

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

MATHEUS ALBERTO SOSSAI

**SISTEMA PARA CRIAÇÃO DE FILTROS PARA TRANSFORMAR
ARQUIVOS DE LOG DA WAUTT EM ARQUIVOS PARA OUTRAS
FERRAMENTAS**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2015

MATHEUS ALBERTO SOSSAI

**SISTEMA PARA CRIAÇÃO DE FILTROS PARA TRANSFORMAR
ARQUIVOS DE LOG DA WAUTT EM ARQUIVOS PARA OUTRAS
FERRAMENTAS**

Trabalho de conclusão de curso apresentada à disciplina de Trabalho de Conclusão de Curso I do curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Luciano Tadeu Esteves Pansanato

CORNÉLIO PROCÓPIO

2015

RESUMO

SOSSAI, Matheus Alberto. SISTEMA PARA CRIAÇÃO DE FILTROS PARA TRANSFORMAR ARQUIVOS DE LOG DA WAUTT EM ARQUIVOS PARA OUTRAS FERRAMENTAS. 38 f. Trabalho de conclusão de curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

Com o surgimento de novas tecnologias, as páginas web puderam se tornar mais próximas de aplicações *desktop*, como é o caso das *Rich Internet Applications*. Diversos testes e avaliações de usabilidade são utilizados com o objetivo de encontrar problemas de interface. Para facilitar a realização destes testes, ferramentas foram desenvolvidas e aperfeiçoadas para capturar a interação do usuário com o sistema. Dentre estas ferramentas, a WAUTT se destaca por capturar informação sobre a interação do usuário com um sistema web no nível de elementos da página e eventos. Este trabalho tem como objetivo apresentar o desenvolvimento de um sistema que permite a filtragem e o pré-processamento de arquivos de *log* importados da WAUTT e possibilitar a exportação para formatos de saída ARFF e CSV para análises posteriores em outras ferramentas.

Palavras-chave: Avaliação de Usabilidade, Análise de Log, Aplicação Internet Rica, Filtragem

ABSTRACT

SOSSAI, Matheus Alberto. SYSTEM FOR CREATION OF FILTERS TO TRANSFORM LOG FILES OF WAUTT FILES TO OTHER TOOLS. 38 f. Trabalho de conclusão de curso – Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

With the emergence of new technologies, web pages could become closer to desktop applications, as in the case of Rich Internet Applications. Several tests and usability assessments are used in order to find interface problems. To facilitate the completion of these tests, tools have been developed and refined to capture user interaction with the system. Among these tools, the WAUTT stands out by capturing information about user interaction with a web system at the level of page elements and events. This work aims to present the development of a system for filtering and pre-processing of log files imported from WAUTT and enable export to ARFF and CSV output formats for further analysis in other tools.

Keywords: Usability Evaluation, Log Analysis, Rich Internet Application, Filtering

LISTA DE FIGURAS

FIGURA 1	– Arquitetura do WAUTT	12
FIGURA 2	– Exemplo de dados capturados	14
FIGURA 3	– Exemplo de arquivo ARFF	16
FIGURA 4	– Protótipo do <i>Login</i>	21
FIGURA 5	– Protótipo do cadastro	21
FIGURA 6	– Protótipo da página de projeto de criação de filtros	22
FIGURA 7	– Protótipo de informações do projeto	22
FIGURA 8	– Protótipo de edição do projeto	23
FIGURA 9	– Protótipo da página de projetos públicos	23
FIGURA 10	– Protótipo de informações de projeto público	24
FIGURA 11	– Diagrama entidade-relacionamento	25
FIGURA 12	– Tela de <i>login</i>	26
FIGURA 13	– Tela de boas vindas	26
FIGURA 14	– Tela de projetos de filtros privados	27
FIGURA 15	– Criação e informações de um projeto	27
FIGURA 16	– Projeto de filtragem	28
FIGURA 17	– Exemplos de tipos de preprocessamento e filtragem	29
FIGURA 18	– Informações de projeto público	30
FIGURA 19	– Visualização de projeto público	30
FIGURA 20	– Resultado ARFF	31
FIGURA 21	– Resultado CSV	32
FIGURA 22	– Gráfico com resultado gerado por CSV	32

LISTA DE SIGLAS

ANSI	<i>American National Standards Institute</i>
CSV	<i>Comma-Separated Values</i>
DOM	<i>Document Object Model</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>HyperText Transfer Protocol</i>
IP	<i>Internet Protocol</i>
JSON	<i>JavaScript Object Notation</i>
PHP	<i>PHP: HyperText Preprocessor</i>
RIA	<i>Rich Internet Applications</i>
SQL	<i>Structured Query Language</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>eXtensible Markup Language</i>
WAUTT	<i>Web Application Usage Tracking Tool</i>
WEKA	<i>Waikato Environment for Knowledge Learning</i>

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
1.2	ORGANIZAÇÃO DO TEXTO	9
2	FUNDAMENTAÇÃO TEÓRICA	10
2.1	ANÁLISE DE <i>LOG</i>	10
2.2	ARQUIVO DE <i>LOG</i> DA WAUTT	11
2.3	SINTAXE DO ARQUIVO DE ENTRADA WEKA	15
3	TRABALHOS RELACIONADOS	17
4	DESENVOLVIMENTO DO PROJETO	19
4.1	RECURSOS UTILIZADOS	19
4.1.1	Linguagens	19
4.1.2	Ferramentas	20
4.2	PROTOTIPAÇÃO	20
4.3	SISTEMA PROPOSTO	23
5	RESULTADOS E DISCUSSÕES	31
6	CONCLUSÃO	33
	REFERÊNCIAS	34
	Apêndice A – EXPORTAÇÃO DO SISTEMA	37
	Anexo A – ARFF GERADO MANUALMENTE	38

1 INTRODUÇÃO

A informação sobre as ações do usuário na interface de um sistema web pode ser capturada por meio de ferramentas (software) específicas. O movimento do mouse, clique e preenchimento de formulário são alguns exemplos dessas ações. A captura da interação do usuário é muito utilizada para obter dados que serão utilizados para avaliação de usabilidade e estudos do comportamento do usuário (PREECE et al., 2012; TURNBULL, 2006; KELLAR; WATTERS, 2005). Esses dados também podem ser usados para justificar a realização de manutenções ou alterações nos sistemas.

O servidor que mantém o sistema web para acesso pelos usuários também registra algumas informações sobre a interação do usuário em um arquivo chamado de arquivo de log. O uso de arquivos de *log* que são armazenados no servidor web para a avaliação de usabilidade é uma estratégia conveniente e confiável (JANSEN, 2006). No entanto, os arquivos de *log* de servidor têm a limitação de conter somente a informação sobre as páginas (URLs) visitadas pelo usuário e não oferecer dados detalhados sobre a interação real do usuário com um sistema web.

Nos sistemas web modernos, usualmente chamados de Aplicações Internet Ricas (de *Rich Internet Applications* - RIAs) (FRATERNALI et al., 2010), a interface pode ser uma única página que inclui subpáginas sob demanda que gerenciam toda a interação do usuário, de maneira semelhante a uma aplicação *desktop*. Esse paradigma evita a atualização de toda a página a cada interação e permite aos sistemas carregar, apresentar e alterar de maneira independente os elementos da página. Assim, os dados sobre a interação do usuário que podem ser armazenados no servidor web não são suficientes para obter informação detalhada sobre o uso real do sistema. No caso de RIAs, o interesse é trabalhar com arquivos de *log* com informação no nível de elementos da página e de eventos associados com a interação do usuário.

A *Web Application Usage Tracking Tool* (WAUTT) (RIVOLLI et al., 2014) é uma ferramenta para capturar informações detalhadas sobre a interação do usuário com uma RIA. A WAUTT é utilizada para capturar a interação do usuário com sistemas web em um computador

cliente e registrar todos os eventos coletados em um arquivo de *log* no servidor para serem usados para avaliação de usabilidade. A captura dos dados ocorre com a inserção de um código JavaScript nas páginas da RIA.

Os arquivos de *log* da interação do usuário com um sistema web que utiliza o paradigma RIA podem ser muito extensos, tornando inviável a análise manual em uma avaliação de usabilidade. Portanto, é necessária a utilização de técnicas de análise de *log* para facilitar o trabalho dos profissionais envolvidos (avaliadores). Por exemplo, as técnicas de mineração de dados e de visualização de informação são alternativas que podem ser empregadas.

No uso do arquivo de *log* da WAUTT como entrada para ferramentas de mineração de dados ou de visualização de informação: o formato (sintaxe) do arquivo de *log* é diferente do formato do arquivo de entrada de dados dessas ferramentas. Para contornar esse problema, a proposta deste trabalho foi desenvolver um sistema interativo para a criação de filtros para transformar o arquivo de *log* da WAUTT em um arquivo de dados no formato de uma ferramenta utilizada para análise de *log*. O sistema que foi desenvolvido deve não somente fornecer uma interface para a criação de filtros, mas também permitir que estes sejam armazenados e tornados públicos para o uso por diferentes usuários. Além disso, um filtro pode servir de base para a construção de outro filtro.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Propor e desenvolver um sistema interativo para a criação de filtros para transformar o arquivo de *log* da WAUTT em arquivos para outras ferramentas.

1.1.2 OBJETIVOS ESPECÍFICOS

- Analisar o formato da entrada de cada grupo de dados capturados pela ferramenta WAUTT.
- Projetar uma proposta de sistema interativo para permitir a criação de filtros e pré-processamento de dados para as entradas do arquivo de *log*.
- Desenvolver o sistema proposto.
- Avaliar a proposta de interface interativa com usuários potenciais.

1.2 ORGANIZAÇÃO DO TEXTO

Este trabalho de conclusão de curso está organizado da seguinte maneira: No Capítulo 2 é apresentada a fundamentação teórica, sendo apresentados conceitos de análise de *log* e as ferramentas WAUTT e *Waikato Environment for Knowledge Learning* (WEKA). No Capítulo 3 é mostrado um trabalho relacionado. O desenvolvimento do projeto é apresentado no Capítulo 4, sendo especificadas as tecnologias e ferramentas utilizadas, a prototipação do projeto e o sistema desenvolvido. Os resultados obtidos são comentados no capítulo 5 e o Capítulo 6 conclui o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos relacionados a esta proposta de projeto de trabalho de conclusão de curso. Na seção 2.1 é explicada a análise de *log* e suas vantagens e desvantagens. Na seção 2.2 é descrita a ferramenta WAUTT e o tipo de *log* produzido. Na seção 2.3 é descrito o WEKA e seu arquivo de entrada.

2.1 ANÁLISE DE LOG

A análise de *log* é um método de avaliação de usabilidade geralmente empregado para sistemas web. Os testes realizados através dessa análise produzem vestígios do real uso do sistema web. Os *logs* podem ser criados tanto por requisições do servidor quanto pela interação do lado do usuário. Além disso, é um teste não intrusivo e não há interferências no comportamento do participante, sendo praticamente invisível, ou seja, os usuários não precisam saber que as informações de uso estão sendo capturadas (CYBIS et al., 2010). Os *logs* são analisados pelos avaliadores para encontrar maneiras de tentar melhorar a usabilidade das páginas e até problemas de funcionamento.

Uma vantagem desse tipo de análise é que não precisa ser realizada de maneira presencial, ou seja, pode ser realizada remotamente. Com isso, o usuário pode estar sendo monitorado sem perceber, o que evita possíveis constrangimentos ao ser monitorado e minimiza a ocorrência de interações artificiais. Além disso, com baixo custo, é possível envolver uma grande quantidade de usuários que estão separados tanto no espaço quanto no tempo (CYBIS et al., 2010; IVORY; HEARST, 2001).

O objetivo é identificar comportamentos e padrões que indicam problemas de usabilidade através da análise dos dados capturados entre usuários e um sistema. Dessa maneira, os dados gerados são, normalmente, relacionados à usabilidade de produtos, caminhos que o usuário realizou, padrões de uso, entre outros (MACK; NIELSEN, 2001).

Em um servidor, geralmente, os registros em arquivos de *log* possuem os seguintes

dados: data e o horário da requisição, a url requisitada, o sistema operacional e o navegador utilizado, o protocolo empregado na requisição e o endereço de IP que faz a requisição (PREECE et al., 2012).

Existem quatro maneiras que as informações adquiridas de um arquivo de *log* podem ser apresentadas (CYBIS et al., 2010):

- **Visitas:** contabiliza o número de usuários, a quantidade de páginas que foram acessadas em um período de tempo, a duração de cada um desses acessos, além do sistema operacional e navegador utilizados e o local de onde partiu a requisição do usuário.
- **Referência:** registra os termos que foram pesquisados para acessar o sistema e a lista de endereços web de onde veio o usuário.
- **Conteúdo:** são as páginas com mais visitas.
- **Navegação:** é todo o caminho que o usuário percorreu para alcançar o seu objetivo.

Os dados que são coletados quando o usuário está realizando atividades em seu próprio ambiente de trabalho são bastante úteis para a identificação de problemas de usabilidade, visto que são decorrentes do uso de um sistema web em situações reais. No entanto, também podem trazer resultados muito superficiais em relação aos objetivos da análise, pois apenas registra a duração e o caminho percorrido pelos usuários em uma visita (CYBIS et al., 2010). Dessa maneira, os *logs* dos servidores são limitados, não trazendo informações detalhadas sobre a interação real do usuário com o sistema (CYBIS et al., 2010).

O uso da ferramenta WAUTT (RIVOLLI et al., 2014) permite contornar essa limitação através da captura de eventos realizados pelo usuário pela interação com elementos contidos em uma página web (SANTANA, 2013).

2.2 ARQUIVO DE LOG DA WAUTT

A WAUTT é uma ferramenta capaz de capturar a interação do usuário com uma aplicação RIA. Existem alguns tipos de papéis que o usuário pode desempenhar no sistema e podem ajudar a entender o funcionamento da arquitetura da WAUTT (RIVOLLI et al., 2014):

- **Desenvolvedor:** adiciona o *script* da WAUTT nas páginas de sua aplicação RIA onde pretende capturar a interação do usuário;

- Avaliador: planeja e aplica a avaliação de uma aplicação RIA e geralmente é especialista em usabilidade.
- Usuário WAUTT: interage com a interface administrativa da WAUTT para consultar as informações adquiridas e configurar a ferramenta; normalmente é o desenvolvedor ou o avaliador.
- Usuário da aplicação: tem a interação capturada na aplicação RIA pela WAUTT.

Na Figura 1 é mostrada a organização da arquitetura da WAUTT, destacando a Interface, o Servidor da WAUTT e o Servidor de Aplicação RIA.

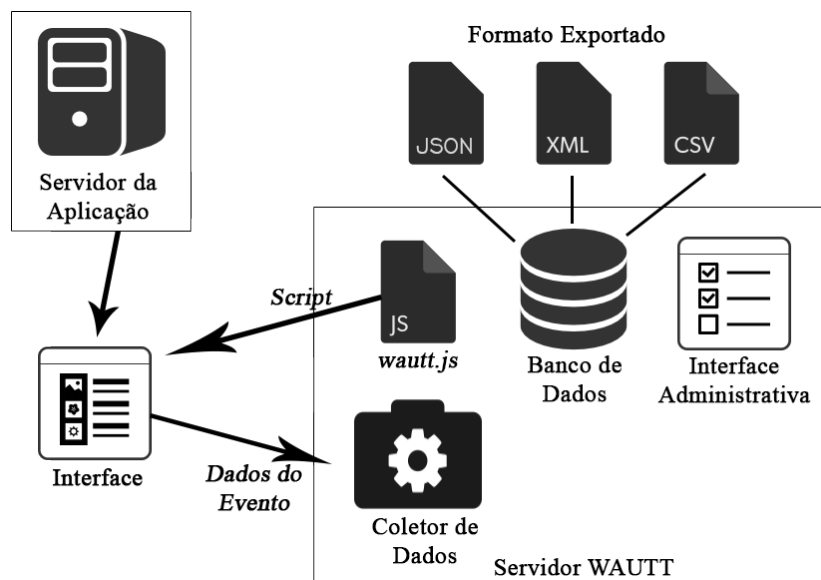


Figura 1: Arquitetura do WAUTT.

Fonte: Adaptado de (RIVOLLI et al., 2014)

A aplicação RIA é hospedada em um servidor web que é o Servidor da Aplicação. Qualquer tecnologia utilizada pela aplicação pode ter a ferramenta WAUTT inserida em seu código *HyperText Markup Language* (HTML). O arquivo “*wautt.js*” é hospedado no servidor WAUTT e é carregado remotamente ao ser referenciado no código da aplicação RIA. Desta forma, a lógica da captura fica centralizada em um único local. Outras ferramentas que capturam a interação do usuário por meio do cliente utilizam abordagens semelhantes, como é o caso do *Google Analytics*¹.

O Servidor WAUTT é responsável por fornecer todo o contato que o desenvolvedor e o avaliador têm com a ferramenta. Esse servidor possui uma interface administrativa que permite

¹<http://www.google.com/analytics/>

visualizar, monitorar e personalizar o que é capturado e exportar para análises detalhadas e específicas por outras ferramentas. O servidor é quem fornece o *script* “*wautt.js*” para capturar os dados da interação do usuário. Além disso, recebe os dados e cria os *logs* em uma base de dados.

Os navegadores e dispositivos são representados pela Interface, com função de visualizar e interagir com a aplicação RIA. As páginas são monitoradas e têm seus dados de captura enviados para o Servidor da WAUTT. Esses dados são armazenados em um *buffer* interno que é enviado periodicamente ao servidor. Na área administrativa é possível determinar de que maneira o *buffer* é enviado: ao atingir um determinado tamanho ou uma quantidade de segundos. Esse processo se faz necessário para diminuir e amenizar o número de requisições *Hypertext Transfer Protocol* (HTTP).

O usuário da WAUTT precisa realizar um cadastro da aplicação no Servidor da WAUTT. Esse registro é necessário para que o desenvolvedor tenha acesso ao código, que contém o “*wautt.js*”, para ser inserido nas páginas, além de ter acesso aos dados capturados e às outras funcionalidades da ferramenta. Nas configurações, por padrão, são habilitados todos os eventos e elementos para a captura.

Existe também um recurso que pode ser configurado na interface administrativa que é o gerenciamento de avaliações e tarefas. Nesse recurso é possível cadastrar avaliações distintas e criar um conjunto de tarefas para cada uma. Dessa maneira, é possível determinar uma tarefa a ser realizada, durante um teste de usabilidade síncrono e presencial, e associar os dados capturados com a tarefa proposta.

Para garantir integridade e consistência dos dados capturados, o *script* “*wautt.js*” é gerado dinamicamente e valida se o domínio cadastrado pelo usuário é o mesmo da requisição. Além disso, para evitar trocas de requisições manuais, são feitas trocas de *tokens* entre o *script* e o Servidor da WAUTT.

Os dados coletados pela WAUTT são armazenados em um banco de dados relacional. Na Figura 2 é mostrado um exemplo de informações contidas nos arquivos de *log* que são exibidos na área administrativa.

Existem associações de cada interação do usuário registrada no banco de dados com os grupos de dados a seguir:

- Dados da sessão: Uma sessão é criada quando o usuário acessa a página da aplicação e é encerrada quando fecha o navegador. Este grupo de dados inclui contém um identificador único, o nome e a versão do navegador e do sistema operacional e o endereço de *Internet*

#40831	-	Time: 2013-05-20T10:47:59.092	Title: Novo Dred	URL: http://10.20.10.165/perolas/editor
Event:	unload	Browser:	Chrome 23	OS: Windows
Resolution X:	1366	Resolution Y:	768	Width: 1366
Height:	667	IP:	10.20.39.205	Tag: BODY
ID:	---	Class:	---	Text: Drediaada Usuário
Position X:	---	Position Y:	---	
#40803	-	Time: 2013-05-20T10:47:57.748	Title: Novo Dred	URL: http://10.20.10.165/perolas/editor
Event:	mousemove	Browser:	Chrome 23	OS: Windows
Resolution X:	1366	Resolution Y:	768	Width: 1366
Height:	667	IP:	10.20.39.205	Tag: A
ID:	---	Class:	---	Text: Finalizar »
Position X:	580	Position Y:	21	
#40780	-	Time: 2013-05-20T10:47:55.918	Title: Novo Dred	URL: http://10.20.10.165/perolas/editor
Event:	mousemove	Browser:	Chrome 23	OS: Windows
Resolution X:	1366	Resolution Y:	768	Width: 1366
Height:	667	IP:	10.20.39.205	Tag: INPUT
ID:	---	Class:	---	Text: ---
Position X:	730	Position Y:	111	
#40750	-	Time: 2013-05-20T10:47:54.450	Title: Novo Dred	URL: http://10.20.10.165/perolas/editor
Event:	mouseup	Browser:	Chrome 23	OS: Windows
Resolution X:	1366	Resolution Y:	768	Width: 1366
Height:	667	IP:	10.20.39.205	Tag: H1
ID:	valortitulo	Class:	---	Text: Aula
Position X:	717	Position Y:	95	
#40748	-	Time: 2013-05-20T10:47:54.318	Title: Novo Dred	URL: http://10.20.10.165/perolas/editor
Event:	mousedown	Browser:	Chrome 23	OS: Windows
Resolution X:	1366	Resolution Y:	768	Width: 1366
Height:	667	IP:	10.20.39.205	Tag: H1
ID:	valortitulo	Class:	---	Text: Aula
Position X:	717	Position Y:	95	

Figura 2: Exemplo de dados capturados.

Protocol (IP).

- Dados da navegação: É possível que uma aplicação carregue outras páginas durante seu uso ou apenas partes de uma única página. Esse grupo de dados inclui: o título da página, a *Uniform Resource Locator (URL)*, os parâmetros, os métodos de solicitação (GET ou POST) e o tamanho da página.
- Dados de evento: Esse grupo varia de acordo com o tipo de evento correspondendo à detalhes específicos dos mesmos. É possível adquirir dados identificados como “quando” (instante de tempo) a interação foi realizada e “o que” (tipo do evento e seus detalhes) foi capturado.
- Dados Document Object Model (DOM): Esse grupo contém os nomes dos elementos e seus atributos. Os atributos presentes em todos os elementos como *id*, *class* e *title*, e em elementos específicos como, em um *input*, *name*, *type* e *value*.
- Dados da tarefa: Caso o recurso estiver sendo utilizado, é possível realizar a análise dos dados capturados de acordo com a tarefa.

Ferramentas adequadas são necessárias para auxiliar na análise quantitativa e na interpretação dos dados após obter os dados capturados das interações do usuário, devido à possibilidade de coletar grande número de dados automaticamente.

Os dados obtidos na captura ficam disponíveis no Servidor WAUTT e é possível exportar para os formatos *JavaScript Object Notation (JSON)*, *eXtensible Markup Language (XML)* e *Comma-Separated Values (CSV)* para que as informações possam passar por análises em ferramentas específicas. Além disso, a WAUTT mostra algumas informações básicas referentes aos dados capturados: detalhes das sessões; número de páginas abertas por sessão; gráficos

de dados capturados por período de tempo; total de dados capturados por evento, elemento da página, tarefa e sessão.

2.3 SINTAXE DO ARQUIVO DE ENTRADA WEKA

WEKA é um programa criado com o propósito de identificar informações a partir de dados brutos. Existem várias tarefas de mineração de dados (*data mining*) suportadas pela ferramenta como: pré-processamento de dados, classificação, agrupamento, regressão, visualização e seleção de características (SUBBULAKSHMI et al., 2012).

O programa pode ser treinado para executar capacidades de aprendizado de máquina e derivar informações úteis em forma de tendências e padrões. O programa opera na indicação que os dados do usuário estão disponíveis como um arquivo ou relação, que significa que cada dado de objeto é descrito por um número fixo de atributos que normalmente são de um tipo específico: nominal, alfanumérico ou valores numéricos (SUBBULAKSHMI et al., 2012).

O WEKA carrega arquivos na extensão ARFF que têm duas seções distintas. São elas: (BOUCKAERT et al., 2013):

- Cabeçalho: possui o nome da relação, uma lista de atributos e seus tipos.
- Dados: lista de dados coletados separados por vírgula.

Um arquivo do tipo ARFF é organizado da seguinte forma e como mostrado no exemplo da Figura 3 (BOUCKAERT et al., 2013):

- Comentários: conteúdo opcional no início do documento para indicação de fonte, significado ou contexto.
- Relação: nome interno do conjunto de dados.
- Atributos nominais: nomes diferentes para distinguir uma instância de outra.
- Atributos numéricos: como o nome sugere, são identificados por um valor numérico que pode ser real, *integer* (inteiro) ou *numeric* (numérico).
- Atributos *string*: é o destino padrão ou variável de classe usado para previsão.
- Dados: são os dados separados por vírgula para cada atributo.

Comentários

```
% This is a toy example, the UCI weather dataset.
% Any relation to real weather is purely coincidental.
```

Relação

```
@relation golfWeatherMichigan_1988/02/10_14days
```

Atributos Nominais

```
@attribute outlook {sunny, overcast, rainy}
@attribute windy {TRUE, FALSE}
```

Atributos Numéricos

```
@attribute temperature real
@attribute humidity real
```

Atributos String

```
@attribute play {yes, no}
```

Dados

```
@data
sunny,FALSE,85,85,no
sunny,TRUE,80,90,no
overcast,FALSE,83,86,yes
rainy,FALSE,70,96,yes
rainy,FALSE,68,80,yes
```

Figura 3: Exemplo de arquivo ARFF.

Fonte: Adaptado de (BOUCKAERT et al., 2013)

3 TRABALHOS RELACIONADOS

Existem outras ferramentas que desempenham funções semelhantes à WAUTT e que também obtêm dados sobre a interação do usuário para serem utilizados em avaliações de usabilidade, como é o caso ErgoMonitor (SCHWERZ et al., 2007), AWUSA (TIEDTKE et al., 2002), WebQuilt (HONG et al., 2001), UsaProxy (ATTERER et al., 2006), WAUTER (BALBO et al., 2005), WebTracker (CHOO et al., 2000), WebRemUSINE (PAGANELLI; PATERNO, 2002), WELFIT (SANTANA; BARANAUSKAS, 2010) e a USABILICS (VASCONCELOS; JR, 2011).

O ErgoMonitor monitora a usabilidade web por meio de *log*. Ele coleta e analisa dados relacionados às interações dos usuários com a página *web* e contidos no *log* de acessos do servidor. Porém, não analisa os dados em aplicações que possuem muitas interações realizadas no cliente. A AWUSA também realiza a análise dos dados de acesso do servidor, mas além disso, considera categorias de páginas e links, determinados ao analisar o código HTML da aplicação. Essa ferramenta utiliza também dados de transações e de *log* adicionais, mas ainda não possui informações suficientes para aplicações RIA.

O WebQuilt utiliza a captura através de um servidor *proxy*, mas difere do *proxy* HTTP tradicional, sendo baseado na URL. Dentro da *string* de consulta da URL, o destino pretendido é codificado e os *links* são redirecionados de uma maneira que as próprias URLs apontam para o *proxy*. Dessa maneira, evita o transtorno de ter que configurar o *proxy* toda vez que for utilizar a ferramenta. Apesar disso, o serviço não captura interações do lado do cliente. Pois o seu foco são técnicas de visualização de informação aplicada nos dados.

A captura de interação realizada pelo UsaProxy se aproximam mais da WAUTT por inserir um *JavaScript* em todas as páginas antes da entrega destas ao cliente. Esse código tem a capacidade de capturar as interações do usuário como a rolagem da página, o redimensionamento da janela, cliques e movimentos do *mouse*, mudança de foco da entrada de dados e o pressionamento de teclas. Além disso, como seu nome sugere, utiliza um *proxy* HTTP e intercepta todo o tráfego e registra todos os dados sobre qualquer solicitação enviada ao servidor

e a resposta do mesmo ao cliente. Os *logs* gerados são enviados periodicamente contendo os dados obtidos. Porém, esta ferramenta não funciona com *HyperText Transfer Protocol Secure* (HTTPS), pois seu conteúdo é criptografado.

O WAUTER é semelhante ao UsaProxy, inserindo um *JavaScript* nas páginas e interceptando os códigos HTML. Porém, o *proxy* é instalado na máquina do cliente para superar a questão prática de acesso ao servidor. Esta ferramenta não captura dados relacionados aos atributos de elementos HTML, nem das folhas de estilos *Cascading Style Sheets* (CSS) e também não funciona com páginas HTTPS.

O WebTracker captura a interação por meio do cliente, ou seja, captura a interação do usuário com o navegador. Além das ações realizadas pelo teclado, captura seleções feitas na barra de ferramentas e menus, que são associadas com a data, a hora e com a página que está aberta no navegador. No entanto, essa ferramenta possui limitações como o rastreamento do clique do *mouse* que somente é capturado ao selecionar um *link* na página.

As ferramentas WebRemUSINE, WELFIT e a USABILICS que realizam a captura por meio da inserção de um *JavaScript* inserido nas páginas para evitar a instalação de programas e configurações de *logs*. Essas ferramentas capturam a interação por meio do cliente. Nas ferramentas WELFIT e USABILICS, o próprio código do *JavaScript* envia os dados, enquanto no WebRemUSINE o *script* passa os dados para um *applet* Java que envia ao servidor no final da sessão. As três aplicações conseguem obter muitos detalhes da interação do usuário com a aplicação. Os dados capturados estão relacionados aos objetos de interação que disparam eventos correspondentes às ações quando manipulados pelo usuário. Uma funcionalidade não implantada nas ferramentas é a captura de eventos definidos pelo usuário. Além disso, dados relacionados aos atributos das folhas de estilo (CSS) não são capturados pelo WELFIT.

Essas ferramentas semelhantes à WAUTT não fornecem funcionalidades para o tratamento dos dados para uso em outras ferramentas de análise. Em geral, os dados são fornecidos em formato bruto, sem qualquer preparação para uso em outra ferramenta que não seja a de origem. Neste trabalho, o objetivo foi o desenvolvimento de um sistema para permitir o uso dos dados obtidos por meio da ferramenta WAUTT em outras ferramentas.

4 DESENVOLVIMENTO DO PROJETO

O sistema desenvolvido para a criação de filtros permite o pré-processamento do arquivo de *log* da WAUTT em uma interface interativa. Para isso, os dados capturados são armazenados no banco de dados no servidor da WAUTT e um arquivo XML é interpretado para a geração dos *logs*. É preciso, então, permitir que o usuário entre com este arquivo XML e escolha quais informações são úteis e de que maneira gostaria de receber estes dados, escolhendo e especificando os dados necessários na tela de edição.

Após isso, com os dados armazenados no banco de dados da aplicação, o usuário deve poder gerar a nova saída esperada que pode ser um arquivo ARFF para usar no WEKA ou um arquivo CSV para facilitar a geração de gráficos e importações por outras ferramentas.

4.1 RECURSOS UTILIZADOS

4.1.1 LINGUAGENS

O sistema foi desenvolvido nas linguagens de programação *PHP: Hypertext Preprocessor* (PHP) e *JavaScript* além da linguagem de marcação XML e HTML, de folhas de estilos (CSS) e de consulta estruturada *Structured Query Language* (SQL). A seguir é descrita cada tecnologia utilizada:

- PHP é uma linguagem de *script* gratuita e de código aberto. Os códigos desta linguagem são executados no servidor, gerando arquivos dinâmicos, podendo coletar dados, modificar dados em um banco de dados, entre diversas outras funções, e o resultado é retornado na forma de um arquivo HTML (W3SCHOOLS, 2015e).
- *JavaScript* é uma linguagem de programação que permite a interatividade com as páginas HTML. Essa linguagem tem seu *script* embutido nestes documentos e traz a interação para o lado do cliente (JAVA, 2015).
- XML é uma linguagem projetada para descrever dados. Essa linguagem é semelhante ao

HTML, mas não possui *tags* definidas, ou seja, o programador é quem deve escolher os nomes (W3SCHOOLS, 2015d).

- HTML é responsável por exibir as informações e é uma linguagem de marcação, ou seja, possui um conjunto de *tags* que especificam diferentes conteúdos dos documentos a serem exibidos (W3SCHOOLS, 2015b).
- CSS é responsável pela maneira que os elementos HTML são mostradas na tela, ou seja, define a formatação de todo o conteúdo (W3SCHOOLS, 2015a).
- SQL é uma linguagem que permite acessar e alterar bancos de dados, além disso é um padrão *American National Standards Institute* (ANSI), tendo seus principais comandos seguidos pela maioria de programas de banco de dados existentes (W3SCHOOLS, 2015c).

4.1.2 FERRAMENTAS

O sistema de gerenciamento de banco de dados utilizado foi o PostgreSQL. Esse sistema tem forte reputação de confiabilidade, integridade de dados e correção, tendo mais de 15 anos de desenvolvimento ativo (GROUP, 2015). Para visualizar e alterar campos e dados do banco de dados, foi utilizada a ferramenta pgAdmin III (TEAM, 2015) e como complemento o programa pgModeler (SILVA, 2015) para a geração de diagramas.

Para a programação do código foi utilizado o Netbeans que é uma IDE focada no desenvolvimento java, móvel e web, mas que fornece um grande conjunto de ferramentas para PHP, C e C++ (MICROSYSTEMS, 2015).

No desenvolvimento dos protótipos foi usada uma ferramenta de criação de *wireframe*, que é um guia visual usado para design de interfaces, a Balsamiq Mockups 3. Ele reproduz a experiência, no computador, de desenhar em um quadro branco (STUDIOS, 2015).

4.2 PROTOTIPAÇÃO

Utilizando a Balsamiq Mockups 3 foram criados protótipos simplificados das principais telas antes do desenvolvimento da aplicação. A Figura 4 mostra a página inicial de login, com os campos de usuário e senha, enquanto a Figura 5 exibe o cadastro simplificado de novos usuários, com os campos de usuário, senha e e-mail.

Após realizar o login, o usuário é direcionado para sua página de projetos de criação

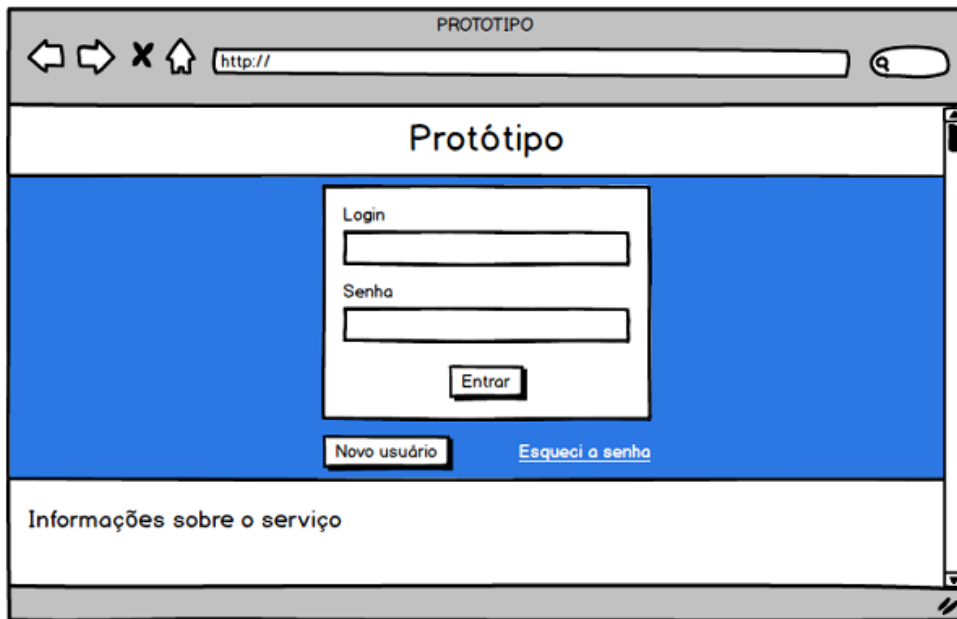


Figura 4: Protótipo do *Login*.



Figura 5: Protótipo do cadastro.

de filtros como é apresentado na Figura 6. Nessa página, as principais funções do sistema são apresentados: criação de um novo projeto e visualizar as informações de um projeto criado, como ilustrado na Figura 7. A *dialog* de informações do projeto contém o nome do projeto, as informações e as opções de editar, tornar público, gerar um arquivo no formato ARFF e excluir o projeto. No protótipo a exportação para CSV não foi especificada, mas foi implementada no projeto final.

Ao clicar na opção que permite a edição do projeto, ou no botão de criação de um novo

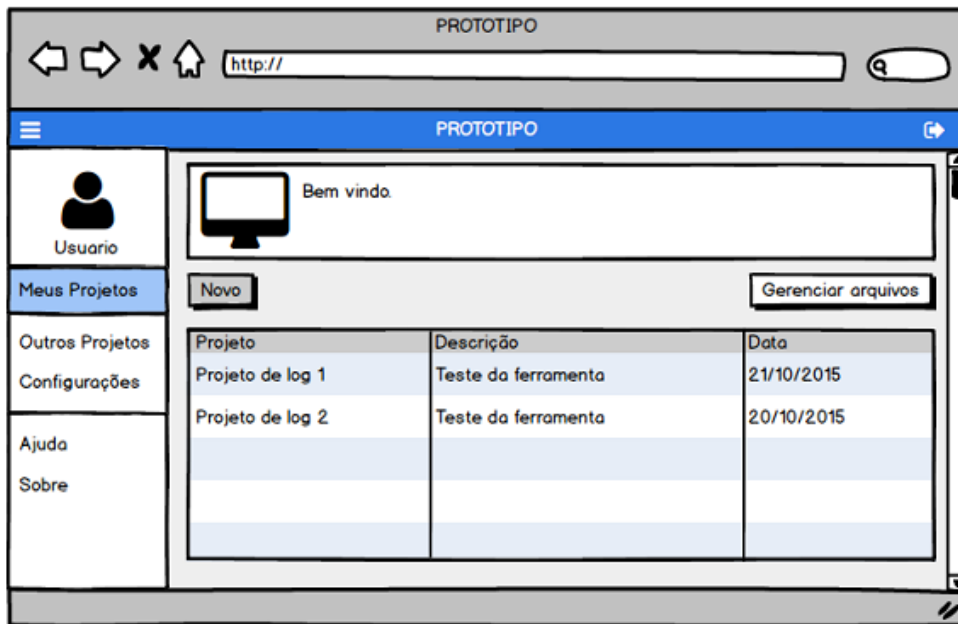


Figura 6: Protótipo da página de projeto de criação de filtros.

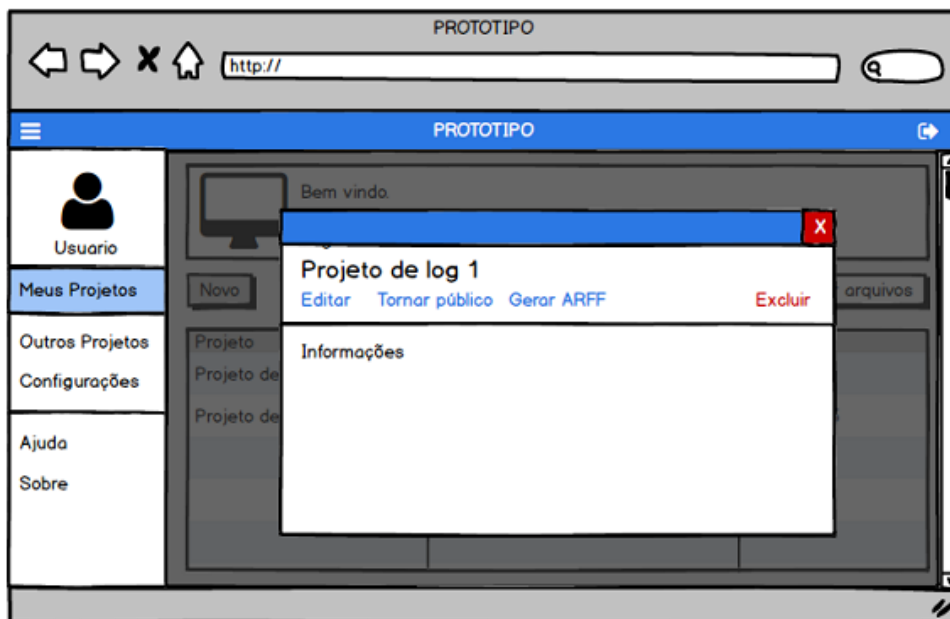


Figura 7: Protótipo de informações do projeto.

projeto, o usuário é redirecionado para a página com o carregamento do arquivo de *log* onde são mostradas as opções de configuração para a geração de arquivos de saída, como é mostrado na Figura 8.

A Figura 9 especifica como é a página que mostra os projetos de criação de filtros públicos, onde qualquer usuário cadastrado pode ver o projeto e cloná-lo, como é visto na Figura 10, que é a janela exibida ao selecionar o projeto na tabela da tela anterior.

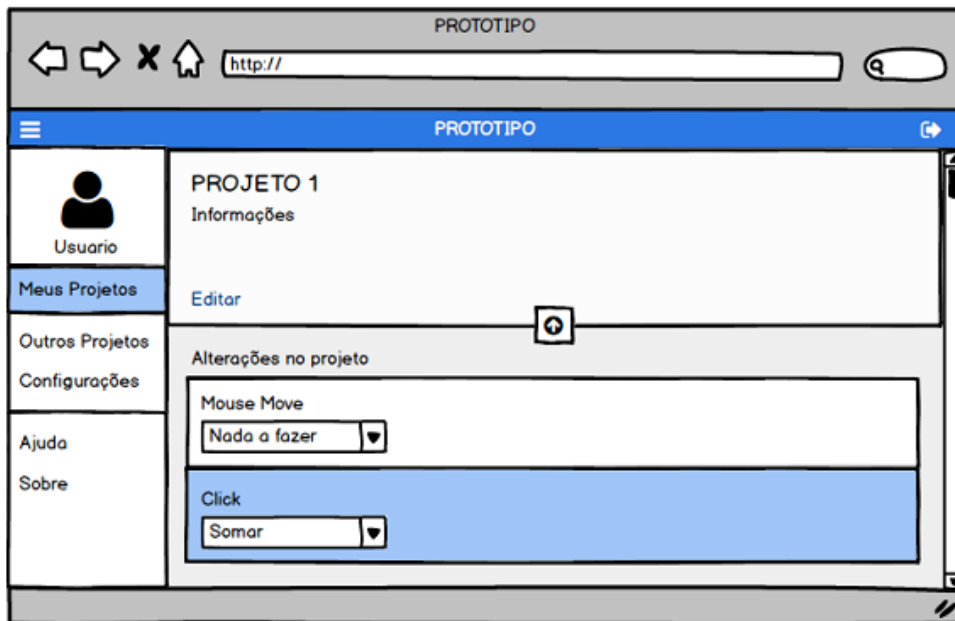


Figura 8: Protótipo de edição do projeto.



Figura 9: Protótipo da página de projetos públicos.

4.3 SISTEMA PROPOSTO

O diagrama entidade-relacionamento é exibido na Figura 11 e mostra oito tabelas de maneira que um usuário pode possuir vários projetos de filtros e um projeto, por sua vez, pode ter vários anexos. As *tags* usadas no projeto têm suas informações necessárias guardadas na tabela *tag*. Esta tabela tem ligação com as tabelas nomeadas de propriedades (são os atributos das *tags*), *subtags* (são as *tags* filhas com resultados) e a tabela de normalização (armazena o

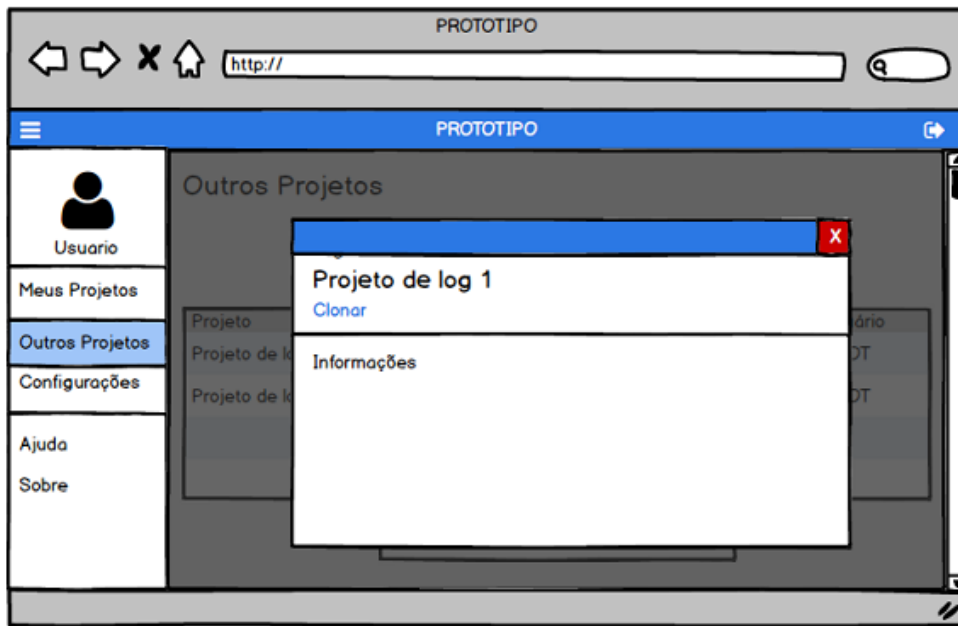


Figura 10: Protótipo de informações de projeto público.

processo e os valores dos filtros que são aplicados em cada *tag*).

O sistema segue o modelo proposto no protótipo com algumas melhorias e modificações. A Figura 12 mostra as páginas de *login* e de cadastro de novos usuários com áreas enumeradas. A área 1, mostra o formulário de *login*, com os campos de usuário e senha, a área 2 mostra o botão que leva até o formulário de cadastro que é mostrado na área 3, possuindo campos de nome, usuário, e-mail, senha e confirmação de senha.

Após o usuário realizar o cadastro e o *login* no sistema, este é direcionado para uma página de boas vindas como é mostrado na Figura 13, dividida em 3 áreas: a 1 mostra o menu lateral contendo o nome do usuário autenticado e as opções de páginas do sistema; a 2 mostra a barra superior que contém o nome da aplicação, um botão para fechar e abrir o menu lateral e um botão para fazer o *logout*; a 3 é a área onde todas as funções do sistema são mostradas e executadas.

Ao selecionar em Meus Projetos no menu, o usuário é direcionado para a página mostrada na Figura 6, onde na área 1 possui um botão para a criação de um novo projeto e na 2, uma tabela com os projetos de filtros criados anteriormente.

A Figura 15 exhibe o cadastro de um novo projeto de filtro na área 1, contendo os campos de nome do projeto, descrição curta (que é aquela que aparece na tabela de projetos), descrição completa, uma opção para marcar o projeto como público (que qualquer pessoa tem acesso) e um campo para importação e seleção de arquivos de *log* no formato XML. A área 2 mostra o *dialog* que é aberto quando algum projeto é selecionado na tabela. Essa área possui

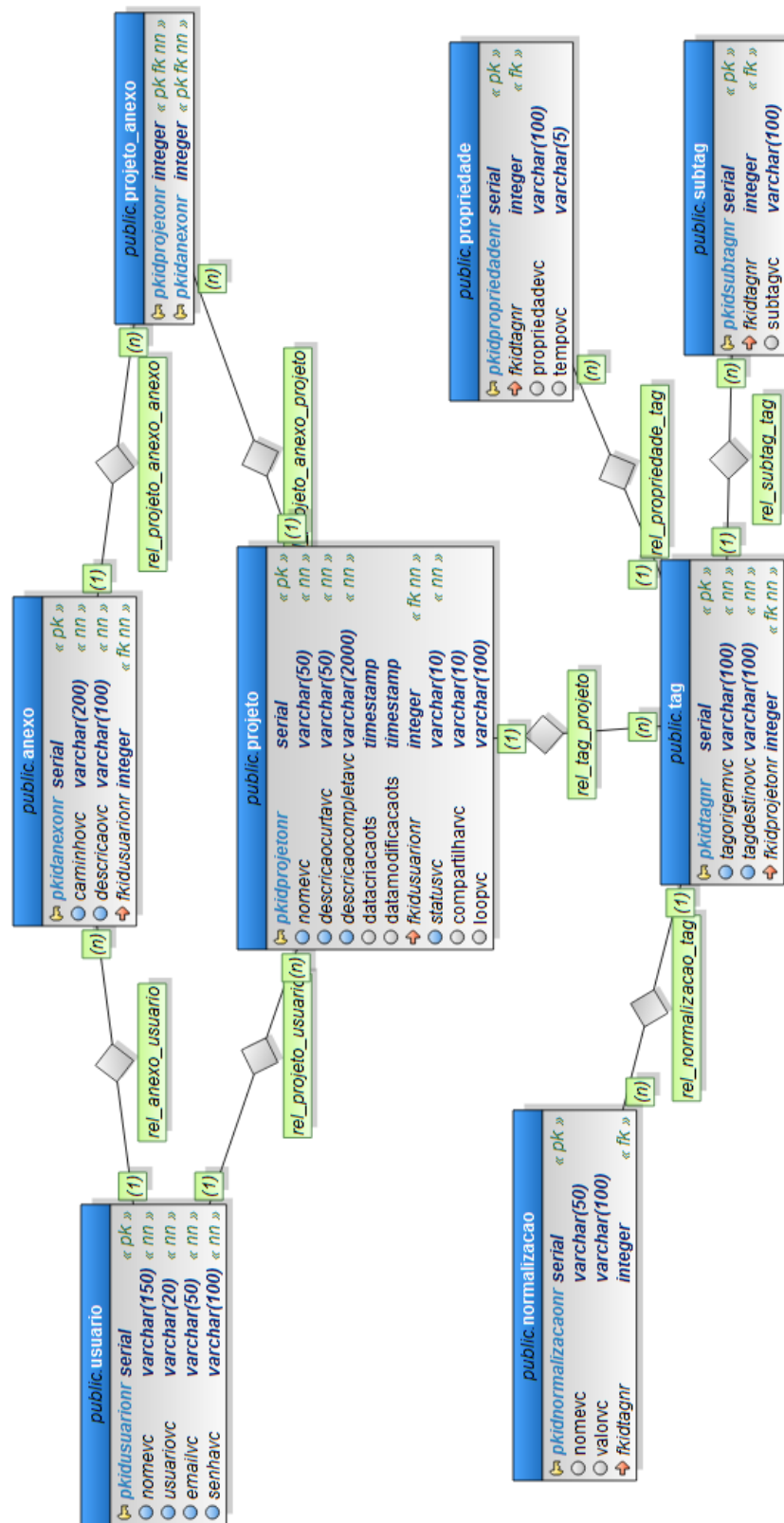


Figura 11: Diagrama entidade-relacionamento.

a descrição curta e completa do projeto, os anexos inclusos, um link onde é possível deixar o projeto público ou privado, além dos botões de editar, exportar para ARFF e exportar para CSV.

The figure consists of two screenshots of the FienoLog WAUTT application interface. The first screenshot, labeled '1', shows the login screen. It features a header 'FienoLog WAUTT' and a form with two input fields: 'Usuário:' and 'Senha:'. Below these fields is a blue 'Entrar' button. At the bottom of the form, there is a 'Cadastrar' button (labeled '2') and a link 'Esqueci a senha'. The second screenshot, labeled '3', shows the registration screen. It has a header 'FienoLog WAUTT' and a form titled 'Cadastrar'. The form contains five input fields: 'Nome:', 'Usuário:', 'Email:', 'Senha:', and 'Repetir a senha:'. Below these fields are two buttons: a blue 'Cadastrar' button and a white 'Cancelar' button.

Figura 12: Tela de login.

The figure shows a screenshot of the FienoLog WAUTT application's welcome screen. At the top, there is a blue header bar with 'Opções' (labeled '2') on the left and 'Sair' on the right. Below the header, the main content area has a light blue background. On the left side, there is a sidebar (labeled '1') containing a user profile icon and a list of menu items: 'Home', 'Meus Projetos', 'Projetos Públicos', 'Configurações', 'Ajuda', and 'Sobre'. The main content area (labeled '3') has a header 'FienoLog WAUTT' and a message '<-- Selecione as opções no menu lateral'. The rest of the main content area is currently empty.

Figura 13: Tela de boas vindas.

Ao criar ou editar um projeto de filtragem, o usuário é direcionado para a página mostrada na Figura 16. Essa página contém as informações do projeto na área 1, assim como as opções de salvar e editar as descrições do projeto. A área 2 é responsável pela adição de *tags* de filtragem que são mostrados na área 3. Assim como nas tabelas do diagrama de classes, nesta área são mostradas as propriedades (atributos) e as *subtags* (*tags* filhas com resultados), além de escolher um nome de exibição e poder escolher o processamento e filtro na seção de normalização.

Dependendo da normalização escolhida, novos campos aparecem, como exemplificado na Figura 17. Esses campos recebem as informações responsáveis pela filtragem e por diferentes



Figura 14: Tela de projetos de filtros privados.

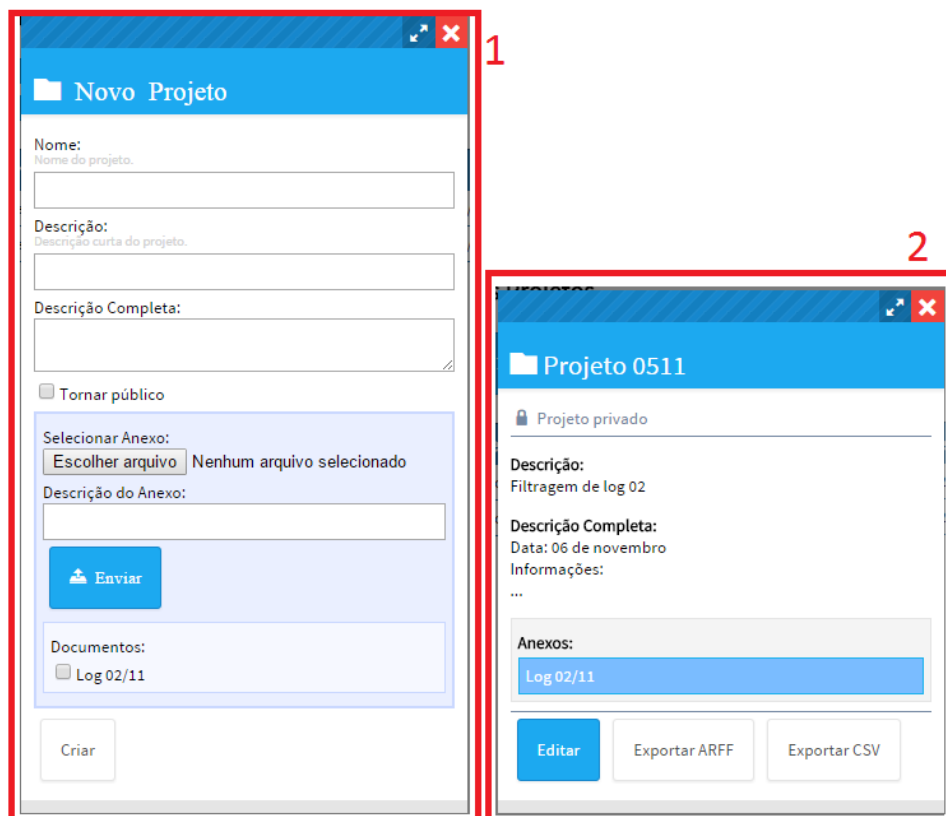


Figura 15: Criação e informações de um projeto.

tipos de processamento que não sejam apenas soma e que possuam algum dado que precisa ser indicado pelo usuário. Além dos exemplos mostrados na imagem, existem ainda as opções de “Somar”, “Somar a partir do 0”, “Somar a partir do 0 filtrando *subtag*”, “Somar a partir do 0 filtrando propriedade” e “Tempo”.

Ao selecionar Projetos Públicos no menu lateral, o usuário é direcionado para uma página semelhante à sua página de projetos, mostrando todos os projetos de filtragem, configu-

Projeto 0511

Salvar
 Editar
 Projeto privado
1

Descrição: Filtragem de log 02

Descrição Completa:
 Data: 06 de novembro
 Informações:
 ...

Anexos:

Log 02/11

▲▲▲

Novo:

Para cada:
 Sessão

2
3

.session.navigation.event

Nome para exportação:

Propriedades

id

time

timezone

examid

examname

taskid

taskname

Subtags

eventname

tagName

text

posx

posy

class

href

button

id

keyCode

style

autocomplete

shift

type

contenteditable

src

unselectable

Normalização

.session.navigation

Nome para exportação:

Propriedades

id

inputtime

lastupdate

timezone

Subtags

url

title

width

height

Normalização

Ocorrência:

Não ocorrência:

Figura 16: Projeto de filtragem.

The image displays three distinct panels, each representing a different preprocessing or filtering method. Each panel is enclosed in a light gray border and contains a title, a dropdown menu, and one or two input fields.

- Panel 1:** Titled "Normalização", it features a dropdown menu set to "Ocorrência". Below it, there are two input fields: "Ocorrência:" with the value "1" and "Não ocorrência:" with the value "0".
- Panel 2:** Also titled "Normalização", its dropdown menu is set to "Somar e dividir". It includes a single input field labeled "Dividir por:" containing the value "100".
- Panel 3:** Titled "Normalização", the dropdown menu is set to "Filtrar e somar". It has one input field labeled "Filtrar por:" which is currently empty.

Figura 17: Exemplos de tipos de préprocessamento e filtragem.

rados como públicos por outros usuários, como é exibido na área 1 da Figura 18. Na área 2 é mostrado o *dialog* com as informações do projeto, assim como nos projetos privados, mas as opções são diferentes: Visualizar, Duplicar, Exportar ARFF e Exportar CSV.

A Figura 19 mostra a página de visualização do projeto, que é igual à página de projetos criados pelo usuário mas com as opções diferentes, possuindo apenas a opção de Duplicar.

Projeto Públicos 1

Nome	Descrição	Usuário	Criado em
Projeto público 2	Descrição do projeto 2	Usuario3	09/11/2015
Projeto Público 01	Descrição do projeto publico	Usuario2	09/11/2015

2

Figura 18: Informações de projeto público.

Figura 19: Visualização de projeto público.

5 RESULTADOS E DISCUSSÕES

Para testar as funcionalidades do sistema, foi utilizado uma parte de um arquivo de *log* gerado pela WAUTT. Dessa maneira, foram separadas apenas duas sessões (*Sessions*), sendo que os arquivos de *log* da WAUTT possuem as seguintes *tags* principais: “*Session*”, “*Navigation*”, “*Browser*”, “*Os*” e “*Event*”.

Os resultados são obtidos para cada sessão do arquivo de *log*, pois é a raiz de todas as outras *tags*. Cada sessão possui uma *tag* “*Navigation*”, “*Browser*”, “*Os*” e várias *tags* “*Event*”. Dessa maneira, é possível verificar aproximadamente o que cada usuário fez, já que cada sessão pode ser considerada um uso por um usuário.

Um resultado compacto da geração de um arquivo ARFF pode ser verificado na Figura 20. Nela é possível notar que segue o padrão estabelecido pela ferramenta WEKA e que o primeiro atributo representa a soma dos valores, o segundo uma soma e divisão por 100 e o terceiro a soma de todos os “*events*” com resultado “*click*”. Esses resultados foram obtidos pelo uso de 3 funções do sistema: “Somar”, “Somar e dividir” e “Filtrar e somar”. Um resultado mais completo e que utiliza funções maiores é visto no Apêndice A.

```
% Documento gerado por FienoLog WAUTT

@relation Resultados

@attribute 'button' real
@attribute 'autocomplete' real
@attribute 'click' real

@data
131,0.86,48
149,0.53,56
```

Figura 20: Resultado ARFF.

Utilizando o mesmo projeto, foi gerado um arquivo CSV, mostrado na Figura 21. Esse

arquivo segue uma estrutura semelhante, mas sem separar os atributos dos dados. Dessa maneira, é simples criar gráficos e realizar qualquer função de programas que editam e visualizam os dados desse tipo de arquivo. Na Figura 22 é mostrado um exemplo de gráfico gerado no software Microsoft Excel com os dados da Figura 21.

Documento gerado por FienoLog WAUTT		
Resultados		
	Sessao 1	Sessao 2
button	131	149
autocomplete	0.86	0.53
click	48	56

Figura 21: Resultado CSV.

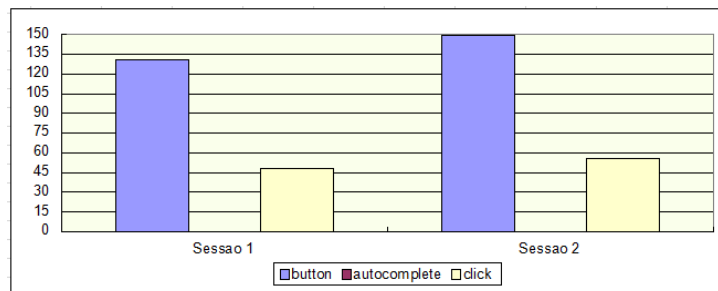


Figura 22: Gráfico com resultado gerado por CSV.

O Anexo A apresenta um resumo de um arquivo ARFF gerado manualmente para ser utilizado no WEKA. Pode-se verificar que os dados do Apêndice A e do Anexo A são semelhantes, ou seja, o sistema desenvolvido exportou os dados filtrando e realizando o pré-processamento dos dados da maneira esperada.

6 CONCLUSÃO

Para facilitar o uso dos dados sobre a interação do usuário capturados pela ferramenta WAUTT, um sistema foi desenvolvido para a criação de filtros para transformar o arquivo de log desta ferramenta em um arquivo de dados no formato de uma ferramenta utilizada para análise de log. O sistema também permite o armazenamento e compartilhamento de filtros para uso por diferentes usuários.

Algumas opções são oferecidas ao usuário em relação à maneira de filtrar os dados após escolher o método de normalização. Para cada *tag* escolhida do arquivo XML de *log* importado, é possível aplicar filtro e normalização. O resultado da filtragem e pré-processamento dos dados são arquivos ARFF para uso no WEKA e CSV para facilitar o uso dos dados em outras aplicações. Esses resultados se mostram válidos pela comparação com um arquivo ARFF gerado manualmente e utilizado em uma avaliação de usabilidade.

Trabalhos futuros podem melhorar a verificação dos dados que podem ser utilizados para cada tipo de normalização e filtragem. Outros formatos de exportação também podem ser disponibilizados e novos tipos de normalização adicionados de acordo com as necessidades dos usuários.

REFERÊNCIAS

- ATTERER, R.; WNUK, M.; SCHMIDT, A. Knowing the user's every move: user activity tracking for website usability evaluation and implicit interaction. 15th International Conference on World Wide Web (WWW 2006), Edinburgh, Escócia, p. 203–212, 2006.
- BALBO, S. et al. Leading web usability evaluations to waster. 11th Australian World Wide Web Conference (AusWeb), Gold Coast, Austrália, 2005.
- BOUCKAERT, R. R. et al. **WEKA Manual for Version 3-6-10**. [S.l.: s.n.], 2013.
- CHOO, C.; DETLOR, B.; TURNBULL, D. **Information Seeking on the Web: An Integrated Model of Browsing and Searching**. [S.l.]: First Monday, 2000.
- CYBIS, W.; BETIOL, A. H.; FAUST, R. **Ergonomia e Usabilidade: Conhecimentos, métodos e aplicações**. 2. ed. São Paulo: Novatec Editora, 2010.
- FRATERNALI, P.; ROSSI, G.; SÁNCHEZ-FIGUEROA, F. Rich internet application. IEEE Internet Computing, v. 14, n. 3, p. 9–12, 2010.
- GROUP, P. G. D. **PostgreSQL**. 2015. Disponível em: <<http://www.postgresql.org/about/>>. Acesso em: 05 de novembro de 2015.
- HONG, J. I. et al. Webquilt: A proxy-based approach to remote web usability testing. ACM Transactions on Information Systems, v. 19, n. 3, p. 263–285, 2001.
- IVORY, M. Y.; HEARST, M. A. The state of the art in automating usability evaluation of user interfaces. ACM Computing Surveys (CSUR), 2001.
- JANSEN, B. J. Search log analysis: What is it; what's been done; how to do it. Library and Information Science Research, 2006.
- JAVA. **Qual é a diferença entre o JavaScript e o Java?** 2015. Disponível em: <https://www.java.com/pt_BR/download/faq/java_javascript.xml>. Acesso em: 22 de abril de 2015.
- KELLAR, M.; WATTERS, C. Studying user behaviour on the web: Methods and challenges. workshop on usage analysis: Combining logging and qualitative methods. ACM Conference on Human Factors in Computing Systems (CHI 2005), Portland, OR, USA, 2005.
- MACK, R. L.; NIELSEN, J. Usability inspection methods. Conference Companion on Human Factors in Computing Systems, Nova York, USA, 2001.
- MICROSYSTEMS, S. **NetBeans**. 2015. Disponível em: <<https://netbeans.org/>>. Acesso em: 10 de outubro de 2015.
- PAGANELLI, L.; PATERNO, F. Intelligent analysis of user interactions with web applications. 7th International Conference on Intelligent User Interfaces, São Francisco, Estados Unidos da América, p. 111–118, 2002.

PREECE, J.; ROGERS, Y.; SHARP, H. **Interaction Design: Beyond Human-Computer Interaction**. 3. ed. Chichester: Wiley, 2012.

RIVOLLI, A.; PEREIRA, D. F.; PANSANATO, L. T. E. Capturing user interaction with rich internet applications for usability evaluation. 13th International Conference WWW/Internet, Porto, Portugal, 2014.

SANTANA, G. A. Uma abordagem para identificação automática de problemas de usabilidade em interfaces de sistemas web através de reconhecimento de padrões. Cornélio Procopio, 2013.

SANTANA, V. F.; BARANAUSKAS, M. C. C. Summarizing observational client-side data to reveal web usage patterns. 25th ACM Symposium on Applied Computing, Sierre, Suíça, 2010.

SCHWERZ, A. L. M.; MORANDINI, M.; SILVA, S. R. A task model proposal for web sites usability evaluation for the ergonomics environment. 12th International Conference on Human-computer Interaction, Beijing, China, p. 1188–1197, 2007.

SILVA, R. A. **pgModeler**. 2015. Disponível em: <<http://pgmodeler.com.br/>>. Acesso em: 10 de outubro de 2015.

STUDIOS, B. **Balsamiq**. 2015. Disponível em: <<https://balsamiq.com/products/mockups/>>. Acesso em: 05 de novembro de 2015.

SUBBULAKSHMI, C. V.; DEEPA, S. N.; MALATHI, N. Comparative analysis of xminer and weka for pattern classification. IEEE International Conference on Advanced Communication Control and Computing Technologies, Índia, 2012.

TEAM, T. pgAdmin D. **pgAdmin**. 2015. Disponível em: <<http://www.pgadmin.org/>>. Acesso em: 10 de outubro de 2015.

TIEDTKE, T.; MARTIN, C.; GERTH, N. Awusa a tool for automated website usability analysis. 9th International Workshop on the Design, Specification and Verification of Interactive Systems, Rostock, Alemanha, p. 251–266, 2002.

TURNBULL, D. Methodologies for understanding web use with logging in context. workshop on logging traces of web activity: The mechanics of data collection. International Conference on World Wide Web (WWW 2006), Edinburgh, Scotland, 2006.

VASCONCELOS, L.; JR, L. B. Usabilics: avaliação remota de usabilidade e métricas baseadas na análise de tarefas. 10th Brazilian Symposium on Human Factors in Computing Systems and 5th Latin American Conference on Human-Computer Interaction, Porto de Galinhas, Brasil, p. 303–312, 2011.

W3SCHOOLS. **CSS Introduction**. 2015. Disponível em: <http://www.w3schools.com/css/css_intro.asp>. Acesso em: 22 de abril de 2015.

W3SCHOOLS. **HTML Introduction**. 2015. Disponível em: <http://www.w3schools.com/html/html_intro.asp>. Acesso em: 22 de abril de 2015.

W3SCHOOLS. **Introduction to SQL**. 2015. Disponível em: <http://www.w3schools.com/sql/sql_intro.asp>. Acesso em: 22 de abril de 2015.

W3SCHOOLS. **Introduction to XML.** 2015. Disponível em:
<http://www.w3schools.com/xml/xml_what_is.asp>. Acesso em: 22 de abril de 2015.

W3SCHOOLS. **PHP 5 Introduction.** 2015. Disponível em:
<http://www.w3schools.com/php/php_intro.asp>. Acesso em: 22 de abril de 2015.

