

VITOR HUGO DE PAULA CARVALHO

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

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

LAVRAS
MINAS GERAIS – BRASIL
2008

VITOR HUGO DE PAULA CARVALHO

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

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Área de Interesse:

Engenharia de *Software*, Desenvolvimento *Web*,
Inteligência de Negócios

Orientador:

Prof. Dr. Ahmed Ali Abdalla Esmín

LAVRAS
MINAS GERAIS – BRASIL
2008

Carvalho, Vítor Hugo de Paula

Track4Web: Uma plataforma inteligente de coleta e análise de dados e interações de usuários na *Web* / Vítor Hugo de Paula Carvalho. Lavras – Minas Gerais, 2008. 81p: il.

Monografia de Graduação – Universidade Federal de Lavras. Departamento de Ciência da Computação.

1. Engenharia de *Software* 2. Desenvolvimento *Web* 3. Inteligência de Negócio
I. CARVALHO, V. H. P. II. Universidade Federal de Lavras III. Título

VITOR HUGO DE PAULA CARVALHO

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

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Aprovada em 17 de Novembro de 2008

Prof. Dr. André Luiz Zambalde

Prof. MSc. Dalton de Sousa

Prof. Dr. Rudini Menezes Sampaio

Prof. Dr. Ahmed Ali Abdalla Esmin
(Orientador)

LAVRAS
MINAS GERAIS – BRASIL
2008

“Não sabendo que era impossível, ele foi lá e fez.” (Jean Cocteau, artista francês)

AGRADECIMENTOS

Agradeço primeiramente a Deus, por ter me dado forças para lutar, paciência para persistir e a glória da conquista.

Agradeço os meus pais, Antônio e Fátima, por sempre confiarem em mim, me apoiarem e sempre estarem tão presentes em minha vida.

Agradeço o carinho das minhas duas irmãs, Dany e Luciane, que sempre estavam dispostas a me ajudar.

Agradeço a minha namorada, Mariana, pelo amor e paciência nos momentos mais complicados dessa jornada.

Agradeço ao meu orientador, Ahmed, pela orientação e confiança em mim depositada e pelo conhecimento compartilhado durante a realização deste trabalho.

Agradeço a todos os professores desta instituição, pela paciência ao tentar nos passar um pouco de seus conhecimentos.

Enfim aos colegas de turma pelo companheirismo e amizade.

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

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 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. Após desenvolvida, espera-se testar a plataforma utilizando *web sites* reais, buscando assim validar o funcionamento deste protótipo.

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 will be available on 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. Once developed, the platform will be submitted to tests with the objective of validating the prototype's work.

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

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS	ii
1. INTRODUÇÃO	1
1.1. Contextualização e Motivação	1
1.2. Objetivo	2
2. <i>WORLD WIDE WEB</i>	3
2.1. História da <i>World Wide Web</i>	3
2.2. Funcionamento da <i>World Wide Web</i>	4
2.3. Navegadores, Padrões e Protocolos	5
3. COLETA DE DADOS NA <i>WEB</i>	7
3.1. Análise de <i>Log</i>	7
3.2. Registro de Acessos	9
3.3. Tecnologias	10
4. ARQUITETURA MVC (<i>MODEL, VIEW, CONTROLLER</i>)	14
4.1. Descrição do Padrão	14
4. PESQUISA, MATERIAIS E MÉTODOS	16
5.1. Natureza da Pesquisa	16
5.2. Materiais	16
5.3. Métodos	17
6. RESULTADOS E DISCUSSÃO	19
6.1. Plataforma <i>Track4Web</i>	19
6.2. Validação da Plataforma	19
6.3. Mecanismo de Coleta	20
6.4. Mecanismo de Análise	25

6.5. Validação da Plataforma	30
7. CONCLUSÕES E TRABALHOS FUTUROS.....	31
7.1. Considerações Finais.....	31
7.2. Trabalhos Futuros.....	31
8. REFERENCIAL BIBLIOGRÁFICO	32

LISTA DE FIGURAS

Figura 4.1 - ARQUITETURA MVC.....	15
Figura 6.1 - ARQUITETURA MVC NO MECANISMO DE COLETA	20
Figura 6.2 - DIAGRAMA DE FUNCIONAMENTO DO MECANISMO DE COLETA..	23
Figura 6.3 - DIAGRAMA DO MECANISMO DE ANÁLISE.....	26
Figura 6.4 - <i>DASHBOARD</i> INTERATIVO DO MECANISMO DE ANÁLISE	27

LISTA DE TABELAS

Tabela 6.1 - DADOS DOS EVENTOS DE <i>LOG</i>	21
Tabela 6.2 - DADOS DOS EVENTOS DE <i>ACTIONS</i>	23

1. INTRODUÇÃO

1.1. Contextualização e Motivação

A utilização da Internet nos dias atuais tem crescido vertiginosamente. Segundo Santos (2008), 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. Considerando a grandiosidade da população brasileira e a respectiva porção que 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 no sentido da inovação e entendem-na como uma oportunidade de expor seus produtos e serviços em um meio de acesso de utilização crescente, com baixo custo e alto poder de influência.

É fato que a tendência da tecnologia é a redução de custos e aumento de possibilidades, partindo deste pressuposto faz-se necessário estudar formas de acompanhar a evolução e utilização de serviços e interesse por produtos na *web*. Além deste acompanhamento, também é interessante estudar a forma com que os usuários utilizam os serviços em questão. 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 Lima (2002), existem várias pesquisas na área de interações de usuários com a *web*, com ênfase em comércio eletrônico. Também existem várias ferramentas comerciais com essa finalidade, porém todas extraem dados de arquivos de *log* de servidores *web* que, de acordo com o autor, compromete a portabilidade e eleva a complexidade ao estabelecer um padrão de interação a ser utilizado por cada ferramenta.

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 mais inferências a partir de estudos e

mineração de dados. A base de dados de tais serviços não está acessível. Com isso, faz-se necessário criar um ambiente que realize a coleta, disponibilize uma interface de consulta e possibilite o acesso direto aos dados com o objetivo de permitir a criação de melhorias que venham a ser úteis para o administrador do *web site*.

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. O intuito futuro desta plataforma é ser expandida para então poder ser aplicada mais ativamente na área de Educação à Distância e também comércio virtual. Dado o caráter futuro desta aplicação, este trabalho abrangerá o desenvolvimento da plataforma que envolve mecanismos de coleta e análise dos dados e interações coletadas além de uma interface de consulta que permita visualizar estes dados com foco na tomada de decisão.

1.2. Objetivo

O principal objetivo deste trabalho foi estudar e desenvolver uma plataforma de coleta de dados e interações de usuários em *sites* da *web*. Além disso, foi criado um ambiente de consulta, no qual é possível, através de um *dashboard* interativo, visualizar os dados coletados permitindo a inferência de informações por parte do administrador do *site* em questão.

2. *WORLD WIDE WEB*

2.1. *História da World Wide Web*

De acordo com o *World Wide Web Consortium* (2008), a *World Wide Web* (“Rede de Alcance Mundial”) é um sistema de documentos em hipermídia que são interligados e executados na Internet. Os documentos podem estar na forma de vídeos, sons, hipertextos e figuras. Para visualizar a informação, usa-se um navegador, que descarrega as informações (chamadas “documentos” ou “páginas”) de servidores *web* (ou “*sites*”) e mostra-os na tela do usuário. O usuário pode então seguir os hyperlinks na página para outros documentos ou mesmo enviar informações de volta para o servidor para interagir com ele. O ato de seguir hyperlinks é comumente chamado de “navegar” ou “surfear” na *Web*.

As idéias por trás da *Web* podem ser identificadas ainda em 1980, na sede da CERN¹, na Suíça, quando Tim Berners-Lee construiu o *Enquire*². Ainda que diferente da *Web* atualmente, o projeto continha algumas das mesmas idéias primordiais, e também algumas idéias da *web* semântica. Seu intento original do sistema foi tornar mais fácil o compartilhamento de documentos de pesquisas entre os colegas.

Lee (1989) escreveu 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. Lee e Cailliau (1990) 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.

Um computador *NeXTcube* foi usado por Lee (1990) como primeiro servidor *web* e também para escrever o primeiro navegador, o *WorldWideWeb*, em 1990. No final do mesmo ano, todas as ferramentas necessárias já haviam sido criadas para o sistema: o navegador, o servidor e as primeiras páginas *web*, que descreviam o próprio projeto. Em

¹ Organização Européia para Investigação Nuclear

² Projeto utilizado para reconhecer e armazenar associações de informações

seis de agosto de 1991, ele postou um resumo no grupo de notícias *alt.hypertext*. Essa data marca a estréia da *Web* como um serviço publicado na Internet.

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. Lee e Fischetti (1999) explicam que sugeriram repetidamente o casamento das tecnologias para membros de ambas as comunidades de desenvolvedores. Como ninguém implementou suas idéias, eles decidiram implementar o projeto por conta própria. No processo, eles desenvolveram um sistema de identificação global e único de recursos, o *Uniform Resource Identifier (URI)*.

Ainda segundo o *World Wide Web Consortium* (2008), 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. Considera-se que a grande virada da *World Wide Web* começou com a introdução do *Mosaic* em 1993, um navegador gráfico criado por um time de desenvolvedores universitários. Antes de seu lançamento, os gráficos não eram freqüentemente misturados com texto em páginas *web*.

2.2. Funcionamento da *World Wide Web*

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.

O navegador então renderiza a página na tela do usuário, assim como descrita pelos arquivos que a compõe.

2.3. Navegadores, Padrões e Protocolos

Ainda nos documentos dispostos no *World Wide Web Consortium* (2008), tem-se a descrição breve do funcionamento dos navegadores e como os padrões possibilitam uma melhor compatibilidade entre os protocolos e serviços.

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.).

Os navegadores mais recentes têm a capacidade de trabalhar também com vários outros protocolos de transferência, como por exemplo, FTP, HTTPS (uma versão criptografada via SSL do HTTP), etc.

A finalidade principal do navegador é fazer o pedido de um determinado conteúdo da *Web* e providenciar a exibição do mesmo. Geralmente, quando o processamento do arquivo não é possível através do navegador, este apenas transfere o arquivo localmente. Quando se trata de texto (*Markup Language* e/ou texto simples) e/ou imagens *bitmaps*, o navegador tenta exibir o conteúdo. Esta é, de fato, a base da *Web*: atender pedidos a arquivos de hipertexto HTML, para o próprio navegador processá-los e exibi-los, juntamente com todos os arquivos dos quais o principal depende.

Os navegadores mais primitivos suportavam somente uma versão mais simples de HTML. O desenvolvimento rápido dos navegadores proprietários, porém, levou à criação de dialetos não-padronizados do HTML, causando problemas de interoperabilidade na

Web. Navegadores mais modernos (tais como o *Internet Explorer*, *Mozilla Firefox*, *Opera* e *Safari*) suportam versões padronizadas das linguagens HTML e XHTML (começando com o HTML 4.01), e mostram páginas de uma maneira uniforme através das plataformas em que rodam.

Alguns dos navegadores mais populares incluem componentes adicionais para suportar *Usenet* e correspondência de *e-mail* através dos protocolos NNTP e SMTP, IMAP e POP3 respectivamente.

3. COLETA DE DADOS NA WEB

3.1. Análise de Log

De acordo com Almeida (2001), existem duas formas de coletar dados e informações a respeito da utilização de um serviço ou *site* da *web*. A primeira 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. O *log* é um mecanismo do servidor *Web* que armazena registros para cada requisição vinda dos visitantes. A maioria dos *logs* de acesso WWW segue o formato CLF (*Common Log Format*) especificado como parte do protocolo HTTP. Cada entrada em um arquivo de *log* tem três campos importantes: o endereço IP de onde a requisição se originou, a descrição da requisição e uma indicação do momento em que esta ocorreu.

Como resultado das interações entre visitantes e um *web site*, Tao e Murtagh (2000) 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, tais como, *ilyx Edge*, *Marketwave*, *Webalizer* e *WebTrends*. Estas ferramentas geram dados estatísticos a partir dos dados brutos armazenados nos arquivos de *log*, resultando em relatórios de totais de visitas às páginas, páginas mais acessadas, data e hora de utilização, etc. Estas informações são úteis em um cenário onde os dados a serem analisados são resultados de fatores quantitativos, tais como, contagem de acessos por domínio e quantificação de uso de um recurso. Entretanto, de acordo com Tao e Murtagh (2000), 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. A principal desvantagem desta metodologia é a necessidade de se fazer a análise do *log* toda vez que as informações mais recentes sobre os

usuários do *web site* forem requeridas. Outro problema é que, normalmente, essas informações não ficam disponíveis *online* para serem consultadas em qualquer lugar onde haja um computador conectado à Internet.

3.1.1. Vantagens

Arquivos de *log* apresentam valiosas vantagens que são particulares a esta abordagem. Pode-se listar:

- **Relatórios de *Spiders* de Mecanismos de Busca**

Saber os padrões de utilização dos *spiders* pode ser de grande valor ao tentar aperfeiçoar um mecanismo de busca. Estes dados podem ser utilizados para otimizar a tecnologia e conteúdo dos *web sites* para adequar às formas com que os *spiders* trabalham.

- **Informações sobre Transferência de Dados**

Arquivos de *log* possibilitam o cálculo da quantidade de dados transferidos pelas solicitações realizadas.

- **Relatórios de Erros de Servidor**

Dados de erros são automaticamente armazenados na maioria dos arquivos de *log* e podem oferecer informações importantes para verificar determinadas funcionalidades e problemas que seriam dificilmente detectados de outra maneira.

3.1.2. Desvantagens

Os arquivos de *log* possuem certa imprecisão de dados e limitações. As informações mais importantes são listadas abaixo:

- ***Cache* de Servidores, *Cache* de Navegador e Servidores *Proxy***

Servidores de *Proxy*, utilizados pela grande maioria das empresas e grandes provedores, podem criar barreiras para a coleta de dados. Para empresas que utilizam medidas baseadas no servidor, servidores *Proxy* podem prevenir a completude dos dados de atingirem o servidor *web*, ocasionando assim a não geração do *log*. O resultado disso é uma imagem incompleta do comportamento dos visitantes.

Efeito semelhante acontece com o *cache* do navegador. Quando um visitante clica no botão “Voltar” ou “Avançar” no navegador, este utilizará uma cópia em *cache* da página armazenada na última visita do mesmo.

- ***Robots e Spiders***

Mesmo que os sistemas de *log* tentem filtrar tráfego gerado por robôs, a atualização dos mesmos para sempre estar tratando novos robôs requer um investimento constante em melhorias. O tráfego gerado por robôs é aparentemente o mesmo que um humano, o que torna difícil entender o que são de fato visitantes e o que são robôs de mecanismos de busca, além também de identificar se campanhas de *marketing* estão mesmo funcionando e sendo acessadas por usuários humanos.

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*. Usualmente estes serviços devem ser desenvolvidos seguindo a arquitetura cliente-servidor. 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. Esta abordagem também permite o controle centralizado das informações de vários servidores.

3.2.1. Vantagens

A realização do registro de acesso por meio de um serviço garante algumas vantagens, dentre as quais se podem citar:

- **Portabilidade**

A implantação de um serviço de registro de acesso não depende necessariamente do servidor onde a aplicação está hospedada, com isso, qualquer *web site* pode utilizar um mecanismo do tipo, sem que seja necessária qualquer configuração avançada.

- **Dados Estratégicos**

Dependendo da forma com que o serviço de registro foi desenvolvido, pode-se capturar um leque vasto de informações a respeito da utilização do *site* por parte do

usuário. Torna-se possível capturar formulários preenchidos, cliques em objetos, etc.

3.2.2. Desvantagens

Pode-se citar como as principais desvantagens desta abordagem:

- **Esforço de desenvolvimento**

Todas as páginas que devem ser rastreadas devem ter o código de captura corretamente posicionado, o que para grandes *sites* pode tomar certo tempo. Ultimamente esta questão tem sido menos importante devido ao fato de que gerenciadores de conteúdo largamente utilizados possuem apenas um arquivo que necessita ser modificado.

- **Códigos de Erro**

A grande maioria de *sites* e serviços *web* necessitaria de configurações adicionais para permitir que uma solução baseada em registro de acesso de páginas colete erros.

- **Informações sobre Download de Arquivos**

As soluções utilizando registro de acesso de páginas apenas permitem o rastreamento do início de downloads realizados, logo é desconhecido se determinado download foi concluído ou não.

- **Java Script Desabilitado**

Na ocasião de um usuário possuir *Java Script* desabilitado em seu navegador, os dados não poderão ser tão facilmente coletados, o que não impede, no entanto que esta situação não possa ser considerada.

3.3. Tecnologias

Abaixo têm-se algumas tecnologias que podem ser utilizadas no desenvolvimento de uma ferramenta de coleta, abordando tanto o lado do cliente (quando houver) como o lado do servidor. As tecnologias utilizadas do lado do servidor abrangem tanto a abordagem de registro de acessos quanto a análise de arquivos de *log*.

- **Cliente**

- *Java Script*

Será responsável por tratar os eventos no cliente, além de coletar e enviar informações por meio de requisições para o servidor.

- **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.

- JAVA, C/C++

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

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

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

Nas próximas subseções estão descritas as tecnologias que foram utilizadas no desenvolvimento deste trabalho.

3.3.1. Java Script

Heilmann (2006) diz 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 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 Dall'oglio

(2007), 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;
- Tipagem fraca
- Sintaxe similar a Linguagem C/C++ e o PERL

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. Thomsom e Willey (2006) afirmam que, atualmente, o *MySQL* é um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo.

Entre os usuários do banco de dados *MySQL* estão: NASA, *Friendster*, Banco Bradesco, *Dataprev*, HP, *Nokia*, *Sony*, *Lufthansa*, U.S. Army, U.S. *Federal Reserve Bank*, *Associated Press*, *Alcatel*, *Slashdot*, *Cisco Systems* e outros.

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;

- Suporte a vários tipos de tabelas (como *MyISAM*, *InnoDB*), cada um específico para um fim;
- Faltam alguns recursos quando comparados com outros banco de dados, como o *PostgreSQL* que aos poucos estão sendo implementados;
- Aceita controle transacional;
- Aceita *Triggers*;
- Aceita *Stored Procedures* e *Functions*;
- Replicação facilmente configurável;
- E tem uma Interface Gráfica feita pela *MySQL LAB*, além de vários outros de outros editores.

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. A utilização e implementação de *cookies* foi um adendo ao HTTP e muito debatida na altura em que surgiu o conceito, introduzido pela *Netscape*, devido às conseqüências de guardar informações confidenciais num computador - já que por vezes pode não ser devidamente seguro, como o uso costumeiro em terminais públicos.

4. ARQUITETURA MVC (*MODEL, VIEW, CONTROLLER*)

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.

Pereira e Zanatta (2006) definem as características da arquitetura MVC e a divide nas seguintes camadas:

- **Model**

Representação do domínio específico da informação em que a aplicação opera. Por exemplo, aluno, professor e turma fazem parte do domínio de um sistema acadêmico. 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.

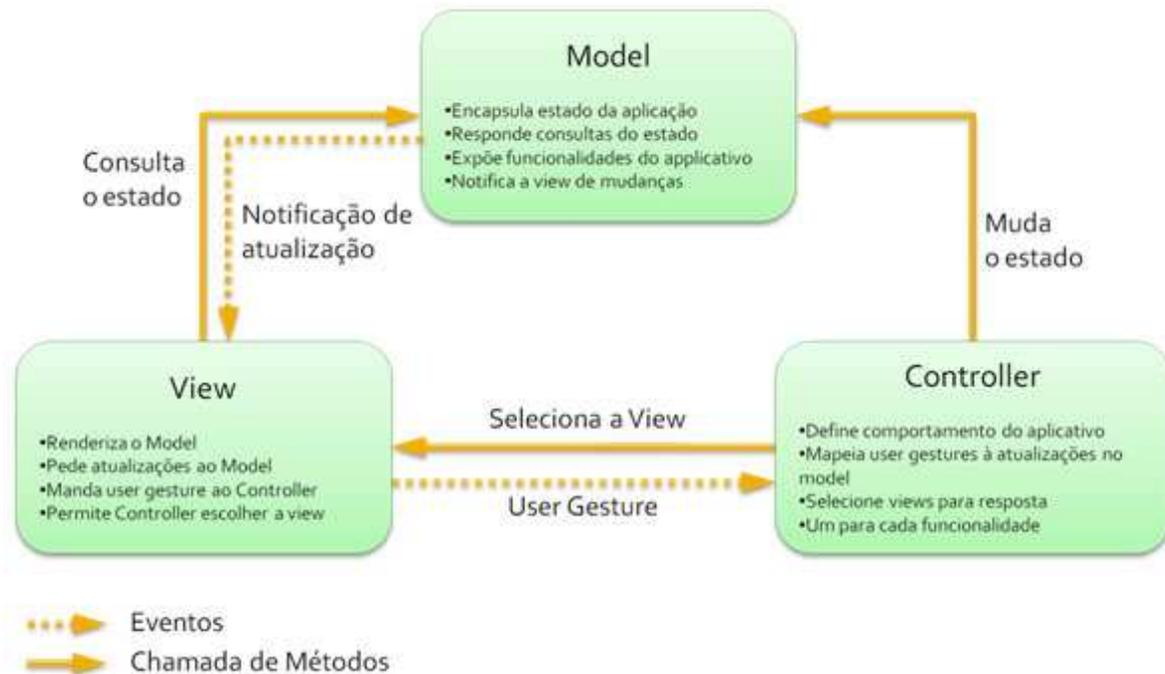


FIGURA 4.1 - ARQUITETURA MVC

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.

4. PESQUISA, MATERIAIS E MÉTODOS

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 Jung (2004) 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 segundo Jacobson (1992), é a partir de experimentações exploratórias que se produzem inovações tecnológicas.

O procedimento utilizado para desenvolver o presente trabalho foi o desenvolvimento experimental de um protótipo. Além disso, utilizou-se também a pesquisa bibliográfica, envolvendo basicamente consultas a livros, artigos científicos e outros trabalhos referentes ao tema.

5.2. Materiais

Este trabalho foi desenvolvido no Núcleo Tecnológico Educacional (NTE) situado no Departamento de Ciência da Computação (DCC) da Universidade Federal de Lavras (UFLA), no período compreendido entre os meses de Agosto e Novembro de 2008.

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 (*Integrated Development Environment*) 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 nos tópicos seguintes:

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 segundo capítulo, 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 descrito quarto capítulo, a plataforma foi desenvolvida seguindo o padrão MVC de desenvolvimento, garantindo uma maior independência dos dados e da classificação e processamento. As tecnologias escolhidas para desenvolvimento foram o PHP, a linguagem no cliente *Java Script* e a base de dados *MySQL*, estas tecnologias foram descritas na terceira seção do terceiro capítulo. Todas as tecnologias mencionadas foram escolhidas devido ao custo de desenvolvimento, visto que todas são *open source*.

As próximas seções deste capítulo abordarão, de uma forma mais específica, os dois mecanismos que compõe a plataforma, além de também ser exposta a forma com que a plataforma foi validada e seu desempenho e utilização validada.

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, interpreta as interações do usuário com o *web site* e gera eventos que são enviados para o controlador no servidor. Este então fica responsável por avaliar as informações recebidas, capturar novos dados e processá-los gerando informações adicionais que poderão então permitir a persistência dos dados na base de dados, etapa esta de responsabilidade da camada *Model*.

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 log.
URL	O endereço do navegador do usuário utilizado para acessar a página geradora do evento de log.
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 RealPlayer 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.

Caso um usuário acidentalmente perca ou voluntariamente apague os *cookies* de sua máquina, ao acessar uma página que utilize o Mecanismo de Coleta será gerada uma nova *hash* de configuração e então, após realizar uma busca na base de dados, verificar-se-á se foi salvo algum *log* com esta *hash* no intervalo de tempo previsto, caso haja, o *hash* de usuário armazenado no banco será utilizado.

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 *GeoIP*³, que, a partir do IP do usuário faz os devidos cálculos de posicionamento. Por se tratar de uma biblioteca gratuita, a mesma às vezes é ineficiente portanto pode haver problemas de identificação incorreta de cidades. O mesmo já é mais difícil de ocorrer para estados e principalmente países.

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.

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

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.
Y	Posição do elemento clicado no eixo Y.

Além das informações acima, uma vez instanciado um objeto da classe *ActionBusiness*, o mesmo é associado à 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.

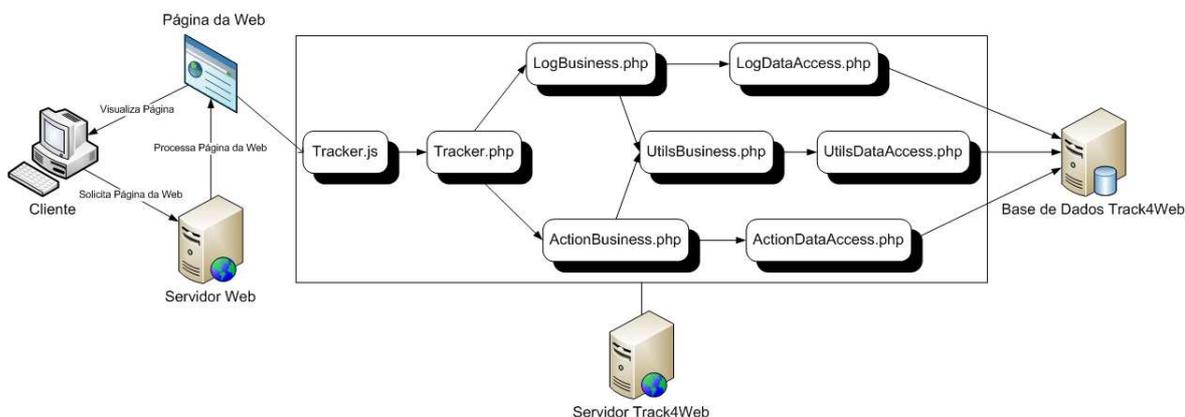


FIGURA 6.2 - DIAGRAMA DE FUNCIONAMENTO DO MECANISMO DE COLETA

Ficam então evidentes, com o auxílio do diagrama da Figura 6.2, os princípios básicos do funcionamento do mecanismo de coleta no contexto deste trabalho.

Tem-se abaixo um fluxo que auxiliará melhor o entendimento deste processo:

- Usuário visualiza documento;
- Documento invoca e executa o processamento do controlador do cliente;
- Controlador do cliente comunica-se com o controlador do servidor;
- Controlador do servidor identifica tipo de evento;
- Após identificar o tipo de evento, os dados seguem seu fluxo até à base de dados.

Tem-se abaixo um pseudo-código do funcionamento do mecanismo de coleta no controlador do servidor:

```
Funcao Coleta(dados) {
    // Informações gerais
    PegaCodigoSite();
    PegaOrigem();
    PegaPagina();
    PegaTitulo();
    PegaIP();
    PegaResolucao();
    PegaUserAgent();
    DecodificaUserAgent();

    // Informações de sistemas de busca e palavra chave
    IdentificaMecanismoDeBusca();
    PegaPalavraChave();

    // Informações de localização
    IdentificaPais();
    IdentificaEstado();
    IdentificaCidade();

    // Informações de plugins do navegador
    IdentificaPluginsNavegador();

    // Dados de sessão e usuário
    IdentificaComputador();
    IdentificaUsuario();
    IdentificaSessao();

    // Verifica se é um log atual
    VerificaExpiracao();

    // Informações do objeto de interação
    PegaObjeto ();
    DadosPosicaoDoObjeto();

    // Salva os dados
    SalvaDados();
}
```

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 do momento em que esta função é chamada, 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.

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

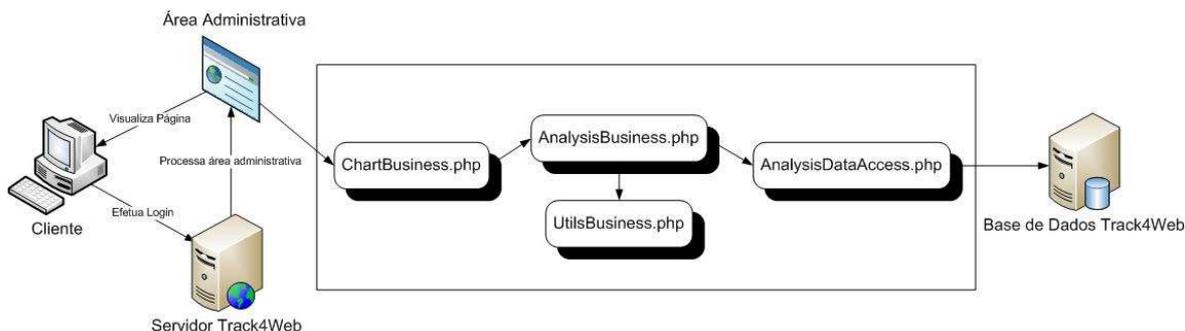


FIGURA 6.3 - DIAGRAMA DO MECANISMO DE ANÁLISE

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 e os processa deixando-os tratáveis para que a camada seguinte possa exibi-los na interface. Esta classe está englobada pela camada *Controller* em uma arquitetura MVC.

- ***UtilsBusiness.php***

Classe de negócio que oferece algumas funcionalidades úteis ao sistema e que foram separadas para que pudessem ser utilizadas por outras camadas na medida em que o desenvolvimento da plataforma evolua. 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.

6.4.1. ***Dashboard Interativo***

Para satisfazer a necessidade de exibir os dados de uma forma interativa e fácil, foi desenvolvido um *Dashboard*, que pudesse oferecer funcionalidades específicas e genéricas

para tratar as informações e disponibilizá-las da melhor maneira possível visando auxiliar e facilitar o processo de tomada de decisão.

A Figura 6.4 mostra a Interface deste *Dashboard*. Algumas informações que nele se encontram serão explanadas no decorrer desta seção.

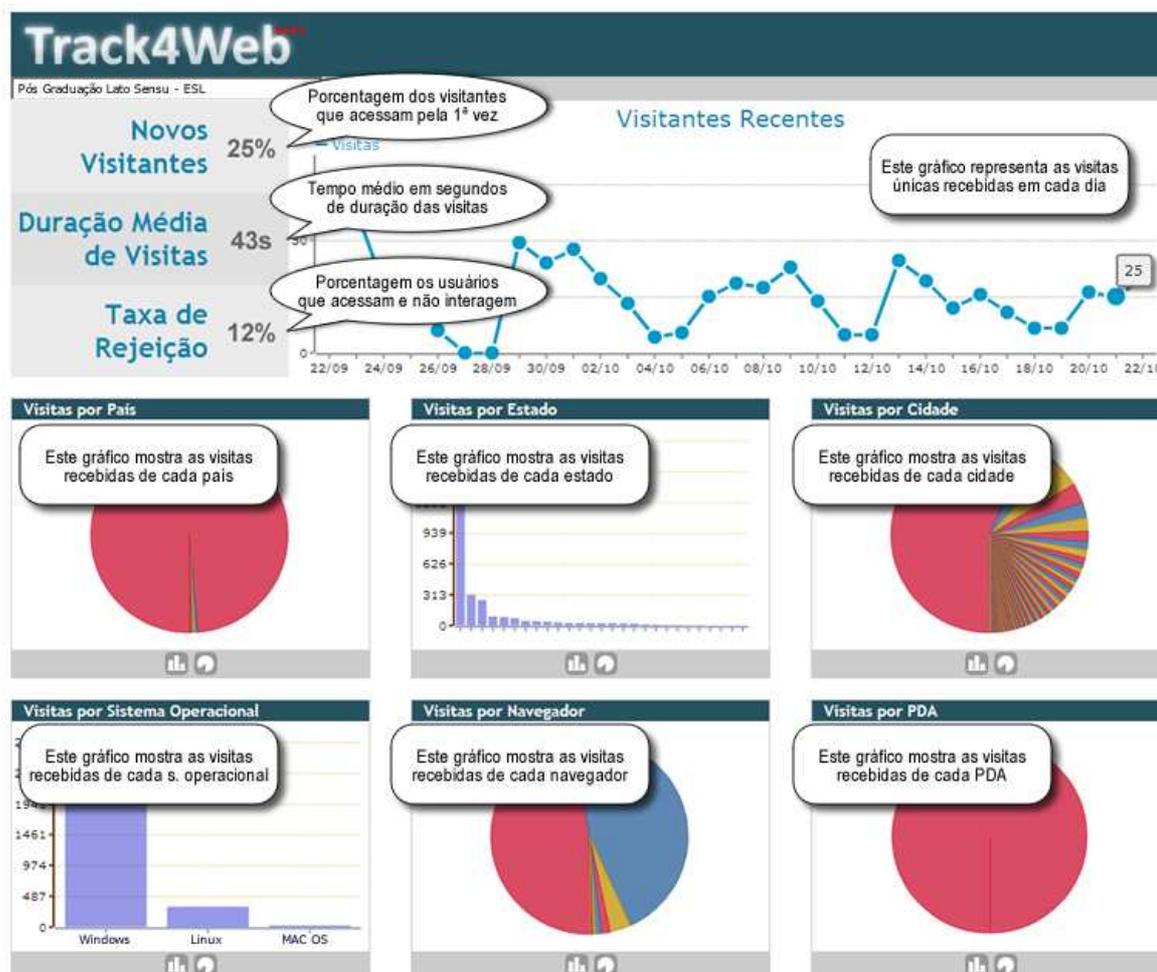


FIGURA 6.4 - DASHBOARD INTERATIVO DO MECANISMO DE ANÁLISE

Alguns *widgets*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, este relatório permite identificar

⁵ Componente de interface gráfica com o usuário.

melhores e piores épocas de utilização do *web site*, bem como oferece informações que possam auxiliar a tomada de decisão.

- **Horário das Visitas**

Pode-se com este relatório acompanhar em qual horário o servidor é mais acessado. É importante salientar que este horário representa o horário no servidor, ou seja, podem-se identificar melhores momentos de se realizar uma manutenção ou mesmo exibir conteúdos distintos.

- **Novos Visitantes**

Este relatório dá a precisa medida da 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. Esta informação se faz importante dependendo do nível de envolvimento do administrador com o *web site* em questão além de facilitar a escolha por tecnologias restritas ou conteúdo utilizado.

- **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*.

6.5. Validação da Plataforma

Durante o período compreendido entre Agosto e Novembro de 2008, a plataforma passou por fases de teste nos servidores do NTE. Neste período foram obtidos vários dados dos quais se tem um resumo a seguir:

- Mais de 15.000 registros de *log*;
- Mais de 12.000 registros de interações (*actions*);
- Usuários de mais de 17 países foram identificados;
- Quase 200 cidades foram identificadas;
- Em tempo de navegação para os sites, foi calculado aproximadamente 4,5 milhões de segundos.

Estes números ajudam a identificar a complexidade do trabalho em questão, além da grandiosidade do universo de dados, com os quais tenta-se trabalhar e entender melhor.

7. CONCLUSÕES E 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). As funcionalidades do mecanismo de análise foram validadas sendo verificada sua correta adaptação perante aos objetivos propostos.

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

ALMEIDA, L. C. **Desenvolvimento de um gerenciador de acessos a web sites.** 2002.
Monografia de Conclusão de Curso. Universidade Federal de Lavras.

CHAKRABARTI, S. **Mining the web: Discovering knowledge from hypertext data.**
2ª Edição, Estados Unidos, Elsevier, 2003.

DALL'OGGIO, P. **PHP Programando com Orientação a Objetos: Inclui Design
Patterns.** 1ª Edição, Brasil, Novatec, 2007.

HEILMANN, C. **Beginning JavaScript with DOM Scripting and Ajax - From Novice
to Professional.** 1ª Edição, Estados Unidos, Apress, 2006.

JACOBSON, I. **Object-Oriented Software Engineering.** Addison-Wesley, 1992.

JUNG, C. F. **Metodologia para pesquisa & desenvolvimento: aplicada a novas
tecnologias, produtos e processos.** Rio de Janeiro, Axcel Books do Brasil Editora,
2004.

LEE, T. B. **Information Management: A Proposal.** 1989. Acessada em Agosto de 2008.
<Disponível em <http://www.w3.org/History/1989/proposal.html>>.

LEE, T. B.; CAILLIAU, R. **WorldWideWeb: Proposal for a HyperText Project.** 1990.
Acessada em Agosto de 2008. <Disponível em <http://www.w3.org/Proposal>>.

LEE, T. B. **The WorldWideWeb browser.** 1990. Acessada em Agosto de 2008.
<Disponível em <http://www.w3.org/People/Berners-Lee/WorldWideWeb>>.

LEE, T. B.; FISCHETTI, M. **Weaving the Web**. 1999. Acessada em Agosto de 2008. <<http://www.w3.org/People/Berners-Lee/Weaving>>.

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.

MARKOV, Z.; LAROSE, D. T. **Data mining the web: Uncovering patterns in web content, structure and usage**. Wiley, 2007.

PEREIRA, N. B. M., ZANATTA, T. **Artigo sobre MVC**. 2006. Faculdade de Informática de Taquara.

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.

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.

THOMSOM, L.; WILLEY, L. **PHP e MySQL – Desenvolvimento Web**. Campus, 2006.

WORLD WIDE WEB CONSORTIUM. **World Wide Web Consortium W3C**. 2008. Acessada em Agosto de 2008. <<http://www.w3.org>>.