não requer alterar o código fonte, servidor ou instalar programas.

- **Q5 - Avaliação Remota**: **a) Sim**, a utilização da ferramenta possibilita a realização de testes sem que os participantes estejam presentes no mesmo ambiente que avaliadores; **b) Não**, para utilizar a ferramenta é necessário estar no mesmo ambiente dos avaliadores;

- **Q6 - Contexto de Avaliações Web**: **a) *Desktop***, pode ser aplicada apenas em aplicações Web no contexto *desktop*, ou seja, em navegadores de computadores de mesa ou *laptops*; **b) *Mobile***, pode ser aplicada apenas em aplicações Web no contexto *mobile*; e **c) Ambos**, podendo ser aplicada tanto em aplicações Web no contexto *mobile*, quanto *desktop*.

2.1.2.5 Síntese

Na síntese foram identificados os principais conceitos de cada estudo primário, que foram organizados em forma de tabela para permitir comparações entre os estudos e classificar categorias. A síntese quantitativa foi baseada na contagem dos estudos primários que foram classificados em cada alternativa das sub-questões de pesquisa. Em seguida foram apresentados os resultados em forma de *bubble-plots*, ou seja, gráficos de dispersão entre sub-questões de pesquisa no formato de bolhas. Por fim, quanto à síntese qualitativa, foram discutidos benefícios e limitações das AUETs.

2.1.3 Resultados

Como pode ser visto na Figura 2, após a condução de pesquisa foram encontrados 1540 novos trabalhos. Dentre eles 471 eram duplicados, 518 foram marcados pela ferramenta por não possuírem as palavras-chave utilizadas na pesquisa, 42 não possuíam *abstracts*, restando 509 artigos a serem lidos por todos os participantes do mapeamento. Dentre esses 509 artigos, 478 foram rejeitados após leitura de título de *abstract* na etapa de triagem, restando 31 artigos de 2009 a 2015 para a etapa de extração.

O mesmo processo foi aplicado aos trabalhos mapeados por Fernandez et al. (FERNANDEZ; INSFRAN; ABRAHÃO, 2011). Dentre os 63 trabalhos que apresentaram métodos de avaliação automáticos, dois trabalhos não foram encontrados e de acordo com o esperado, não haviam trabalhos duplicados. Os 61 trabalhos restantes foram submetidos a uma triagem, sendo 37 rejeitados com base no título e *abstract*. Os 24 artigos restantes foram concatenados aos 31 trabalhos identificados com base nas buscas realizadas. Um trabalho referente a 2009 foi encontrado nos dois mapeamentos, totalizando assim 54 artigos a serem extraídos os dados.

Foi avaliado o índice de concordância entre os participantes baseado no índice Fleiss Kappa (FLEISS; COHEN, 1973). Os valores do índice Kappa e suas respectivas interpretações são apresentados na Tabela 2.

Tabela 2 – Interpretação dos valores do índice Fleiss Kappa (FLEISS; COHEN, 1973).

Valor do Kappa	**Interpretação**
< 0	Sem concordância
0 - 0.19	Concordância pobre
0.20 - 0.39	Concordância razoável
0.40 - 0.59	Concordância moderada
0.60 - 0.79	Concordância substancial
0.80 - 1.00	Concordância quase perfeita

Todos obtiveram um bom grau de concordância, mas apenas os participantes mais experientes em usabilidade obtiveram o maior grau de concordância segundo a escala do Fleiss Kappa presente na Tabela 3.

Tabela 3 – Concordância entre os participantes na etapa de triagem segundo o Índice Fleiss Kappa (FLEISS; COHEN, 1973).

Comparação	**Índice Fleiss Kappa**	**Descrição da Concordância**
Participantes 1 e 2	0,84	Quase Perfeita
Participantes 1 e 3	0,69	Substancial
Participantes 2 e 3	0,73	Substancial
Participantes 1, 2 e 3	0,76	Substancial

Além da concordância entre os participantes é importante observar quais critérios de exclusão mais utilizados. Um trabalho pode ser excluído por mais de um critério de exclusão simultaneamente. A Tabela 4 sumariza as exclusões baseadas nos critérios de exclusão.

Outras avaliações quantitativas foram realizadas visando identificar quais grupos de pesquisa mais trabalham com ferramentas que visam automatizar avaliações de usabilidade, além da evolução das publicações no decorrer dos anos. Dentre os países, pesquisadores norte-americanos apareceram em 10 trabalhos, enquanto italianos apareceram em nove e brasileiros em sete, compondo assim os três países que mais publicaram ferramentas para automatizar avaliações de usabilidade de sistemas Web. Dentre os grupos de pesquisa, a UNICAMP e a Università di Salerno fazem parte de quatro trabalhos, cada uma, sendo os grupos de destaque na área de pesquisa. A distribuição das publicações no decorrer dos anos pode ser vista na Figura 3, possibilitando perceber há pouca diferença na quantidade de trabalhos na área no decorrer dos anos, apesar dos primeiros trabalhos mapeados surgirem em 2001 e a maior quantidade de trabalhos ser em 2011.

Os trabalhos mapeados foram classificadas em conformidade com as sub-questões de pesquisa. Todos os trabalhos mapeados podem ser visualizados em duas tabelas que

Tabela 4 – Resultados obtidos na triagem de trabalhos relevantes. Um artigo excluído pode se encaixar em um ou mais critérios.

Critério	Etapa 1	Etapa 2	Total
Não apresenta um método ou ferramenta que avalia a usabilidade	475	34	509
Não foca em usabilidade	431	3	434
Não apresenta método(s) (semi)automáticos	217	7	224
Não apresenta uma ferramenta	166	16	182
Não aborda domínio Web	114	1	115
Trabalho introdutório ou não conclusivo	1	0	1
Trabalho duplicado ou publicado em fontes distintas	1	0	1
Não escrito em inglês	1	0	1

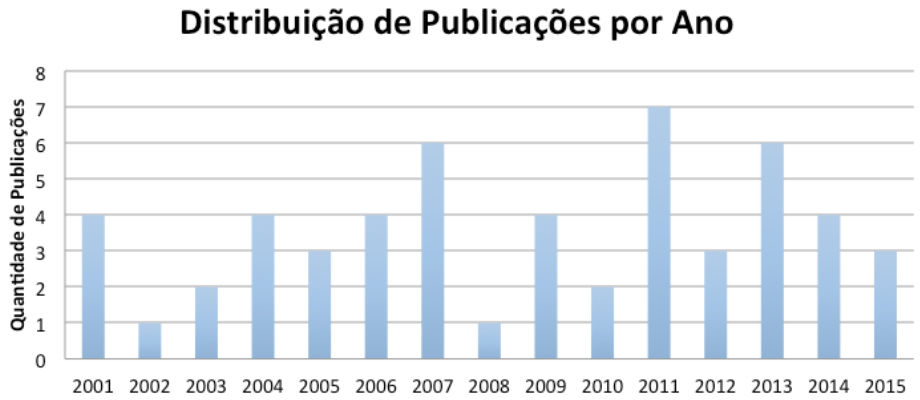


Figura 3 – Distribuição das publicações mapeadas no decorrer dos anos.

contém cada trabalho e as suas respectivas classificações em cada sub-questão. A Tabela 7 possui os 24 trabalhos que foram extraídos do mapeamento previamente realizado por Fernandez et al. (FERNANDEZ; INSFRAN; ABRAHÃO, 2011). A Tabela 8 apresenta os demais 30 trabalhos provenientes das buscas por novos trabalhos na área, entre 2009 e 2015.

Dentre os trabalhos mapeados, Os resultados obtidos na sub-questão Q1 (Abordagem para Automação) reforça que todos os trabalhos de ferramentas que foram classificados possuem ao menos um tipo de automação. A abordagem para automação mais identificada, presente em aproximadamente 96% dos trabalhos, foi a "Análise Automática", que identifica automaticamente problemas de usabilidade. Em seguida, a "Captura Automática" foi identificada em cerca de 57% dos trabalhos. Por fim, apenas 11% dos trabalhos apresentou uma ferramenta que realizasse sugestões de como resolver os problemas de usabilidade ("Críticas Automáticas").

De acordo com os resultados obtidos na sub-questão Q2 (Método), percebe-se a

dominância de ferramentas baseadas em teste de usabilidade, sendo contemplado em mais de 72% das ferramentas. A predominância desse tipo de avaliação de usabilidade ocorre, dentre outros motivos, pois oferece *feedbacks* da utilização de usuários reais e por ser bastante oneroso realizar testes sem automatizar nenhuma parte do processo.

Ao realizar um cruzamento dos métodos (Q2) utilizados e das abordagens de automação (Q1) é possível perceber que apesar da concentração de métodos baseados em testes, apenas dois realizam sugestões de melhorias (Figura 4). Os métodos que mais sugerem correções e melhorias automaticamente são baseados em técnicas de inspeção de usabilidade.

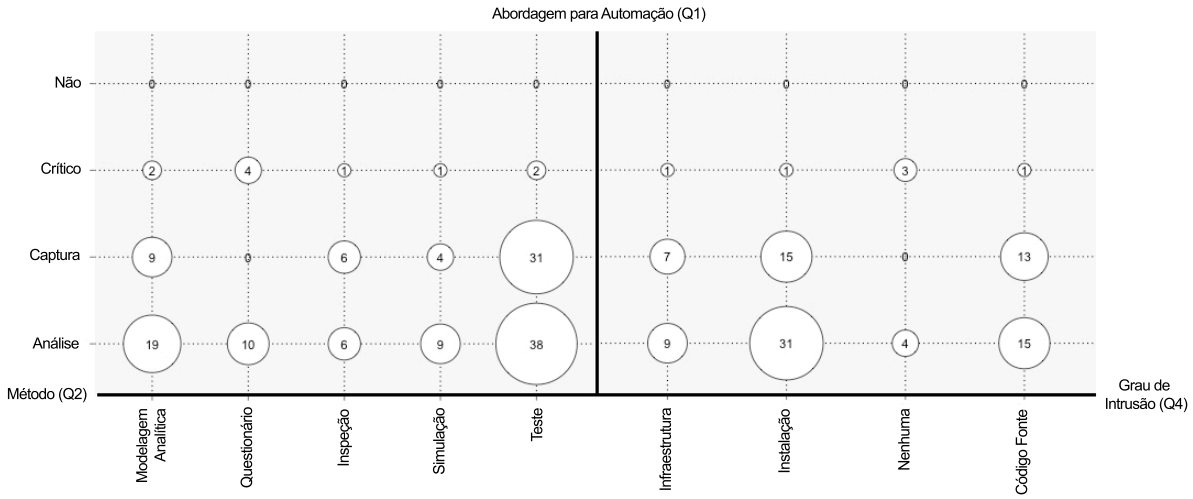


Figura 4 – Mapa Sistemático - abordagem de automação (Q1), método (Q2) e grau de intrusão (Q4).

Observando os resultados obtidos na sub-questão Q3, referente ao nível de esforço necessário para utilizar as ferramentas que automatizam avaliações de usabilidade na Web é perceptível a dominância de ferramentas que necessitam que usuários utilizem o sistema sob teste (formal ou informalmente), contemplando 83% das ferramentas.

Segundo os resultados obtidos na sub-questão Q4 (Grau de Intrusão), percebe-se que apenas cerca de 9% das ferramentas dispensam quaisquer instalações ou alterações no sistema a ser avaliado. Dentre os tipos de intrusão, a mais comum é a necessidade de instalação de algum *software*, presente em cerca de 59% dos trabalhos mapeados. A necessidade de alterações no código fonte do sistema está presente em aproximadamente de 28% dos trabalhos.

Uma forma de contornar a necessidade de alterações no código fonte é a partir de alterações na infraestrutura utilizada pelo sistema avaliado. Por exemplo, ao invés de modificar o código da aplicação torna-se necessário configurar servidores *proxy* que alteram o código fonte automaticamente. Essa abordagem é interessante caso haja restrições de modificações no código, mas podem influenciar na segurança e na performance da aplicação,

impactando desta forma na sua usabilidade. Essa abordagem foi utilizada em cerca de 17% das ferramentas mapeadas.

A sub-questão Q5 (Avaliação Remota) visa identificar quais ferramentas que automatizam avaliações de usabilidade de sistemas Web podem ser utilizadas de forma remota e quais exigem que o avaliador esteja no mesmo ambiente dos participantes da avaliação. Espera-se que a maioria das ferramentas permitam que a execução ocorra remotamente. O resultado obtido é que todas as ferramentas mapeadas permitem que a execução ocorra de forma remota. Logo, percebe-se que a característica "permite avaliações remotas" não é um diferencial de uma ferramenta em relação às demais.

Devido à crescente utilização de dispositivos móveis com acesso à internet, foi proposto avaliar quais os contextos que podem ser avaliados pelas ferramentas (sub-questão Q6): *desktop*, *mobile* ou ambos. A grande maioria das ferramentas, cerca de 87%, permite apenas avaliações em navegadores de computadores de mesa ou *notebooks*. Aproximadamente 11% das ferramentas permitem avaliar a usabilidade de aplicações Web sendo utilizadas tanto em *desktops*, como em *tablets* e *smartphones*. Apenas uma ferramenta foi concebida exclusivamente para o contexto móvel: WUP (BURZACCA; PATERNÒ, 2013). Percebe-se que o contexto *mobile* até o momento foi pouco explorado.

A Figura 5 apresenta um gráfico para a combinação da classificação dos trabalhos em relação às sub-questões: abordagem de automação, método, esforço necessário e contexto de aplicações web.

Dentre os resultados obtidos durante o mapeamento dos trabalhos é possível perceber uma carência em especial de ferramentas aplicáveis ao contexto Web *mobile* e que façam sugestões de melhorias de forma automaticamente (crítica automática).

2.1.3.1 Limitações do Mapeamento Sistemático

Um trabalho de mapeamento pode estar sujeito a uma série de problemas que limitam suas conclusões. Por conta disso, é apresentado a seguir alguns potenciais problemas que podem estar atrelados a este tipo de trabalho:

- **Viés de publicação**: essa ameaça está relacionada à predominância de trabalhos com algum resultado positivo, uma vez que, em geral são mais publicados que os resultados negativos. Ou seja, podem existir ferramentas desenvolvidas mas que não tiveram artigos associados. Para reduzir o problema foram realizadas buscas em periódicos e anais de conferências, uma vez que utilizar apenas periódicos poderia gerar um viés, visto que publicações sobre ferramentas são mais associadas a eventos;

- **Viés de seleção**: essa ameaça está relacionada à distorção de análises estatísticas, por conta dos critérios de seleção de publicações. Pensando em minimizar esse

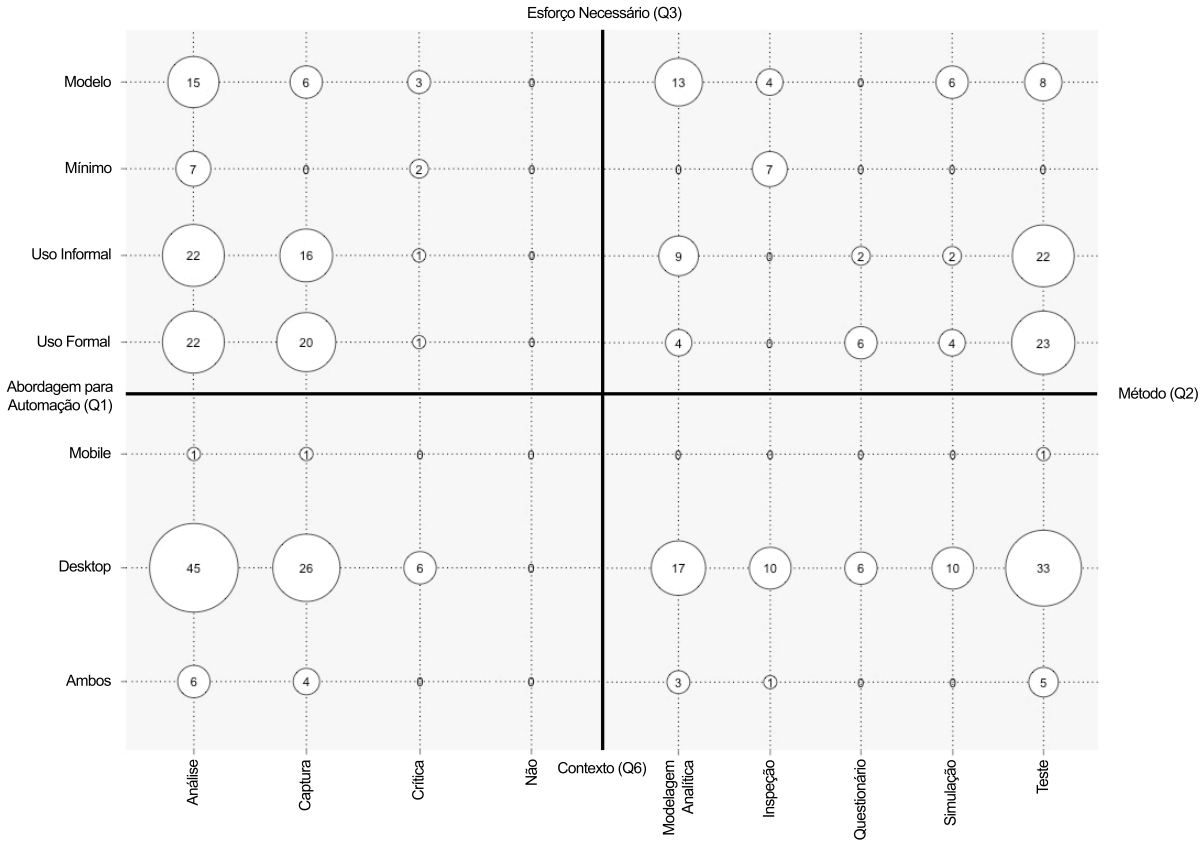


Figura 5 – Mapa Sistemático - abordagem de automação (Q1), método (Q2), esforço necessário (Q3) e contexto de aplicações web (Q6).

problema, foram utilizados apenas critérios de exclusão, visando alcançar o maior número possível de trabalhos que envolvessem o domínio de Web AUETs;

- **Imprecisão / Má classificação**: refere-se à possibilidade de extrações diferentes de acordo com cada revisor. Para reduzir o problema, o trabalho deve ser conduzido por no mínimo dois pesquisadores. Este mapeamento foi conduzido por três pesquisadores, que tiveram sua concordância avaliada.

2.2 Principais Trabalhos Relacionados

Apesar dos diversos trabalhos relacionados à abordagem proposta, alguns são destacáveis por possuírem métodos semelhantes ao proposto. Dentre eles, as ferramentas UsAGE, USABILICS e WELFIT merecem destaque.

A ferramenta *User Action Graphing Effort* (UsAGE) (UEHLING; WOLF, 1995) é um dos trabalhos pioneiros na automatização de avaliações de usabilidade. A ferramenta é baseada na comparação de um usuário "experiente" e um "novato", identificando pontos de diferença nas suas interações. Com essa ferramenta é possível avaliar *softwares desktop* que possuem interfaces construídas a partir da ferramenta TAE Plus, que dentre outras funcio-

nalidades, possibilita a gravação automática das ações realizadas na interface. A UsAGE deve ser utilizada durante sessões de testes laboratoriais. A partir dos *logs* capturados são realizadas comparações automáticas que geram um grafo contendo as ações realizadas por cada usuário. A abordagem proposta baseou-se na UsAGE, entretanto com foco em avaliações de aplicações Web e comparações de grupos de usuários "experientes" e "novatos" ou de sessões de "referência" e demais sessões. Outro diferencial é a classificação das ações, tornando mais simples a interpretação dos resultados. Essa ferramenta provavelmente não foi mapeada por Fernandez *et al.* (FERNANDEZ; INSFRAN; ABRAHÃO, 2011) por não abordar sistemas Web.

Outra ferramenta que utiliza um método semelhante ao proposto é a USABILICS, que realiza avaliações remotas e semiautomáticas de usabilidade de aplicações Web. Ao criar cada tarefa do teste com a ferramenta, são definidas as ações esperadas. Em seguida, a ferramenta compara as ações esperadas com as ações realizadas por usuários durante os testes, calculando a similaridade entre essas sequências de eventos. Para capturar os eventos dos usuários, são necessárias alterações no código fonte da aplicação a ser testada. Como resultado, a ferramenta calcula o índice de usabilidade de cada tarefa (VASCONCELOS; BALDOCHI JR., 2012). Apesar das semelhanças, a UseSkill *Control* gera tabelas e grafos comparando as ações realizadas por "novatos" e "experientes", facilitando a identificação de possíveis problemas de usabilidade, além de não ser intrusiva.

A WELFIT (SANTANA; BARANAUSKAS, 2015) é uma ferramenta que suporta testes remotos/não-remotos, síncronos/assíncronos e formais/informais. Ela realiza a captura de *logs* automaticamente do lado do cliente, exigindo alteração do código fonte da aplicação. Durante a comparação automática dos *logs*, a ferramenta leva em consideração a distância, a partir da heurística *Sequence Alignment Method* (SAM) e o tempo médio de cada evento. Os resultados obtidos a partir da ferramenta são em forma de relatórios estatísticos e grafos dos eventos capturados.

Apesar da abordagem da WELFIT ser semelhante à proposta neste trabalho, ela apresenta problemas caso haja uma grande massa de *logs*, a legibilidade dos grafos gerados por ela fica comprometida, dificultando a identificação pontual dos problemas de usabilidade. Para amenizar esse problema, a UseSkill permite a personalização dos eventos capturados e a configuração da visualização de grafos, além de apresentar tabelas contendo os *logs* classificados e detalhados.

Geng e Tian (GENG; TIAN, 2015a) apesar de não terem desenvolvido uma ferramenta, eles propuseram um método para identificar problemas de usabilidade relacionados à navegação baseado na comparação de padrões de uso reais e preditos. Os padrões de uso reais são extraídos utilizando-se algoritmos de *Web Usage Mining* em *logs* de servidores *Web*. Os padrões de uso preditos são obtidos através da simulação do comportamento ideal do usuário utilizando-se modelos cognitivos. As diferenças encontradas entre esses

dois padrões de uso são usadas para descobrir problemas e sugerir ações corretivas para melhorar a usabilidade. Porém, a utilização de *logs* de servidor, apesar de mais simples, não fornece informações detalhadas, limitando-se a calcular métricas relacionadas às diferenças nos padrões de uso reais e preditos. Além disso, o desenvolvimento de modelos cognitivos é uma tarefa complexa e que exige especialistas na área.

De uma forma resumida, as principais diferenças e contribuições dos trabalhos relacionados evidenciados podem ser aferidos na Tabela 5.

Tabela 5 – Interpretação dos valores do índice Fleiss Kappa (FLEISS; COHEN, 1973).

Atributo	**UsAGE**	**USABILICS**	**WELFIT**	**Geng e Tian**	**UseSkill**
Artefato gerado	ferramenta	ferramenta	ferramenta	método	ferramenta
Contexto	*Desktop*	Web	Web	Web	Web
Origem dos *logs*	cliente	servidor	cliente	servidor	cliente
Relatórios	grafos	métricas	estatística e grafos	métricas	grafos, métricas e listas
Avaliação Web *mobile*	não	não	não	não	sim
Abordagem de automação	captura e análise	análise e crítica	captura e análise	análise	captura e análise

2.3 Considerações Finais

Este capítulo apresentou a extensão de mapeamento sistemático, com detalhes sobre o processo de identificação, triagem, sumarização e discussão dos principais trabalhos relacionados à automação de avaliações de usabilidade de sistemas Web. Esse estudo fornece uma visão geral da linha de pesquisa do trabalho realizado, agrupando os resultados e identificando lacunas passíveis de investigação.

Tabela 6 – Estudos Primários Selecionados.

Id	Referência	Id	Referência
s01	(SCHWERZ; MORANDINI; SILVA, 2007)	s28	(MOSQUEIRA-REY et al., 2009)
s02	(SCHOLTZ, 2001)	s29	(GRIGERA; GARRIDO; RIVERO, 2014)
s03	(ATTERER; SCHMIDT, 2005)	s30	(CALLEROS; GARCÍA; VANDERDONCKT, 2013)
s04	(ALONSO-RÍOS et al., 2009)	s31	(PAUL; YADAMSUREN; ERDELEZ, 2012)
s05	(XU; XU, 2007)	s32	(CASSINO; TUCCI, 2011b)
s06	(VANDERDONCKT; BEIREKDAR; NOIRHOMME-FRAITURE, 2004)	s33	(DINGLI; CASSAR, 2014)
s07	(WEST; LEHMAN, 2006)	s34	(COLETI; MORANDINI; NUNES, 2013)
s08	(OBENDORF; WEINREICH; HASS, 2004)	s35	(VARGAS; WEFFERS; ROCHA, 2011b)
s09	(BEDNARIK et al., 2004)	s36	(CHYNAŁ; SOBECKI; SZYMAŃSKI, 2014)
s10	(RAMLI; JAAFAR, 2008)	s37	(NEBELING; SPEICHER; NORRIE, 2013)
s11	(KATSANOS; TSELIOS; AVOURIS, 2006)	s38	(CASSINO; TUCCI, 2011a)
s12	(PAGANELLI; PATERNÒ, 2002)	s39	(VARGAS; WEFFERS; ROCHA, 2011a)
s13	(ATTERER; WNUK; SCHMIDT, 2006)	s40	(DHOUIB; TRABELSI; ABDALLAH, 2013)
s14	(NORMAN; PANIZZI, 2006)	s41	(CASSINO et al., 2015)
s15	(CHENG-YING; YAN-SHENG, 2004)	s42	(GENG; TIAN, 2015b)
s16	(QI; REYNOLDS; PICARD, 2001)	s43	(DAVIS; SHIPMAN, 2011)
s17	(CHI et al., 2003)	s44	(BURZACCA; PATERNÒ, 2013)
s18	(BLACKMON; KITAJIMA; POLSON, 2005)	s45	(POUR; CALVO, 2011)
s19	(LÓPEZ; FAJARDO; ABASCAL, 2007)	s46	(VASCONCELOS; JR, 2012)
s20	(LISTER, 2001)	s47	(ALBANESI et al., 2011)
s21	(CHATLEY et al., 2003)	s48	(HARMS; GRABOWSKI, 2014)
s22	(LI; KIT, 2005)	s49	(CASSINO; TUCCI, 2010)
s23	(NAKAMICHI et al., 2007)	s50	(SPEICHER; BOTH; GAEDKE, 2013)
s24	(PASCUAL; DÜRSTELER, 2007)	s51	(KIURA; OHIRA; MATSUMOTO, 2009)
s25	(POWER; PETRIE; MITCHELL, 2009)	s52	(SANTANA; BARANAUSKAS, 2015)
s26	(VARGAS; WEFFERS; ROCHA, 2010)	s53	(HONG; LANDAY, 2001)
s27	(HUMAYOUN et al., 2012)	s54	(BUCHHOLZ et al., 2007)

Tabela 7 – Estudos Primários selecionados para Extração dos Dados provenientes do mapeamento realizado por Fernandez et al. (FERNANDEZ; INSFRAN; ABRAHÃO, 2011), com trabalhos de 1996 a 2009.

Id	**Q1**				**Q2**					**Q3**				**Q4**				**Q6**		
	a	b	c	d	a	b	c	d	e	a	b	c	d	a	b	c	d	a	b	c
s01		x	x		x			x			x	x			x			x		
s02		x	x		x		x						x	x		x		x		
s03			x			x		x			x					x		x		
s04			x	x		x				x							x	x		
s05		x	x		x				x			x		x				x		
s06			x			x				x						x		x		
s07		x	x		x		x						x			x		x		
s08		x	x		x		x						x			x		x		
s09				x		x		x			x						x	x		
s10		x	x		x		x					x	x			x		x		
s11			x					x	x		x					x		x		
s12		x	x		x				x		x		x			x		x		
s13		x		x	x			x					x		x			x		
s14		x	x	x	x		x						x	x				x		
s15		x	x		x			x			x	x			x			x		
s16		x	x		x				x				x			x		x		
s17			x		x				x		x		x				x	x		
s18			x			x		x	x		x					x		x		
s19		x	x		x		x					x	x		x	x		x		
s20			x		x							x	x			x		x		
s21		x			x				x				x			x		x		
s22			x					x	x		x					x		x		
s23		x	x		x							x	x			x		x		
s24			x		x				x			x			x	x		x		

Tabela 8 – Estudos Primários selecionados para Extração dos Dados após Condução da Pesquisa entre 2009 e 2015.

Id	**Q1**				**Q2**					**Q3**				**Q4**				**Q6**		
	a	b	c	d	a	b	c	d	e	a	b	c	d	a	b	c	d	a	b	c
s25			x		x								x	x		x		x		
s26			x		x			x				x				x		x		
s27		x	x		x			x				x		x		x		x		
s28			x	x		x			x		x						x	x		
s29		x	x		x							x		x				x		
s30			x			x				x							x	x		
s31		x	x		x								x			x		x		
s32			x			x				x						x		x		
s33			x			x				x						x				x
s34		x	x		x								x			x		x		
s35			x		x		x					x				x		x		
s36		x	x		x							x		x				x		
s37		x	x		x			x			x		x	x						x
s38			x					x			x					x		x		
s39			x		x			x				x				x		x		
s40			x	x		x				x						x		x		
s41			x			x				x				x				x		
s42		x	x	x	x			x			x	x			x			x		
s43			x		x			x			x	x			x	x				x
s44		x	x		x								x	x		x			x	
s45		x	x		x								x		x			x		
s46		x	x		x			x			x		x	x				x		
s47		x	x		x							x	x			x		x		
s48		x	x		x							x	x	x				x		
s49			x					x			x					x		x		
s50		x	x		x			x				x		x				x		
s51		x	x		x							x		x				x		
s52		x	x		x							x		x						x
s53		x	x		x							x			x					x
s54		x	x		x			x					x		x					x

3 Abordagem Proposta

Com o intuito de reduzir os custos e a complexidade envolvidos em avaliações de usabilidade de sistemas Web, este trabalho propõe uma abordagem composta por um método e uma ferramenta para automatizar parte desse tipo de avaliação. O método propõe diretrizes com base na captura e análise de *logs* de utilização. A ferramenta, denominada UseSkill, é responsável por implementar os conceitos presentes no método, possibilitando a sua utilização em ambientes de produção, além de permitir avaliar os resultados obtidos com o uso da abordagem.

3.1 Método

O método proposto neste trabalho baseia-se na captura remota de ações realizadas por usuários de sistemas Web e na comparação entre as sessões de uso consideradas "adequadas" (um padrão de uso considerado bom) e "inadequadas" (um padrão de uso considerado ruim). A ideia por trás do método é identificar quais partes das funcionalidades influenciam negativamente na utilização dos usuários e fazem com que eles divirjam entre si, apontando assim possíveis pontos problemáticos.

Para que seja possível capturar os dados corretos, analisá-los e apoiar na identificação de problemas de usabilidade, o método proposto possui quatro etapas: captura dos *logs* de utilização, preparação dos dados, análise dos dados e geração de relatórios. A Figura 6 apresenta as etapas do método proposto.

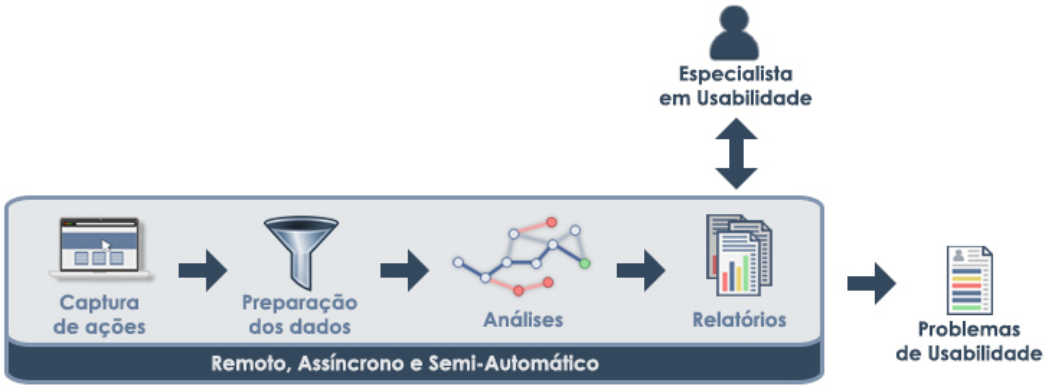


Figura 6 – Abordagem proposta para avaliar a usabilidade de sistemas Web.

Para entender um pouco melhor as atividades realizadas por especialistas em usabilidade e por ferramentas que implementem o método proposto, a Figura 7 mostra as atividades, as transições, os objetos envolvidos, além das raias representando as etapas da avaliação.

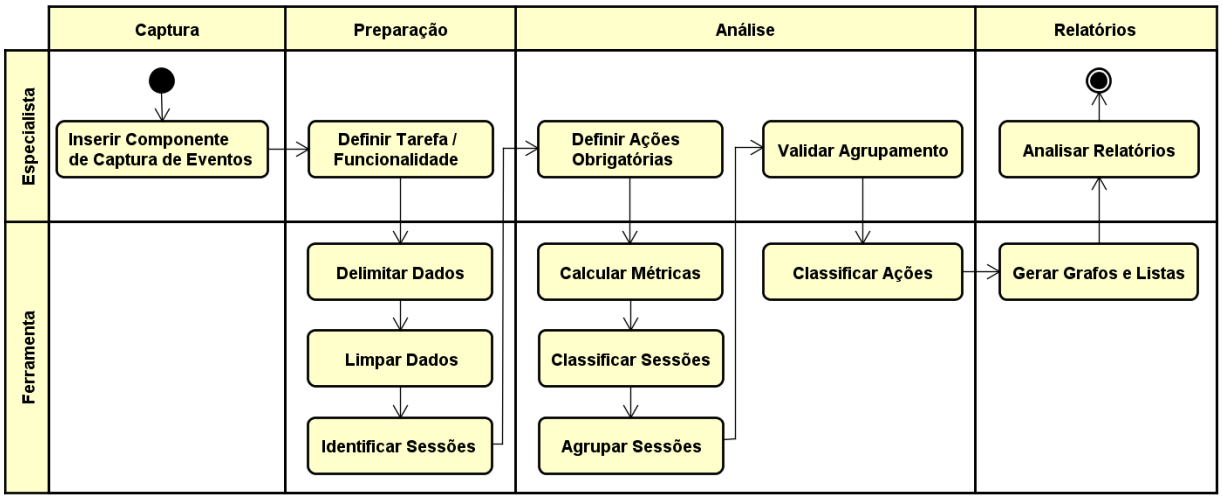


Figura 7 – Diagrama de atividade contendo os objetos, ações, transições e as etapas da avaliação com base no método proposto.

3.1.1 Captura de *logs*

Diferentemente de testes laboratoriais, que avaliam um usuário por vez, este método propõe que as avaliações sejam remotas e assíncronas, o que dispensa a presença de especialistas e usuários no local e no instante da avaliação. Essas características permitem que diversos usuários sejam avaliados simultaneamente sem acrescer custos com translado e preparação de ambiente, além de simplificar a logística durante a avaliação (BRUUN et al., 2009).

Para permitir que as avaliações sejam remotas, assíncronas e com etapas automatizadas, os *logs* devem possuir informações importantes sobre as ações realizadas durante as utilizações do sistema. A captura destas ações ocorre por meio de um Componente de Captura de Eventos em *Javascript* que interage com *navegadores Web*. Ele é personalizável, permitindo capturar informações específicas que podem ser úteis durante a análise dos *logs*. Entretanto, dentre as informações capturadas, algumas são obrigatórias por serem cruciais para diferenciá-las e permitirem realizar comparações entre as utilizações do sistema, tais como:

- **Tempo**: horário que a ação ocorreu;
- **Tipo da ação realizada**: clique, preenchimento de campo, *mouseover*, etc.;
- **Elemento que sofreu a ação**: botão, *link*, campo de texto, etc.;
- **Onde a ação ocorreu**: qual página, geralmente utiliza-se a URL, mas também podem ser utilizados metadados;
- **Quem fez a ação**: um identificador de qual usuário realizou a ação.

3.1.2 Preparação dos dados

Para que os dados de utilização (*logs*) capturados estejam aptos durante a análise, é necessário que eles sejam organizados em um padrão e que as informações indesejadas sejam removidas. A preparação dos dados proposta no método baseia-se em definições do pré-processamento de dados em *Web Usage Mining* (MOBASHER, 2006).

A primeira fase da preparação é a identificação de quais ações capturadas fazem parte da funcionalidade a ser analisada. Essa delimitação dos dados é importante para que cada funcionalidade possa ser analisada isoladamente. Em seguida ocorre a limpeza dos dados, removendo tipos de ações indesejadas. Por exemplo, durante uma avaliação, caso não haja a necessidade de acompanhar quando há "*mouseover*" em algum elemento da interface, esse tipo de ação deve ser desconsiderada, para diminuir os dados ruidosos.

Com os dados delimitados e limpos, ocorre a identificação das sessões de uso. Cada sessão representa um conjunto de ações realizadas por determinado usuário ao utilizar uma funcionalidade de um sistema. Seu conceito é semelhante à de uma utilização da funcionalidade. Um usuário, por exemplo, pode utilizar diversas vezes a mesma funcionalidade do sistema, gerando assim diversas sessões de uso.

Resumidamente, a preparação dos dados corresponde à etapa em que as ações capturadas são transformadas nas sessões de uso a serem avaliadas.

3.1.3 Análise dos dados

Com as sessões (conjuntos de ações) capturadas, o próximo passo é calcular métricas para medir a qualidade de uso de cada sessão. O método proposto sugere o cálculo de duas métricas: eficácia e eficiência. A seleção delas foi baseada nas métricas SUM (SAURO; KINDLUND, 2005) e MUSiC (BEVAN, 1995), além de serem relacionadas diretamente ao conceito de usabilidade proposto na ISO 9241-11 (ISO, 1998).

O cálculo da eficácia baseia-se na quantidade de ações obrigatórias (AO) realizadas pelo usuário ao utilizar determinada funcionalidade. Por exemplo, considere o caso de uma funcionalidade que possui cinco ações a serem realizadas obrigatoriamente pelo usuário para que ele a utilize corretamente, mas o usuário realizou apenas três. A eficácia é o percentual de ações obrigatórias realizadas na sessão. Nesse caso hipotético, a eficácia seria de 60%. A Fórmula 3.1 apresenta como a eficácia é calculada.

$$Efica_s = \frac{AO_s * 100}{AO} \tag{3.1}$$

A variável AO_s representa a quantidade de ações obrigatórias contidas na sessão e a variável AO representa o total de ações obrigatórias da funcionalidade. O valor da eficácia da sessão ($Efica_s$) varia entre 0 e 100. Para a funcionalidade, o cálculo da eficácia

é a média das eficácias das sessões, como pode ser visto na Fórmula 3.2, onde s representa as sessões ds usuários que utilizaram a funcionalidade.

$$Efica_f = \frac{\sum_{s=1} Efica_s}{s} \tag{3.2}$$

A eficiência é a proporção entre a eficácia e o esforço demandado, nesse caso medido em função do tempo e quantidade de ações realizadas. Caso um usuário tenha atingido todos os objetivos, mas com alto tempo, sua sessão de uso terá eficácia alta e eficiência baixa. Para calcular a eficiência de uma sessão de uso é necessário calcular a eficácia e dividir sobre a quantidade de ações e de tempo despendido na sessão, como pode ser visto na Fórmula 3.3.

$$Efici_s = \frac{Efica_s}{(\frac{A_s}{mA_ok}) * (\frac{T_s}{mT_ok})} \tag{3.3}$$

A variável A_s equivale à quantidade de ações da sessão, mA_ok é a quantidade de ações da sessão que foi realizada corretamente e com menor número de ações, T_s é o tempo (em segundos) despendido durante a sessão, mT_ok é tempo da sessão correta que foi realizada mais rapidamente. Para o cálculo da eficiência de uma funcionalidade é utilizada a média das eficiências das sessões ($Efici_s$), segundo a Fórmula 3.4.

$$Efici_f = \frac{\sum_{s=1} Efici_s}{s} \tag{3.4}$$

Em seguida, com as métricas calculadas para cada uma das sessões, elas são classificadas como “boas” ou “ruins”, de acordo com as suas eficácias e eficiências. Caso a sessão tenha bons índices de eficácia e eficiência, ela deve ser classificada como uma “boa” sessão. Caso contrário, se os índices forem baixos, a sessão deve ser classificada como “ruim”.

Em situações onde o cálculo das métricas é inviável, outra estratégia possível é classificar os usuários como: “experientes”, ou seja, já conhecedores do funcionamento do sistema sob avaliação; e “novatos”, que não possuem tanta intimidade com o sistema e, por conta disso, são mais suscetíveis a esbarrar em problemas de usabilidade.

Desta forma, as sessões de usuários “experientes” são consideradas “boas” e as dos usuários “novatos” são “ruins”. Segundo Nielsen, os usuários que já aprenderam a utilizar o sistema tendem a melhorar suas performances, dessa forma, caso não seja possível mensurar a qualidade de cada sessão, avaliar a experiência dos usuários é uma saída (NIELSEN; LANDAUER, 1993).

Com as sessões classificadas, a ideia é agrupá-las em: Grupo de Sessões Referência (GSR), contendo as utilizações das funcionalidades do sistema de maneira esperada; ou

Grupo das Demais Sessões (GDS), que não lograram êxito, realizaram ações demasiadamente ou demoraram muito para finalizar a sessão. Após a classificação das sessões em GSR e GDS é possível comparar tais grupos a fim de encontrar diferenças entre eles. Essas comparações servem como base para a classificação de cada uma das ações contidas nas sessões.

As ações que foram mais frequentes no GSR são classificadas como "obrigatórias" (AO), ou seja, passos que os usuários devem realizar para utilizar a funcionalidade corretamente. As demais ações contidas no GSR e que não foram classificadas como AO, são as ações "corretas" (AC). As ações mais frequentes no GDS e que não estão entre as mais frequentes no GSR são as "problemáticas" (AP). Por fim, as ações contidas no GDS, que não fazem parte do GSR e que não foram classificadas como AP, são consideradas "alertas" (AA).

A classificação das ações é uma etapa importante para gerar relatórios que apontem para as partes de funcionalidades com maior possibilidade de possuírem problemas de usabilidade. As ações AP e AA servem para dar indícios de onde estão os locais problemáticos e as métricas apontam quais sessões enfrentaram mais dificuldades.

3.1.4 Geração de relatórios

A partir das métricas e classificações realizadas, devem ser gerados relatórios que facilitem a análise e interpretação dos dados por parte de especialistas em usabilidade. A proposta baseia-se na possibilidade de ter uma visão geral da usabilidade e ao mesmo tempo permitir análises aprofundadas em determinadas utilizações.

Os relatórios para análises rápidas e que permitem aos especialistas terem noções gerais da usabilidade são baseados em grafos. Os nós dos grafos representam as ações realizadas, sendo que os nós redondos correspondem às ações mais frequentes e os quadrados são as ações que não estão entre as mais frequentes. As arestas do grafo apontam quais caminhos foram percorridos pelos usuários, as cores dos nós indicam a classificação de cada ação, a largura das arestas apontam quais caminhos foram mais percorridos e o tamanho de cada nó (diâmetro ou tamanho do lado) representa quais ações foram mais realizadas.

Esses grafos são versáteis, podendo ser utilizados, por exemplo, para visualizar cada sessão de uso ou para apresentar as ações mais realizadas na funcionalidade. A Figura 8 exemplifica um grafo de uma sessão de uso, baseado no método proposto.

Apesar dos grafos apresentarem uma visão geral das utilizações, para identificar problemas de usabilidade são necessários mais detalhes das ações realizadas. O método sugere a utilização de listas contendo as ações realizadas sequencialmente e a possibilidade de selecionar ações para ver suas informações capturadas, como ação, elemento, local, horário e usuário. Com isso é possível, por exemplo, verificar quanto tempo o usuário

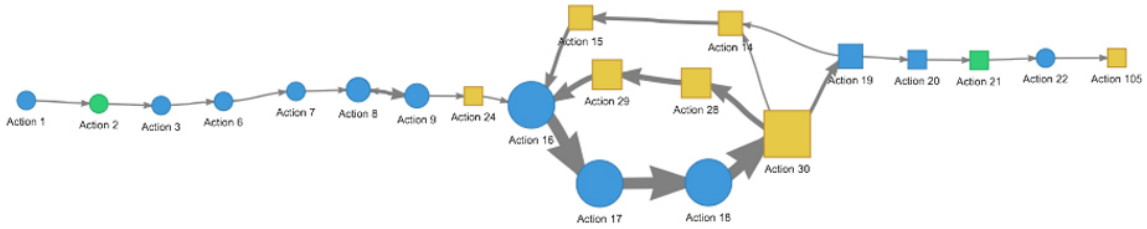


Figura 8 – Grafo exemplificando uma sessão de uso. Os nós azuis são AO, os amarelos AA e os verdes AC. Esse grafo não possui AP.

demorou entre uma ação e outra, ou identificar quais elementos receberam mais ações repetidamente.

De forma resumida, o método aqui descrito baseia-se na captura de *logs* de utilização com dados sobre o tempo, o tipo de ação realizada, o elemento, a página e o usuário. Em seguida identifica as sessões de uso e remove dados indesejados. A partir das sessões, as métricas eficácia e eficiência são calculadas, possibilitando o agrupamento das sessões de uso no GSR ou GDS. A comparação dos grupos serve de base para a classificação das ações realizadas. Por fim, são gerados grafos e listas de ações que apontam indícios para especialistas em usabilidade detectarem possíveis problemas.

3.2 UseSkill

A ferramenta UseSkill implementa o método proposto para auxiliar avaliações de usabilidade. Ela permite a realização de avaliações de usabilidade em contextos controlados e em ambientes de produção. Inicialmente foi proposto o módulo UseSkill *Control* (USC), que visa auxiliar a realização de testes de usabilidade remotos e que necessita da definição de roteiros, tarefas e questionários. Esse tipo de avaliação, em um contexto controlado, é classificada como formal, ou seja, requer a execução de tarefas específicas, previamente selecionadas por um especialista (IVORY; HEARST, 2001).

O módulo USC auxilia durante a etapa de concepção, execução e análise dos resultados do teste. A USC baseia-se na comparação das interações entre os grupos compostos por usuários "experientes" e "novatos", apontando indícios de problemas. A USC possibilita a realização de questionários intercalados entre tarefas e permite que usuários enviem *feedbacks*, capturando dados sobre a satisfação ao utilizar o sistema.

Além do USC, este trabalho também propõe o módulo UseSkill *OnTheFly* (USOTF),

que contempla avaliações de usuários em seu ambiente de produção, executando suas atividades do dia a dia. Essa necessidade surgiu pois convidar usuários, criar roteiros e preparar o sistema para ser testado envolve custos e complexidade logística. Dessa forma, a USOTF não necessita da definição de roteiros, nem convidar usuários a realizar um conjunto de tarefas visando apenas avaliar suas interações. Apesar de não coletar informações sobre a satisfação durante o uso do sistema, a análise dos usuários é baseada em suas atividades corriqueiras.

Para entender melhor como a ferramenta foi projetada e desenvolvida, a Seção 3.2.1 apresenta a Arquitetura da UseSkill. Em seguida, a Seção 3.2.2 apresenta a UseSkill *Control* e a Seção 3.2.3 apresenta a UseSkill *OnTheFly*, detalhando o funcionamento dos módulos e apresentando seus desafios e limitações.

3.2.1 Arquitetura

Geralmente a definição da arquitetura precede o desenvolvimento do software, desempenhando papel importante na comunicação entre desenvolvedores e partes interessadas (*stakeholders*). Corresponde à definição de quais componentes e como eles interagem entre si, auxiliando a redução de retrabalho durante o desenvolvimento do software (PRESSMAN, 2005). Esta seção apresenta a arquitetura da ferramenta UseSkill, contemplando os módulo USC e USOTF.

A Figura 9 apresenta um diagrama contendo os componentes presentes na ferramenta proposta. Há uma divisão em dois blocos principais, contendo um conjunto de componentes, o bloco “UseSkill” e outro “Ferramentas Auxiliares”, além de um componente desacoplado de ambos, denominado “Captura de Eventos”.

O bloco UseSkill representa a aplicação contida no servidor Web da ferramenta proposta. Como pode ser observado, a sua arquitetura segue o padrão de desenvolvimento MVC (*Model-View-Controller*), visando a separação entre a interface do usuário e a lógica do sistema. As camadas Modelo, Visão e Controle exercem esta divisão de funcionalidades. No padrão MVC, o Modelo realiza a manipulação dos dados internos de uma aplicação e se comunica especialmente com o armazenamento de dados. A camada de Visão apresenta a interface do usuário, realizando chamadas ao Controlador, que por sua vez busca os dados do Modelo desejado. Por fim, a camada de Controle é responsável por funcionalidades que envolvem o comportamento da aplicação, controla os fluxos entre as camadas de Visão e Modelo, e gera a resposta ao usuário.

Dentro do bloco UseSkill, além das *Views* e Controladores, há dois módulos denominados UseSkill *Control* e UseSkill *OnTheFly*. Nesse módulo são definidos os componentes responsáveis pelo Modelo, o processamento dos *logs*, geração de Relatórios e os DAOs (*Data Access Object*), responsáveis pela persistência dos dados, separando as regras de

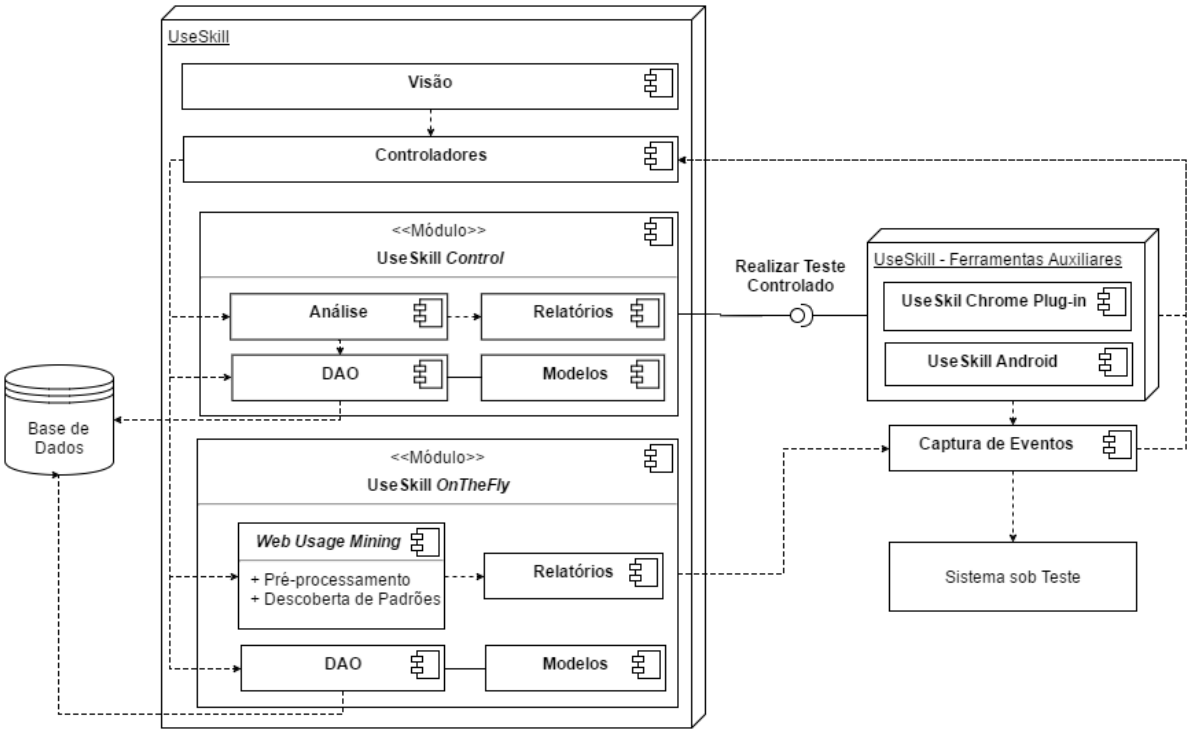


Figura 9 – Diagrama de Componentes da UseSkill, agrupando módulos internos e ferramentas auxiliares.

negócio das regras de acesso a banco de dados.

Além do grande bloco da UseSkill, há o bloco de Ferramentas Auxiliares, que contém a implementação de ferramentas que auxiliam na realização de testes controlados na UseSkill *Control*. Essas ferramentas auxiliares realizam a inserção do Componente de Captura de Eventos no sistema sob avaliação e realizam a comunicação com os Controladores da UseSkill. O Componente de Captura de Eventos também faz chamadas para os Controladores da UseSkill, mas apenas quando o teste é por meio da UseSkill *OnTheFly*, que não possui ferramentas auxiliares.

Quanto às tecnologias utilizadas, a Figura 10 apresenta com mais detalhes quais tecnologias são utilizadas na UseSkill, divididas de acordo com seus respectivos módulos (USC e USOTF), além de agrupar quais tecnologias são da camada de *Front-end* e *Back-end*. O *Front-end* representa os componentes manipulados pelo usuários, enquanto o *Back-end* são os componentes presentes no servidor.

O *Back-end* foi construído sobre a infraestrutura provida pela *Amazon Elastic Compute Cloud - EC2*[1], auxiliando na configuração de servidores de forma redimensionável para aplicações Web. Adotou-se também Java como linguagem de programação, em conjunto com o *framework* VRaptor[2], que possui boa documentação e padrões simples que permitem a criação de aplicações *RESTFul* sem grandes dificuldades. Para o SGBD

[1] https://aws.amazon.com/ec2/
[2] http://www.vraptor.org/

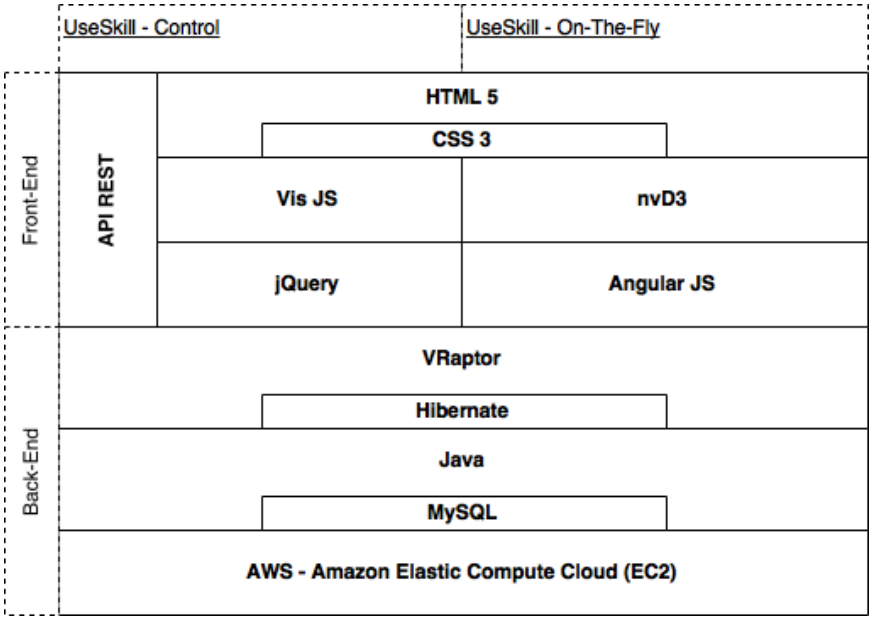


Figura 10 – Diagrama com as principais tecnologias utilizadas durante o desenvolvimento da UseSkill.

(Sistema de Gerenciamento de Banco de Dados) foi utilizado o MySQL[3], juntamente com o *framework* Hibernate[4], um dos padrões amplamente adotados ao utilizar o VRaptor. A API (*Application Programming Interface*) REST é o canal de comunicação entre as ferramentas auxiliares e o servidor da UseSKill.

Quanto à camada de comunicação com os usuários (*Front-end*), diferentemente do *Back-end*, os módulos UseSkill *Control* e *OnTheFly* possuem boa parte das tecnologias distintas. A USC por ter sido desenvolvida primeiro foi utilizado apenas o *framework* jQuery[5], que auxilia a manipulação de elementos do DOM (*Document Object Model*) e a biblioteca VisJS[6], que auxilia na geração de grafos dos relatórios. A USOTF utiliza o *framework* AngularJS[7], que além de auxiliar na manipulação de DOM, também utiliza o padrão MVC, facilitando o desenvolvimento e tornando o código mais testável, além da biblioteca nvD3[8] para a geração de gráficos. As interfaces de ambos os módulos são semelhantes, pois utilizam as mesmas folhas de estilo (CSS ou *Cascading Style Sheets*).

Por fim, a Figura 11 apresentada um diagrama de classe que contém as principais classes do módulo UseSkill *Control* e os relacionamentos entre si.

De acordo com o diagrama de classe da Figura 11 constata-se que os testes são compostos essencialmente por Tarefas, um Questionário e um grupo de usuários convidados a participar do Teste. A relação direta entre Usuário e Teste serve para associar qual usuário

[3] https://www.mysql.com/
[4] http://hibernate.org/orm/
[5] https://jquery.com/
[6] http://visjs.org/
[7] https://angularjs.org/
[8] http://nvd3.org/

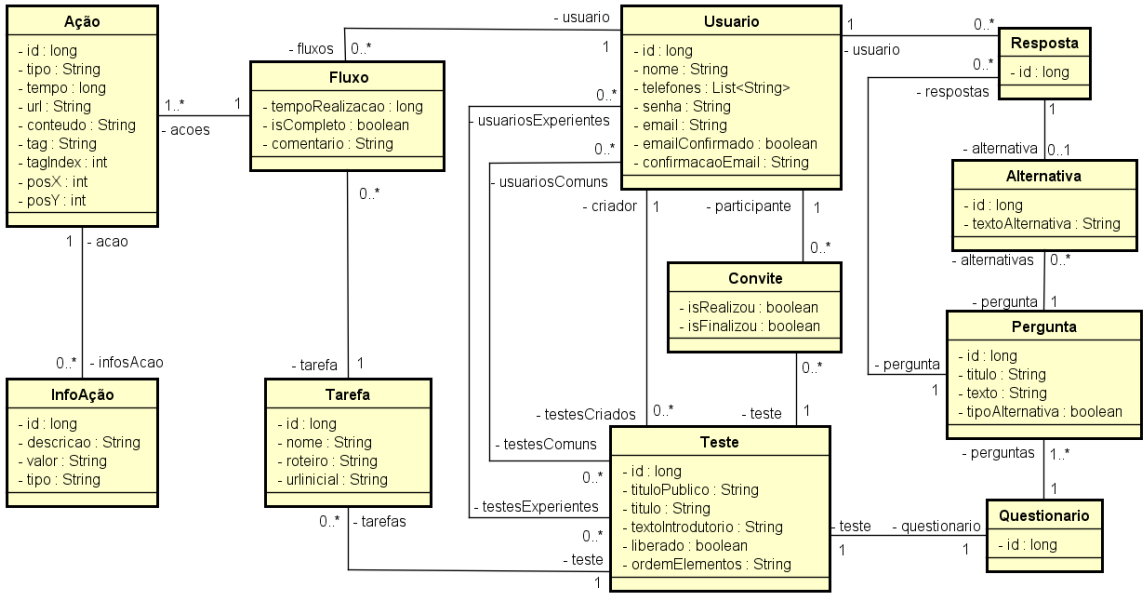


Figura 11 – Diagrama de Classe contendo as principais classes da UseSkill *Control*.

criou o teste. Os Convites relacionam os Usuários que realizarão ou realizaram o Teste. Quanto às perguntas, há a possibilidade destas serem subjetivas ou possuírem Alternativas. Cada Usuário que participa de um Teste gera um conjunto de Ações relacionadas a uma Tarefa. Esse é justamente o Fluxo ou Sessão realizada por determinado usuário. Cada Usuário convidado pode realizar apenas uma vez um Fluxo de forma completa.

O Componente para Captura de Eventos é idêntico para os dois módulos, alterando apenas a forma de inserção dele no sistema sob avaliação. O componente é escrito em *JavaScript* e que "escuta" os eventos realizados na interface Web. Os eventos são armazenados sequencialmente antes de serem enviados periodicadamente ao servidor da UseSkill. Os eventos são transformados em ações quando são identificados o tipo de ação, o horário, o endereço da página, a *tag* HTML, as posições X e Y do elemento que sofreu a ação e um conjunto de informações auxiliares que podem ser enviadas e personalizadas, de acordo com o método proposto.

Como o componente é baseado em *JavaScript*, para evitar conflitos entre nomes de funções e variáveis na página, o componente utiliza a definição de funções anônimas, onde as variáveis são definidas apenas no escopo interno à função do componente. Quanto à compatibilidade com os *browsers*, o componente é compatível com as versões mais recentes dos navegadores Chrome[9], Internet Explorer[10], Safari[11] e Firefox[12].

[9] https://www.google.com/chrome/
[10] https://www.microsoft.com/download/internet-explorer.aspx
[11] http://www.apple.com/safari/
[12] https://www.mozilla.org/firefox/new/

3.2.2 UseSkill *Control*

Baseado no método e na arquitetura propostos, foi desenvolvida a ferramenta UseSkill *Control* (USC), apoiando a realização de testes de usabilidade remotamente em sistemas Web de forma não intrusiva, sem exigir alterações manuais no sistema a ser testado. Concebida em uma plataforma de computação em nuvem, a ferramenta pode ser utilizada por diversos usuários simultaneamente.

A ferramenta é voltada para testes em contextos controlados, ou seja, onde há uma definição clara de qual roteiro deve ser seguido: quais tarefas os participantes devem realizar e quais questionários devem ser respondidos. Para capturar os dados dos usuários, a USC possui ferramentas auxiliares que inserem o Componente de Captura de Eventos em páginas Web.

Para o contexto Web em computadores de mesa e *laptops* é necessário que os participantes instalem um *plug-in* que atualmente está disponível para o *web browser* Chrome. Para o contexto Web *mobile* é necessário instalar um aplicativo para dispositivos que utilizam o sistema operacional Android.

Uma desvantagem dessas ferramentas auxiliares é que elas restringem o escopo para navegadores e sistemas operacionais específicos. O *plug-in* foi desenvolvido para o Chrome por ser um dos *web browser* mais utilizados mundialmente. Da mesma forma, o aplicativo foi desenvolvido para Android também por ser bastante utilizado em dispositivos móveis mundialmente (ZHOU; NEAMTIU; GUPTA, 2015).

O desenvolvimento de ferramentas auxiliares é uma forma não intrusiva de realizar a captura, diferentemente de outras abordagens que exigem mudanças no sistemas. Por serem não intrusivas, essas ferramentas auxiliares não interferem no funcionamento do sistema sob teste, e apresentam as tarefas e seus respectivos roteiros que devem ser executados pelos usuários, guiando a avaliação.

O processo de uso da ferramenta é composto por 4 atividades (Figura 12): criar testes, participar de teste, gerar relatórios e analisar os relatórios gerados.

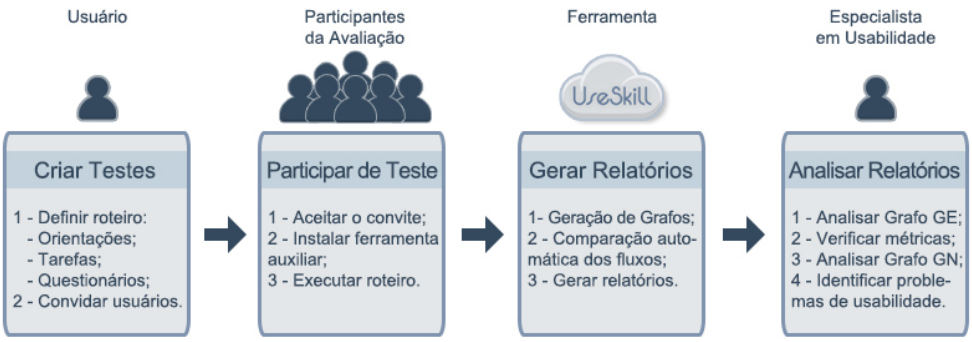


Figura 12 – Processo de uso da UseSkill *Control*.

A seguir são apresentados o funcionamento e o método de avaliação do módulo USC. Primeiramente são apresentados detalhes de como a ferramenta gera os relatórios a serem analisados por especialistas, além do embasamento do módulo no método proposto (etapa de gerar relatórios na Figura 12). Em seguida são apresentadas as demais etapas realizadas durante as avaliações: criar testes, participar deles e analisar os relatórios gerados pela ferramenta.

3.2.2.1 Método de Avaliação

O módulo UseSkill *Control* foi concebido com base no método de avaliação proposto. A seguir as etapas de avaliação são detalhadas e como elas foram implementadas no módulo USC.

3.2.2.1.1 Componente de Captura de *Logs*

Ferramentas auxiliares apoiam a captura dos dados e a condução das avaliações, apresentando o roteiro com suas tarefas e perguntas sequencialmente, de acordo com a definição do avaliador ao criar o teste.

A inserção do Componente de Captura no sistema sob avaliação é realizada por tais ferramentas auxiliares de forma transparente ao usuário. As ferramentas, por serem integradas aos navegadores, controlam a abertura de novas páginas e os conteúdos delas. Assim, ao abrir uma nova página a ser avaliada, a ferramenta auxiliar insere o *JavaScript* responsável por capturar as ações e enviar para o servidor da UseSkill.

Os dados enviados pelas ferramentas auxiliares são os descritos no método, exceto as informações sobre o usuário. No caso da USC o usuário é identificado de acordo com o usuário cadastrado na ferramenta UseSkill. A identificação de usuário "novato" ou "experiente" também é realizada implicitamente pela ferramenta, considerando o tipo de convite enviado pelo avaliador.

3.2.2.1.2 Preparação dos Dados

A etapa de preparação dos dados na USC é composta basicamente pelo cadastro de testes com suas tarefas. Ao cadastrar cada tarefa é possível identificar quais tipos de dados serão ignorados durante a análise dos *logs*. A delimitação dos dados é realizada de acordo com a tarefa que o usuário está realizando. Desta forma, os dados são delimitados e limpos.

A identificação de uma sessão de uso ocorre quando um usuário inicia uma tarefa e o seu fechamento ocorre quando o usuário clica no botão de finalizar a tarefa, independente de ter atingido ou não o objetivo.

3.2.2.1.3 Análise dos Dados

Devido às tarefas não possuírem a definição de pontos obrigatórios, as métricas de eficácia e eficiência prescritas no método tornam-se inviáveis. Para calcular a eficácia é necessário identificar quão bem o usuário realizou a tarefa. Sem a definição de marcos a serem atingidos é inviável mensurar a eficácia do uso com o método proposto.

Desta forma, além da ausência de tais métricas, as sessões são classificadas como "boas" e "ruins" de acordo com o tipo do convite realizado. Se os usuários forem convidados como "experientes", suas sessões compõem o grupo de sessões referência (GSR). As sessões dos usuários "novatos" são as do grupo de demais sessões (GDS).

Dentre os fluxos de usuários "experientes", o que possuir menor quantidade de ações é definido como caminho ótimo (CO). Em caso de fluxos com tamanhos iguais, o critério de desempate é o tempo. Em seguida a ferramenta realiza a etapa de classificação de ações, onde elas passam por uma sequência de verificações e, ao final, são classificadas como "obrigatórias" (AO), "corretas" (AC), "alertas" (AA) e "problemáticas" (AP).

Após a classificação das ações, a USC calcula as seguintes métricas: percentual de ações em relação ao caminho ótimo (PACO), percentual de "ações obrigatórias" (PAO), percentual de ações contidas no GSR (PAGR) e percentual de ações redundantes ou problemáticas (PARP).

3.2.2.1.4 Relatórios

Após a classificação das ações e o cálculo de algumas métricas, a USC gera relatórios para apoiar os avaliadores. Dentre eles, a ferramenta gera um grafo concatenando as ações realizadas pelo GSR e, em seguida, é gerado outro grafo contendo as execuções dos usuários do GDS. Os vértices dos grafos representam as ações e as arestas correspondem à sequência entre as ações.

As ações são coloridas de acordo com suas classificações: as "alertas" são amarelas, "obrigatórias" são azuis, "corretas" são verdes e "problemáticas" são vermelhas. Caso haja muitas ações no grafo, é possível filtrá-las de acordo com o tipo de ação e quantidade de vezes que ela foi realizada. Com os grafos criados, as ações repetidas são marcadas e o caminho ótimo encontrado.

Para cada sessão de utilização a ferramenta também gera um grafo, permitindo avaliar como o usuário se comportou ao realizar a tarefa. Junto ao grafo é apresentada uma tabela de detalhamento de ações, contendo todas as ações realizadas e suas respectivas classificações.

3.2.2.2 Funcionamento

Para que avaliações sejam realizadas com apoio da USC é necessário seguir um passo a passo. Esta Seção apresenta todas as etapas realizadas para a identificação de indícios de problemas de usabilidade com apoio da ferramenta.

3.2.2.2.1 Cadastrar Teste e Tarefas

A definição de testes é importante para que seja possível realizar avaliações de usabilidade controladas. Em cada teste há um texto para contextualizar os participantes, uma lista de tarefas e perguntas a serem realizadas, além de um conjunto de usuários experientes e novatos a serem convidados.

Para cada tarefa do teste é necessário atribuir um título, um roteiro (textual) detalhando o que deve ser realizado, além do endereço no sistema Web para execução da tarefa. A ferramenta também permite cadastrar perguntas antes e/ou depois de cada tarefa, definindo um *roadmap* de execução. Cada pergunta possui um título, um texto para a pergunta e, caso a resposta seja objetiva, uma lista de alternativas. A Figura 13 apresenta a interface da ferramenta durante esta etapa.

Figura 13 – Interface da UseSkill *Control* durante a definição das tarefas e perguntas de um teste.

Após definir as tarefas e questionários da avaliação é necessário convidar os usuários participantes. Durante o envio do convite, o usuário deve ser classificado como experiente ou novato. Em seguida, os usuários recebem um e-mail de notificação para que iniciem a avaliação com a UseSkill *Control*.

3.2.2.2.2 Participar de Teste

Cada usuário convidado pode aceitar ou recusar a participação em um teste. Caso o usuário aceite o convite, a ferramenta irá capturar *logs* de utilização durante a avaliação.

As ferramentas auxiliares para captura de ações possuem algumas funcionalidades em comum:

- **Listagem de Testes**: as ferramentas auxiliares são responsáveis por listar os testes disponíveis para os usuários, permitindo aceitar ou recusar tais convites de testes;
- **Auxiliar Usuários**: durante a execução de cada teste, as ferramentas auxiliares são responsáveis por abrir as tarefas e questionários de acordo com a ordem definida na sua criação;
- **Capturar *logs* de Interação**: captura de *logs* de interação dos usuários com a interface durante os testes. Para isso, as ferramentas inserem o Componente de Captura de Eventos que captura eventos na interface e os envia para a ferramenta auxiliar. Ao final de cada tarefa, as ações são encaminhadas para o servidor da UseSkill *Control*.

A Figura 14 apresenta um teste sendo executado com apoio do *plugin* instalado no navegador Chrome. A lista contendo os testes disponíveis para o usuário aparece ao clicar no ícone da UseSkill, no canto superior direito do navegador. Ao aceitar é aberta uma nova aba contendo a primeira atividade (tarefa ou questionário) a ser realizada.

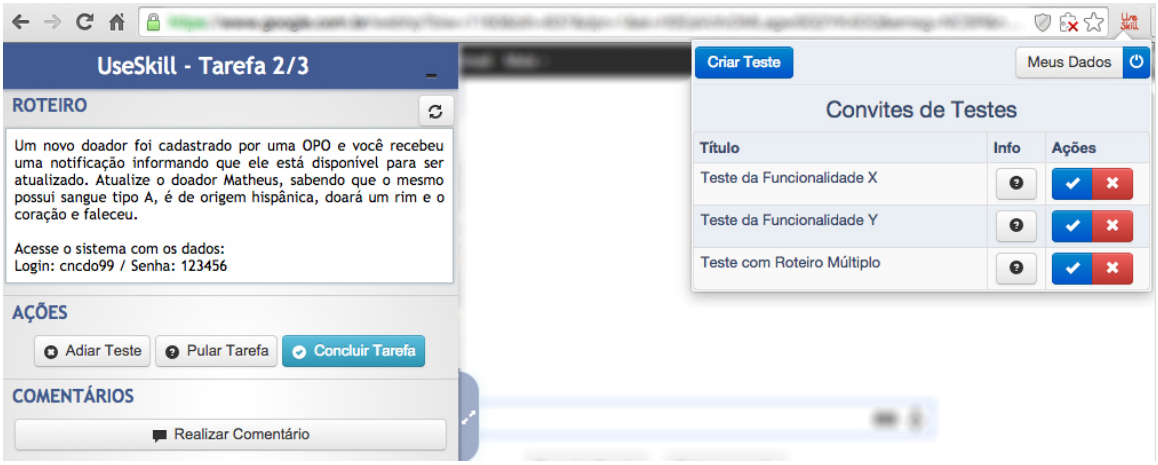


Figura 14 – Interface da ferramenta auxiliar integrada ao navegador Chrome. Do lado esquerdo é apresentado o roteiro da tarefa e as ações disponíveis. No canto superior direito há uma lista de convites de testes.

Durante a execução de tarefas é inserida uma aba lateral esquerda, que contém o roteiro, botões para “Adiar Teste”, “Pular Tarefa” e “Concluir Tarefa”. Nessa aba também é possível realizar comentários. Apesar de ocupar um bom espaço na interface, é possível minimizar essa aba durante a realização do teste, diminuindo o impacto da ferramenta durante a execução do teste. O funcionamento mais detalhado das possíveis ações são:

- **Adiar Teste**: descarta quaisquer informações capturadas e armazenadas para a sessão em execução (*logs*, respostas dos questionários, etc.) e retorna para a tela da listagem de testes. Nesse caso o teste é apenas adiado e o usuário ainda poderá realizá-lo em outra oportunidade, porém terá de recomeçá-lo;

- **Pular Tarefa**: caso o usuário não consiga concluir a tarefa. As informações capturadas da tarefa são armazenadas e enviadas para o servidor, mas o fluxo da tarefa é marcado como incompleto;

- **Concluir Tarefa**: quando o usuário julga ter realizado todas as ações descritas no roteiro e deseja avançar para a próxima tarefa ou pergunta. Essa ação envia todas as informações capturadas da tarefa para o Servidor, marcando-a como concluída;

- **Realizar Comentário**: o usuário pode realizar a qualquer momento um comentário livre. Essa opção permite capturar as impressões do usuários durante os testes, podendo realizar críticas, sugestões ou qualquer outra impressão sobre o sistema avaliado.

Após a participação dos usuários são gerados relatórios comparando suas execuções. A seguir são apresentados detalhes de como ocorre essa comparação e quais dados são gerados pela UseSkill *Control* que permitem identificar possíveis problemas de usabilidade.

3.2.2.2.3 Analisar Relatórios

A análise de relatórios é a etapa responsável por apresentar as principais contribuições dos relatórios gerados pela ferramenta. Eles são gerados a partir dos conjuntos de ações capturadas em cada uma das tarefas do teste. Dentre os suportes dados pela ferramenta, são apresentadas listas contendo as ações realizadas por usuários e suas respectivas classificações, além de grafos contendo as ações mais realizadas por tarefa. Com todos os relatórios gerados, um especialista em usabilidade deve analisá-los para identificar possíveis pontos com problemas de usabilidade.

Primeiramente, o grafo do GSR (experientes) deve ser analisado, verificando se os usuários experientes realizaram a tarefa adequadamente ou se cometeram falhas ou redundâncias. Essa análise inicial além de identificar problemas de usabilidade com base apenas em ações de usuários experientes, é determinante para que sejam gerados bons relatórios, pois se o GSR cometer falhas ou redundâncias, os usuários do GDS (novatos) podem cometer os mesmos problemas, porém serão mascarados, pois não haverá diferenças entre experientes e novatos em locais com problemas.

Em seguida, as métricas, em especial as médias e desvios padrão da quantidade de ações e tempo, serão usadas como indícios sobre a complexidade da tarefa e sobre a

discrepância entre as execuções do mesmo grupo de usuários. As métricas das sessões também permitem avaliar o quão bom foi cada execução. A PACO, PAGE e PAO indicam a eficácia da sessão em relação aos experientes, sendo a última de suma importância para verificar se usuário passou por ações classificadas como obrigatórias. Apesar das métricas citadas, elas podem ocultar problemas com redundância, que são apontadas pela métrica PARP.

A terceira etapa é analisar o grafo do GDS, identificando vértices na cor vermelha com os maiores raios, indicando quais ações "problemáticas" foram mais realizadas. A tabela contendo o detalhamento das ações permite uma análise mais aprofundada dos problemas e de suas causas. Além da tabela de detalhamento das ações realizadas pelo usuário, a ferramenta disponibiliza outra tabela contendo todas as ações obrigatórias da tarefa e destacando as realizadas pelo usuário. Essa tabela auxilia na identificação de quais ações obrigatórias foram menos realizadas, apontando quais locais em que os usuários mais encontraram dificuldades, além da completude de cada usuário por tarefa.

Para complementar a análise de qualidade de uso, a ferramenta permite ao avaliador ler comentários enviados durante os testes e relacionar os resultados às respostas dos questionários. Com esses dados disponíveis em nuvem é possível avaliar a satisfação ao utilizar sistemas Web.

3.2.2.3 Desafios e Limitações

Apesar das grandes vantagens providas pela UseSkill *Control* é necessário ressaltar suas limitações. A ferramenta realiza avaliações especificamente para sistemas Web, sendo assim sua primeira restrição. Para sanar parte dessa restrição, a UseSkill *Control* permite a avaliação Web *desktop* e Web *mobile*. Entretanto, o foco continua sendo Web, ou seja, não é possível realizar testes em aplicações nativas do sistema como nas abordagens propostas por Lettner (LETTNER; HOLZMANN, 2012) e Kluth (KLUTH; KREMPELS; SAMSEL, 2014).

A disponibilidade de ferramentas auxiliares para captura de eventos apenas para dispositivos que utilizam o Sistema Operacional *Android* ou para computadores com o navegador Chrome também se configura como uma limitação. A necessidade de possuir usuários experientes e novatos também pode dificultar seu uso, pois nem sempre será fácil encontrar usuários com níveis de experiência distintos disponíveis para avaliar o sistema.

Outra limitação é que a qualidade dos relatórios gerados depende diretamente da qualidade das execuções realizadas por usuários. Se usuários experientes errarem durante a execução das tarefas, eles podem mascarar problemas de usabilidade, pois não haverá diferença entre experientes e novatos em pontos problemáticos. Para amenizar tal problema, usuários experientes ou o responsável pela avaliação podem descartar ou ignorar fluxos do GSR que possuírem problemas.

3.2.3 UseSkill *OnTheFly*

Para auxiliar avaliações de usabilidade sem a necessidade de criar testes formais, roteiros e convites, foi concebido um módulo da UseSkill que realiza a captura e análise de *logs* durante a utilização de sistemas em produção, ou seja, *on the fly*. O módulo USOTF (UseSkill *OnTheFly*) utiliza o mesmo Componente de Captura de Eventos da UseSkill *Control*, entretanto sem a necessidade de ferramentas auxiliares.

Apesar dos auxílios e facilidades providos pela UseSkill *Control*, a USOTF visa reduzir ainda mais a complexidade e os custos da realização de avaliações de usabilidade. O módulo realiza a captura dos dados em tempo real de utilização, sem a necessidade dos participantes alocarem tempo apenas para avaliar a usabilidade do sistema.

Os dados capturados representam a utilização do sistema em contexto real, durante seu dia a dia, pois são capturados enquanto o usuário está no ambiente real de utilização, não em ambientes controlados. Essas características também facilitam a realização de avaliações de usabilidade periódicas, permitindo visualizar se houve melhora ou piora na qualidade de uso das funcionalidades do sistema.

A forma para capturar os dados de uso também possui diferenças significativas em relação à USC. Na USOTF não há roteiros, tarefas pré-determinadas, nem ferramentas auxiliares para inserir o Componente de Captura de Eventos no sistema. Devido a isso, o componente deve ser inserido no código fonte do sistema, podendo assim ser restrito a partes do sistema específicas ou a todas as funcionalidades dele.

Para analisar os *logs* de funcionalidades específicas do sistema e permitir a análise delas separadamente é necessário delimitar na USOTF onde as funcionalidades começam e terminam. A ferramenta USOTF baseia-se na criação de testes, que são compostos por um conjunto de funcionalidades e de janelas temporais de avaliações, que permitem avaliar versões específicas do sistema. O módulo também identifica quais funcionalidades são as mais utilizadas do sistema.

Devido ao grande volume de dados capturados, a análise da usabilidade é realizada com apoio de técnicas de mineração e agrupamento de dados. A ideia é baseada na método proposto e assemelha-se à análise da USC, entretanto não há definição de usuários "experientes" e "novatos", sendo necessária a classificação das sessões "boa" e "ruins" de acordo com as métricas eficácia e eficiência de uso. Os relatórios gerados também são baseados em grafos e listagens de ações realizadas, porém com peculiaridades em relação aos da USC.

A seguir são descritos o método de avaliação da ferramenta e em seguida são apresentados quais os passos necessários para utilizá-la. Por fim, são apresentados alguns desafios e limitações da ferramenta.

3.2.3.1 Método de Avaliação

O módulo UseSkill *OnTheFly*, assim como o USC, foi concebido baseando-se no método de avaliação proposto nesta abordagem. A Figura 15 apresenta as etapas necessárias para avaliar a usabilidade de funcionalidades com a USOTF.

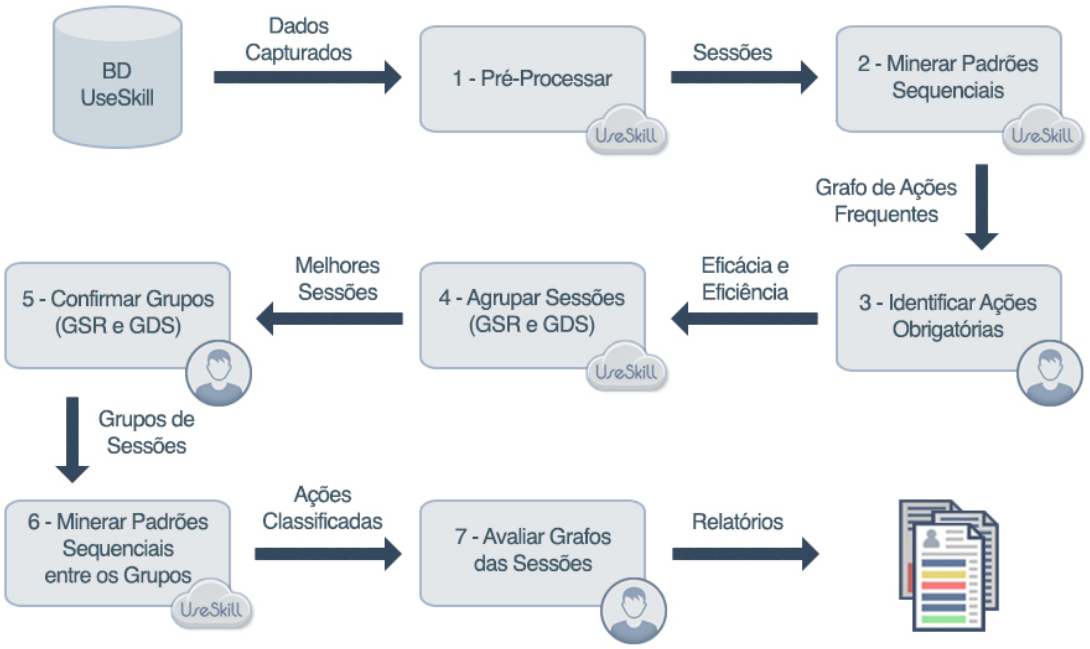


Figura 15 – Passos para avaliar usabilidade por meio da UseSkill OnTheFly. Cada caixa representa uma etapa, o ícone no canto inferior direito informa se é realizado pelo avaliador ou pela ferramenta e cada seta contém o resultado da etapa.

As etapas percorridas durante avaliações de usabilidade são detalhadas logo a seguir. Primeiramente é necessário descrever como ocorre a captura dos dados utilizados na análise. Em seguida, o pré-processamento é descrito na seção de preparação dos dados. A seção de análise dos dados discorre sobre a mineração de dados, agrupamentos e classificações realizadas. Por fim, a seção de relatórios aborda sobre os resultados gerados e que apoiam especialistas em usabilidade na identificação de problemas.

3.2.3.1.1 Componente de Captura de *Logs*

A captura de *logs* de utilização é realizada a partir da inserção do Componente de Captura de Eventos no sistema a ser avaliado. É necessário configurá-lo de acordo com as necessidades da avaliação de cada sistema. O Algoritmo 1 exemplifica um código de configuração do Componente de Captura da USOTF.

Dentre as informações definidas durante a configuração da USOTF, destacam-se a *client* e *version*. A *client* deve ser igual à abreviação definida durante a criação do teste. Essa informação é importante para que a ferramenta identifique quais *logs* correspondem

Algoritmo 1 Configuração da integração via código fonte com a USOTF

```
1: useskill-capt-onthefly({
2: onthefly: true, //para scripts inseridos via onthefly
3: timetosend: 60, //tempo em segundos para enviar os logs para a UseSkill
4: captureback: false, //captura de eventos de voltar
5: capturehashchange: false, //capturar eventos de mudança de hash
6: client: "Abrev", //abreviação do sistema
7: version: 1, //versão para controle do log
8: username: function()  return USER.username; , //informações do usuário logado
9: debug: false //apresentar dados no console
10: });
```

a cada teste. A *version* também é importante, pois caso o usuário altere quais eventos são capturados ou outra informação importante, o usuário consegue controlar quando cada configuração ficou vigente.

O Componente de Captura de Eventos utilizado é o mesmo da UseSkill *Control* e realiza a captura de eventos Web *desktop* e *mobile*, identificando o tipo de dispositivo durante a utilização de cada usuário. Com isso, a USOTF auxilia tanto avaliações em computadores de mesa, como em dispositivos móveis.

A primeira estratégia utilizada para enviar os dados capturados para o servidor da UseSkill foi que a cada carregamento de uma nova página os dados seriam enviados. Entretanto, esse tipo de envio era intrusivo e impactava na performance do sistema, pois antes de carregar a nova página era necessário enviar os dados capturados para o servidor. Nos casos onde o usuário realizava muitas ações, o envio dos dados demorava, dando a impressão de lentidão ao sistema.

Devido a esse problema foi alterada a abordagem para enviar periodicamente os dados capturados, enquanto o usuário navega nas páginas. A configuração *timetosend* define a periodicidade em segundos para que haja o envio das informações para o servidor da UseSkill. Caso o usuário demore menos tempo que o necessário para o envio dos dados, o componente de captura armazena as informações para tentar novamente quando carregar outra página do sistema. Outro momento de envio ocorre quando o usuário fecha a aba do navegador que possui o sistema sob avaliação.

3.2.3.1.2 Preparação dos Dados

Após as definições de como ocorre a captura dos dados é necessário explanar sobre como os dados são tratados e preparados para as análises. A primeira etapa é o cadastro das funcionalidades a serem avaliadas. Durante o cadastro deve ser informado um título para cada funcionalidade, o tempo máximo permitido entre a realização de duas ações e os tipos de ações a serem desconsiderados.

O cadastro de tipos de ações a serem desconsiderados serve como filtro dos dados, removendo as ações que não são consideradas úteis para a análise. Esse filtro é aplicado apenas durante a análise da funcionalidade, ou seja, ignorando os dados capturados que por ora são indesejados. Esses dados apesar de ignorados não são excluídos, podendo ser avaliados futuramente.

O tempo máximo entre as ações é o limite que o usuário tem para ficar sem realizar ações no sistema. Esse limite serve para identificar quais sessões de uso não foram concluídas e quais usuários pararam de utilizar o sistema durante a realização de uma funcionalidade.

Após tais definições, os usuários precisam informar as ações iniciais e finais da funcionalidade. Essas ações servem para identificar com exatidão quando os usuários começaram a utilizar a funcionalidade e quando os mesmos chegaram ao final da funcionalidade. Nem sempre uma funcionalidade possui apenas um ponto de início ou fim, então a ferramenta permite cadastrar diversas ações iniciais e finais.

Com as funcionalidades criadas e suas ações iniciais e finais atribuídas, a próxima etapa é criar janelas temporais para analisar as funcionalidades do sistema. A ideia das janelas temporais é possibilitar que os avaliadores analisem a usabilidade de versões específicas do sistema. Dessa forma, a criação de janelas temporais necessita apenas da definição de data inicial e final.

As análises das funcionalidades ocorrem após a seleção de uma janela temporal específica. Com isso, avaliar uma funcionalidade em uma janela temporal permite delimitar quais dados capturados correspondem à funcionalidade e ao período desejado. Os dados indesejados são ignorados e as sessões são identificadas e classificadas seguindo as seguintes regras:

- **Completa**: usuário começa com uma ação inicial e termina em uma ação final;
- **Reinício**: começa em uma ação inicial, mas o usuário não realiza nenhuma das ações finais e faz novamente uma ação inicial;
- **Limiar**: começa em uma ação inicial, mas não passa novamente por ações finais ou iniciais. Elas são encerradas após os usuários ultrapassarem o limite de tempo de realização entre uma ação e outra;
- **Erro**: começa com uma ação inicial, mas não passa novamente por ações finais ou iniciais e o usuário não realizou ações que tenham ultrapassado o limiar. Geralmente são sessões de usuários que ainda estavam realizando a funcionalidade ou que os dados finais da utilização ainda não foram enviados para a UseSkill.

Como resultado dessa etapa são geradas as sessões classificadas. Ela é representada como a Etapa 1 do método de avaliação da USOTF presente na Figura 15, denominado no fluxograma como “Pré-Processar”.

3.2.3.1.3 Análise dos Dados

A primeira parte da análise é a mineração de padrões sequenciais frequentes (*Frequent Sequential Patterns* ou FSP) no universo de todas as sessões identificadas. A ideia dessa etapa é encontrar subsequências de ações frequentes em todas as sessões identificadas. Dessa forma é possível observar quais ações foram mais realizadas sequencialmente durante as utilizações da funcionalidade. Essas ações servem como indícios de pontos onde os usuários devem passar obrigatoriamente para realizar a funcionalidade corretamente (ações obrigatórias) ou de pontos problemáticos onde muitos usuários estão enfrentando dificuldades.

Devido ao grande volume de dados, é inviável identificar os FSP por meio da geração de todas as combinações possíveis para identificar a melhor. Há um grupo de algoritmos de mineração de dados voltados especificamente para tal necessidade. Dentre os diversos algoritmos de mineração de padrões sequenciais frequentes, foi utilizado o algoritmo CM-SPADE. Ele é baseado no algoritmo Apriori e no SPADE. O CM-SPADE foi selecionado por ser mais rápido que os algoritmos originais e consumir menos memória ao minerar padrões sequenciais frequentes (FOURNIER-VIGER et al., 2014).

Para executá-lo é necessário definir três parâmetros: o suporte mínimo, o tamanho mínimo dos padrões e quais sessões serão mineradas. O suporte mínimo é um conceito proveniente de regras de associação (AGRAWAL; SRIKANT, 1994; FOURNIER-VIGER et al., 2014) e corresponde à fração de sessões que contém o padrão identificado. Caso o suporte mínimo seja 90%, serão considerados apenas os padrões que aparecem em ao menos 9 a cada 10 sessões mineradas. O tamanho mínimo representa a quantidade mínima de ações que o padrão sequencial deve possuir para ser válido. A identificação das sessões define o escopo dos dados a serem minerados.

Na abordagem utilizada, a definição do suporte mínimo, do tamanho mínimo e do grupo de sessões ocorre de forma automática, visando gerar um grafo contendo os padrões sequenciais encontrados. Durante a análise automática, a ferramenta busca por FSP com ao menos cinco ações em todas as sessões realizadas. O suporte mínimo inicial é de 100%, ou seja, aparecer em todas as sessões. Caso não encontre FSPs, a ferramenta minera novamente com 75% e em seguida com 50%. Se ainda assim não encontrar padrões, a ferramenta não gera o grafo de padrões sequenciais. A Figura 16 exemplifica um grafo gerado pela ferramenta após a mineração de FSPs, que representa a Etapa 2 do método de avaliação.

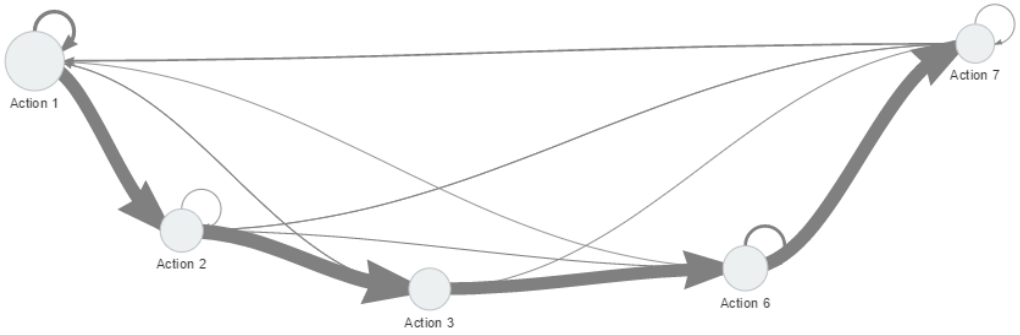


Figura 16 – Grafo contendo padrões sequenciais frequentes antes da classificação das ações pelo avaliador.

Com o grafo em mãos, o avaliador deve classificar as ações identificadas, sendo esta a Etapa 3. No exemplo da Figura 17, as duas primeiras e a última ação do padrão identificado foram classificadas pelo avaliador como ação obrigatória (AO), possuindo coloração azulada, enquanto a terceira ação foi classificada como alerta (AA), de cor amarela, e a quarta ação foi classificada como problemática (AP), com cor avermelhada.

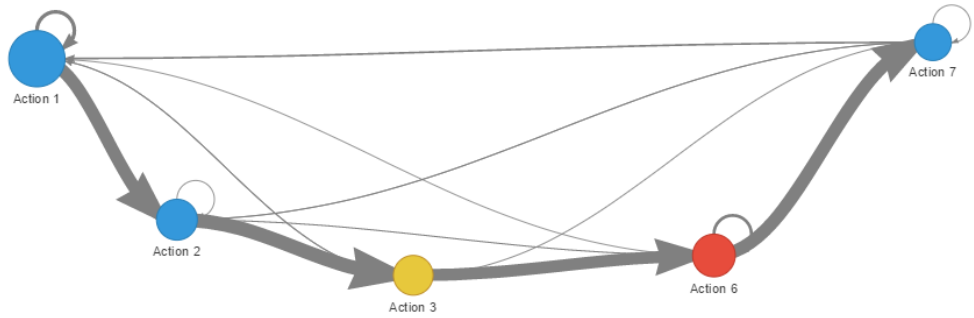


Figura 17 – Grafo contendo padrões sequenciais frequentes após a classificação das ações pelo avaliador.

Com as ações do grafo classificadas é possível calcular as métricas de eficácia e eficiência para cada sessão e para a funcionalidade. O cálculo das métricas segue as definições do método proposto. A primeira a ser calculada é a eficácia, onde em cada sessão são contabilizadas quantas ações obrigatórias foram realizadas (Fórmula 3.1). Caso haja quatro ações obrigatórias definidas na funcionalidade e em uma sessão foram realizadas apenas duas, sua eficácia é de 50%. A eficiência considera o esforço realizado, ponderando a eficácia atingida em relação à quantidade de ações e tempo despendidos, como pode ser visto na Fórmula 3.3.

A Etapa 4 se refere ao agrupamento das sessões de acordo com as métricas de eficácia e eficiência. A ideia do agrupamento é identificar um grupo de sessões consideradas como referência para as demais sessões. Esse grupo GSR deve ser formado pelas sessões com melhores índices de eficácia e eficiência. Para realizar esse agrupamento foi utilizado o algoritmo k-means (MACQUEEN et al., 1967), por ser um algoritmo simples e rápido.

A ideia é identificar diversos grupos de sessões e selecionar o grupo com centróide mais próximo do ponto máximo de eficácia e eficiência como GSR. As demais sessões são classificadas como pertencentes ao grupo GDS.

Para executar o k-means na abordagem é necessário definir dois parâmetros: número mínimo de *clusters* e o limiar da distância euclidiana entre os centróides. O número mínimo de *clusters* é a quantidade de *clusters* que o k-means identificará na sua primeira realização. O limiar da distância serve para verificar se não há dois grupos com distâncias euclidianas semelhantes em relação ao ponto ótimo de eficácia e eficiência (ponto [100; 100]). Caso o limiar não seja atingido, um novo agrupamento é realizado com a quantidade de *clusters* acrescida de mais um.

$$Dist_s = \sqrt{(100 - Efici_s)^2 + (100 - Efica_s)^2} \tag{3.5}$$

A Formula 3.5 apresenta como é calculada a distância euclidiana para cada sessão de uso. A variável $Efici_s$ representa a eficiência, e $Efica_s$ é a eficácia da sessão de uso. Ambas variam entre 0 e 100. Dentre todas as distâncias euclidianas das sessões, as duas menores são comparadas e é verificado se a diferença entre elas é menor que o limiar definido.

Como resultado, a ferramenta gera um gráfico apresentando todas as sessões agrupadas como GSR e GDS. A Figura 18 exemplifica um gráfico gerado pela ferramenta, onde os pontos laranjas corresponde ao GDS e os azuis ao GSR. O ponto selecionado (fchagas-4) corresponde a uma sessão com eficácia 42.86% e eficiência 55.57%.

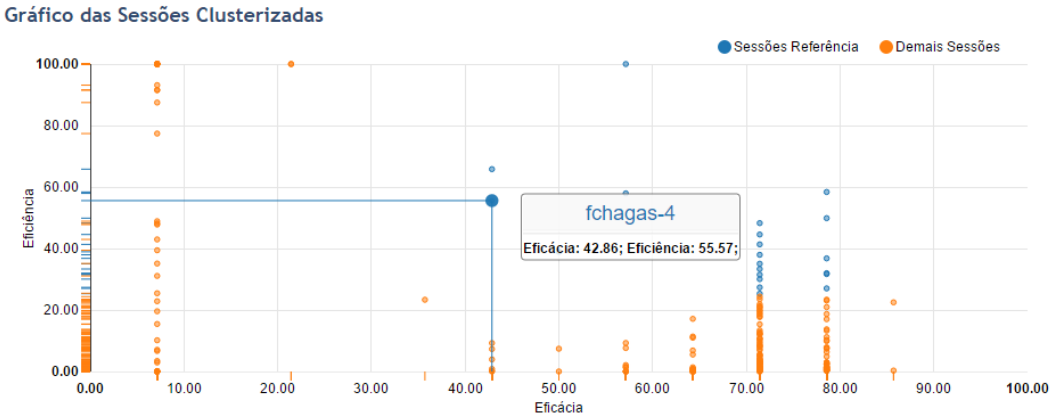


Figura 18 – Gráfico contendo as sessões agrupadas de acordo com as métricas eficácia e eficiência. A sessão "fchagas-4" pertencente ao GSR está em destaque no gráfico.

Após o agrupamento automático, a ferramenta permite ao avaliador alterar manualmente sessões caso não concorde com o agrupamento. Essa confirmação dos grupos corresponde à Etapa 5 do método de avaliação da USOTF. A Figura 19 apresenta um

exemplo de gráfico de sessões após as modificações realizadas pelo avaliador, onde a sessão fchagas-4 deixou de pertencer ao GSR.

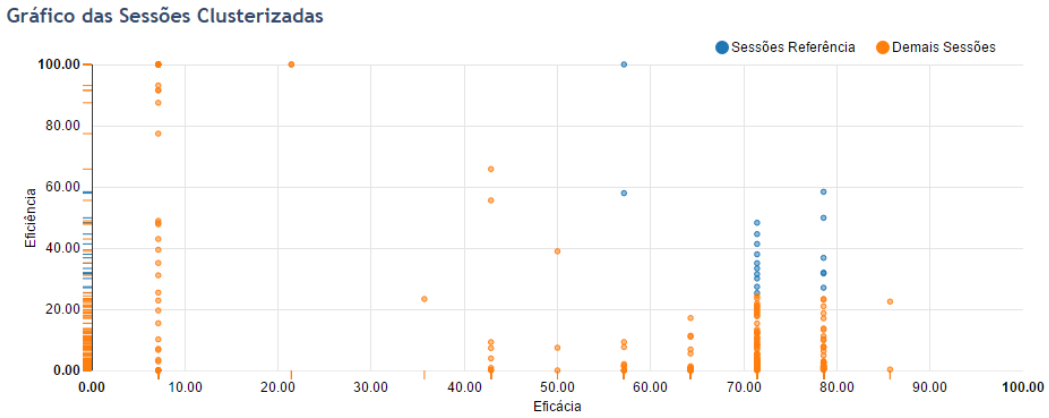


Figura 19 – Gráfico contendo as sessões agrupadas de acordo com as métricas eficácia e eficiência após interferência do avaliador.

A Etapa 6 utiliza novamente o algoritmo para mineração de FSPs. Entretanto, agora a mineração ocorre isoladamente para cada um dos grupos. A ideia é identificar as ações mais realizadas apenas em sessões de referência e nas demais sessões. O CM-SPADE é executado duas vezes, com valores pré-definidos empiricamente, mas que o avaliador pode ajustar de acordo com sua necessidade. Os valores iniciais são: suporte mínimo de 75% e quantidade mínima de 4 ações nos FSPs. O resultado dessa etapa é a classificação das ações:

- **Obrigatórias (AO)**: ações presentes nos padrões sequenciais frequentes do GSR. Dentre as melhores sessões, essas ações foram as que mais apareceram sequencialmente;
- **Corretas (AC)**: ações presentes no GSR que não estão presentes nos padrões sequenciais do GSR. São as demais ações do GSR que não foram classificadas como obrigatórias;
- **Problemáticas (AP)**: ações presentes nos padrões sequenciais frequentes do GDS que não estão presentes no GSR. São as ações presentes apenas no grupo das demais sessões e que fazem parte das ações mais realizadas sequencialmente no GDS;
- **Alertas (AA)**: ações presentes no GDS que não estão presentes no GSR e não fazem parte dos padrões sequenciais frequentes do GSR. São as ações presentes apenas no grupo das demais sessões e que não foram classificadas como problemáticas.

Essas classificações servem para apontar indícios de possíveis pontos problemáticos durante a avaliação detalhada das sessões.

3.2.3.1.4 Relatórios

Após todas as etapas durante a análise dos dados, a ferramenta apresenta uma lista contendo todas as sessões e suas respectivas informações: o usuário que realizou; suas métricas de eficácia e eficiência; tempo e quantidade de ações despendidos; e sua classificação como sessão completa, reinício, limiar ou erro. A Figura 20 apresenta a lista gerada pela ferramenta ordenada pela métrica "eficiência".

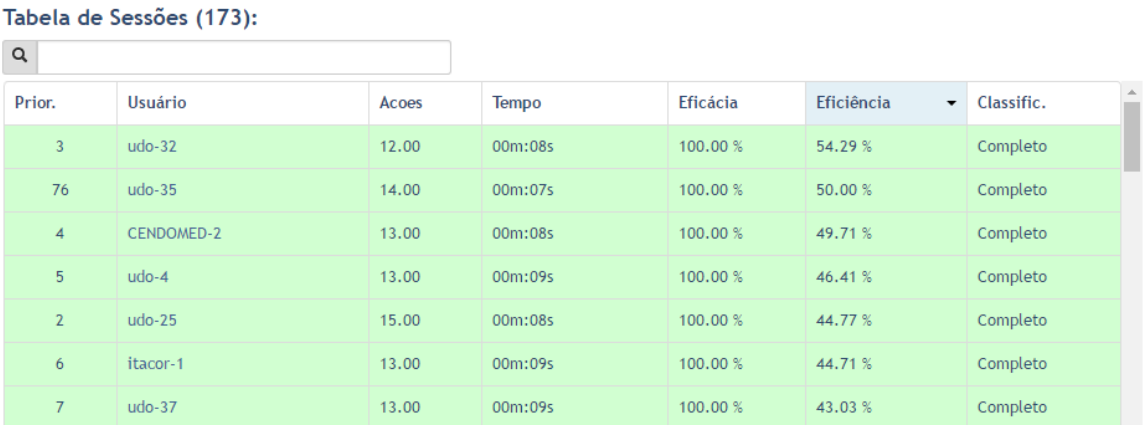


Tabela de Sessões (173):

Prior.	Usuário	Acoes	Tempo	Eficácia	Eficiência	Classific.
3	udo-32	12.00	00m:08s	100.00 %	54.29 %	Completo
76	udo-35	14.00	00m:07s	100.00 %	50.00 %	Completo
4	CENDOMED-2	13.00	00m:08s	100.00 %	49.71 %	Completo
5	udo-4	13.00	00m:09s	100.00 %	46.41 %	Completo
2	udo-25	15.00	00m:08s	100.00 %	44.77 %	Completo
6	itacor-1	13.00	00m:09s	100.00 %	44.71 %	Completo
7	udo-37	13.00	00m:09s	100.00 %	43.03 %	Completo

Figura 20 – Lista de sessões de uso ordenadas pela métrica "eficiência". Ao clicar na sessão é apresentado o detalhamento da sessão.

Ao selecionar uma sessão desejada, a ferramenta apresenta um grafo contendo as ações que foram realizadas. É possível visualizar também todas as ações em forma de lista e identificar quais ações obrigatórias não foram identificadas na sessão. O grafo permite que o usuário percorra sequencialmente as ações realizadas pelo usuário, apresentando informações detalhadas de cada ação e permitindo alterar a classificação que foi dada durante a análise automática. A Figura 21 apresenta um exemplo de grafo gerado pela USOTF para representar uma sessão. Os botões no canto inferior direito. abaixo do grafo, permitem que o usuário visualize as ações realizadas na mesma ordem que foram realizadas pelo usuário.

Ao selecionar um nó do grafo ou da lista de ações, a ferramenta apresenta o detalhamento da ação. A Figura 22 contém o exemplo de uma ação detalhada, onde é possível visualizar informações sobre o elemento, o local e o momento que a ação aconteceu, além da possibilidade de classificar a ação.

Com apoio desses relatórios ocorre a Etapa 7 do método, onde espera-se que os avaliadores consigam identificar pontos onde possíveis problemas ocorreram durante a utilização do sistema.

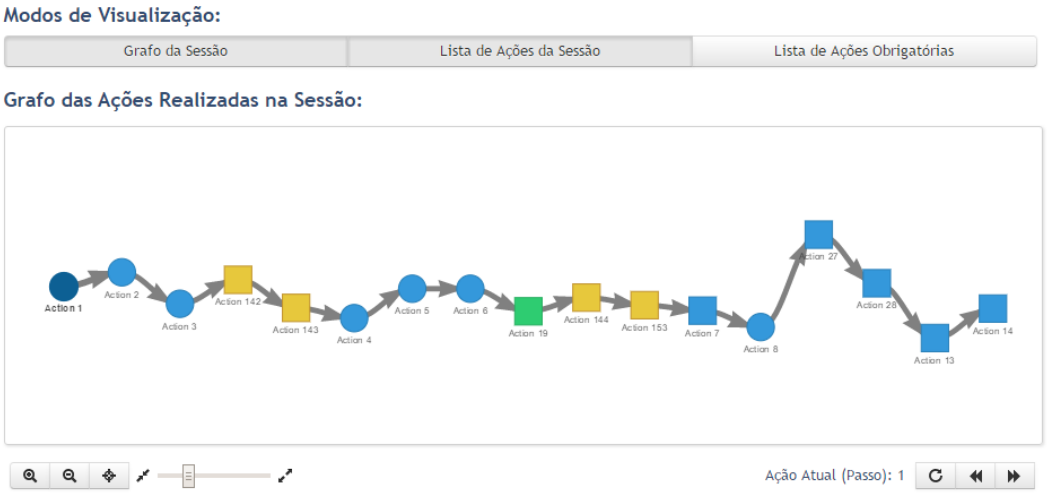


Figura 21 – Grafo de uma sessão específica na UseSkill *OnTheFly*.

Figura 22 – Detalhamento de ação da sessão durante análise aprofundada.

3.2.3.2 Funcionamento

Para avaliar a usabilidade de sistemas Web com apoio da USOTF há um passo a passo a ser seguido. Esta seção apresenta as etapas necessárias, além de outras funcionalidades disponibilizadas pela ferramenta que podem ajudar os avaliadores. Diferentemente da USC que é necessário convidar usuários a participarem do teste, a USOTF é integrada por meio do código fonte, evitando a etapa de convidar participantes e a instalação de uma ferramenta auxiliar para guiar os usuários.

3.2.3.2.1 Identificar Funcionalidades Mais Utilizadas

Para apoiar a definição de quais funcionalidades devem ser avaliadas, a USOTF apresenta as funcionalidades que foram mais utilizadas em determinado período de tempo. A identificação das funcionalidades depende basicamente da definição do período (data inicial e final) e quais dados serão utilizados para identificar as funcionalidades.

Aplicações Web baseadas no estilo arquitetural REST (*Representational State Transfer*) (FIELDING; TAYLOR, 2002) permitem identificar as funcionalidades de acordo

com a URL acessada. No caso de sistemas que não possuem padrões de URL que permitam distinguir funcionalidades é possível mapeá-las por meio de informações referentes a metadados. A Figura 23 apresenta o resultado gerado pela UseSkill durante um experimento de um sistema para gestão de plano de saúde.

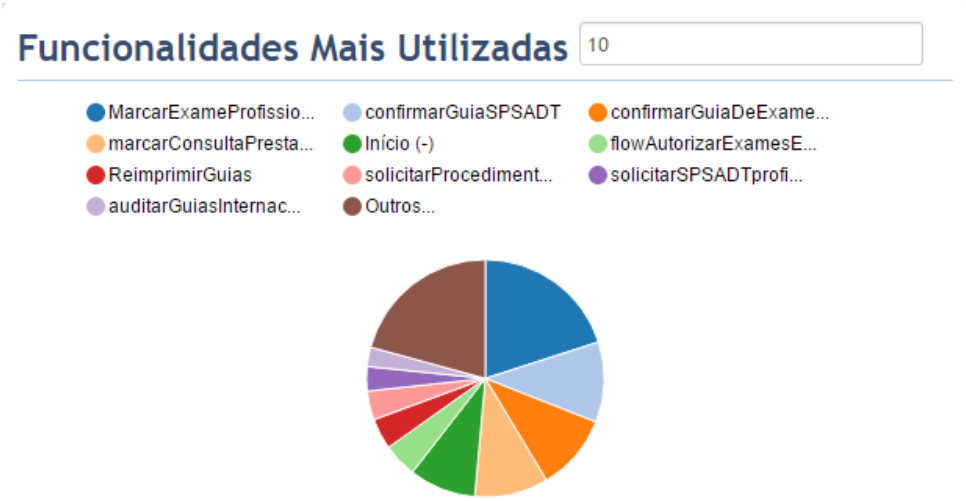


Figura 23 – Identificar funcionalidades mais utilizadas com apoio da UseSkill *OnTheFly*.

3.2.3.2.2 Cadastrar Testes e Funcionalidades

O primeiro passo para avaliar a usabilidade de sistemas por meio do módulo *OnTheFly* da UseSkill é criar um "Teste". Para isso é necessária a definição de um título, uma abreviação do nome do cliente e a URL do sistema. O título serve para descrever o teste a ser realizado e a URL identifica qual o endereço do sistema que está sendo avaliado. A abreviação do cliente serve para distinguir as ações capturadas entre os testes cadastrados na USOTF.

Após a definição do "Teste" é necessário cadastrar quais funcionalidades serão avaliadas. Cada Funcionalidade possui um título, que a descreve, e alguns dados referentes à forma de avaliar os *logs* capturados na funcionalidade, como o limite de tempo máximo entre ações e tipos de ações a serem desconsideradas durante a análise. A Figura 24 apresenta a interface da ferramenta durante o cadastro de uma Funcionalidade no módulo USOTF.

Ao cadastrar uma funcionalidade é possível definir também o "Agrupamento de Ações Obrigatórias". Ele deve ser preenchido apenas após a definição das ações obrigatórias, que ocorre durante a análise da funcionalidade.

O agrupamento serve para identificar ações obrigatórias que são similares. Por exemplo, para realizar determinada funcionalidade o usuário tem que selecionar um item, mas essa seleção pode ser feita clicando nos elementos X ou Y. Nesse caso clicar em X ou

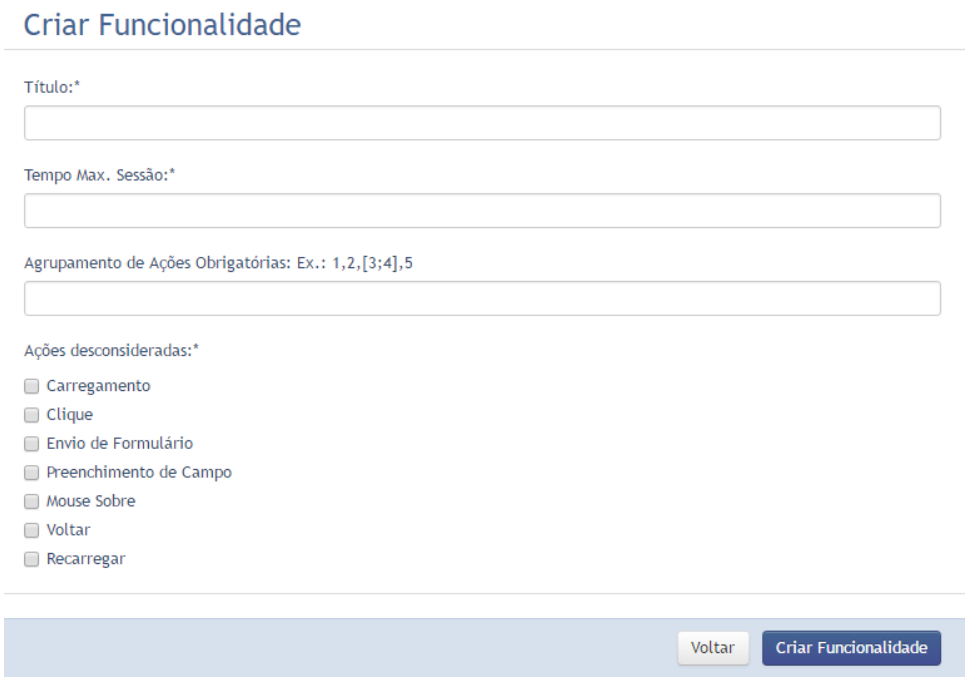


Figura 24 – Definição de uma Funcionalidade no módulo UseSkill *OnTheFly*.

Y é equivalente para a realização da tarefa, sendo assim necessário agrupá-las por meio desse campo.

Em seguida, para cada "Funcionalidade" é necessário definir suas ações iniciais e finais. Cada ação cadastrada possui as seguintes informações:

- **Momento**: pode ser início, fim ou obrigatória (durante a realização). Identifica qual tipo de ação está sendo cadastrada;
- **Tipo de Ação**: qual o tipo da ação desejada, podendo ser "carregamento de página", "clique", "preenchimento de campo", dentre outros tipos de ações;
- **Descrição**: uma breve descrição da ação para facilitar sua interpretação ao ver os detalhes da funcionalidade;
- **Elemento (XPath, ou *XML Path Language*)**: permite identificar qual elemento no HTML do sistema sofreu a ação por meio de XPath [13];
- **URL (*Uniform Resource Locator*)**: identificador de qual página o elemento faz parte. Para sistemas que não alteram a URL durante sua navegação é possível atribuir um conjunto de informações para identificar a página especificamente (metadados).

Com as funcionalidades cadastradas resta apenas cadastrar as "Janelas Temporais" para que sejam realizadas as avaliações. As avaliações realizadas ocorrem após a definição

[13] XPath: XML Path Language, é uma linguagem de consulta (*Query Language*) para selecionar nós de um documento XML bastante utilizada para identificar unicamente um elemento

de qual janela temporal será analisada. O histórico de avaliações contendo as métricas e quantidade de sessões identificadas na última avaliação é salvo na janela temporal. A Figura 25 apresenta os dados das avaliações realizadas em determinada janela temporal

Figura 25 – Histórico de avaliações de uma janela temporal entre os dias 07/03/16 e 21/03/16.

3.2.3.2.3 Analisar Relatórios

Dentre os principais apoios da ferramenta na identificação de problemas de usabilidade são: grafos contendo padrões sequenciais frequentes, listas de sessões com métricas calculadas e grafos de sessões com ações classificadas.

Os grafos contento as ações presentes nos padrões sequenciais frequentes são úteis para identificar os caminhos mais percorridos pelos usuários durante a utilização do sistema. A ferramenta permite gerar grafos do resultado de minerações considerando todas as sessões ou de sessões filtradas de acordo com suas classificações. Por exemplo, o grafo de ações dos FSPs de sessões completas apresentam boa parte das ações obrigatórias a serem realizadas ao utilizar a funcionalidade. De forma análoga, o grafo contendo sessões de reinício apresentam as ações mais frequentes e permitem identificar os principais motivos de usuários não alcançarem ações finais.

As listas de sessões realizadas, contendo suas respectivas métricas e classificações, permitem ao avaliador realizar diversas análises. Por exemplo, análises das sessões com melhores eficácias e eficiências permitem identificar quais ações são realizadas quando a funcionalidade é utilizada da melhor maneira. As sessões classificadas como “reinício” são úteis para identificar por que os usuários não finalizam a funcionalidade e a reiniciam. No caso de sessões “completas” com alta eficácia e baixa eficiência é possível averiguar

o porquê que alguns usuários apesar de terem realizado a funcionalidade corretamente, demoraram ou realizaram ações demasiadamente.

Por fim, o relatório mais importante para entender com detalhes cada uma das sessões são os grafos individuais. Eles são gerados ao final da avaliação com a USOTF e contém as ações classificadas, além de permitirem ao avaliador navegar nas ações de acordo com as ações realizadas sequencialmente pelo usuário. A identificação de problemas pode ser, por exemplo, a partir da visualização de ciclos de ações ou de um aglomerado de ações classificadas como "alerta" ou "problemática". A funcionalidade de percorrer as ações sequencialmente permite que nós que possuam mais de uma aresta de saída não causem confusão ao avaliador. Esses casos ocorrem quando o usuário realiza mais de uma vez alguma ação e em seguida realiza ações distintas.

3.2.3.3 Desafios e Limitações

Apesar das diversas contribuições da UseSkill *OnTheFly*, ela possui algumas limitações e desafios a serem superados. Dentre eles, o apoio fornecido para avaliar a usabilidade de sistemas Web leva em consideração apenas eficácia e eficiência, não conseguindo avaliar a satisfação dos usuários. Outro ponto importante inerente à captura de *logs* em ambientes reais é que, apesar dos usuários estarem no contexto de uso do dia a dia, nem sempre o usuário está focado ao realizar determinada funcionalidade, podendo impactar na quantidade de ações realizadas e no tempo para finalizá-la.

Para gerar relatórios mais completos é necessário que haja uma quantidade razoável de *logs* capturados. Caso o sistema seja novo e possua poucos usuários, avaliar a usabilidade de suas funcionalidades baseado na captura de *logs* em ambiente real pode demorar tanto quanto outros métodos de avaliação de usabilidade. A necessidade de alterar o sistema a ser testado também é uma limitação da ferramenta proposta, pois nem sempre é simples e de fácil acesso ao código fonte do sistema a ser testado, embora a alteração a ser realizada é bastante simples e limita-se à inclusão do componente de captura no sistema.

Outro desafio para o método proposto é a definição de ações iniciais, finais e obrigatórias. Essas definições impactam diretamente no cálculo da eficácia e eficiência das sessões, ou seja, influenciando diretamente nos relatórios gerados. A identificação dessas ações nem sempre é simples, podendo ser complexa e com custo benefício baixo durante a primeira avaliação realizada com a ferramenta.

Por fim, um fator complicador para utilizar a USOTF é a necessidade de avaliadores com conhecimento no sistema sob avaliação, na ferramenta UseSkill e em conceitos relacionados a usabilidade. O conhecimento no sistema sob avaliação é importante, pois os detalhes das ações apresentam informações técnicas e se o usuário não remeter ao sistema, não fica clara a identificação de problemas.

3.3 Considerações Finais

Este capítulo apresentou a abordagem proposta, que contém o método e a ferramenta UseSkill. Inicialmente o método foi apresentado, pois ele embasou a concepção da ferramenta e seus dois módulos principais. Quanto à ferramenta, este capítulo descreve sobre a arquitetura e as tecnologias utilizadas, além de detalhar o funcionamento e a forma de avaliação de usabilidade de cada um dos módulos principais da ferramentas: Control e OnTheFly. O primeiro corresponde ao módulo para avaliações em contextos controlados, permitindo definir tarefas e perguntas; enquanto a OnTheFly avalia sistemas Web em ambiente de produção, monitorando as funcionalidades enquanto os usuários reais a utilizam. O próximo capítulo apresenta as avaliações empíricas realizadas para mensurar o apoio da UseSkill durante avaliações de usabilidade de sistemas Web.

4 Avaliações

4.1 UseSKill *Control*

Para avaliar a ferramenta UseSkill *Control* foi realizado um estudo experimental. O objetivo do experimento foi avaliar a realização de testes de usabilidade laboratoriais e de avaliações de usabilidade com o apoio da USC.

Para o experimento, foi utilizado um sistema real de gestão de cooperativas médicas, com clientes em vários estados do Brasil. Ele possui cerca de 130 funcionalidades e aproximadamente 80 KLOC (*Kilo Lines of Code*). As variáveis independentes são as técnicas para avaliação de usabilidade e as variáveis dependentes são o tempo e a quantidade de problemas identificados. O tempo, registrado em minutos, envolve não apenas a execução das tarefas, mas também o tempo gasto com a avaliação por especialistas. Foram contabilizados todos os problemas de usabilidade identificados pelas técnicas, independente da criticidade do mesmo.

Por dificuldades inerentes à realização do experimento, foi realizado um *quasi-experiment*, pois os participantes foram escolhidos de forma não aleatória e baseada na conveniência, de acordo com a disponibilidade dos participantes (WOHLIN et al., 2012). Dentre os participantes, 29 são alunos de graduação do curso de Ciência da Computação e três são desenvolvedores do sistema de gestão de cooperativas médicas.

4.1.1 Hipóteses

O experimento observou as seguintes hipóteses nulas e suas hipóteses alternativas correspondentes:

$H_0$1: Não há diferença entre o tempo despendido no teste laboratorial e na USC.

H_A1: O tempo despendido na avaliação da USC é diferente que no teste laboratorial.

$H_0$2: Não há diferença entre o custo (em horas de trabalho) com especialistas no teste laboratorial e na avaliação da USC.

H_A2: O custo (em horas de trabalho) com especialistas na avaliação da USC é diferente do custo do teste laboratorial.

$H_0$3: Não há diferença entre a quantidade de problemas de usabilidade identificados no teste laboratorial e na avaliação da USC.

H_A3: A quantidade de problemas de usabilidade identificados na avaliação da USC é diferente que no teste laboratorial.

4.1.2 Desenho Experimental

O experimento foi planejado seguindo o desenho de "um fator com dois tratamentos completamente aleatórios"; os participantes atribuídos aos dois grupos foram selecionados aleatoriamente, embora a seleção dos participantes do experimento tenha sido feita por conveniência. O Grupo 1 possui cinco alunos de graduação que avaliaram o sistema Web usando o método de teste laboratorial. O Grupo 2 possui 27 pessoas que utilizaram a UseSkill *Control* para avaliarem o sistema Web (24 alunos, desempenharam o papel de usuários novatos, e três desenvolvedores que desempenharam o papel de usuários experientes).

Tabela 9 – Desenho experimental aplicado.

Grupos	**Contextualização**	**Laboratorial**	**UseSkill *Control***
Grupo 1	X	X	
Grupo 2	X		X

Foram selecionados mais participantes para o Grupo 2, pois ao realizar avaliações remotas é possível acrescentar participantes sem aumentar significativamente o custo (BASTIEN, 2010). Ambos os grupos passaram por um processo de contextualização acerca do sistema de gestão de cooperativas médicas. Adicionalmente, os membros do Grupo 2 também foram apresentados à UseSkill *Control*. Foi frisado que os participantes deveriam apenas avaliar o sistema Web e não a ferramenta USC. A Tabela 1 apresenta o desenho do experimento.

4.1.3 Execução e Análise

Cada execução foi analisada, mensurando os tempos gastos e a quantidade de problemas de usabilidade identificados. O levantamento de problemas de usabilidade durante o teste laboratorial se deu a partir de observações anotadas em planilhas por um especialista durante o teste, além de análises dos vídeos gravados e questionários pós teste. Um total de 13 problemas de usabilidade distintos foram identificados durante a execução do teste laboratorial.

Para identificar problemas de usabilidade com a UseSkill *Control*, foram utilizados os grafos, listas de ações e métricas. Quanto ao tempo gasto em cada um dos métodos, foram comparados os tempos referentes à execução da avaliação e análise dos dados. A etapa de planejamento das duas técnicas foi realizada em conjunto, sendo desconsiderada nessa comparação.

Quanto ao tempo para realizar cada tarefa, a média do teste laboratorial foi 7,5 minutos, enquanto na UseSkill *Control* a média dos experientes e novatos foi 8,9. Observou-se que os participantes do teste laboratorial foram mais focados, realizando as tarefas em um período de tempo menor que os participantes da UseSkill *Control*.

Apesar da pequena diferença entre os tempos de execução para cada tarefa, o tempo total de execução do teste laboratorial com cinco participantes foi de 2 horas e 52 minutos. No caso da UseSkill *Control* os 24 alunos utilizaram 1 hora e 19 minutos, e os três desenvolvedores levaram 49 minutos, totalizando 2 horas e 8 minutos.

A análise dos dados coletados e dos vídeos do teste laboratorial durou 4 horas e 47 minutos, enquanto a análise das tabelas da UseSkill *Control* consumiu 3 horas e 26 minutos. Ao contabilizar todo o tempo consumido com execução e análise, o teste laboratorial necessitou de 7 horas e 39 minutos, enquanto a UseSkill *Control* consumiu 5 horas e 34 minutos, uma redução de 27,3% no tempo necessário para a avaliação, vide Tabela 2.

Tabela 10 – Duração das avaliações realizadas.

Etapas	**Laboratorial**	**UseSkill *Control***
Execução	2 horas e 52 minutos	2 horas e 08 minutos
Análise	4 horas e 47 minutos	3 horas e 26 minutos
Total	7 horas e 39 minutos	5 horas e 34 minutos

4.1.4 Discussão

O teste laboratorial consumiu no total 7 horas e 39 minutos, enquanto a UseSkill *Control* exigiu 5 horas e 34 minutos para sua realização, uma redução de 27,3% no tempo necessário. Esse resultado nos leva a considerar a rejeição da hipótese $H_0 1$, porém, como temos apenas uma observação, por conta do formato da avaliação, não é possível obter uma conclusão assertiva nesse momento.

Quanto ao gasto com especialistas, o teste laboratorial exigiu a presença do especialista em um período total de 7 horas e 39 minutos, pois além da análise dos dados, nesse tipo de teste é necessária a presença de especialistas durante a realização do teste. A UseSkill *Control* exigiu 3 horas e 26 minutos do especialista para a análise dos dados, reduzindo em 55% o custo com especialistas. Da mesma forma que a hipótese anterior, esses dados nos sugerem a rejeição da hipótese $H_0 2$, mas a quantidade de observações impossibilita uma asserção quanto a esse fato.

Em relação à quantidade de problemas detectados, o teste laboratorial detectou 13 problemas distintos, enquanto a UseSkill *Control* detectou apenas 10, uma redução de 23,1%. Desta forma existe uma sugestão de rejeição da hipótese $H_0 3$.

Os resultados do estudo indicam que o método desenvolvido, juntamente com sua ferramenta de apoio, podem ser considerados uma boa opção para empresas de desenvolvimento de *software*, que por conta da dificuldade de conseguirem especialistas presenciais, por conta do alto custo envolvido com o teste e da falta de direcionamento para que ele aconteça, não conseguem realizar testes de usabilidade laboratoriais.

As evidências apontadas pelo estudo experimental ainda são bastante preliminares, por conta das ameaças à validade associadas à avaliação. A principal ameaça está associada à validade interna, que define se o relacionamento entre os tratamentos usados e os resultados é causal. Dada a quantidade de observações presentes no estudo, é difícil afirmar isso com certeza. O custo de realização da avaliação foi alto, pois demandou muito tempo dos participantes, além de necessitar da participação de especialistas no assunto. Isso dificulta a realização de um estudo mais aprofundado que envolva a realização de diferentes avaliações, em várias funcionalidades e com vários participantes interagindo.

Com relação às ameaças a validade externa, que definem a capacidade de generalização dos resultados obtidos, é possível notar que também existem limitações. A seleção dos participantes foi feita por conveniência, o que limita a capacidade de generalização dos resultados observados. Outro fator destacável é que os participantes não trabalham com gestão de cooperativas médicas e podem ter ficados confusos com alguns conceitos, dificultando o uso do sistema e por conseguinte, impactando nos resultados. O ideal seria realizar essa avaliação com pessoas do contexto do sistema. Por conta disso foi proposta a extensão da UseSkill para tratar com ambientes de produção.

Apesar do estudo apresentar indícios de que a UseSkill Control auxilia em avaliações de usabilidade de sistemas Web, e que pode ser uma boa alternativa em relação a testes laboratoriais, o estudo carece de maior rigor. Após a realização desta avaliação, o foco da pesquisa foi a concepção da UseSkill OnTheFly, tornando assim inviável a realização de avaliações e estudos experimentais mais robustos e controlados para a UseSkill Control.

4.2 UseSKill *OnTheFly*

Avaliações empíricas são de suma importância para validar trabalhos científicos. Existem três tipos principais de estratégias utilizadas para avaliar empiricamente em Engenharia de Software: *surveys*, estudos de caso e experimentos (WOHLIN et al., 2012).

Surveys possuem foco na coleta de informações dos participantes, visando descrever, comparar ou explicar seus conhecimentos, atitudes e comportamentos. Estudos de caso realizam a investigação de uma instância de um fenômeno dentro de seu contexto real. Experimento controlado manipula um fator ou variável do ambiente estudado com base na aleatorização. Em experimentos, diferentes tratamentos são aplicados, mantendo variáveis constantes, e medindo os efeitos sobre parâmetros de saída. Esta pesquisa empírica pode

ser classificada como *Quasi-experiment*, pois apesar das semelhanças com um experimento controlado, a atribuição de tratamentos aos indivíduos não foi aleatória, baseando-se em características dos participantes (WOHLIN et al., 2012).

Os objetivos deste estudo experimental são: avaliar se a ferramenta UseSkill *OnThe-Fly* contribui na identificação de problemas de usabilidade em um sistema Web; identificar quais tipos de problemas de usabilidade são encontrados com apoio da ferramenta; e comparar os resultados da UseSkill em relação a outro método de avaliação de usabilidade relevante.

O método selecionado para comparar com a UseSkill foi o *Cognitive Walkthrough* (CW) ou Passo a passo Cognitivo. Avaliações baseadas em CW são realizadas por uma equipe de especialistas em usabilidade que executam determinadas tarefas na aplicação, discutindo questões relacionadas à usabilidade a partir da utilização de um protótipo ou versão específica do sistema (POLSON et al., 1992). O CW baseia-se na ideia de que os usuários geralmente preferem aprender um sistema usando-o, em vez de, por exemplo, estudá-lo por um manual. O método CW foi selecionado por ser capaz de avaliar funcionalidades específicas em um software e por possuir baixo custo (FERNANDEZ; ABRAHÃO; INSFRAN, 2012).

Para alcançar os objetivos da pesquisa, as seguintes questões guiaram as avaliações realizadas:

- **QP1**: a USOTF identifica problemas de usabilidade, segundo especialistas na área?
- **QP2**: a quantidade de problemas de usabilidade distintos identificados com apoio da USOTF é diferente se comparado com avaliações baseadas em CW?
- **QP3**: a relevância dos problemas de usabilidade identificados com apoio da USOTF é diferente dos problemas encontrados em avaliações baseadas em CW?
- **QP4**: os tipos de problemas de usabilidade identificados com apoio da USOTF são diferentes dos tipos identificados por meio de CW?

Dado os objetivos e questões de pesquisa, o relato sobre este estudo experimental foi dividido em quatro partes, baseando-se na organização utilizada por Thelin e Wohlin (THELIN; RUNESON; WOHLIN, 2003): preparação, onde a escolha dos participantes, do sistema sob avaliação, dos papéis e informações sobre as classificações adotadas são detalhadas; planejamento, apresentando as variáveis, hipóteses, desenho experimental e ameaças à validade; operação, que relata como foram realizadas as avaliações que compõem este estudo experimental; e análises, explanando os resultados e as respostas das questões de pesquisa.

4.2.1 Preparação

Esta seção descreve a preparação necessária para conduzir o estudo experimental, os sujeitos que atuam nele, os contextos das avaliações e as definições dos tipos de problemas e relevância utilizados.

4.2.1.1 Participantes e Papéis

Este estudo experimental baseia-se em avaliações realizadas com apoio da UseSkill *OnTheFly* e em avaliações com o método CW. A definição dos participantes foi realizada de acordo com os dois tipos de avaliações distintas. O Grupo X avaliou o sistema utilizando um método baseado em CW. Ele foi composto por dois *designers* de interface que trabalham na empresa responsável pelo sistema sob avaliação. O Grupo Y realizou avaliações com apoio da UseSkill, sendo composto por apenas um avaliador que possui conhecimento sobre a ferramenta, sobre o sistema a ser avaliado e sobre usabilidade.

Devido à pequena quantidade de profissionais com conhecimento sobre usabilidade, a seleção dos participantes foi baseada na conveniência.

4.2.1.2 Contexto do Experimento

Esta seção apresenta informações sobre o ambiente das avaliações, o sistema que foi selecionado para o experimento e sobre quais funcionalidades foram avaliadas. A seleção do sistema ocorreu de acordo com a conveniência e as avaliações ocorreram em ambientes isolados, sem interferência externa. Os participantes puderam realizar questionamentos sobre regras de negócio do sistema a qualquer momento.

O sistema Web selecionado, denominado iHealth, é utilizado para gerir planos de saúde. Ele foi avaliado tanto com apoio da USOTF, como por meio do CW. Para avaliar com a UseSkill *OnTheFly* foi necessária a inserção do Componente de Captura de Eventos no código fonte do sistema iHealth. Foram capturados dados de utilização do sistema em ambiente de produção durante 4 meses, totalizando aproximadamente 3 milhões de ações capturadas.

Apesar da captura de *logs* ter armazenado dados de utilização durante quatro meses, foram consideradas apenas duas semanas, devido ao ciclo de vida iterativo e incremental do sistema iHealth. Esse corte temporal foi necessário, pois caso o período avaliado fosse muito extenso, os *logs* capturados seriam refentes a versões diferentes das mesmas funcionalidades. Nesse estudo experimental a mesma versão do sistema foi avaliada, tanto na UseSkill quanto por meio do CW.

Após a definição da janela temporal de duas semanas, foram escolhidas quais funcionalidades seriam avaliadas. A seleção objetivou escolher as partes do sistema mais utilizadas pelos usuários no dia a dia. A UseSkill auxiliou essa seleção apresentando uma

lista de funcionalidades de acordo com o número de ações realizadas durante o período selecionado.

Dentre as funcionalidades do sistema iHealth, as cinco mais utilizadas foram pré-selecionadas para compor a avaliação. Entretanto, a quarta mais utilizada era inviável mapear todas as suas ações finais, pois era referente ao portal inicial do sistema, sendo assim removida do experimento. Das quatro funcionalidades pré-selecionadas restantes, a solicitação de procedimentos odontológicos necessita que o usuário realize outras duas antes de executá-la: agendar e confirmar consulta odontológica, sendo essas duas incluídas no estudo experimental e fechando as seis funcionalidades que foram avaliadas.

Tabela 11 – Funcionalidades utilizadas nas avaliações com a USOTF. As colunas representam a posição no ranking das mais utilizadas, o total de ações realizadas e o percentual dessas ações em relação ao total do período.

Posição	**Funcionalidade**	**Qtd. Ações**	**Percentual**
01	Marcar Exame	101.012	24,53%
02	Confirmar Exame	40.739	09,89%
03	Marcar Consulta	34.192	08,30%
05	Solicitar Procedimento Odontológico	26.870	06,52%
15	Agendar Consulta Odontológica	5.118	01,24%
17	Confirmar Consulta Odontológica	3.261	00,79%

A Tabela 11 apresenta as funcionalidades selecionadas, além da posição de cada uma em um *ranking* das mais utilizadas, a quantidade de ações capturadas e o percentual do total de ações capturadas por funcionalidade. Nessa tabela é possível observar que o volume das seis funcionalidades corresponde a 51,27% do total de ações capturadas durante as duas semanas.

4.2.1.3 Classificação dos Problemas

A classificação dos problemas de usabilidade foi baseada nas 10 heurísticas de Nielsen (NIELSEN, 1994). Elas são bastante utilizadas na literatura e serviram de base para a definição de *frameworks* complexos que objetivam classificar problemas (ANDRE et al., 2001; YUSOP; GRUNDY; VASA, 2015). Devido à sua importância na área e simplicidade, as heurísticas de Nielsen foram adotadas nesse estudo experimental. Cada problema identificado foi comparado com tais heurísticas e em seguida classificados. Elas são:

- **Visibilidade do estado do sistema**: certificar que a interface sempre informe ao usuário o que está acontecendo;
- **Linguagem do usuário**: a comunicação do sistema precisa ser contextualizada ao usuário, não utilizar palavras do sistema, que não fazem sentido para o usuário;

- **Saídas claras (abortar ou desfazer)**: facilitar "saídas de emergência" para o usuário, além de permitir desfazer ou refazer a ações no sistema;
- **Consistência e padrões**: nunca identificar a mesma ação com ícones ou palavras distintas. Coisas similares deve ser tratadas da mesma maneira, facilitando a identificação pelo usuário;
- **Prevenir erros**: um *design* cuidadoso que possa prevenir erros é melhor que boas mensagens, então o sistema deve antecipar-se aos erros dos usuários;
- **Reconhecimento ao invés de recordação**: a interface deve oferecer ajuda contextual e informações capazes de orientar os usuários, evitando acionar constantemente a memória deles;
- **Flexibilidade e eficiência (atalhos)**: o sistema precisa ser fácil para usuários leigos e flexível o bastante para se tornar ágil para usuários avançados;
- **Diálogos simples e *design* minimalista**: os textos e imagens devem ser diretos e naturais, presentes apenas nos momentos em que são necessários;
- **Mensagens de erro claras e objetivas**: as mensagens devem ajudar os usuários a reconhecer, diagnosticar e sanar erros;
- **Ajuda e documentação**: explicações, ajudas e documentações são importantes para orientar os usuários em caso de dúvida. Elas devem ser visíveis e facilmente acessadas.

4.2.1.4 Relevância dos Problemas

Quanto à relevância dos problema, ela foi subdividida em 4 métricas. A primeira, sobre a severidade do problema, considerou a escala proposta por Nielsen (NIELSEN, 1994) e que foi sumarizada na Tabela 12.

Tabela 12 – Escala de severidade de problemas de usabilidade, baseada na proposta de Nielsen (NIELSEN, 1994).

Severidade	Tipo	Descrição
0	Sem importância	Não afeta a operação da interface
1	Cosmético	Não há necessidade imediata de solução
2	Simples	Problema de baixa prioridade (pode ser reparado)
3	Grave	Problema de alta prioridade (deve ser reparado)
4	Catastrófico	Muito grave, deve ser reparado de qualquer forma

Após a definição do grau de severidade, cada problema de usabilidade também foi classificado quanto a sua frequência, impacto e persistência. As notas foram valores inteiros entre 0 e 4, de forma análoga à avaliação de severidade, onde quanto maior a nota,

mais relevante é o atributo. Para evitar problemas de interpretação, a definição desses valores seguiram os seguintes conceitos (NIELSEN, 1994):

- **Frequência**: se é comum ou raro, se acontecem em muitas funcionalidades ou muitas etapas de uma funcionalidade. Considera em quantos locais esse problema ocorre;
- **Impacto**: se é fácil ou difícil de ser superado pelos usuários;
- **Persistência**: se os usuários podem superar apenas uma vez, quando eles sabem sobre o problema, ou se os usuários são incomodados repetidamente pelo problema.

4.2.2 Planejamento

4.2.2.1 Variáveis

As variáveis do experimento foram separadas em dois grupos: independentes e dependentes. As variáveis independentes são as controladas pelos pesquisadores durante o experimento, podendo impactar nas dependentes. Neste estudo experimental, os métodos utilizados para avaliar a usabilidade do sistema iHealth (USOTF e CW) representam as variáveis independentes.

As variáveis dependentes são as que medimos para verificar o efeito dos tratamentos utilizados no estudo. As variáveis: quantidade, relevância e tipo de problemas são as variáveis a serem mensuradas, sendo assim as variáveis dependentes do estudo.

4.2.2.2 Hipóteses

Observando as questões de pesquisa, as hipóteses deste estudo experimental foram definidas:

- **Hipótese nula,** $H1_0$: não foram identificados problemas reais de usabilidade com apoio da USOTF, segundo especialistas em usabilidade, ou $H1_0$: Problemas (USOTF) = 0. **Hipótese alternativa,** $H1_1$: Problemas (USOTF) $\neq$ 0;
- **Hipótese nula,** $H2_0$: não há diferença entre a quantidade de problemas distintos identificados com a USOTF e por meio do CW, ou $H2_0$: Qtd. Problemas (USOTF) = Qtd. Problemas (CW). **Hipótese alternativa,** $H2_1$: Qtd. Problemas (USOTF) $\neq$ Qtd. Problemas (CW);
- **Hipótese nula,** $H3_0$: não há diferença entre a relevância dos problemas identificados com a USOTF e por meio do CW, ou $H3_0$: Relevância (USOTF) = Relevância (CW). **Hipótese alternativa,** $H3_1$: Relevância (USOTF) $\neq$ Relevância (CW);

- **Hipótese nula,** $H4_0$: não há diferença entre os tipos de problemas identificados com a USOTF e por meio do CW, ou $H4_0$: Tipos de Problemas (USOTF) = Tipos de Problemas (CW). **Hipótese alternativa,** $H4_1$: Tipos de Problemas (USOTF) $\neq$ Tipos de Problemas (CW);

4.2.2.3 Desenho

O desenho de um experimento descreve como os testes são organizados e executados. Mais formalmente, podemos definir experimento como um conjunto de testes dos tratamentos (WOHLIN et al., 2012). Este estudo experimental foi dividido em duas avaliações de usabilidade a serem comparadas, sendo uma com apoio da UseSkill e a outra com base no método *Cognitive Walkthrough*.

Antes de iniciar cada avaliação, os participantes preencheram um questionário de caracterização, em seguida avaliaram o sistema e, por fim, preencheram outro questionário pós-experimento. O desenho experimental utilizado foi "um fator com dois tratamentos" (WOHLIN et al., 2012), entretanto a seleção dos participantes foi não-aleatória e desbalanceada, ou seja, com número de participantes distintos nas avaliações e escolhidos com base em características relacionados à experiência em usabilidade, na ferramenta USOTF e no sistema iHealth.

Essas características tornam o estudo um *Quasi-Experiment*. Apesar de ser semelhante a um experimento convencional, a atribuição dos tratamentos aos indivíduos não foi aleatória, emergindo das características dos sujeitos ou objetos (WOHLIN et al., 2012). A Tabela 13 apresenta o desenho adotado.

Tabela 13 – Etapas realizadas de acordo com os grupos de participantes do estudo experimental.

Etapa	**Grupo X**	**Grupo Y**
1	Questionário de Caracterização	
2	Avaliação com *Cognite Walkthrough*	Avaliação com a UseSkill
3	Questionário de Pós-Experimento	

Por fim, após as etapas acima, os participantes do Grupo X avaliaram os problemas identificados com a UseSkill.

4.2.2.4 Limitações e Ameaças à Validade

Uma questão fundamental sobre um experimento é quão válidos seus resultados são. É importante considerar a questão da validade desde a fase de planejamento, a fim de obter a validade adequada dos resultados (WOHLIN et al., 2012). O trabalho aqui descrito enfrentou diversas limitações que geram ameaças à validade.

A primeira limitação é que os tratamentos aplicados (UseSkill e CW) pertencem a "tipos" e "classes" de métodos de avaliação de usabilidade distintos. O primeiro pertence à classe "teste", enquanto o segundo faz parte da classe "avaliação heurística" (IVORY; HEARST, 2001). Substituir o método de "avaliação heurística" por outra ferramenta do mesmo "tipo de método"(teste de usabilidade remoto) implica em:

- Selecionar outra ferramenta: identificar artigos e ferramentas do mesmo "tipo de método" e que estivessem disponível para utilização. A seleção seria realizada por autores da UseSkill, o que poderia enviesar a escolha;
- Selecionar especialistas: após a seleção da ferramenta, seria difícil encontrar especialista nela. Pesquisadores ligados à UseSkill não poderiam se aprofundar na ferramenta, pois seria outra ameaça à validade.

Além das limitações sobre os métodos utilizados, a quantidade de avaliadores experientes em usabilidade disponíveis para o experimento é pequena, sendo outra ameaça à validade. Por conta da pequena quantidade de participantes fica inviável realizar análises estatísticas nos resultados, impactando na generalização dos resultados obtidos.

Outra limitação ocorreu quanto à escolha da estratégia utilizada neste estudo experimental. A realização de um estudo de caso foi inviabilizado por não haver *software houses* locais que façam avaliações de usabilidade durante o ciclo de vida de seus sistemas e por limitações perante à autonomia para gerar *backlog* para os sistemas avaliados com a UseSkill.

Por fim, algumas limitações da ferramenta também impactaram neste estudo, mais especificamente na seleção das funcionalidades do sistema que foram avaliadas. Uma das funcionalidades mais utilizadas pelos usuários, segundo o componente de captura, foi a "Início". Entretanto, por não possuir ações iniciais e finais bem definidas, foi inviável realizar a avaliação desta funcionalidade.

4.2.3 Operação

Esta seção apresenta como ocorreu cada etapa das avaliações: a avaliação com CW, realizada pelo Grupo X; a avaliação com apoio da UseSkill, realizada pelo Grupo Y; e por fim, a avaliação dos problemas encontrados com a UseSkill, realizada pelo Grupo X.

4.2.3.1 Avaliação com *Cognitive Walkthrough*

Os participantes do Grupo X realizaram avaliações baseadas no método *Cognitive Walkthrough*. Originalmente esse método é utilizado para determinar o nível de usabilidade por meio de um ou mais especialistas em usabilidade que "caminham" através de um

conjunto funcionalidades do site, respondendo a algumas perguntas relacionadas às suas expectativas de comportamento dos usuários (POLSON et al., 1992).

As diferenças entre o método original e o utilizado neste estudo experimental são: as avaliações foram realizadas enquanto o sistema era utilizado por um especialista no iHealth, diferentemente do CW original que os avaliadores interagem com o sistema diretamente; e os avaliadores não foram guiados por perguntas específicas, ficando livres para identificar os problemas de acordo com suas experiências e conhecimentos. Essas modificações ocorreram para reduzir o tempo do experimento e evitar efeitos de fadiga nos participantes.

Durante a avaliação, os participantes puderam pausar a execução para realizar questionamentos e fazer anotações sobre os problemas encontrados. Para cada problema identificado, cada avaliador preencheu um formulário com os seguintes campos:

- **Título**: denominação para o problema encontrado. Por exemplo, "erro ao preencher campo A";
- **Funcionalidade**: em quais funcionalidades do sistema o problema foi identificado;
- **Etapa**: em quais partes das funcionalidades o problema foi identificado;
- **Descrição**: uma breve descrição sobre o problema, detalhando o que é, além de como e onde ocorre;
- **Identificação**: como o problema foi identificado pelo avaliador.

As funcionalidades "Marcar Consulta Eletiva", "Solicitar Exame", "Confirmar Exame", "Marcar Consulta Odontológica", "Confirmar Consulta Odontológica" e "Solicitar Tratamento Odontológico" foram avaliadas enquanto um usuário experiente utilizada o sistema. Para cada problema identificado, os avaliadores também deram notas relacionadas à relevância dos problemas (severidade, frequência, impacto e persistência).

4.2.3.2 Avaliação com a UseSkill

O Grupo Y ficou responsável pela avaliação com a USOTF. Inicialmente as funcionalidades foram cadastradas na ferramenta e em seguida foram avaliadas sequencialmente, semelhante à avaliação com CW. A Seção 3.2.3 contém a Figura 15 que apresenta as etapas realizadas para avaliar a usabilidade das funcionalidades por meio da UseSkill.

A primeira etapa da avaliação é o pré-processamento, onde os *logs* presentes no banco de dados da UseSkill são pré-processados e são identificadas as sessões de uso. Cada sessão é composta por um conjunto de ações realizadas pelo usuário ao utilizar o sistema.

As sessões podem ser finalizadas em uma ação final, inicial (em caso de reinício) ou caso ultrapasse um limite de tempo configurado.

O pré-processamento é uma etapa realizada sem a intervenção do avaliador, de forma automática pela UseSkill. Uma das limpezas nos *logs* é realizada por meio de ações a serem desconsideradas. Nenhuma funcionalidade necessitava de ações do tipo "*mouse* sobre", então elas foram desconsideradas. Além disso, todas as funcionalidades consideraram o valor de 20 minutos para o limite máximo de tempo entre uma ação e outra. Caso exceda esse limite, a ferramenta considera como o fim de uma sessão. A definição empírica do limite de 20 minutos considerou que nenhum usuário demore mais que isso entre uma ação e outra, a não ser que tenha parado de utilizar o sistema.

Em seguida, a UseSkill aplicou o algoritmo CM-SPADE (FOURNIER-VIGER et al., 2014) para minerar os padrões sequenciais sobre os resultados do pré-processamento. Cada ação do grafo gerado foi avaliada isoladamente, buscando identificar as ações obrigatórias ou problemáticas nessas ações frequentes. A Figura 26 exemplifica o grafo gerado pela ferramenta durante a análise da funcionalidade "Marcar Exame", e em seguida o grafo após o avaliador classificar as ações obrigatórias (nós azuis no grafo).

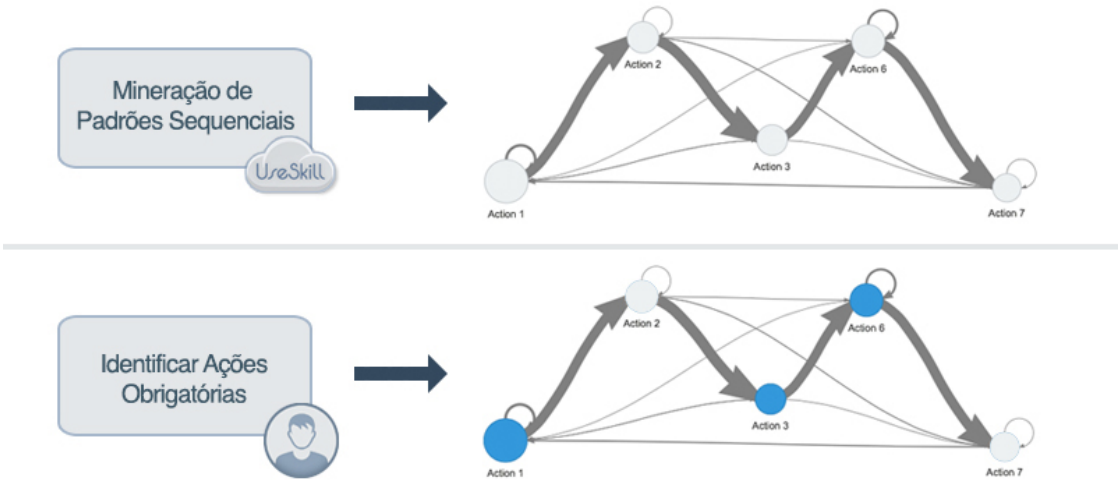


Figura 26 – Grafo de ações frequentes gerado pela ferramenta e após avaliador identificar ações obrigatórias.

Em seguida, após a ferramenta calcular as métricas eficácia e eficiência de cada sessão, baseando-se na definição das ações obrigatórias, as sessões foram agrupadas em: sessões referência (GSR) e demais sessões (GDS). O agrupamento busca identificar as sessões com melhor eficácia e eficiência para o GSR e as demais sessões compõem o GDS. A Figura 27 ilustra o resultado gerado pela ferramenta.

Com base no agrupamento automático gerado pela ferramenta, o avaliador alterou apenas em casos que haviam sessões distantes do ponto máximo de eficácia e eficiência. Com os grupos definidos, a ferramenta UseSkill os comparou para dar sugestões de classificação

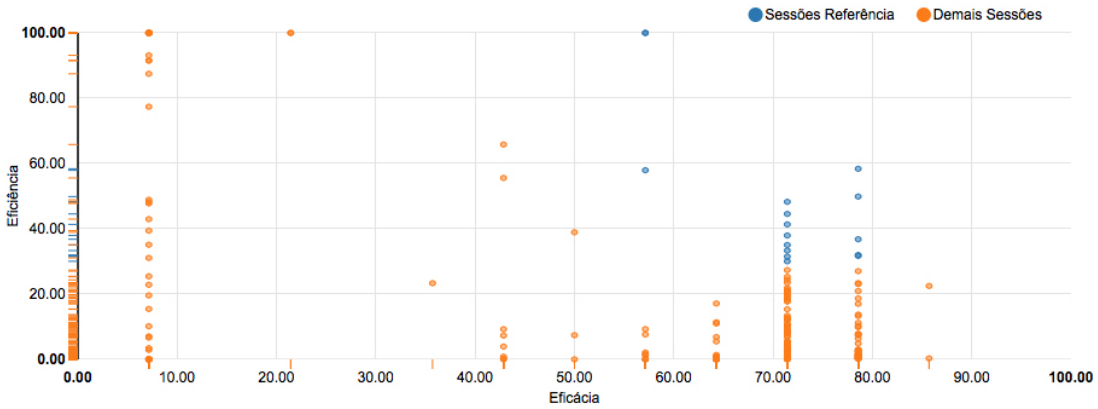


Figura 27 – Gráfico gerado pela ferramenta contendo o conjunto de melhores sessões agrupadas de acordo com a eficácia e eficiência.

das ações. Para isso, a ferramenta utilizou novamente algoritmos de mineração de padrões sequenciais para classificar todas as ações capturadas.

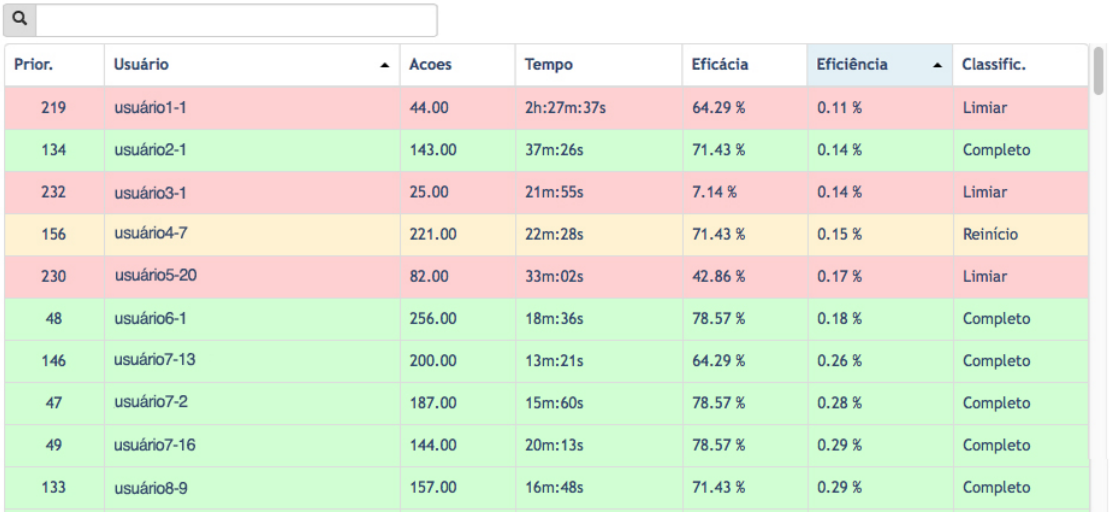
Após essas etapas, todas as ações da funcionalidade foram classificadas, permitindo ao avaliador verificar cada sessão individualmente e buscar por problemas de usabilidade. Entretanto, devido ao grande volume de dados capturados, era inviável avaliar todas as sessões. A soma das sessões identificadas nas funcionalidades que foram classificadas pela ferramenta totalizou 792 sessões.

Devido ao esforço para avaliar as sessões isoladamente, o avaliador adotou a seguinte estratégia: avaliar duas das melhores sessões para verificar quais as ações foram necessárias para utilizar a funcionalidade com boa eficácia e eficiência; em seguida, avaliar quatro sessões de usuários que concluíram a funcionalidade e que possuíam as piores eficiências, em busca do porquê da demora e da grande quantidade de ações; por fim, foram avaliadas três sessões de "reinício" com os piores índices de eficiência, focando em identificar o que causou a repetição. A Figura 28 corresponde à lista de sessões apresentada pela ferramenta, permitindo ordenar de acordo com a métrica desejada.

Em cada sessão, o avaliador focou nas ações classificadas como "alerta" ou "problemática". Outro indício foi o tamanho dos nós nos grafos, pois quanto maior eles forem, significa que as ações que eles representam foram mais realizadas naquela sessão. Além disso, ciclos nos grafos permitiram identificar conjuntos de ações repetitivas ou problemáticas nas funcionalidades. Para facilitar a interpretação das ações dos gráficos, foi utilizada a funcionalidade de percorrer as ações sequencialmente, simulando como o usuário as realizou durante a utilização do sistema.

4.2.3.3 Avaliação de Concordância com a UseSkill

Após a avaliação com base no método CW, os avaliadores do Grupo X analisaram os problemas encontrados pelo Grupo Y com apoio da UseSkill. Essa etapa visou mensurar



Prior.	Usuário	Acoes	Tempo	Eficácia	Eficiência	Classific.
219	usuário1-1	44.00	2h:27m:37s	64.29 %	0.11 %	Limiar
134	usuário2-1	143.00	37m:26s	71.43 %	0.14 %	Completo
232	usuário3-1	25.00	21m:55s	7.14 %	0.14 %	Limiar
156	usuário4-7	221.00	22m:28s	71.43 %	0.15 %	Reinício
230	usuário5-20	82.00	33m:02s	42.86 %	0.17 %	Limiar
48	usuário6-1	256.00	18m:36s	78.57 %	0.18 %	Completo
146	usuário7-13	200.00	13m:21s	64.29 %	0.26 %	Completo
47	usuário7-2	187.00	15m:60s	78.57 %	0.28 %	Completo
49	usuário7-16	144.00	20m:13s	78.57 %	0.29 %	Completo
133	usuário8-9	157.00	16m:48s	71.43 %	0.29 %	Completo

Figura 28 – Lista de sessões apresentada pela ferramenta, ordenada pela eficiência.

a concordância do Grupo X em relação aos problemas encontrados com apoio da UseSkill.

Cada problema identificado pelo Grupo Y foi detalhado da mesma forma que os problemas encontrados com CW. Todos os problemas possuíam o título, a funcionalidade, a etapa, a descrição, uma sugestão de melhoria, além das evidências que levaram à identificação do problema com a USOTF. Para cada problema, os avaliadores do Grupo X informaram se concordam ou não, o porquê e deram notas para os atributos de relevância do problema (severidade, frequência, impacto e persistência).

4.2.4 Análises

Após o detalhamento sobre a preparação, o planejamento e a operação do estudo experimental, esta seção apresenta os resultados obtidos. Inicialmente serão apresentados os resultados dos questionários de caracterização e do pós-experimento. Em seguida, as hipóteses e questões de pesquisa são respondidas.

4.2.4.1 Questionário de Caracterização

A primeira análise é referente à caracterização dos participantes. Ela é uma etapa importante por auxiliar na compreensão dos resultados, além de permitir verificar se há diferenças consideráveis entre os participantes. A caracterização foi baseada em quatro conjuntos de perguntas: informações gerais, conhecimentos sobre usabilidade, planos de saúde e sobre o sistema iHealth.

De acordo com a Tabela 14, percebe-se que os avaliadores do Grupo X e do Grupo Y possuem experiência semelhante na área de usabilidade e o mesmo conhecimento sobre o sistema iHealth. Houve uma pequena divergência quanto ao conhecimento sobre usabilidade, experiência em avaliações de usabilidade e conhecimento sobre plano de saúde.

Tabela 14 – Resultados do questionário de caracterização.

Critério	**Grupo X**			**Grupo Y**
	Aval. A	**Aval. B**	**Média**	**Aval. C**
Experiência (anos)	3	5	4	4
Conhecimento sobre Usabilidade (valores entre 0 a 4)	3	2	2,5	3
Experiência Avaliações de Usabilidade (valores entre 0 a 4)	2	1	1,5	2
Conhecimento sobre plano de Saúde (valores entre 0 a 4)	2	2	2	3
Conhecimento sobre o sistema iHealth (valores entre 0 a 4)	4	4	4	4
Experiência nas funcionalidades avaliadas (valores entre 0 a 4)	0	0	0	4

O único critério bastante destoante entre os grupos é a experiência dos avaliadores nas funcionalidades de foram avaliadas.

Para avaliar com apoio da UseSkill, um dos pré-requisitos é ter conhecimento sobre o sistema sob avaliação, justificando a necessidade do avaliador C possuir tanto conhecimento nas funcionalidades do sistema. Entretanto, essa diferença entre os grupos pode interferir nos resultados das avaliações.

4.2.4.2 Questionário Pós-Experimento

Ao término das avaliações, os participantes preencheram um questionário onde avaliaram a execução do experimento, a usabilidade do sistema iHealth, o tempo que foi disponibilizado, o roteiro utilizado e também puderam dar *feedbacks* sobre a avaliação. A escala utilizada foi do tipo *likert* entre 0 e 4, onde: 0 é muito ruim; 1 é ruim; 2 é moderado; 3 é bom; e 4 é muito bom (ALLEN; SEAMAN, 2007).

A Tabela 15 apresenta todas as notas dadas pelos avaliadores. A média das notas referentes à condução do experimento foi 2,78. Considerando a escala *likert* adotada, a nota foi entre moderada e boa. A média das notas atribuída para a usabilidade do sistema iHealth foi 1,66. Desta forma, a usabilidade foi considerada entre ruim e moderada.

Os avaliadores também ficaram livres para dar *feedbacks* em relação ao experimento, à usabilidade do iHealth e sobre a UseSkill. Quanto aos resultados obtidos com a UseSkill, os comentários foram: “Achei interessante, mas foi meu primeiro contato, preciso usá-la mais vezes. Entretanto, de fato, pareceu uma boa opção para documentar e identificar problemas de usabilidade”; “Os problemas identificados com apoio da UseSkill demonstram que a ferramenta pode ser utilizada como um passo preliminar em testes de usabilidade, revelando problemas de uso do sistema testado em produção”. Um dos avaliadores deu um *feedback* quanto às avaliações realizadas: “Senti falta de fazer o teste manipulando

Tabela 15 – Resultados do questionário preenchido após as avaliações. A escala utilizada para todos os critérios foi *likert* entre 0 e 4, onde 0 é muito ruim e 4 muito bom (ALLEN; SEAMAN, 2007).

Critério	**Grupo X**		**Grupo Y**
	Aval. A	**Aval. B**	**Aval. C**
Execução do Experimento	3	3	3
Tempo do Experimento	2	2	2
Qualidade do Roteiro	4	3	3
Média sobre a condução do experimento	3,00	2,67	2,67
Usabilidade do iHealth	2	2	2
Clareza das informações	1	2	2
Facilidade de uso	2	3	3
Evitar erros ao utilizar	0	2	0
Apoiar correção de erros	0	2	1
Satisfação de uso	1	3	1
Eficácia de uso	2	3	2
Eficiência de uso	1	2	1
Média sobre a usabilidade do iHealth	1,12	2,37	1,50

diretamente o sistema, para ter realmente a vivência de quem o utiliza no dia a dia, mas para isso precisaríamos de muito mais tempo".

4.2.4.3 Identificação de Problemas com apoio da UseSkill (QP1)

A primeira questão de pesquisa a ser respondida foi a **QP1**: *a UseSkill identifica problemas de usabilidade, segundo especialistas na área?* Para respondê-la, a seguir serão apresentados exemplos de problemas identificados com apoio da ferramenta e a análise de cada problema realizada por *designers* de interface com experiência moderada em usabilidade.

Para cada problema exemplificado são apresentadas as seguintes informações: título, descrição, sugestão de melhoria, etapa e funcionalidade em que o problema foi identificado. Além dos detalhes do problema, são apresentadas as evidências fornecidas pela UseSkill que viabilizaram a identificação de tais problemas. O primeiro a ser identificado foi:

- **Título**: grande esforço ao solicitar exames;
- **Funcionalidade / Etapa**: solicitar exame / criar guia;
- **Descrição**: é necessário realizar uma grande quantidade de ações para solicitar exames no iHealth. Caso o usuário deseje solicitar n exames, ele tem que realizar as seguintes ações n vezes: preencher o campo referente ao código do procedimento, clicar no exame desejado que foi retornado em uma lista, digitar a quantidade desejada e, por fim, clicar no botão inserir;

- **Sugestão de melhoria**: ao preencher o código e selecionar o procedimento desejado, ele já deveria ser adicionado à lista de exames com uma quantidade padrão e sem a necessidade de clicar em inserir. Caso seja necessário mudar a quantidade ou excluir um procedimento, o usuário poderia fazer isso na lista de exames adicionados.

A identificação desse problema foi a partir da análise de grafos de sessões. A Figura 29 contém dois grafos que foram analisados e apresentaram o mesmo problema. O tamanho dos cinco maiores nós azuis (ações obrigatórias) e o verde (ações corretas) presentes nos ciclos indicam que tais ações foram repetidas diversas vezes. As pequenas ações amarelas contidas no meio dos ciclos são os cliques nos diferentes exames que foram solicitados. O grafo inferior representa uma sessão que demorou 37 minutos e 143 ações para solicitar 10 exames no sistema.

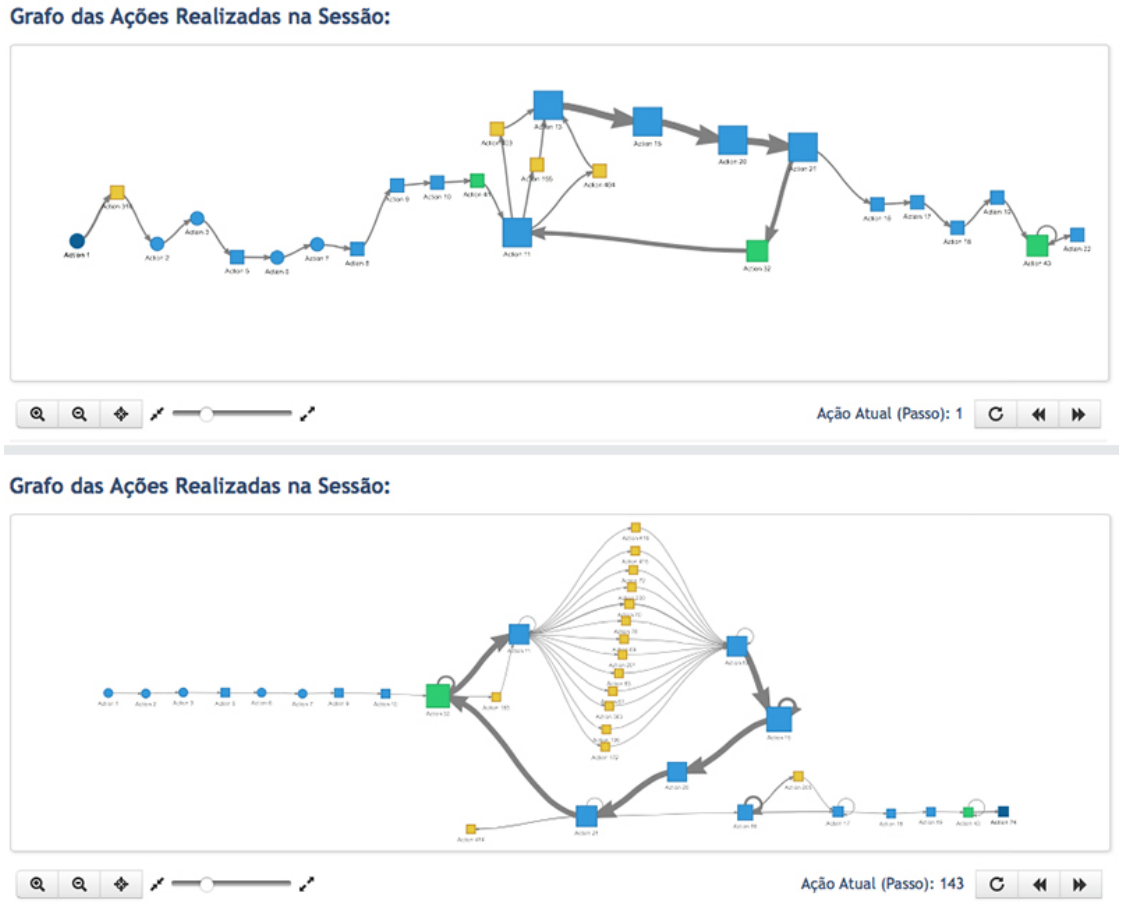


Figura 29 – Grafos de sessões que concluíram a execução da funcionalidade. O grafo superior obteve índice de eficiência de 41,97% e o inferior de 0,14%.

Outro problema identificado com a UseSkill foi:

- **Título**: antecipar ao problema de selecionar um médico ou especialidade inválida;
- **Funcionalidade / Etapa**: marcar consulta / criar guia;

- **Descrição**: caso ocorra algum problema em relação aos pré-requisitos para seleção da especialidade ou do médico, e o usuário tentar avançar, um erro será lançado e ele terá que refazer boa parte do processo de marcação de consulta;
- **Sugestão de melhoria**: ao selecionar o médico ou especialidade que não são compatíveis, o usuário deveria ser impedido de tentar prosseguir e informado por meio de uma mensagem contextualizando sobre o problema.

Esse problema foi identificado por meio da análise realizada em grafos de sessões "completas" e com pouca eficiência. A Figura 30 apresenta uma das sessões que possibilitou identificar o problema. No grafo há ciclos de ações que possuem em comum a ação de clicar no botão "Avançar" (nó "Action 13", o maior nó verde do grafo). Entretanto percebe-se que o usuário tentou avançar diversas vezes sem êxito. Ele esbarrou em problemas 3 vezes (observa-se que há três arestas saindo da ação "Avançar" e apontando para nós que retornam para ela). Apenas na quarta tentativa o usuário logrou êxito e conseguiu concluir a funcionalidade.

De acordo com as ações realizadas pelo usuário ao tentar contornar o problema, percebe-se que ocorreram erros nos dados do médico e da especialidade. Dessa forma não estava claro para o usuário tais problemas, e ele tentou avançar diversas vezes sem saber se sua solicitação estava correta ou problemática.

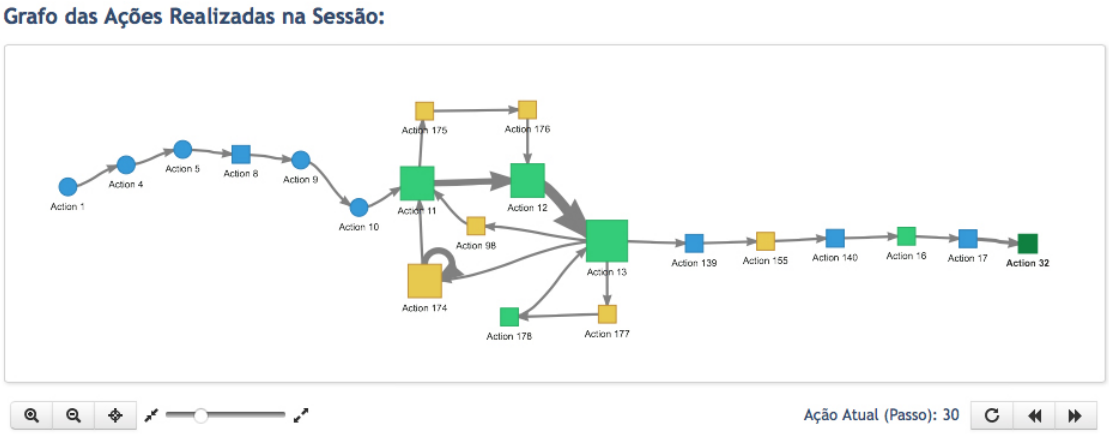


Figura 30 – Grafo de sessão que usuário tentou avançar três vezes até lograr êxito. A sessão obteve 2,37% de eficiência em uma escala de 0 a 100.

O problema a seguir foi identificado por meio do grafo inicial, gerado após a mineração dos padrões sequenciais:

- **Título**: ordem dos campos "CPF" e "número do cartão do beneficiário" está invertida;
- **Funcionalidade / Etapa**: marcar consulta / buscar beneficiário;
- **Descrição**: em 219 sessões de 254, a segunda ação a ser realizada foi clicar no campo para preencher o número do cartão do beneficiário ao invés do CPF;

- **Sugestão de melhoria**: ao invés da funcionalidade iniciar com foco no CPF, deveria focar no campo do cartão do beneficiário, pois a maioria dos usuários identificou o beneficiário por ele.

Este problema apesar de ser simples, impactou em mais de 85% das utilizações da funcionalidade. A Figura 31 apresenta resultados obtidos por meio da UseSkill: o grafo contendo as ações presentes nos padrões sequenciais frequentes e a classificação atribuída pelo avaliador; o detalhamento da ação selecionada; e a listagem das sessões que a ação selecionada foi realizada.

O grafo de padrões sequenciais frequentes possui 4 ações obrigatórias, na cor azul, e duas "corretas', em tons de verde. A primeira ação verde representa o foco automático dado pelo sistema no campo de CPF. A segunda ação verde está selecionada, por isso possui um tom mais escuro, e representa o clique no campo "número do cartão do beneficiário". Elas foram classificadas como "corretas" para destacá-las, apesar de que ao menos uma delas deve ser realizada para prosseguir na funcionalidade. Como a ação de clicar no campo do "número do cartão do beneficiário" foi realizada na maioria das vezes, se ele já viesse com foco automático ao invés do CPF, o usuário economizaria pelo menos uma ação logo ao início do uso da funcionalidade.

A seguir é descrito outro problema de usabilidade identificado que é destacável:

- **Título**: permitir solicitar o mesmo tratamento odontológico a diversos elementos dentários;
- **Funcionalidade / Etapa**: solicitar tratamento odontológico / selecionar tratamentos;
- **Descrição**: ao solicitar tratamentos odontológicos, em alguns casos é necessária a solicitação do mesmo tratamento para diversos elementos. Um caso comum é o de restaurações, que geralmente são realizadas em diversos dentes. Entretanto o sistema exige que o tratamento seja adicionado a um elemento ou (hemi)arcada por vez. Esse processo acaba sendo muito demorado e propenso a problemas, pois após todo o processo pode ser que ocorra um problema e não seja possível finalizar a solicitação;
- **Sugestão de melhoria**: deveria haver um componente onde após a seleção do tratamento seja permitido selecionar quais os elementos desejados.

Esse problema acarreta em um grande esforço para a solicitação de tratamentos odontológicos. Ele também foi identificado a partir de visualizações de sessões com baixa eficiência. Na sessão apresentada na Figura 32, o usuário realizou 330 ações e demorou 27 minutos e 21 segundos para solicitar a mesma restauração em 10 dentes.

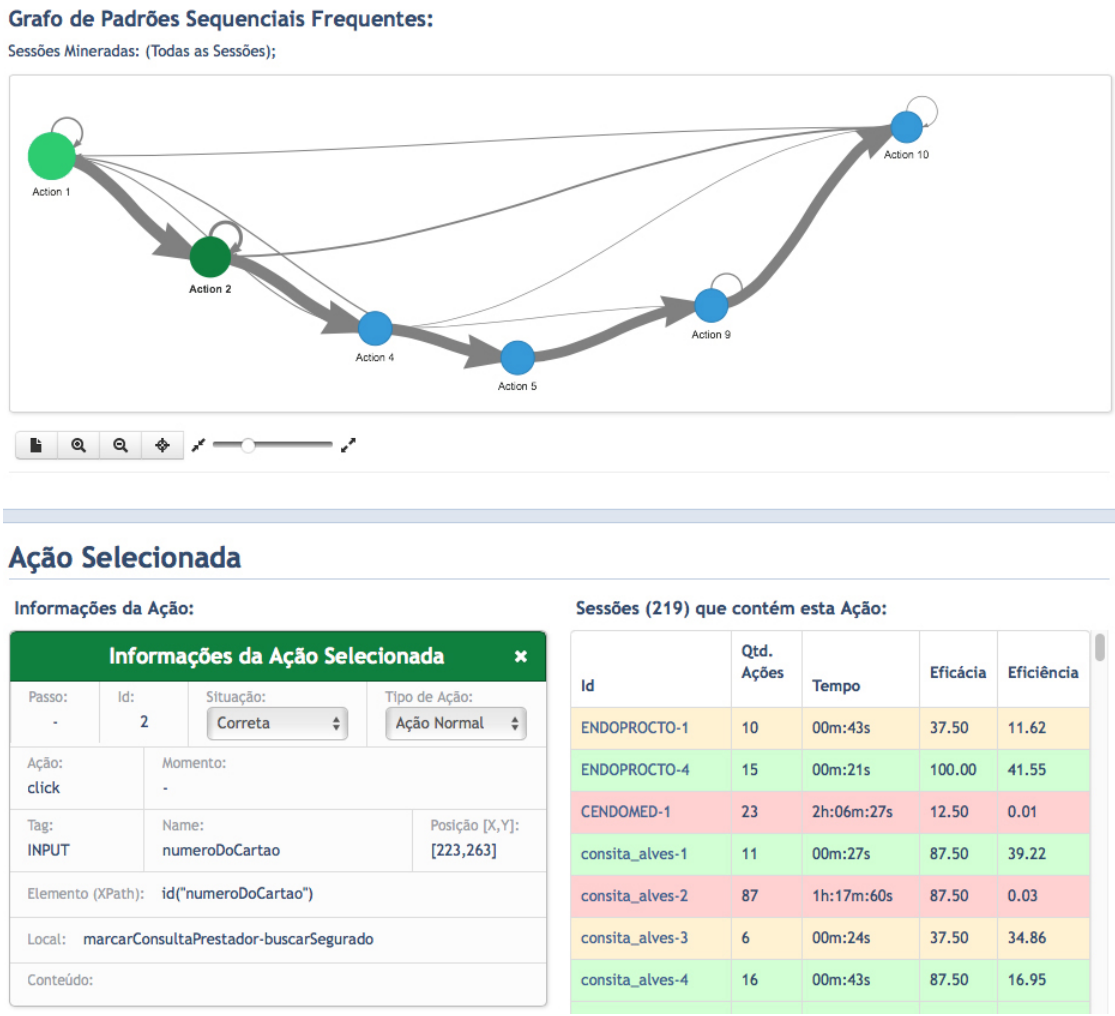


Figura 31 – Grafo com ações contidas nos padrões sequenciais frequentes da funcionalidade de Marcar Consulta, detalhes da ação selecionada e lista de sessões que contém a ação.

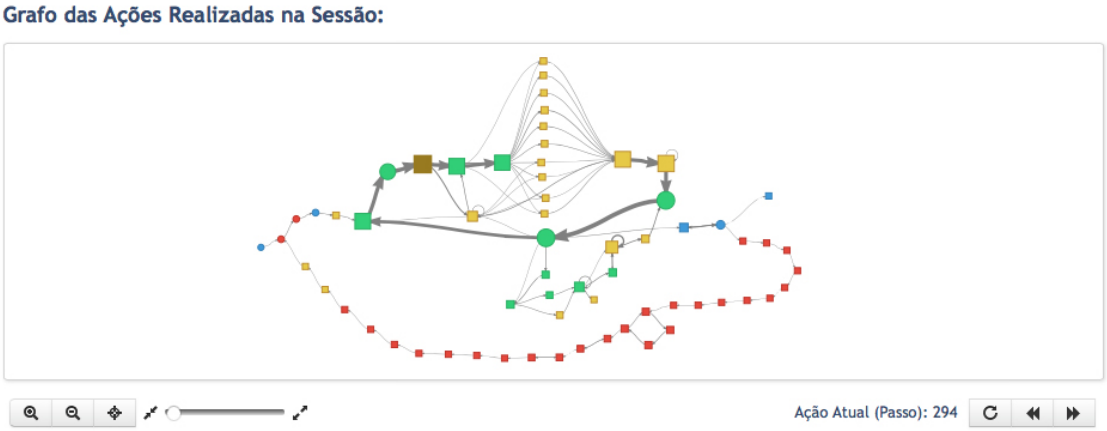


Figura 32 – Grafo de sessão com eficácia 100% e eficiência 0,08%, apresentando problemas ao solicitar o mesmo tratamento a diversos elementos dentários.

Na sessão apresentada, o usuário tentou solicitar o mesmo tratamento a 10 dentes e ao concluir ocorreu um problema. Os 10 nós amarelos em paralelo representam a seleção dos

elementos distintos, e o ciclo com os nós amarelos e verdes com os maiores diâmetros/lados representam as ações necessárias entre a seleção de um dente e outro. Após todo o esforço para selecionar todos os elementos desejados, o usuário não conseguiu concluir a solicitação.

Para resolver o problema durante a solicitação, o usuário foi para a funcionalidade "Confirmar Tratamento Odontológico" e em seguida na "Imprimir Guia" antes de refazer todo o processo de solicitação dos 10 tratamentos de novo. Os nós vermelhos são as ações realizadas fora do fluxo de solicitação.

Ao final da avaliação realizada com apoio da UseSkill, o avaliador do Grupo Y identificou 10 problemas de usabilidade. Para validar tais problemas, eles foram analisados pelo Grupo X logo após a avaliação das mesmas funcionalidades com o método *Cognitive Walkthrough*. Para cada um dos 10 problemas, os avaliadores informaram se concordam ou não, além de atribuir notas sobre a relevância dos problemas.

O avaliador B concordou com todos os problemas identificados com apoio da UseSkill e o avaliador A discordou de apenas um deles. De acordo com o avaliador A, o problema intitulado "ordem dos campos "CPF" e "número do cartão do beneficiário" está invertida" não é um problema de usabilidade, e sim uma proposta de melhoria para aumentar a flexibilidade de uso.

O índice Kappa foi utilizado para medir a concordância entre os avaliadores. Os valores podem variar entre -1 e 1, onde 1 representa uma concordância perfeita, 0 é o que seria esperado por acaso, e valores negativos indicam potencial desacordo sistemático (VIERA; GARRETT et al., 2005). O índice Kappa entre os avaliadores foi 0,94, representando uma concordância próxima à ideal. A Tabela 16 apresenta a matriz de confusão comparando os resultados dos dois avaliadores.

Tabela 16 – Matriz de confusão utilizada para cálculo do índice Cohen's Kappa (VIERA; GARRETT et al., 2005).

		Avaliador A		
		Sim	**Não**	**Total**
Avaliador B	**Sim**	9	1	10
	Não	0	0	0
	Total	9	1	10

Considerando os resultados obtidos, a resposta para a **QP1** é: *sim, segundo profissionais com experiência em usabilidade, a ferramenta apoia a identificação de problemas de usabilidade em sistemas Web*. A hipótese $H1_0$ foi rejeitada, visto que foram identificados problemas com apoio da UseSkill.

4.2.4.4 Quantidade (QP2)

A **QP2** foi a segunda questão a ser respondida: *a quantidade de problemas de usabilidade distintos identificados com apoio da UseSkill é diferente do que utilizando CW?* Para respondê-la é necessário verificar quantos problemas distintos foram identificados com apoio das duas técnicas.

O avaliador A identificou 11 problemas e o avaliador B identificou 10, ambos do Grupo X, que basearam-se no método CW. Dentre os 21 problemas encontrados, 5 eram iguais entre os dois avaliadores, restando assim 16 problemas distintos identificados por meio de CW.

O avaliador C, do Grupo Y, identificou 10 problemas com apoio da UseSkill. Destes problemas, 3 eram iguais a problemas identificados pelos avaliadores do Grupo X, totalizando assim 7 novos problemas não encontrados por meio de CW. A Tabela 17 sumariza a quantidade de problemas identificados durante o estudo experimental, que totalizou 23 problemas distintos identificados com os dois métodos.

Tabela 17 – Quantidade de problemas identificados durante o estudo experimental

Descrição	**Quantidade de Problemas**
Avaliador A	11
Avaliador B	10
Problemas iguais entre A e B	5
Total CW	16
Avaliado C (Total USOTF)	10
Problemas iguais entre CW e USOTF	3
Total de problemas distintos	23

Apesar da quantidade de problemas identificados por cada avaliador ser parecida, de acordo com a quantidade de problemas identificados com apoio de cada método, a resposta da *QP2* é: *sim, a quantidade de problemas distintos identificados com apoio da UseSkill foi menor que a quantidade de problemas identificados com CW*. Desta forma, a hipótese nula $H2_0$ também foi rejeitada.

4.2.4.5 Relevância (QP3)

A questão de pesquisa **QP3** é: *a relevância dos problemas de usabilidade identificados com apoio da UseSkill é diferente do que utilizando CW?* Para obter essa resposta, todos os 23 problemas identificados no experimento foram avaliados pelo Grupo X.

A pontuação de cada problema de usabilidade no que se refere à sua severidade, frequência, impacto e persistência foi definida em uma escala de 0 a 4. A Tabela 18 sumariza os resultados obtidos com as notas dadas pelos avaliadores.

Tabela 18 – Média das notas atribuídas para severidade, frequência, impacto e persistência. As notas foram separadas por cada avaliador e por método que apoiou a identificação. As "em comum" são as notas dos problemas que foram encontrados pelos avaliadores A e B.

Avaliador	**Prob.**	**Sever.**	**Frequênc.**	**Impacto**	**Persist.**
Avaliador A (*CW*)	11	2,27	2,64	2,55	3,00
Avaliador B (*CW*)	10	1,40	3,50	1,80	1,60
Média dos Avaliadores (*CW*)	-	1,83	3,07	2,17	2,30
Avaliador A (*CW em comum com Aval. B*)	5	2,20	3,80	2,40	3,00
Avaliador B (*CW em comum com Aval. A*)	5	1,20	3,80	1,80	1,40
Média dos Avaliadores (*CW em comum*)	-	1,70	3,80	2,10	2,20
Avaliador A (*USOTF*)	9	3,56	3,44	3,56	3,44
Avaliador B (*USOTF*)	10	1,90	3,10	2,00	1,80
Média dos Avaliadores (*USOTF*)	-	2,73	2,77	3,27	2,78

Ao comparar as notas atribuídas pelos dois avaliadores nos 5 problemas identificados por ambos, percebe-se que o avaliador A atribuiu notas maiores que o avaliador B. Entretanto, mesmo com essa diferença nas notas, ambos consideraram os problemas identificados com a UseSkill mais severos, impactantes e persistentes, mas que ocorrem com menor frequência em relação aos identificados por eles mesmo durante avaliação do sistema com CW.

Desta forma, a resposta da *QP3* é: *sim, a severidade, o impacto e a persistência dos problemas identificados com a UseSkill aparentam serem maiores, enquanto a frequência é menor que os problemas identificados com CW*. Desta forma, a hipótese nula $H3_0$ foi rejeitada.

4.2.4.6 Tipos de Problemas (QP4)

Por fim, a questão de pesquisa **QP4** é: *os tipos de problemas de usabilidade identificados com apoio da UseSkill são diferentes dos tipos identificados por meio de CW?* Para obter uma resposta é necessário classificar os problemas em tipos e em seguida compará-los.

Os problemas encontrados foram classificados segundo as 10 heurísticas de Nielsen, verificando qual heurística cada problema está mais relacionado. A Tabela 19 apresenta um

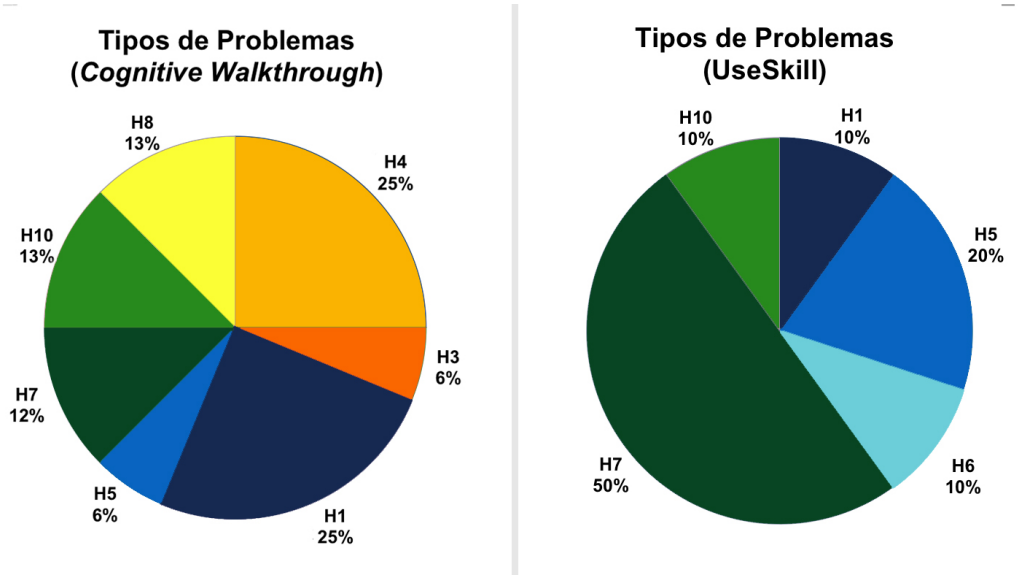


Figura 33 – Gráfico de setores com os tipos de problemas identificados durante o estudo experimental.

código para cada heurística, uma breve descrição e a quantidade de problemas encontrados pelos avaliadores e com apoio da UseSkill.

Tabela 19 – Problemas encontrados classificados segundo Nielsen.

Problema	Código	CW	UseSkill
Visibilidade do estado do sistema	H1	4 (25,00%)	1 (10,00%)
Linguagem do usuário	H2	0 (0,00%)	0 (0,00%)
Saídas Claras (abortar ou desfazer)	H3	1 (06,25%)	0 (0,00%)
Consistência e padrões	H4	4 (25,00%)	0 (0,00%)
Prevenir erros	H5	1 (06,25%)	2 (20,00%)
Reconhecimento ao invés de recordação	H6	0 (0,00%)	1 (10,00%)
Flexibilidade e eficiência (atalhos)	H7	2 (12,50%)	5 (50,00%)
Diálogos simples e design minimalista	H8	2 (12,50%)	0 (0,00%)
Mensagens de erro claras e objetivas	H9	0 (0,00%)	0 (0,00%)
Ajuda e documentação	H10	2 (12,50%)	1 (10,00%)

Dentre os 16 problemas encontrados por meio do CW, 50% são relacionados à visibilidade do estado do sistema (H1) e à ausência de consistências e padrões (H4), ou seja, problemas relacionados diretamente à interface do sistema. No caso dos 10 problemas de usabilidade encontrados com a UseSkill, 70% são ligados a prevenção de erros (H5) e flexibilidade (H7), que impactam diretamente na qualidade de uso do sistema. A Figura 33 apresenta gráficos de setores com os tipos de problemas encontrados.

Observando o gráfico da Figura 33 percebe-se que problemas do tipo H3, H4 e H8 foram identificados apenas por meio de CW e problemas do tipo H6 somente com apoio da UseSkill. Percebe-se também que além da diferença na quantidade e na relevância dos problemas, os métodos podem ser empregados para identificar problemas distintos.

Assim, a resposta da *QP4* é: *sim, foram identificados problemas distintos com os dois métodos. Cada método possui pontos fortes em tipos de problemas distintos.* Desta forma, a hipótese nula $H4_0$ foi rejeitada.

4.2.5 Análise de Correlação entre Eficácia e Eficiência

Devido à relação direta da métrica eficácia durante o cálculo da eficiência das sessões foi realizada uma breve análise de correlação entre tais métricas. Foi utilizada a medida de correlação de Pearson, que mede a força e a direção de uma relação linear entre duas variáveis em um gráfico de dispersão. O valor varia entre +1 e -1 (BENESTY et al., 2009). Para interpretar o seu valor foi utilizada a seguinte escala:

- **Exatamente -1**: relação linear perfeita de descida;
- **Entre -0.7 e maior que -1**: relação linear forte de descida;
- **Entre -0.5 e maior que -0.7**: relação linear moderada de descida;
- **Entre -0.3 e maior que -0.5**: relação linear fraca de descida;
- **Menor que 0.3 e maior que -0.3**: sem relação linear;
- **Entre 0.3 e menor que 0.5**: relação linear fraca de subida;
- **Entre 0.5 e menor que 0.7**: relação linear moderada de subida;
- **Entre 0.7 e menor que 1**: relação linear forte de subida;
- **Exatamente 1**: relação linear perfeita de subida;

Os resultados obtidos durante a análise das funcionalidades são descritos na Tabela 20. Em três funcionalidades os dados não apresentaram relação linear entre as métricas, entretanto em uma houve relação fraca e duas com relação moderada. A análise dos valores dos coeficientes de Pearson permitiram concluir que não há correlação forte entre as métricas, devido aos resultados variarem entre não haver relação linear e haver relação moderada.

4.2.6 Considerações Finais

Como resultado do estudo experimental realizado, apesar da baixa generalização dos resultados estatisticamente, todas as hipóteses nulas foram rejeitadas. Percebe-se também que a ferramenta foi responsável por apoiar a identificação de sete problemas que não foram encontrados por meio de CW. A abordagem proposta não substitui avaliações já existentes, mas complementa com problemas que impactam diretamente na utilização

Tabela 20 – Valores do coeficiente de Pearson dos resultados obtidos nas análises.

Funcionalidade	**Coeficiente de Pearson**
Marcar Consulta Eletiva	0.123
Solicitar Exame	-0.593
Confirmar Exame	0.243
Marcar Consulta Odontológica	-0.266
Confirmar Consulta Odontológica	-0.376
Solicitar Tratamento Odontológico	-0.507

do sistema. As notas atribuídas à severidade demonstram que os problemas identificados com apoio da UseSkill são tão relevantes quanto os por meio de CW.

A abordagem proposta apoia a avaliação de usabilidade de sistemas Web por meio de dados de utilização. O estudo experimental mostrou indícios de que os problemas identificados são referentes à qualidade do uso do sistema, enquanto os problemas identificados por meio de CW mostraram-se voltados para correções e melhorias em pontos referentes à interface do sistema. A UseSkill, no estudo experimental, identificou problemas com foco relacionado à eficácia e eficiência do uso, enquanto o CW identificou principalmente problemas relacionados à satisfação dos usuários.

Apesar das limitações e ameaças à validade deste estudo experimental, percebe-se que a abordagem proposta tem potencial para apoiar a avaliação de usabilidade em sistemas Web. O método proposto possibilita a avaliação de como usuários se comportam na aplicação sem a necessidade de toda a complexidade e custos de testes de usabilidade laboratoriais. Além disso, a ferramenta permite visualizar quais funcionalidades são mais utilizadas, apoiando na priorização de avaliações de usabilidade.

5 Considerações Finais

A importância e a necessidade de avaliar a usabilidade de sistemas Web de forma simples e a baixo custo motivou a realização deste e de diversos outros trabalhos. Realizou-se um estudo detalhado sobre as formas de automatizar avaliações de usabilidade, além de um mapeamento sistemático visando identificar as principais ferramentas que avaliam a usabilidade de sistemas Web apresentadas na literatura. Os resultados do mapeamento nortearam a concepção da abordagem aqui apresentada, que tem por objetivo apoiar avaliações de usabilidade de sistemas Web remotamente.

A abordagem proposta aqui é composta por um método, com duas variações de aplicação, e uma ferramenta, denominada UseSkill, constituída de dois módulos principais. O método baseia-se na divisão dos sessões de uso de um sistema, classificando-as em "adequadas" (um padrão de uso considerado bom) e "inadequadas" (um padrão de uso considerado ruim), comparando as ações realizadas em tais sessões, automaticamente, e indicando possíveis problemas de usabilidade. A ferramenta implementada para dar suporte ao método possui dois módulos distintos: o módulo UseSkill *Control* (USC), que apoia a avaliação de usabilidade de sistemas Web remotamente de forma assíncrona, em contexto controlado e que não necessita de alterações no sistema a ser testado; e o módulo denominado UseSkill *OnTheFly* (USOTF), que também apoia avaliações de usabilidade remotamente de forma assíncrona a partir da captura de *logs* de interação dos usuários na aplicação, mas em ambiente de produção, a partir da mineração de dados de uso, permitindo assim que avaliações possam ser realizadas constantemente, visando identificar possíveis pontos suspeitos de terem problemas de usabilidade.

O estudo experimental realizado no contexto controlado demonstrou que a ferramenta UseSkill *Control* pode reduzir os custos e a complexidade da realização de avaliações de usabilidade. O resultado ainda é preliminar, mas parece indicar uma tendência. Apesar da ferramenta ter detectado menos problemas de usabilidade que a abordagem laboratorial, a UseSkill detectou todos os problemas impeditivos, que influenciaram diretamente na eficácia e eficiência da utilização.

O estudo experimental feito em um ambiente de produção, que comparou avaliações realizadas com apoio da UseSkill *OnTheFly* em relação ao método *Cognitite Walkthrough*, apresentou indícios interessantes. Mesmo tendo encontrado menos problemas com o apoio da ferramenta, a severidade, o impacto e a persistência dos problemas identificados com a UseSkill *OnTheFly* eram maiores. Os tipos de problemas encontrados também aparentam ser diferentes, uma vez que o método de inspeção encontrou mais problemas relacionados à interface do sistema, enquanto que a USOTF encontrou mais problemas que impactam

diretamente na qualidade de uso do sistema.

Em resumo, os resultados obtidos indicam que ambos métodos podem auxiliar as avaliações de usabilidade, reduzindo o custo e tempo associado, ao mesmo tempo em que auxiliam a identificação de problemas, com um nível de qualidade bastante acentuado.

5.1 Desafios e Limitações

Apesar dos resultados animadores obtidos até o momento, há diversos desafios e limitações em relação à abordagem proposta. A primeira limitação da abordagem é a sua aplicação especificamente para o contexto Web. No caso da UseSkill *Control*, há a necessidade de ferramentas auxiliares, que guiam o usuário durante a realização das avaliações e capturam as interações realizadas entre o usuário e o sistema. A disponibilidade de ferramentas auxiliares para captura de eventos também é outra limitação, visto que atualmente são suportados apenas dispositivos móveis que utilizam o Sistema Operacional *Android* e computadores com o navegador Chrome.

Uma limitação da UseSkill *Control* é a necessidade de possuir usuários experientes e novatos sempre que for realizar avaliações de usabilidade. Isso tende a ser mais impactante em sistemas pequenos, pois nem sempre será fácil encontrar usuários com níveis de experiência distintos disponíveis para avaliar o sistema. Outra limitação é que a qualidade dos resultados da abordagem depende diretamente da qualidade das execuções dos usuários.

Quanto aos desafios e limitações do módulo UseSkill *OnTheFly*, a maior limitação da ferramenta é a impossibilidade de avaliar questões referentes à satisfação dos usuários, pois apenas *logs* de interação são analisados, descartando assim as expressões faciais e os *feedbacks* dos usuários.

Outra limitação da USOTF é que ao capturar *logs* em ambientes reais, apesar dos usuários estarem no ambiente real de uso, nem sempre eles estão focados, impactando diretamente na qualidade dos resultados gerados. Além disso, para avaliar a usabilidade utilizando o módulo USOTF é necessário possuir uma quantidade razoável de *log* capturado, inviabilizando a sua aplicação a curto prazo.

A necessidade de alterar o sistema a ser testado também é uma limitação da UseSkill *OnTheFly*. Um desafio para a USOTF é a definição de ações iniciais, finais e obrigatórias. Essas definições também impactam diretamente nas métricas das sessões e influenciam nos resultados das avaliações de usabilidade. Identificar essas ações nem sempre é simples, dificultando a realização da primeira avaliação com a ferramenta.

Apesar dos desafios e limitações da abordagem proposta, a maior dificuldade encontrada neste trabalho foi a realização de um estudo experimental robusto. Desde a dificuldade e complexidade em avaliar a USOTF em relação a outra ferramenta do

mesmo método de avaliação de usabilidade, até a quantidade de avaliadores experientes em usabilidade disponíveis. A pequena quantidade de participantes inviabilizou a realização de análises estatísticas nos resultados, impactando na generalização dos resultados obtidos.

5.2 Trabalhos Futuros

Com relação a trabalhos futuros, dois pontos merecem destaque. O primeiro deles é a melhoria dos módulos implementados para apoiar a abordagem desenvolvida. A usabilidade dos módulos necessita de melhorias. Essa preocupação é importante, pois a usabilidade das ferramentas de avaliação também é essencial. Elas não podem exigir grandes esforços por parte dos especialistas e dos usuários que participam de avaliações (SANTANA; BARANAUSKAS, 2015).

Um outro ponto que merece atenção refere-se aos estudos experimentais. É necessário ampliar os estudos realizados, de forma a torna-los mais relevantes, além de permitir uma melhor visualização dos resultados. No entanto, realizar tais estudos é algo caro e difícil, uma vez que os recursos para sua execução são escassos.

Com relação à parte acadêmica, ainda existem trabalhos a concluir. O mapeamento sistemático apresentado neste trabalho está em preparação para ser enviado para o *Journal Information and Software Technology*. Em paralelo também está sendo preparado um artigo para o *Journal Interacting with Computers*, apresentando a abordagem proposta e os estudos experimentais realizados.

Além disso, encontra-se em tramitação o registro de software referente ao módulo UseSkill Control. Devemos realizar a solicitação de registro também do módulo UseSkill OnTheFly, além de disponibilizá-lo na Web, para uso amplo por parte da comunidade.

Referências

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: *Proceedings of the 20th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (VLDB '94), p. 487–499. ISBN 1-55860-153-8. Disponível em: <http://dl.acm.org/citation.cfm?id=645920.672836>. Citado 2 vezes nas páginas 16 e 58.

ALBANESI, M. G. et al. Towards semi-automatic usability analysis through eye tracking. In: ACM. *Proceedings of the 12th International Conference on Computer Systems and Technologies*. [S.l.], 2011. p. 135–141. Citado na página 34.

ALLEN, I. E.; SEAMAN, C. A. Likert scales and data analyses. *Quality Progress*, American Society for Quality, v. 40, n. 7, p. 64, 2007. Citado 3 vezes nas páginas 13, 84 e 85.

ALONSO-RÍOS, D. et al. An html analyzer for the study of web usability. In: IEEE. *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. [S.l.], 2009. p. 1224–1229. Citado na página 34.

ANDRE, T. S. et al. The user action framework: a reliable foundation for usability engineering support tools. *International Journal of Human-Computer Studies*, Elsevier, v. 54, n. 1, p. 107–136, 2001. Citado na página 75.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. [S.l.], 2007. p. 1027–1035. Citado na página 17.

ATTERER, R.; SCHMIDT, A. Adding usability to web engineering models and tools. In: *Web Engineering*. [S.l.]: Springer, 2005. p. 36–41. Citado na página 34.

ATTERER, R.; WNUK, M.; SCHMIDT, A. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. In: ACM. *Proceedings of the 15th international conference on World Wide Web*. [S.l.], 2006. p. 203–212. Citado na página 34.

BAKER, S. et al. Automated usability testing using hui analyzer. In: IEEE. *Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on*. [S.l.], 2008. p. 579–588. Citado na página 2.

BALBO, S. et al. The importance of including users in clinical software evaluation: what usability can offer in home monitoring. Health Informatics Society of Australia, 2008. Citado 2 vezes nas páginas 10 e 25.

BASTIEN, J. C. Usability testing: a review of some methodological and technical aspects of the method. *International Journal of Medical Informatics*, Elsevier, v. 79, n. 4, p. e18–e23, 2010. Citado na página 70.

BEDNARIK, R. et al. Development of the tup model-evaluating educational software. In: IEEE. *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on*. [S.l.], 2004. p. 699–701. Citado na página 34.

BENESTY, J. et al. Pearson correlation coefficient. In: _____. *Noise Reduction in Speech Processing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 1–4. ISBN 978-3-642-00296-0. Disponível em: <http://dx.doi.org/10.1007/978-3-642-00296-0_5>. Citado na página 94.

BEVAN, N. Measuring usability as quality of use. *Software Quality Journal*, Springer, v. 4, n. 2, p. 115–130, 1995. Citado 4 vezes nas páginas 10, 12, 13 e 39.

BLACKMON, M. H.; KITAJIMA, M.; POLSON, P. G. Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. In: ACM. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 2005. p. 31–40. Citado na página 34.

BRUUN, A. et al. Let your users do the testing: a comparison of three remote asynchronous usability testing methods. In: ACM. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [S.l.], 2009. p. 1619–1628. Citado 2 vezes nas páginas 10 e 38.

BUCHHOLZ, G. et al. Model-based usability evaluation–evaluation of tool support. In: *Human-Computer Interaction. Interaction Design and Usability*. [S.l.]: Springer, 2007. p. 1043–1052. Citado na página 34.

BURZACCA, P.; PATERNÒ, F. Remote usability evaluation of mobile web applications. In: *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*. [S.l.]: Springer, 2013. p. 241–248. Citado 2 vezes nas páginas 30 e 34.

BYRNE, M. D. et al. The tangled web we wove: A taskonomy of www use. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 1999. (CHI '99), p. 544–551. Citado na página 11.

CALLEROS, J. M. G.; GARCÍA, J. G.; VANDERDONCKT, J. Advance human–machine interface automatic evaluation. *Universal access in the information society*, Springer, v. 12, n. 4, p. 387–401, 2013. Citado na página 34.

CASSINO, R.; TUCCI, M. Valuta: A tool to specify and verify interactive visual applications. In: *Management of the Interconnected World*. [S.l.]: Springer, 2010. p. 403–410. Citado na página 34.

CASSINO, R.; TUCCI, M. Developing usable web interfaces with the aid of automatic verification of their formal specification. *Journal of Visual Languages & Computing*, Elsevier, v. 22, n. 2, p. 140–149, 2011. Citado na página 34.

CASSINO, R.; TUCCI, M. An integrated environment to design and evaluate web interfaces. In: *Information Technology and Innovation Trends in Organizations*. [S.l.]: Springer, 2011. p. 245–253. Citado na página 34.

CASSINO, R. et al. Empirical validation of an automatic usability evaluation method. *Journal of Visual Languages & Computing*, Elsevier, v. 28, p. 1–22, 2015. Citado na página 34.

CASTILLO, J. C.; HARTSON, H. R.; HIX, D. Remote usability evaluation: can users report their own critical incidents? In: ACM. *CHI 98 Cconference Summary on Human Factors in Computing Systems*. [S.l.], 1998. p. 253–254. Citado na página 9.

CHATLEY, R. et al. Visual methods for web application design. In: IEEE. *Human Centric Computing Languages and Environments, 2003. Proceedings. 2003 IEEE Symposium on*. [S.l.], 2003. p. 242–244. Citado na página 34.

CHEN, S.-C.; YEN, D. C.; HWANG, M. I. Factors influencing the continuance intention to the usage of web 2.0: An empirical study. *Computers in Human Behavior*, v. 28, n. 3, p. 933 – 941, 2012. Citado na página 2.

CHENG-YING, M.; YAN-SHENG, L. Testing and evaluation for web usability based on extended markov chain model. *Wuhan University Journal of Natural Sciences*, Springer, v. 9, n. 5, p. 687–693, 2004. Citado na página 34.

CHI, E. H. et al. The bloodhound project: automating discovery of web usability issues using the infoscentπ simulator. In: ACM. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 2003. p. 505–512. Citado na página 34.

CHILANA, P. K. et al. Post-deployment usability: a survey of current practices. In: ACM. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. [S.l.], 2011. p. 2243–2246. Citado na página 2.

CHITRAA, V.; DAVAMANI, D.; SELVDOSS, A. A survey on preprocessing methods for web usage data. *arXiv preprint arXiv:1004.1257*, 2010. Citado 3 vezes nas páginas 14, 15 e 17.

CHYNAŁ, P.; SOBECKI, J.; SZYMAŃSKI, J. M. Application of network analysis in website usability verification. In: *Intelligent Information and Database Systems*. [S.l.]: Springer, 2014. p. 392–401. Citado na página 34.

COLETI, T. A.; MORANDINI, M.; NUNES, F. d. L. dos S. Analyzing face and speech recognition to create automatic information for usability evaluation. In: *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*. [S.l.]: Springer, 2013. p. 184–192. Citado na página 34.

CONTE, T. et al. Usability evaluation based on web design perspectives. In: *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on*. [S.l.: s.n.], 2007. p. 146–155. Citado na página 8.

DAVIS, P. A.; SHIPMAN, F. M. Learning usability assessment models for web sites. In: ACM. *Proceedings of the 16th international conference on intelligent user interfaces*. [S.l.], 2011. p. 195–204. Citado na página 34.

DHOUIB, A.; TRABELSI, A.; ABDALLAH, H. B. Eiserwebs: An evaluation tool for interactive systems based on web services. In: IEEE. *Information and Communication Technology and Accessibility (ICTA), 2013 Fourth International Conference on*. [S.l.], 2013. p. 1–4. Citado na página 34.

DINGLI, A.; CASSAR, S. An intelligent framework for website usability. *Advances in Human-Computer Interaction*, Hindawi Publishing Corp., v. 2014, p. 5, 2014. Citado na página 34.

DIX, A. *Human-computer interaction*. [S.l.]: Springer, 2009. Citado na página 8.

DRAY, S.; SIEGEL, D. Remote possibilities?: international usability testing at a distance. *interactions*, ACM, v. 11, n. 2, p. 10–17, 2004. Citado na página 10.

DYBÅ, T.; DINGSØYR, T. Empirical studies of agile software development: A systematic review. *Information and software technology*, Elsevier, v. 50, n. 9, p. 833–859, 2008. Citado na página 24.

FERNANDEZ, A.; ABRAHÃO, S.; INSFRAN, E. A systematic review on the effectiveness of web usability evaluation methods. In: IET. *Evaluation & Assessment in Software Engineering (EASE 2012), 16th International Conference on*. [S.l.], 2012. p. 52–56. Citado na página 73.

FERNANDEZ, A.; INSFRAN, E.; ABRAHÃO, S. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 53, n. 8, p. 789–817, 2011. Citado 12 vezes nas páginas 13, 3, 7, 19, 21, 22, 24, 25, 26, 28, 32 e 35.

FIELDING, R. T.; TAYLOR, R. N. Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, ACM, v. 2, n. 2, p. 115–150, 2002. Citado na página 63.

FLEISS, J. L.; COHEN, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, Sage Publications, 1973. Citado 4 vezes nas páginas 13, 24, 27 e 33.

FOURNIER-VIGER, P. et al. Fast vertical mining of sequential patterns using co-occurrence information. In: *Advances in Knowledge Discovery and Data Mining*. [S.l.]: Springer, 2014. p. 40–52. Citado 3 vezes nas páginas 16, 58 e 81.

FRANCISCO, L.; BENITTI, F. B. V. Usability evaluation in brazil: A systematic mapping. In: IEEE. *Information Systems and Technologies (CISTI), 2014 9th Iberian Conference on*. [S.l.], 2014. p. 1–7. Citado na página 2.

GENG, R.; TIAN, J. Improving web navigation usability by comparing actual and anticipated usage. *IEEE Transactions on Human-Machine Systems*, IEEE, v. 45, n. 1, p. 84–94, 2015. Citado na página 32.

GENG, R.; TIAN, J. Improving web navigation usability by comparing actual and anticipated usage. *Human-Machine Systems, IEEE Transactions on*, IEEE, v. 45, n. 1, p. 84–94, 2015. Citado na página 34.

GRIGERA, J.; GARRIDO, A.; RIVERO, J. M. A tool for detecting bad usability smells in an automatic way. In: *Web Engineering*. [S.l.]: Springer, 2014. p. 490–493. Citado na página 34.

HARMS, P.; GRABOWSKI, J. Usage-based automatic detection of usability smells. In: *Human-Centered Software Engineering*. [S.l.]: Springer, 2014. p. 217–234. Citado na página 34.

HILBERT, D. M.; REDMILES, D. F. Extracting usability information from user interface events. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 32, n. 4, p. 384–421, dez. 2000. ISSN 0360-0300. Citado 2 vezes nas páginas 7 e 11.

HOCHHEISER, H.; SHNEIDERMAN, B. Using interactive visualizations of www log data to characterize access patterns and inform site design. *Journal of the American Society for Information Science and Technology*, Wiley Online Library, v. 52, n. 4, p. 331–343, 2001. Citado na página 12.

HONG, J. I.; LANDAY, J. A. Webquilt: a framework for capturing and visualizing the web experience. In: ACM. *Proceedings of the 10th international conference on World Wide Web*. [S.l.], 2001. p. 717–724. Citado na página 34.

HUMAYOUN, S. R. et al. A model-based approach to ongoing product evaluation. In: ACM. *Proceedings of the International Working Conference on Advanced Visual Interfaces*. [S.l.], 2012. p. 596–603. Citado na página 34.

ISO. Ergonomic requirements for office work with visual display terminals (VDTs) - part 11: Guidance on usability, ISO 9241-11. *International Organization for Standardization*, 1998. Citado 3 vezes nas páginas 1, 7 e 39.

ISO. Software engineering - product quality - part 1: Quality model, ISO/IEC 9126-1. *International Organization for Standardization*, 2000. Citado na página 7.

IVORY, M. Y.; HEARST, M. A. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, ACM, v. 33, n. 4, p. 470–516, 2001. Citado 11 vezes nas páginas 1, 2, 7, 9, 10, 11, 17, 19, 25, 42 e 79.

JORGENSEN, M.; SHEPPERD, M. A systematic review of software development cost estimation studies. *Software Engineering, IEEE Transactions on*, IEEE, v. 33, n. 1, p. 33–53, 2007. Citado na página 24.

KATSANOS, C.; TSELIOS, N.; AVOURIS, N. Infoscent evaluator: a semi-automated tool to evaluate semantic appropriateness of hyperlinks in a web site. In: ACM. *Proceedings of the 18th Australia conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*. [S.l.], 2006. p. 373–376. Citado na página 34.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. 2007. Citado na página 19.

KIURA, M.; OHIRA, M.; MATSUMOTO, K.-i. Webjig: An automated user data collection system for website usability evaluation. In: *Human-Computer Interaction. New Trends*. [S.l.]: Springer, 2009. p. 277–286. Citado na página 34.

KLUTH, W.; KREMPELS, K.-H.; SAMSEL, C. Automated usability testing for mobile applications. *International Conference on Web Information Systems and Technologies*, p. 149 – 156, 2014. Citado na página 53.

LAW, E. et al. Towards a shared definition of user experience. In: *CHI '08 Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2008. (CHI EA '08), p. 2395–2398. Citado na página 1.

LETTNER, F.; HOLZMANN, C. Usability evaluation framework: Automated interface analysis for android applications. In: *Proceedings of the 13th International Conference on Computer Aided Systems Theory - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012. (EUROCAST'11), p. 560–567. ISBN 978-3-642-27578-4. Citado na página 53.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. In: *Soviet physics doklady*. [S.l.: s.n.], 1966. v. 10, n. 8, p. 707–710. Citado na página 23.

LI, C.-h.; KIT, C.-c. Web structure mining for usability analysis. In: IEEE. *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*. [S.l.], 2005. p. 309–312. Citado na página 34.

LISTER, M. Usability testing software for the internet. In: ACM. *CHI'01 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2001. p. 17–18. Citado na página 34.

LÓPEZ, J. M.; FAJARDO, I.; ABASCAL, J. Towards remote empirical evaluation of web pages usability. In: *Human-Computer Interaction. Interaction Design and Usability*. [S.l.]: Springer, 2007. p. 594–603. Citado 2 vezes nas páginas 2 e 34.

MABROUKEH, N. R.; EZEIFE, C. I. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, ACM, v. 43, n. 1, p. 3, 2010. Citado na página 15.

MACLEOD, M.; RENGGER, R. The development of drum: A software tool for video-assisted usability evaluation. *People and Computers*, Cambridge University Press, p. 293–293, 1993. Citado na página 12.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado 2 vezes nas páginas 17 e 59.

MOBASHER, B. *Web usage mining*. [S.l.]: Chapter, 2006. v. 12. Citado 4 vezes nas páginas 13, 14, 17 e 39.

MOSQUEIRA-REY, E. et al. A multi-agent system based on evolutionary learning for the usability analysis of websites. In: *Intelligent Agents in the Evolution of Web and Applications*. [S.l.]: Springer, 2009. p. 11–34. Citado na página 34.

MUELLER, C. J. et al. An economical approach to usability testing. In: IEEE. *Computer Software and Applications Conference, 2009. COMPSAC'09. 33rd Annual IEEE International*. [S.l.], 2009. v. 1, p. 124–129. Citado na página 10.

NAKAMICHI, N. et al. Webtracer: A new web usability evaluation environment using gazing point information. *Electronic Commerce Research and Applications*, Elsevier, v. 6, n. 1, p. 63–73, 2007. Citado na página 34.

NEBELING, M.; SPEICHER, M.; NORRIE, M. C. Crowdstudy: General toolkit for crowdsourced evaluation of web interfaces. In: ACM. *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*. [S.l.], 2013. p. 255–264. Citado na página 34.

NIELSEN, J. Usability inspection methods. In: ACM. *Conference companion on Human factors in computing systems*. [S.l.], 1994. p. 413–414. Citado 5 vezes nas páginas 13, 8, 75, 76 e 77.

NIELSEN, J. *Designing web usability: The practice of simplicity*. Thousand Oaks, CA, USA: New Riders Publishing, 2000. Citado na página 2.

NIELSEN, J.; LANDAUER, T. K. A mathematical model of the finding of usability problems. In: ACM. *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. [S.l.], 1993. p. 206–213. Citado 5 vezes nas páginas 1, 2, 8, 9 e 40.

NIELSEN, J.; NORMAN, D. The definition of user experience. *nngroup,[Online]. Available: http://www.nngroup.com/articles/definition-user-experience/.[Accessed 20 July 2015]*, 2015. Citado na página 1.

NORMAN, K. L.; PANIZZI, E. Levels of automation and user participation in usability testing. *Interacting with computers*, Oxford University Press, v. 18, n. 2, p. 246–264, 2006. Citado na página 34.

OBENDORF, H.; WEINREICH, H.; HASS, T. Automatic support for web user studies with scone and tea. In: ACM. *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. [S.l.], 2004. p. 1135–1138. Citado na página 34.

PAGANELLI, L.; PATERNÒ, F. Intelligent analysis of user interactions with web applications. In: ACM. *Proceedings of the 7th international conference on Intelligent user interfaces*. [S.l.], 2002. p. 111–118. Citado 3 vezes nas páginas 3, 12 e 34.

PALANQUE, P. et al. Supporting usability evaluation of multimodal man-machine interfaces for space ground segment applications using petri net based formal specification. In: *Ninth International Conference on Space Operations, Rome, Italy*. [S.l.: s.n.], 2006. Citado na página 8.

PASCUAL, V.; DÜRSTELER, J. C. Wet: a prototype of an exploratory search system for web mining to assess usability. In: IEEE. *Information Visualization, 2007. IV'07. 11th International Conference*. [S.l.], 2007. p. 211–215. Citado na página 34.

PATERNÒ, F.; SANTORO, C. Remote usability evaluation: Discussion of a general framework and experiences from research with a specific tool. In: LAW, E.-C.; HVANNBERG, E.; COCKTON, G. (Ed.). *Maturing Usability*. [S.l.]: Springer London, 2008, (Human-Computer Interaction Series). p. 197–221. Citado na página 8.

PAUL, A.; YADAMSUREN, B.; ERDELEZ, S. An experience with measuring multi-user online task performance. In: IEEE. *Information and Communication Technologies (WICT), 2012 World Congress on*. [S.l.], 2012. p. 639–644. Citado na página 34.

PETERSEN, K. et al. Systematic mapping studies in software engineering. In: SN. *12th International Conference on Evaluation and Assessment in Software Engineering*. [S.l.], 2008. v. 17, n. 1. Citado 3 vezes nas páginas 11, 19 e 20.

POLSON, P. G. et al. Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *International Journal of man-machine studies*, Elsevier, v. 36, n. 5, p. 741–773, 1992. Citado 2 vezes nas páginas 73 e 80.

POUR, P. A.; CALVO, R. A. Towards a generic framework for automatic measurements of web usability using affective computing techniques. In: *Affective computing and intelligent interaction*. [S.l.]: Springer, 2011. p. 447–456. Citado na página 34.

POWER, C.; PETRIE, H.; MITCHELL, R. A framework for remote user evaluation of accessibility and usability of websites. *Universal Access in Human-Computer Interaction. Addressing Diversity*, Springer, p. 594–601, 2009. Citado na página 34.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005. Citado na página 43.

QI, Y.; REYNOLDS, C.; PICARD, R. W. The bayes point machine for computer-user frustration detection via pressuremouse. In: ACM. *Proceedings of the 2001 workshop on Perceptive user interfaces*. [S.l.], 2001. p. 1–5. Citado na página 34.

QUINN, G. B. et al. Rcsb pdb mobile: ios and android mobile apps to provide data access and visualization to the rcsb protein data bank. *Bioinformatics*, Oxford Univ Press, v. 31, n. 1, p. 126–127, 2015. Citado na página 1.

RAMLI, R. B. M.; JAAFAR, A. B. e-rue: A cheap possible solution for usability evaluation. In: IEEE. *Information Technology, 2008. ITSim 2008. International Symposium on*. [S.l.], 2008. v. 3, p. 1–5. Citado na página 34.

REDISH, J. Expanding usability testing to evaluate complex systems. *Journal of Usability Studies*, v. 2, n. 3, p. 102–111, 2007. Citado na página 2.

SANTANA, V. F. de; BARANAUSKAS, M. C. C. Welfit: A remote evaluation tool for identifying web usage patterns through client-side logging. *International Journal of Human-Computer Studies*, Elsevier, v. 76, p. 40–49, 2015. Citado 7 vezes nas páginas 2, 3, 10, 11, 32, 34 e 99.

SAURO, J.; KINDLUND, E. A method to standardize usability metrics into a single score. In: ACM. *Proceedings of the SIGCHI conference on Human factors in computing systems*. [S.l.], 2005. p. 401–409. Citado 2 vezes nas páginas 13 e 39.

SCHOLTZ, J. Adaptation of traditional usability testing methods for remote testing. In: IEEE. *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*. [S.l.], 2001. p. 8–pp. Citado na página 34.

SCHWERZ, A. L.; MORANDINI, M.; SILVA, S. R. D. A task model proposal for web sites usability evaluation for the ergomonitor environment. In: *Human-Computer Interaction. Interaction Design and Usability*. [S.l.]: Springer, 2007. p. 1188–1197. Citado na página 34.

SHAHABI, C. et al. Knowledge discovery from users web-page navigation. In: *Research Issues in Data Engineering, 1997. Proceedings. Seventh International Workshop on*. [S.l.: s.n.], 1997. p. 20–29. Citado na página 17.

SHNEIDERMAN, B. *Designing the user interface: strategies for effective human-computer interaction*. [S.l.]: Addison-Wesley Reading, MA, 1992. v. 3. Citado na página 9.

SIOCHI, A. C.; EHRICH, R. W. Computer analysis of user interfaces based on repetition in transcripts of user sessions. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 9, n. 4, p. 309–335, 1991. Citado na página 12.

SPEICHER, M.; BOTH, A.; GAEDKE, M. Was that webpage pleasant to use? predicting usability quantitatively from interactions. In: *Current Trends in Web Engineering*. [S.l.]: Springer, 2013. p. 335–339. Citado na página 34.

SRIVASTAVA, J. et al. Web usage mining: Discovery and applications of usage patterns from web data. *ACM SIGKDD Explorations Newsletter*, ACM, v. 1, n. 2, p. 12–23, 2000. Citado 3 vezes nas páginas 13, 14 e 17.

SSEMUGABI, S.; VILLIERS, R. D. A comparative study of two usability evaluation methods using a web-based e-learning application. In: ACM. *Proceedings of the 2007 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*. [S.l.], 2007. p. 132–142. Citado na página 2.

THELIN, T.; RUNESON, P.; WOHLIN, C. An experimental comparison of usage-based and checklist-based reading. *Software Engineering, IEEE Transactions on*, IEEE, v. 29, n. 8, p. 687–704, 2003. Citado na página 73.

UEHLING, D. L.; WOLF, K. User action graphing effort (usage). In: ACM. *Conference companion on human factors in computing systems*. [S.l.], 1995. p. 290–291. Citado 2 vezes nas páginas 12 e 31.

VANDERDONCKT, J.; BEIREKDAR, A.; NOIRHOMME-FRAITURE, M. Automated evaluation of web usability and accessibility by guideline review. In: *Web Engineering*. [S.l.]: Springer, 2004. p. 17–30. Citado na página 34.

VARGAS, A.; WEFFERS, H.; ROCHA, H. da. Discovering and analyzing patterns of usage to detect usability problems in web applications. In: *Intelligent Systems Design and Applications (ISDA), 2011 11th International Conference on*. [S.l.: s.n.], 2011. p. 575–580. Citado 2 vezes nas páginas 1 e 34.

VARGAS, A.; WEFFERS, H.; ROCHA, H. V. d. Analyzing user interaction logs to evaluate the usability of web applications. In: IEEE. *Web Society (SWS), 2011 3rd Symposium on*. [S.l.], 2011. p. 61–67. Citado na página 34.

VARGAS, A.; WEFFERS, H.; ROCHA, H. V. da. A method for remote and semi-automatic usability evaluation of web-based applications through users behavior analysis. In: ACM. *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*. [S.l.], 2010. p. 19. Citado na página 34.

VASCONCELOS, L. G. de; BALDOCHI JR., L. A. Towards an automatic evaluation of web applications. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2012. (SAC '12), p. 709–716. Citado na página 32.

VASCONCELOS, L. G. de; JR, L. A. B. Towards an automatic evaluation of web applications. In: ACM. *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. [S.l.], 2012. p. 709–716. Citado na página 34.

VIERA, A. J.; GARRETT, J. M. et al. Understanding interobserver agreement: the kappa statistic. *Fam Med*, v. 37, n. 5, p. 360–363, 2005. Citado 2 vezes nas páginas 13 e 90.

WEST, R.; LEHMAN, K. Automated summative usability studies: An empirical evaluation. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2006. (CHI '06), p. 631–639. Citado 2 vezes nas páginas 10 e 34.

WOHLIN, C. et al. *Experimentation in software engineering*. [S.l.]: Springer, 2012. Citado 4 vezes nas páginas 69, 72, 73 e 78.

XU, L.; XU, B. Applying agent into intelligent web application testing. In: IEEE. *Cyberworlds, 2007. CW'07. International Conference on*. [S.l.], 2007. p. 61–65. Citado na página 34.

YUSOP, N. S. M.; GRUNDY, J.; VASA, R. Reporting usability defects: Limitations of open source defect repositories and suggestions for improvement. In: *Proceedings of the ASWEC 2015 24th Australasian Software Engineering Conference*. New York, NY, USA: ACM, 2015. (ASWEC ' 15 Vol. II), p. 38–43. ISBN 978-1-4503-3796-0. Disponível em: <http://doi.acm.org/10.1145/2811681.2811689>. Citado na página 75.

ZAKI, M. J. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, v. 42, n. 1, p. 31–60, 2001. ISSN 1573-0565. Citado na página 16.

ZHOU, B.; NEAMTIU, I.; GUPTA, R. A cross-platform analysis of bugs and bug-fixing in open source projects: Desktop vs. android vs. ios. In: *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*. New York, NY, USA: ACM, 2015. (EASE '15), p. 7:1–7:10. Citado na página 47.