

TRACK4WEB: UMA PLATAFORMA INTELIGENTE DE COLETA E ANÁLISE DE DADOS E INTERAÇÕES DE USUÁRIOS NA WEB

Vítor Hugo de Paula Carvalho¹
Ahmed Ali Abdalla Esmin²
Departamento de Ciência da Computação
Universidade Federal de Lavras
{ ¹vitorhugo@comp.ufla.br, ²ahmed@ufla.br }

Resumo

Com o crescimento desenfreado da utilização da internet e surgimento de tecnologias e novos paradigmas, surge a necessidade de controlar de uma forma mais apurada o desempenho e utilização destes *web sites* e serviços. Com este intuito, o presente trabalho propõe o desenvolvimento de uma plataforma que operará sobre qualquer *web site*, realizando a coleta dos dados e interações realizadas pelos usuários. Uma vez coletadas tais informações, elas serão disponibilizadas em um ambiente de consulta, criado para facilitar a análise e avaliação do *web site*, além de auxiliar os seus administradores na tomada de decisões estratégicas.

Palavras-Chave: Coleta de Dados; Análise de Dados; Coleta de Interações; *Web*.

TRACK4WEB: AN INTELLIGENT PLATFORM FOR DATA AND WEB USERS' INTERACTIONS COLLECTION AND ANALYSIS

Abstract

With the growth of the use of the Internet and emerging technologies and new paradigms, it is necessary to control in a more accurate way the performance and use of web sites and services. Within this context, this work proposes the development of a platform that will operate on any web site, collecting data and users' interactions. Once collected such information, there will be available an environment where all data are shown, making it easy the analysis and evaluation of the web site by its administrators as well as being the basis for strategic decisions.

Keywords: Data Collection; Data Analysis; Interaction's Collection; *Web*.

1. INTRODUÇÃO

A utilização da Internet nos dias atuais tem crescido vertiginosamente. Segundo[1], mais da metade de toda população brasileira informou já ter tido acesso a um computador, sendo que, um total de 34% da população brasileira tem acesso à internet. Conclui-se que cada vez mais, a internet se afirma como um grande meio de comunicação, capaz de atingir massas, alterar hábitos e mudar conceitos e atitudes. As empresas têm percebido essa mudança e entendem-na como uma oportunidade de expor seus produtos e serviços com baixo custo.

Levando em base estas informações, faz-se necessário estudar formas de acompanhar a evolução e utilização de serviços e interesse por produtos na *web*. Através da coleta das formas de interação dos usuários com o serviço ou *site*, é possível entender e aplicar melhorias com o intuito de satisfazer melhor as necessidades dos usuários. Neste contexto, entende-se que interação é um termo geral utilizado para classificar eventos específicos emitidos por usuários em qualquer aplicação ou *site* da *web*, sob o ponto de vista da plataforma *Track4Web*.

De acordo com [2], existem várias pesquisas na área de interações de usuários com a *web*, com ênfase em comércio eletrônico.

Existem ferramentas de monitoramento de *web sites* disponíveis para serem utilizadas, porém tais ferramentas apenas disponibilizam informações estatísticas, não possibilitando com isso que possam ser feitas maiores inferências a partir de estudos e mineração de dados sobre os dados coletados. A base de dados de tais serviços não está acessível.

Para aplicar algoritmos de inteligência de negócios e mineração de dados sobre as informações coletadas, é necessário que os dados estejam disponíveis e possam ser exportados quando necessário.

2. WORLD WIDE WEB

A *World Wide Web* (“Rede de Alcance Mundial”) é um sistema de documentos em hipermídia que são interligados e executados na Internet.

As idéias por trás da *Web* podem ser identificadas ainda em 1980, na sede da CERN¹, na Suíça, quando Tim Berners-Lee construiu o Enquire².

Em [3], o autor escreve uma proposta de gerenciamento de informação, que referenciava o Enquire e descrevia um sistema de informação mais elaborado. Este sistema basear-se-ia em sistemas de hipertexto distribuído para solucionar o problema de perda de informação de sistemas complexos. Já em [4], os autores propuseram o projeto da *World Wide Web* em um âmbito mais formal, no qual seriam criados os *softwares* necessários, como o servidor de dados e o navegador.

O conceito crucial do hipertexto originou-se em projetos da década de 1960, como o projeto *Xanadu* e o NLS. A idéia revolucionária de Lee seria a união do hipertexto e a Internet. É explicado em [5] que os autores sugeriram repetidamente o casamento das tecnologias para membros de ambas as comunidades de desenvolvedores.

Sistemas anteriores diferenciavam-se da *Web* em alguns aspectos. Na *Web* um *hyperlink* é unidirecional enquanto trabalhos anteriores somente tratavam ligações bidirecionais. Isso tornou possível criar um *hyperlink* sem qualquer ação do autor do documento sendo ligado, reduzindo significativamente a dificuldade em implementar um servidor *Web* e um navegador. Por outro lado, o sistema unidirecional é responsável por o que atualmente chama-se *hyperlink* quebrado, isto é, um *hyperlink* que aponta para uma

página não disponível devido à evolução contínua dos recursos da Internet com o tempo.

Diferente de sistemas anteriores como o *HyperCard*, a *World Wide Web* não era *software* proprietário, tornando possível a criação de outros sistemas e extensões sem a preocupação de licenciamento. Em 30 de abril de 1993, a CERN anunciou que a *World Wide Web* seria livre para todos, sem custo.

O processo de visualizar uma página *web* ou outro recurso disponibilizado normalmente se inicia ao digitar uma URL no navegador ou seguindo (acessando) um *hyperlink*. Primeiramente, a parte da URL referente ao servidor *web* é separada e transformada em um endereço IP, por um banco de dados da Internet chamado *Domain name system* (DNS). O navegador estabelece então uma conexão TCP-IP com o servidor *web* localizado no endereço IP retornado.

O próximo passo é o navegador enviar uma requisição HTTP ao servidor para obter o recurso indicado pela parte restante da URL (retirando-se a parte do servidor). No caso de uma página *web* típica, o texto HTML é recebido e interpretado pelo navegador, que realiza então requisições adicionais para figuras, arquivos de formatação, arquivos de script e outros recursos que fazem parte da página.

Os Navegadores *Web*, ou *Web Browsers* se comunicam geralmente com servidores *Web* (podendo hoje em dia se comunicar com vários tipos de servidor), usando principalmente o protocolo de transferência de hiper-texto HTTP para efetuar pedidos de arquivos, e processar respostas vindas do servidor. Estes arquivos são por sua vez identificados por uma URL. O navegador tem a capacidade de ler vários tipos de arquivo, sendo nativo o processamento dos mais comuns (HTML, XML, JPEG, GIF, PNG, etc.), e os restantes possíveis através de *plugins* (*Flash*, *Java*, etc.).

3. COLETA DE DADOS NA WEB E TECNOLOGIAS UTILIZADAS

3.1. Análise de Log

De acordo com [6], existem duas formas de coletar dados e informações a respeito da utilização de um serviço ou *site* da *web*. Uma abordagem para o descobrimento de informações sobre os usuários de um *web site* seria a análise dos arquivos de *log* do servidor *Web*. Estes arquivos contêm um histórico completo dos arquivos acessados pelos clientes.

¹ Organização Européia para Investigação Nuclear

² Projeto utilizado para reconhecer e armazenar associações de informações

Como resultado das interações entre visitantes e um *web site*, Em [7], os autores dizem que um arquivo de *log* contém muita informação sobre os comportamento dos usuários dentro do *site*, que, se totalmente explorada, pode melhorar os serviços para o cliente e o desempenho do *web site*.

Há muitas ferramentas de análise de *log* disponíveis na Internet. De acordo com [7], o desempenho destas ferramentas cai rapidamente quando o tamanho do conjunto de dados aumenta. Além disso, grande parte das pessoas que não têm seus *web sites* hospedados em servidores dedicados, não têm permissão para acessar os arquivos de *log* do servidor *web*, uma vez que estes arquivos também contêm informações estratégicas sobre todos outros os *web sites* hospedados naquele mesmo servidor.

3.2. Registro de Acessos

Outra abordagem que permite a coleta de dados na *web* baseia-se na metodologia de coleta dos dados no cliente, capturando informações por meio dos objetos criados pelo navegador do usuário ao visitar um *web site*. No navegador do usuário ao ser acessada a página que implemente a coleta, é realizada uma requisição para o servidor que recebe, interpreta e trata os dados.

Neste contexto, a solução de registro de acesso se faz mais portátil e permite que o esforço de implantação da mesma seja reduzido, uma vez que apenas o *web site* deve ser editado de forma a inserir o código que realize a comunicação com o servidor.

3.3. Tecnologias

Abaixo, têm-se algumas opções que podem ser utilizadas para desenvolver uma arquitetura de coleta, tanto no lado do cliente, quanto no lado do servidor.

- **Cliente**

- **Java Script**

Será responsável por tratar os eventos no cliente, além de coletar e enviar informações por meio de requisições para o servidor. De acordo com [8], esta tecnologia é a única capaz de oferecer as possibilidades necessárias para a coleta efetiva dos dados e interações do usuário.

- **Servidor**

- **PHP, ASP, ASP.Net, JSP e CGI**

Será responsável por receber e tratar as requisições realizadas pelo cliente de acordo com os eventos específicos.

- **MySQL, PostgreSQL, SQL Server e INTERBASE**

Base de dados responsável por salvar os dados e informações coletadas.

- **JAVA, C/C++**

Será responsável por tratar os eventos salvos no *log* do servidor *web*.

Com base nas tecnologias expostas acima, e levando em consideração os custos agregados de cada uma, mostra-se nas próximas subseções características das linguagens utilizadas para realizar o desenvolvimento deste trabalho.

3.3.1. Java Script

Em [8] é dito que *Java Script* é uma linguagem de programação criada pela *Netscape* em 1995, que a princípio se chamava *LiveScript*, para atender, principalmente, as seguintes necessidades:

- Validação de formulários no lado cliente (programa navegador);
- Interação com a página.

Assim, foi feita como uma linguagem de script. *Java Script* tem sintaxe semelhante à do *Java*, mas é totalmente diferente no conceito e no uso.

1. Oferece tipagem dinâmica - tipos de variáveis não são definidos;
2. É interpretada, ao invés de compilada;
3. Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral);
4. Oferece bom suporte a expressões regulares (característica também comum a linguagens de script).

3.3.2. PHP

PHP (um acrônimo recursivo para "PHP: *Hypertext Preprocessor*") é uma linguagem de programação de computadores interpretada, livre e, de acordo com [9], é muito utilizada para gerar conteúdo dinâmico na *World Wide Web*, sendo uma linguagem orientada a objetos.

A linguagem PHP é uma linguagem de programação de domínio específico, ou seja, seu escopo se estende a um campo de atuação que é o desenvolvimento *web*, embora tenha variantes como o PHP-GTK. Seu propósito principal é de implementar soluções *web* velozes, simples e eficientes.

Algumas características da linguagem:

- Velocidade e robustez
- Estruturado e orientação a objeto
- Portabilidade - independência de plataforma - escreva uma vez, rode em qualquer lugar;

3.3.3. MySQL

O *MySQL* é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* - Linguagem de Consulta Estruturada) como interface. Em [10], seus autores afirmam que, atualmente, o *MySQL* é um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

Algumas características:

- Portabilidade (suporta praticamente qualquer plataforma atual)
- Compatibilidade (existem drivers ODBC, JDBC e .NET e módulos de interface para diversas linguagens de programação, como *Delphi*, *Java*, *C/C++*, *Python*, *Perl*, *PHP*, *ASP* e *Ruby*)
- Excelente desempenho e estabilidade;
- Pouco exigente quanto a recursos de hardware;
- Facilidade de uso;
- É um *Software Livre*;

3.3.4. Cookies

Um *cookie* é um grupo de dados trocados entre o navegador e o servidor de páginas, colocado num arquivo de texto criado no computador do usuário. A sua função principal é a de manter a persistência de sessões HTTP.

4. ARQUITETURA MVC (MODEL, VIEW, CONTROLLER)

A arquitetura MVC é um padrão de desenvolvimento de *software*, cujo principal preceito é a separação dos dados (*Model*), da lógica de negócio (*Controller*) e da visualização (*View*).

4.1. Descrição do Padrão

É comum dividir a aplicação em camadas separadas: a apresentação (interface), domínio e acesso a dados. Em MVC a camada de apresentação também é separada do *View* e do *Controller*.

Os autores de [11] definem as características da arquitetura MVC e a divide nas seguintes camadas:

• **Model**

Representação do domínio específico da informação em que a aplicação opera. A camada de acesso a dados (banco de dados) está encapsulada na camada *Model*.

• **Controller**

Processa e responde eventos, geralmente ações do usuário, e pode invocar alterações no *Model*.

• **View**

Renderiza os dados e informações, normalmente em forma de interface de usuário.

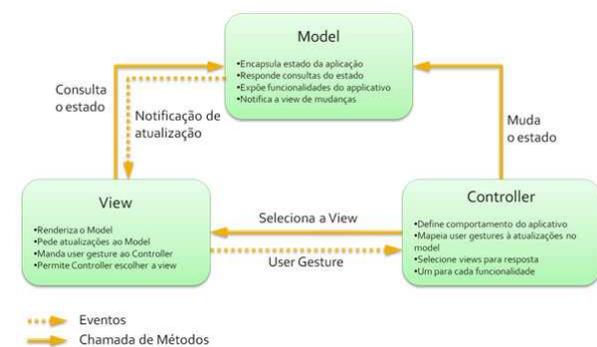


FIGURA 4.1 - ARQUITETURA MVC

A Figura 4.1 demonstra melhor a organização da arquitetura, demonstrando o papel de cada camada, bem como a relação existente realizada por uma sobre a outra.

De uma forma mais prática, voltada para o ambiente *Web*, pode-se interpretar o padrão MVC da seguinte forma:

- **View**: Páginas com resposta HTML;
- **Controller**: Regras de negócio, processamento de informações e dados;
- **Model**: representação dos dados e persistência em base de dados.

5. PESQUISA, MATERIAIS E MÉTODOS

Neste capítulo será apresentado o tipo de pesquisa feita neste trabalho e forma em que foi feita a construção e desenvolvimento da plataforma *Track4Web*.

5.1. Natureza da Pesquisa

O presente trabalho tem como base desenvolver uma plataforma capaz de ser utilizada em qualquer *web site* na Internet, e podendo a partir daí, monitorar a utilização e interações do usuário com o *web site*. Portanto esta pesquisa é do tipo tecnológica, uma vez que se utiliza de técnicas existentes em engenharia de *software* para o desenvolvimento desta plataforma que será responsável pela coleta de dados e interações de usuários e uma Interface Gráfica de análise dos dados armazenados proporcionando assim um ambiente de tomada de decisão integrado.

Segundo [12] e observando o método científico tem-se que este projeto é de natureza tecnológica, com objetivos de caráter exploratório, utilizando procedimentos experimentais. A finalidade deste trabalho é a produção de um software e como [13], é a partir de experimentações exploratórias que se produzem inovações tecnológicas.

5.2. Materiais

A plataforma de desenvolvimento da *Track4Web* foi um notebook dotado de tecnologia *Intel® Centrino® Duo*, com processador *Intel® Core™ 2 Duo T7300* com 2 GHz, 2 GB de Memória RAM DDR2 e adaptador gráfico *Intel® Graphic Media Accelerator X3100*.

O servidor hospedeiro da plataforma é um servidor virtual em uma máquina *Dell PowerEdge 2900*.

As ferramentas necessárias para o funcionamento da plataforma são:

- Servidor *Web* (preferencialmente *Apache*)
- PHP
- *MySQL*

A principal ferramenta de desenvolvimento foi o *Zend Development Environment (Integrated Development Environment) Zend Studio 5.5 Trial Version*, que acopla em si um ótimo *debugger* PHP. A ferramenta foi utilizada principalmente para o desenvolvimento do código das camadas de negócio. Também foi utilizado o *Adobe Dreamweaver CS3 Trial Version* como ferramenta gráfica de desenvolvimento com um maior apelo na parte gráfica, sendo utilizado no desenvolvimento da Interface de consulta.

Os requisitos funcionais necessários para que a plataforma colete os dados de forma eficaz se resume à necessidade de habilitar-se o *JavaScript* no

navegador dos usuários, o que normalmente já é habilitado por padrão.

Para acesso à Interface de consulta, recomenda-se resolução igual ou superior à 1024x768, plugin *Adobe Flash Player* e navegador recente.

5.3. Métodos

Os métodos adotados para o cumprimento do objetivo deste trabalho estão dispostos nas seguintes subseções desta seção.

5.3.1. Estudo da *World Wide Web*

Nesta etapa, realizou-se um estudo sobre a *World Wide Web*, analisando a forma com que as informações se mostram organizadas. Também em paralelo foi feito um estudo da organização interna de um *site*, os elementos gráficos e eventos.

5.3.2. Estudo da Coleta de Dados na *Web*

Uma vez mapeada a forma de organização de um *site*, fez-se possível estudar formas de coletas dados a partir destes. Foram identificadas as maneiras possíveis e definida a forma com que se daria a coleta.

5.3.3. Estudo e Definição das Tecnologias

Nesta etapa, foi realizado um estudo e definiram-se as tecnologias que seriam necessárias para desenvolver a plataforma. É importante salientar que o principal ponto forte das tecnologias escolhidas para este trabalho é o custo, uma vez que são tecnologias abertas de utilização livre, sem necessidade de licenças ou compra de *softwares*.

5.3.4. Definição da Plataforma

Tendo todos os estudos sido realizados, parte-se para a definição e planejamento da plataforma. Talvez a etapa mais importante visto que a plataforma deve oferecer desempenho aliado à confiabilidade.

5.3.5. Desenvolvimento da Plataforma

O desenvolvimento da plataforma iniciou-se após uma rigorosa pesquisa bibliográfica e que pudesse oferecer conceitos e informações suficientes para identificar pontos a serem focalizados quando da definição dos dados a serem coletados e formas de disponibilizá-los posteriormente.

A fase de desenvolvimento foi dividida em duas etapas que compõe os mecanismos da plataforma:

- Mecanismo de Coleta de Dados e Interações de Usuários
- Mecanismo de Análise e Consulta de Informações

O desenvolvimento da plataforma como um todo ocorreu seguindo o padrão MVC de desenvolvimento, descrito no capítulo anterior.

5.3.6. Aplicação e Teste da Plataforma

Após o desenvolvimento da plataforma, esta então será submetida a teste, para verificar se realmente alcançou-se o objetivo esperado. Para teste, a plataforma será disponibilizada para coletar dados de usuários de *sites* que já estão disponíveis na *web*.

6. RESULTADOS E DISCUSSÃO

Este capítulo tem por objetivo apresentar de forma sucinta, porém coesa a plataforma obtida como proposto por este trabalho. As duas seções deste capítulo mostram cada mecanismo que compõe a plataforma.

6.1. Plataforma *Track4Web*

Baseando-se no exposto no capítulo 2, a plataforma *Track4Web* foi desenvolvida como sendo uma ferramenta de auxílio para administradores de *web sites* na tomada de decisão e acompanhamento online do desempenho de seu *web site* ou serviço *web*.

Conforma exposto na terceira seção do segundo capítulo, a plataforma foi desenvolvida seguindo o padrão MVC de desenvolvimento, garantindo uma maior independência dos dados e da classificação e processamento.

6.2. Validação da Plataforma

Para verificar se a plataforma atende às expectativas, a mesma foi disponibilizada em quatro *web sites*. Todos eles estão hospedados no Núcleo Tecnológico Educacional. Os *web sites* foram:

- **Administração de Sistemas de Informação**
Disponível em <http://www.nte.ufla.br/asi>
- **Engenharia de Software com Ênfase em Software Livre**
Disponível em <http://www.nte.ufla.br/esl>
- **Informática em Educação**
Disponível em <http://www.nte.ufla.br/ied>
- **Tecnologia de Redes de Computadores**
Disponível em <http://www.nte.ufla.br/rde>

6.3. Mecanismo de Coleta

O Mecanismo de Coleta é o componente mais importante da plataforma, sendo o responsável por interpretar interações do usuário com qualquer *web site* e disparar eventos, que serão interpretados pela camada de controle que reside no servidor.

Com base na organização da plataforma MVC, é possível verificar que não existe exatamente uma interface no Mecanismo de Coleta, porém existem dois controladores, um reside no cliente e outro no servidor.

Pode-se então exibir uma representação das camadas do Mecanismo de Coleta na Figura 6.1.



FIGURA 6.1 - ARQUITETURA MVC NO MECANISMO DE COLETA

Existem dois eventos que disparam ações no servidor, ambos estão descritos abaixo com os respectivos dados transferidos.

6.3.1. Eventos de *Log*

Eventos de *Log* são eventos que ocorrem todas as vezes que um usuário carrega uma página da *web* que utilize a plataforma *Track4Web* e conseqüentemente utiliza o Mecanismo de Consulta.

Os dados que trafegam do cliente quando da ocorrência de um evento de *Log* são exibidos abaixo na Tabela 6.1.

TABELA 6.1 - DADOS DOS EVENTOS DE *LOG*

Dado	Descrição
Código do Site	Código do <i>Site</i> que informa à qual <i>site</i> aquele <i>Log</i> pertence. Todos os <i>sites</i> estarão previamente cadastrados na plataforma.
Referer	Página de origem, ou seja, a página que o usuário estava antes de acessar a página que gerou o evento de <i>log</i> .
URL	O endereço do navegador do usuário utilizado para acessar a página geradora do evento de <i>log</i> .
Título	O título da página que acaba de ser carregada.

Resolução de Tela	Informa a resolução de tela do cliente.
Flash	Informa se o usuário possui ou não o <i>plugin</i> do <i>Flash</i> instalado.
Java	Informa se o usuário possui ou não o <i>plugin</i> <i>Java</i> instalado.
PDF	Informa se o usuário possui ou não o <i>plugin</i> PDF instalado.
QuickTime	Informa se o usuário possui ou não o <i>plugin</i> do <i>QuickTime</i> instalado.
RealPlayer	Informa se o usuário possui ou não o <i>plugin</i> do <i>RealPlayer</i> instalado.
Windows Media Player	Informa se o usuário possui ou não o <i>plugin</i> do <i>Windows Media Player</i> instalado.

Após ser instanciado um objeto da classe *LogBusiness*, passa-se a processar as informações recebidas em conjunto com as informações que são coletadas pela própria classe de negócios que são:

- **IP do usuário**
- **UserAgent do navegador do usuário**

De posse das informações acima mencionadas, é possível através de métodos implementados na classe *LogBusiness* identificar algumas informações importantes que são as seguintes:

- **Navegador do usuário**
- **Sistema Operacional do usuário**
- **PDA do usuário (se aplicável)**
- **País de origem do acesso**
- **Estado de origem do acesso**
- **Cidade de origem do acesso**

Além das informações acima, também é necessário preocupar-se com a segurança e persistência das informações. Partindo desta necessidade, os seguintes dados também são obtidos e salvos para identificar a sessão e o cliente que acessa aquela página.

- **Hash de Configuração do computador do usuário**

Armazena informações referentes à configuração do computador do usuário, e permite então identificar o mesmo usuário caso ele perca o *cookie* identificador de usuário ou o *cookie* de identificação da sessão.

- **Hash de Usuário**

Armazena uma string de *hash* com informações únicas do usuário. Este *hash* é salvo em um *cookie* na máquina do cliente, e expira em 30 dias.

- **Hash de Sessão**

Armazena uma string de *hash* com informações únicas da sessão do usuário, este *hash* também é salvo em um *cookie* na máquina do cliente, e expira em 5 minutos.

De forma análoga para o *hash* de sessão, porém com um tempo de expiração menor, se já existir um *hash* de usuário nos últimos 5 minutos, atribui-se o *hash* de sessão como sendo o que já existe no banco de dados.

Para realizar a busca de País, Estado e Cidade de origem do acesso, utilizou-se uma biblioteca gratuita denominada GeoIP³, que, a partir do IP do usuário faz os devidos cálculos de posicionamento.

Para identificar o navegador, sistema operacional e caso aplicável o PDA do usuário, utilizou-se um algoritmo de identificação do *UserAgent* particular, desenvolvido pelo autor do trabalho que será detalhado nos apêndices deste.

6.3.2. Eventos de Actions

Eventos de *Actions* são eventos que ocorrem todas as vezes que um usuário dá um clique em algum elemento importante do *site*. Considera-se um elemento importante como sendo imagens, botões, links, etc.

Os dados que trafegam do cliente quando da ocorrência de um evento de *Action* são exibidos abaixo na Tabela 6.2.

TABELA 6.2 - DADOS DOS EVENTOS DE ACTIONS

Dado	Descrição
URL	O endereço do navegador do usuário utilizado para acessar a página geradora do evento de log.
Link	Qual o objeto sendo clicado, com uma referência a este.
Destino	Se for um link, informa qual o destino do mesmo.
X	Posição do elemento clicado no eixo X.

³ Disponível em <http://www.maxmind.com>

Y Posição do elemento clicado no eixo Y.

Além das informações acima, uma vez instanciado um objeto da classe *ActionBusiness*, o mesmo é associado a um objeto de *Log (LogBusiness)*. Com isto, garante-se que existe um rastreamento completo das ações e páginas carregadas pelos usuários dentro de um *web site*.

A Figura 6.2 apresenta um diagrama do funcionamento do Mecanismo de Coleta.

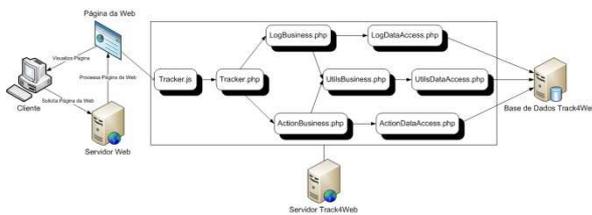


FIGURA 6.2 - DIAGRAMA DE FUNCIONAMENTO DO MECANISMO DE COLETA

6.3.3. Utilização do Mecanismo de Coleta

Para implementar o mecanismo de coleta em um *web site*, deve-se simplesmente inserir o seguinte código no código HTML da página.

```

<!-- Script Track4Web - Site: ESL -->
<script language="javascript"
type="text/javascript" src="http://tm-
licesa.dcc.ufla.br/~vitor/Track4Web/Tracker
.js"></script>
<script language="javascript"
type="text/javascript">
var TrackId = 3;
CollectData();
</script>

```

A variável *TrackId* deve ser ajustada com o código de cadastro do *web site* no banco de dados.

A chamada da função *CollectData()* realiza as configurações necessárias e inicia a coleta de dados e interações do usuário. A partir daí, todas as interações do usuário com o *web site* são coletadas e armazenadas.

6.4. Mecanismo de Análise

O Mecanismo de Análise é responsável por resgatar os dados da base de dados e disponibilizá-los de forma a permitir a análise das informações geradas com foco na tomada de decisão por parte do administrador do *web site*.

Neste mecanismo não existem apenas as camadas de negócio, como acontecia no Mecanismo de Coleta, também foi criado um *dashboard*⁴ interativo para facilitar a visualização dos dados por parte do administrador.

Como visto na seção 6.2 deste capítulo, a plataforma foi disponibilizada e dados foram coletados de quatro *web sites*. O Mecanismo de Análise, que opera sobre os dados coletados, encontra-se disponível para consulta em

<http://tm-licesa.dcc.ufla.br/~vitor/Track4WebGUI/>.

A Figura 6.3 exibe o diagrama do Mecanismo de Análise desenvolvido para a plataforma.

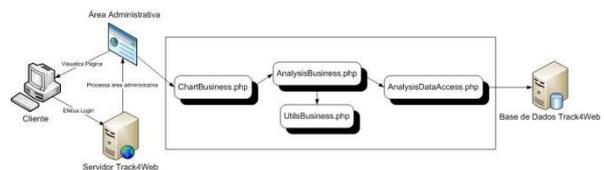


FIGURA 6.3 - DIAGRAMA DO MECANISMO DE ANÁLISE

Como pode ser visto na Figura 6.3, o Mecanismo de Análise possui os seguintes componentes:

- **ChartBusiness.php**

Classe de negócio responsável por tratar os dados e disponibilizá-los para interface de forma a exibir gráficos, tabelas e outros tipos de visualização de informação. Esta classe está englobada pela camada *Controller* em uma arquitetura MVC.

- **AnalysisBusiness.php**

Classe de negócio que faz o tratamento dos dados obtidos do Banco de Dados. Esta classe está englobada pela camada *Controller* em uma arquitetura MVC.

- **UtilsBusiness.php**

Classe de negócio que oferece algumas funcionalidades úteis ao sistema. Esta classe está englobada pela camada *Controller* em uma arquitetura MVC.

- **AnalysisDataAccess.php**

Classe de acesso a dados que fornece métodos de persistência com o banco de dados. Esta classe fica sendo a representação da camada *Model* em um arquitetura MVC.

⁴ Conjunto de indicadores para suporte analítico utilizado como apoio à tomada de decisão.

6.4.1. Dashboard Interativo

Para satisfazer a necessidade de exibir os dados de uma forma interativa e fácil, foi desenvolvido um *Dashboard*, que oferece funcionalidades para tratar as informações e disponibilizá-las visando auxiliar e facilitar o processo de tomada de decisão.

A Figura 6.4 mostra a Interface deste *Dashboard*.

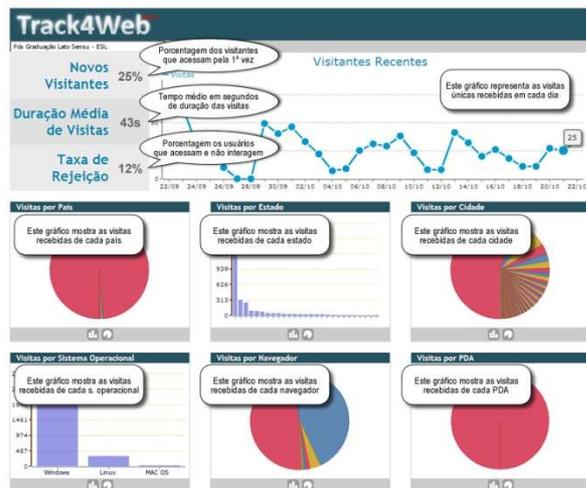


FIGURA 6.4 - DASHBOARD INTERATIVO DO MECANISMO DE ANÁLISE

Alguns *widjets*⁵ foram criados para facilitar a interpretação das informações, podem-se citar alguns dos mais importantes:

- **Visitantes Recentes**

Oferece uma visão geral da utilização do *web site*, disponibilizando o número de acessos únicos nos últimos dias.

- **Horário das Visitas**

Pode-se com este relatório acompanhar em qual horário o servidor é mais acessado.

- **Novos Visitantes**

Este relatório dá a precisa noção do número e porcentagem do total de usuários que são novos no *web site*, permitindo com isso rastrear a aceitabilidade e projeção do *web site* no contexto desejado.

- **Taxa de Rejeição**

Esta importante informação permite rastrear e identificar a aceitabilidade do *web site* perante os usuários, basicamente identifica-se como uma rejeição

um usuário que entra no *site*, porém não prossegue no mesmo, visualizando informações e visitando páginas posteriores. Esta informação foi deduzida de forma empírica e a sua utilização será avaliada.

- **Tempo de Visitas**

Esta informação também permite avaliar a aderência do conteúdo do *web site*, caso o usuário permaneça mais tempo no *web site*, isto provavelmente significa que o mesmo gostou do conteúdo e está interagindo com o mesmo.

- **Hits por Item**

Como pode ser visto no dashboard, existem vários *widjets* que mostram a porcentagem de *Hits* por determinadas características como Navegador, Sistema Operacional, etc.

- **Páginas de Origem e Saída**

Esta informação é interessante, pois mostra de qual origem partem as visitas para o *web site* em questão, se a maioria das visitas vem de algum patrocinador, mecanismo de *marketing*, etc. Também é possível saber para onde os usuários partem depois de visitar o *web site*, podendo com isso inferir informações importantes para áreas de *marketing* e comercial.

- **Mecanismo de Busca e Palavras Chave**

Esta informação ajuda na divulgação do *web site* por meio de mecanismos de busca, exibindo de quais mecanismos partem a maioria das visitas e também as palavras chaves utilizadas nos mesmos para alcançar o *web site*.

7. CONCLUSÕES E TRABALHOS FUTUROS

Neste capítulo serão apresentadas as conclusões finais do desenvolvimento da plataforma e também idéias para trabalhos futuros.

7.1. Considerações Finais

O objetivo deste trabalho foi desenvolver uma versão inicial da plataforma assim como a verificação das funcionalidades mediante aplicação do mesmo em alguns *web sites*. Dessa forma, pode-se observar que o objetivo principal foi alcançado com sucesso, uma vez que a plataforma está disponibilizada operando sobre os *web sites* dos cursos de pós-graduação *Lato sensu* hospedados no servidor do Núcleo Tecnológico Educacional (NTE).

Outro objetivo era gerar uma plataforma que fosse fracamente acoplada, ou seja, que fosse possível

⁵ Componente de interface gráfica com o usuário.

ser disponibilizada facilmente em qualquer ambiente, o que também foi realizado.

7.2. Trabalhos Futuros

Como trabalhos futuros, pode-se citar a validação da plataforma no ambiente de Educação à Distância (EaD), o que ocorrerá com vistas ao desenvolvimento de uma nova ferramenta de apoio à gestão e tomada de decisão em EaD. Esta nova ferramenta se beneficiará da plataforma desenvolvida uma vez que seus componentes serão conectados a esta última, formando assim uma nova plataforma, que deverá ser disponibilizada em breve.

Outra possibilidade consiste em aplicar a plataforma e implementar novas funcionalidades voltadas ao comércio eletrônico, também agindo como plataforma de tomada de decisão sobre a área de e-commerce.

8. REFERENCIAL BIBLIOGRÁFICO

- [1] SANTOS, R. S. Pela primeira vez mais da metade da população já teve acesso ao computador. 2007. In: Pesquisa sobre o uso das Tecnologias da Informação e da Comunicação no Brasil: TIC Domicílios e TIC Empresas 2007. São Paulo : Comitê Gestor da Internet no Brasil, 2008.
- [2] LIMA, J. C. ArchCollect: uma arquitetura para coleta, transformação, carregamento e apresentação de interações de usuários Web. 2002. Tese de Mestrado. Instituto Tecnológico de Aeronáutica.
- [3] LEE, T. B. Information Management: A Proposal. 1989. Acessada em Agosto de 2008. <Disponível em <http://www.w3.org/History/1989/proposal.html>>.
- [4] LEE, T. B.; CAILLIAU, R. WorldWideWeb: Proposal for a HyperText Project. 1990. Acessada em Agosto de 2008. <Disponível em <http://www.w3.org/Proposal>>.
- [5] LEE, T. B.; FISCHETTI, M. Weaving the Web. 1999. Acessada em Agosto de 2008. <<http://www.w3.org/People/Berners-Lee/Weaving>>.
- [6] ALMEIDA, L. C. Desenvolvimento de um gerenciador de acessos a web sites. 2002. Monografia de Conclusão de Curso. Universidade Federal de Lavras.
- [7] TAO, F.; MURTAGH, F. Towards knowledge discovery from WWW log data. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'00). IEEE, Las Vegas, Nevada, 2000.
- [8] HEILMANN, C. Beginning JavaScript with DOM Scripting and Ajax - From Novice to Professional. 1ª Edição, Estados Unidos, Apress, 2006.
- [9] DALL'OGGIO, P. PHP Programando com Orientação a Objetos: Inclui Design Patterns. 1ª Edição, Brasil, Novatec, 2007.
- [10] THOMSOM, L.; WILLEY, L. PHP e MySQL – Desenvolvimento Web. Campus, 2006.
- [11] PEREIRA, N. B. M., ZANATTA, T. Artigo sobre MVC. 2006. Faculdade de Informática de Taquara.
- [12] JUNG, C. F. Metodologia para pesquisa & desenvolvimento: aplicada a novas tecnologias, produtos e processos. Rio de Janeiro, Axcel Books do Brasil Editora, 2004.
- [13] JACOBSON, I. Object-Oriented Software Engineering. Addison-Wesley, 1992.