

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

# Sistema de Monitoramento de Encomendas e acompanhamento pós vendas

Matheus Bratfisch

Florianópolis, novembro de 2012

Matheus Bratfisch

*Sistema de Monitoramento de Encomendas  
e acompanhamento pós vendas*

Trabalho de conclusão de curso apresentado para obtenção do Grau de Bacharel em Ciências da Computação pela Universidade Federal de Santa Catarina.

Orientador:  
Leandro José Komosinski

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Florianópolis, novembro de 2012

Matheus Bratfisch

Sistema de Monitoramento de Encomendas e acompanhamento pós  
vendas

Trabalho de Conclusão de Curso submetido e apresentado a Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciências da Computação e aprovado pela banca examinadora constituída por:

---

Prof. Leandro José Komosinski

---

Prof. Frank Siqueira

---

Prof. Patricia Della M<sup>e</sup>a Plentz

# *Resumo*

Empresas realizam vendas todos os dias. Muitas destas empresas acreditam que o processo de venda está terminado assim que a "venda" está concluída. Estas realizam vendas por internet, telefone e outros e consideram a venda concluída logo que o produto é enviado. Mas o mercado já não é bem assim. Os consumidores, que realizam compras ficam cada vez mais exigentes. Com a popularização de tecnologias estes consumidores passaram a gerar feedbacks, opiniões, comentando sobre a marca e gerando uma importante rede de informações sobre as empresas que venderam os produtos. Com esses dois fatores em mente, pode-se perceber claramente que a venda não está terminada após o pagamento ou envio do produto. Um acompanhamento realizado após a mesma é necessário e importante.

Este trabalho propõe uma solução que visa auxiliar os comerciantes a realizar um acompanhamento pós-venda de qualidade, averiguando a opinião direta do consumidor, tornando assim possível que a empresa melhore seus processos internos e criando uma certa confiança com seus clientes. A proposta é a criação de um sistema que monitore automaticamente as encomendas enviadas por uma empresa, e assim que a mesma for entregue realize uma pesquisa de satisfação do cliente.

Foi desenvolvido um sistema de monitoramento de encomendas online que permite o cadastro de códigos de rastreio dos Correios, o sistema por sua vez monitora de forma autônoma a evolução deste código de rastreio. O sistema permite que sejam cadastrados e-mails a cada encomenda e assim que a mesma é entregue, o sistema realiza uma pesquisa de satisfação com perguntas customizáveis. Para melhorar a compreensão do que cada código de rastreio significa, foi criado um sistema de etiquetas para cada entrada no sistema. Visando a facilidade e a compreensão da localização das encomendas, o sistema de mapas do Google (Google Maps) é utilizado para exibir os locais onde a encomenda já esteve e para onde vai.

# *Lista de Figuras*

1	Antecedentes e Consequências da confiança numa empresa e no vendedor [Doney e Cannon 1997] . . . . .	p. 11
2	SCRUM Process [SCRUM] . . . . .	p. 17
3	Diagrama demonstrando o método geral de busca. . . . .	p. 21
4	Diagrama demonstrando o método específico de busca dos correios. . . . .	p. 22
5	Modelagem do Banco de dados Relacional . . . . .	p. 23
6	Página inicial do Sistema . . . . .	p. 24
7	Página de Usuário logado no Sistema . . . . .	p. 26
8	Erro durante tentativa de visualização de código de rastreio com status Conferido . . . . .	p. 27
9	Exemplo de estatísticas. . . . .	p. 28
10	Página inicial do Sistema . . . . .	p. 29
11	Página com autenticação do usuário . . . . .	p. 30
12	Página de Usuário logado no Sistema . . . . .	p. 30
13	Exemplo de estatísticas. . . . .	p. 31
14	Página de configurações do usuário . . . . .	p. 31

# *Lista de Tabelas*

- 1 Tabela demonstrando verificação de eficiência de consulta (parte 1) . . . p.24
- 2 Tabela demonstrando verificação de eficiência de consulta (parte2) . . . p.25

# *Sumário*

<b>1</b>	<b>Introdução</b>	p. 8
1.1	Motivação . . . . .	p. 8
1.2	Objetivos Gerais . . . . .	p. 8
1.3	Objetivos Específicos . . . . .	p. 9
1.4	Estrutura do Trabalho . . . . .	p. 9
<b>2</b>	<b>Descrição do Problema e de Soluções existentes</b>	p. 10
2.1	Relação Cliente com o Vendedor . . . . .	p. 10
2.2	Descrição do Problema . . . . .	p. 11
2.3	Descrição de Soluções existentes . . . . .	p. 11
2.3.1	PackageTrackr . . . . .	p. 12
2.3.2	Anywhere ECT - Integração com os correios . . . . .	p. 12
<b>3</b>	<b>Metodologia e Tecnologias</b>	p. 13
3.1	Linguagens e Ferramentas . . . . .	p. 13
3.1.1	HTML . . . . .	p. 13
3.1.2	JavaScript . . . . .	p. 13
3.1.3	PHP . . . . .	p. 14
3.1.4	MySQL . . . . .	p. 14
3.1.5	GetText . . . . .	p. 15
3.2	Metodologia de Desenvolvimento Ágil . . . . .	p. 16
3.2.1	Desenvolvimento com SCRUM . . . . .	p. 16

<b>4 Sistema de Monitoramento de Encomendas e acompanhamento pós venda</b>	p. 19
4.1 Descrição do sistema . . . . .	p. 19
4.2 Solução Proposta . . . . .	p. 20
4.2.1 Análise de Requisitos . . . . .	p. 20
4.2.2 SPRINT - Desenvolvimento - 1 . . . . .	p. 20
4.2.3 SPRINT - Modelagem do Banco de dados . . . . .	p. 21
4.2.4 SPRINT - Desenvolvimento - 2 . . . . .	p. 22
4.2.5 SPRINT - Otimização de consultas . . . . .	p. 23
4.2.6 SPRINT - Desenvolvimento - 3 . . . . .	p. 25
4.2.7 SPRINT - Análise de satisfação do cliente . . . . .	p. 26
4.2.8 SPRINT - Desenvolvimento - 4 . . . . .	p. 27
4.2.9 SPRINT - Ajustes finais . . . . .	p. 28
4.3 Visão Geral . . . . .	p. 29
4.4 Análise de um caso de uso . . . . .	p. 31
4.4.1 Informações sobre a empresa . . . . .	p. 31
4.4.2 Análise anterior a utilização do software . . . . .	p. 32
4.4.3 Análise posterior a utilização do software . . . . .	p. 32
<b>5 Conclusão e Trabalhos futuros</b>	p. 33
5.1 Conclusão . . . . .	p. 33
5.2 Trabalhos Futuros . . . . .	p. 33
<b>Referências</b>	p. 34
<b>Anexo A – Artigo</b>	p. 35



# ***1 Introdução***

A relação entre um vendedor e um comprador não termina quando uma venda é realizada. A relação que se intensifica depois de uma venda vem crescendo e esta determina a escolha do comprador numa próxima oportunidade. A venda, então, meramente consome o ponto quando o casamento começa. Quão bom este relacionamento é depende de quão bem o vendedor gerencia o seu relacionamento, [Levitt 1983] e que demonstra a grande importância do acompanhamento pós venda. Com a popularização de lojas de e-commerce essa relação fica ainda mais dinâmica, aumentando a necessidade de manter-se uma relação boa com os clientes.

## **1.1 Motivação**

O acompanhamento pós venda se torna cada vez mais importante para o comprador e para o vendedor. Com isso percebemos a necessidade de uma ferramenta que auxilie neste processo novamente tornando o comprador mais satisfeito e possibilitando que o vendedor melhore cada vez mais.

Com a popularização de vendas por telefone e internet, as transportadoras tem se tornado parte importante do processo de compra e venda e então surge a necessidade de monitorar o estado das encomendas.

## **1.2 Objetivos Gerais**

Este trabalho tem como principal objetivo a criação de um sistema de monitoramento dos sistemas de encomendas, possibilitando uma análise instantânea do nível de satisfação do usuário.

## 1.3 Objetivos Específicos

Para atingir os objetivos gerais, os seguintes objetivos específicos são propostos:

- Criação de um Sistema de Monitoramento de entregas de encomendas.
- Adição de funcionalidades que permitam realizar pesquisa de satisfação do usuário de forma automática.
- Exibição das respostas das pesquisas de satisfação.

## 1.4 Estrutura do Trabalho

Este trabalho divide-se em 5 capítulos. O primeiro nos trouxe a Introdução e a motivação para a realização do mesmo, o segundo descreve o problema e traz soluções existentes. O capítulo 3 analisa algumas soluções existentes e propõe uma solução mais completa. O capítulo 4 demonstra o desenvolvimento deste trabalho, com uma visão geral do sistema e relata um caso de uso que auxiliou no desenvolvimento deste. O capítulo 5 apresenta as conclusões e descreve possibilidades de trabalhos futuros.

## *2 Descrição do Problema e de Soluções existentes*

### **2.1 Relação Cliente com o Vendedor**

As relações entre Cliente e Vendedor, sua qualidade e como evoluí-la é um assunto bastante estudado pela área de marketing. Esta muitas vezes propõe modelos a serem seguidos para que a mesma tenha sucesso. Infelizmente, a maioria das pesquisas e estratégias de marketing tratam a troca entre comprador e vendedor como um evento discreto, não como uma relação contínua, [Dwyer e Oh 1987] ou seja, esta transação é tratada como um evento único que termina com a concretização da venda. Essas relações contínuas assumem diferentes formas, [Dwyer e Oh 1987].

Alguns modelos levantam diversos fatores a serem acompanhados e modificados para que uma relação entre Cliente e Vendedor obtenha sucesso. Um dos fatores citados é a confiança, que também pode ser chamada de credibilidade. Este é formado por diversos fatores como:

- Reputação
- Tamanho
- Características do vendedor
- Tempo do "relacionamento"
- Performance na entrega
- Preço
- Performance do Produto/Serviço
- Entre outros.

Na Figura 1 é demonstrada a interação entre esses diversos fatores na decisão de escolha do comprador como um diagrama. A mesma demonstra que a credibilidade do vendedor é formada pela reputação e tamanho do fornecedor, a experiência e a força do vendedor e esses fatores por sua vez refletem na decisão de compra do consumidor.

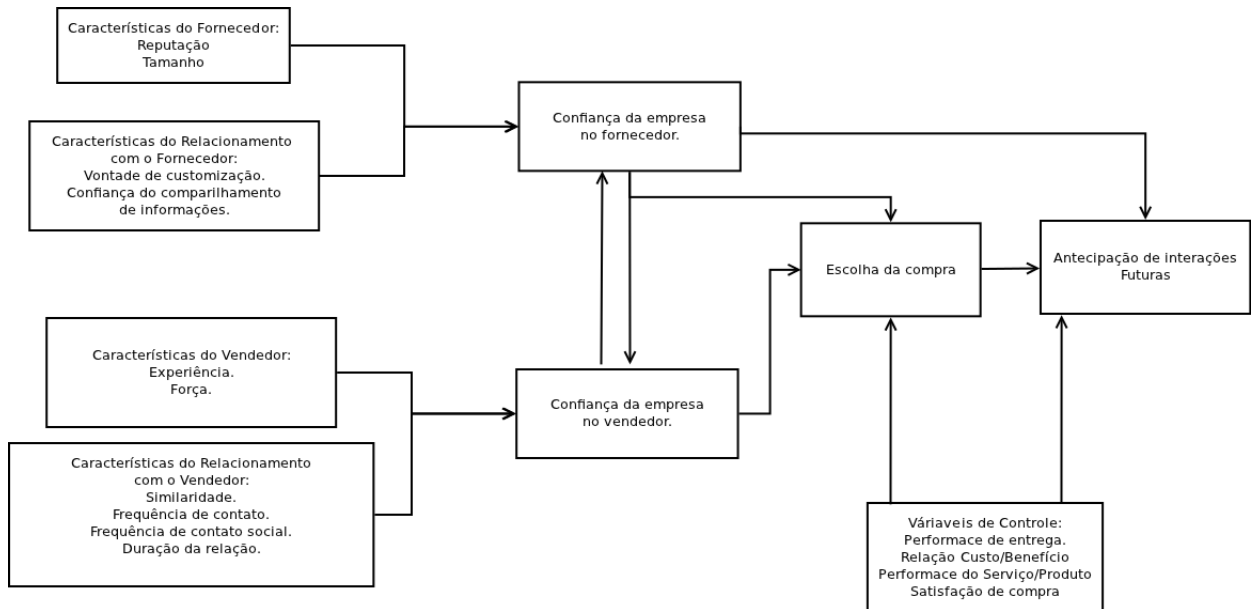


Figura 1: Antecedentes e Conseqüências da confiança numa empresa e no vendedor [Doney e Cannon 1997]

## 2.2 Descrição do Problema

Verificado que a relação entre um vendedor e um comprador é um processo contínuo que deve ser tratado como um relacionamento. Levando em consideração os vendedores online (lojas, e-commerce) é percebido a falta de um mecanismo que auxilie estes vendedores a manter uma boa relação com os clientes. Após o produto comprado pelo cliente ser enviado o vendedor não toma mais atitudes e deixa a relação esfriar, esta parte do processo pode ser trabalhada para que a relação entre ambos melhore e o cliente sinta mais confiança no vendedor que se mostra presente mesmo depois do envio do produto.

## 2.3 Descrição de Soluções existentes

Foi realizada uma pesquisa de sistemas semelhantes que pudessem suprir as necessidades que este trabalho tenta suprir, sistemas com as mesmas funcionalidades que este trabalho, não foram encontrados durante as pesquisas realizadas. Duas soluções encontra-

das que implementam parcialmente a nossa solução foram encontrados e serão descritos, estes pareciam ser os mais completos e semelhantes por isso, estes são apresentados.

### **2.3.1 PackageTrackr**

Lançado em Janeiro de 2008, o PackageTrackr teve como motivação as deficiências das diversas transportadoras. O mesmo é uma solução para monitorar pacotes de maneira automática. Este possui muitas funcionalidades e suporte a diversas transportadoras. [PackageTrackr]

Algumas de suas funcionalidades que assumem mais destaque são:

- Suporte a diversas transportadoras (Fedex, USPS, DHL, Australian Post, Canadian Post, etc)
- Acesso via Smartphone (Android, Iphone)

### **2.3.2 Anywhere ECT - Integração com os correios**

A empresa Ledcorp, criada a quase 10 anos, com especialidade em acessibilidade e comunicação, mobilidade, rastreamento e monitoração de veículos, cargas e pessoas, integração de sistemas e outros. Possui um módulo chamado de Anywhere ECT. Este realiza controle e monitoramento do andamento do processo de entrega de encomendas pelos Correios. [Anywhere ECT - Integração com os Correios]

Utilizando este junto a outras ferramentas pode-se obter funcionalidades extras como acompanhar em um mapa a evolução do status das postagens, definir SLA (Service Level Agreement), possibilidade de notificação via SMS, integração com redes sociais.

Ele também pode ser integrado a qualquer ERP de forma direta ou por EDI, atualizando diretamente o status de entrega de forma automatizada.

## ***3 Metodologia e Tecnologias***

### **3.1 Linguagens e Ferramentas**

#### **3.1.1 HTML**

HTML (*HyperText Markup Language*) é a principal linguagem de marcação para páginas web. A partir desta, os navegadores sabem como renderizar e posicionar textos, imagens e definir características dos mesmos. Esta foi desenvolvida pelo *World Wide Web Consortium* e WHATWG (*Web Hypertext Application Technology Working Group*). A mesma utiliza elementos html que são formados por <elemento>e normalmente fechada por </elemento>, em alguns casos a mesma utiliza uma notação simplificada sendo <elemento />. Cada elemento pode possuir diversos parâmetros específicos que definem mais precisamente as características de um elemento. A primeira menção a HTML surgiu em 1991 por Berners-Lee que a definia com apenas 18 elementos e era bastante simples. A primeira versão "padronizada" surgiu em 1995 com o RFC1866. Atualmente a mesma continua sendo amplamente utilizada e se encontra na versão 4.01. A nova versão (HTML 5) já esta sendo escrita e atualmente é um rascunho. Este trabalho fará uso da versão 4.01, priorizando a compatibilidade entre navegadores.

Esta linguagem de marcação é utilizada no sistema para definir a interface gráfica do sistema, posicionando os elementos, definindo cores, tamanhos de textos, etc.

#### **3.1.2 JavaScript**

JavaScript é uma linguagem de script baseada em ECMAScript, sua primeira versão surgiu em 1995. É a linguagem mais utilizada para programação client-side em navegadores web, [JavaScript]. Esta é uma linguagem multi-paradigmas, suporta orientação a objetos, imperativa e funcional. Novas tecnologias como o Ajax [JavaScript], fizeram com que a mesma tomasse destaque novamente. Ela foi desenvolvida inicialmente por Brendan

Eich, com o nome Mocha. O uso mais comum desta é escrever funções que estão junto ao HTML e interagem com os objetos DOM (*Document Object Model*), [JavaScript]. Alguns exemplos simples de sua utilização são:

- Validação de formulários.
- Alteração de imagens a partir de eventos.
- Transmissão de informações sobre os hábitos de um usuário numa página Web.

O JavaScript é utilizada no sistema para tornar a linguagem de marcação interativa, permitindo que ações em um elemento específico altere a estrutura de marcação sem a necessidade de carregar a página novamente, por exemplo, a exibição de detalhes de um item e a remoção de uma encomenda da listagem do usuário.

### 3.1.3 PHP

O PHP (*PHP: Hypertext Processor*, originalmente: *Personal Home Page*), é uma linguagem utilizada no lado do servidor com propósitos gerais que originalmente foi projetada como uma linguagem para desenvolvimento Web para produzir páginas dinâmicas. O seu desenvolvimento começou em 1994 quando o programador Rasmus Lerdorf desenvolveu um conjunto de Scripts PERL para manter sua página web pessoal. O seu lançamento oficial depois de diversas mudanças aconteceu em 1998 com o PHP3. PHP é Software Livre sobre a licença PHP. Atualmente a sua última versão é a 5.4, que foi lançada dia 1o de Março de 2012. [PHP] Ele pode ser utilizado com grande parte de sistemas operacionais e servidores web.

A linguagem de programação PHP é quem realiza a maior parte das funções deste sistema, adicionando informações ao banco de dados, consultando informações do banco de dados, fazendo iterações para exibição de itens e diversas outras funções.

### 3.1.4 MySQL

As informações são armazenadas em um banco de dados do modelo relacional do tipo MySQL. A sua primeira versão foi lançada em 1995 e atualmente é o banco de dados relacional mais utilizado no mundo, [MySQL]. Este é código aberto com a licença GPL (Licença Pública Geral). Sua consolidação também é demonstrada por meio de sistemas famosos que utilizam do mesmo como: Joomla, Wordpress, Drupal entre outros. Sua

última versão estável é a 5.5.13 lançada em 13 de Abril de 2012. Com essas informações temos que este é um banco de dados consolidado e que continua a ser desenvolvido e aprimorado. O MySQL possui diversas funcionalidades como:

- Multi Plataforma
- Gatilhos
- Visões
- Suporte a SSL
- Diferentes formas de armazenamento como MyISAM, InnoDB entre outros
- Suporte a replicação

Este banco de dados é o responsável por armazenar de forma persistente as informações inseridas no sistema, armazenar a relação entre essas informações e também permitir a recuperação destas.

### 3.1.5 GetText

Em sistemas atuais a internacionalização do software é algo de grande importância, já que serviços online estão disponíveis para o mundo todo. Um software internacionalizado permite que a entrada de dados seja feita em mais de uma língua e a interface do mesmo também deve sofrer adaptações, demonstrando sua interface na língua que o usuário definir. O sistema utilizado para permitir a internacionalização é o GetText. Este é um sistema multi plataforma que teve sua primeira versão lançada em 1995, [GetText], que também está integrado ao PHP, facilitando assim a internacionalização. A internacionalização é feita criando um arquivo para cada linguagem com as sequências de caracteres a serem exibidas para cada sequência da língua padrão. Após definida a linguagem a ser utilizada, o gettext utiliza desses arquivos de línguas para converter a língua padrão para a língua desejada.

Para tornar o sistema internacionalizado a biblioteca GetText é utilizada. A mesma é responsável por fazer o mapeamento de textos de uma língua para outra.



## 3.2 Metodologia de Desenvolvimento Ágil

A utilização de uma metodologia para o desenvolvimento de software é essencial para conseguir entregar o software desejado no prazo e sem erros. Estas metodologias começaram a aparecer nos anos 60 definindo técnicas para criar software com sucesso. A metodologia Ciclo de Vida de desenvolvimento de Sistemas é considerado a mais antiga técnica formalizada. Em paralelo às técnicas clássicas, existem as técnicas de desenvolvimento ágil, que reduzem um pouco a burocracia e possuem outros valores.

Métodos ágeis são uma reação aos métodos tradicionais de desenvolvimento de software. No desenvolvimento de maneira tradicional o desenvolvimento começa com a definição e documentação de um completo conjunto de requisitos. A indústria e a tecnologia se movem muito depressa e estes requisitos mudam constantemente e os clientes, por sua vez, se tornam cada vez mais incapazes de declarar suas necessidades de forma definitiva, [Cohen e Costa]. Estes métodos ágeis são normalmente um conjunto de diferentes técnicas que compartilham os mesmos valores e princípios básicos. Alguns dos princípios do Desenvolvimento Ágil:

”Indivíduos e interações mais que processos e ferramentas  
 Software em funcionamento mais que documentação abrangente  
 Colaboração com o cliente mais que negociação de contratos  
 Responder a mudanças mais que seguir um plano  
 Ou seja, mesmo havendo valor nos itens à direita  
 valorizamos mais os itens à esquerda.” [Agile Manifesto]

### 3.2.1 Desenvolvimento com SCRUM

Para atingir os objetivos propostos deste trabalho a metodologia SCRUM foi adotada. Este, por sua vez, é um método incremental e iterativo para desenvolvimento ágil de software. O SCRUM possui etapas bem definidas que podem ser vistas na Figura 2.

Este processo possui etapas bem definidas. O Product BackLog define uma lista de requisitos de sistema normalmente demonstrados na forma de histórias e são ordenadas de acordo com risco, valor para o negócio entre outros. Já o Sprint BackLog define quais requisitos do sistema serão solucionados em uma determinada Sprint. Uma Sprint é a unidade básica do desenvolvimento no SCRUM e esta tem duração de 1 semana a 1 mês. Quando uma determinada SPRINT é terminada é realizada uma reunião de revisão para

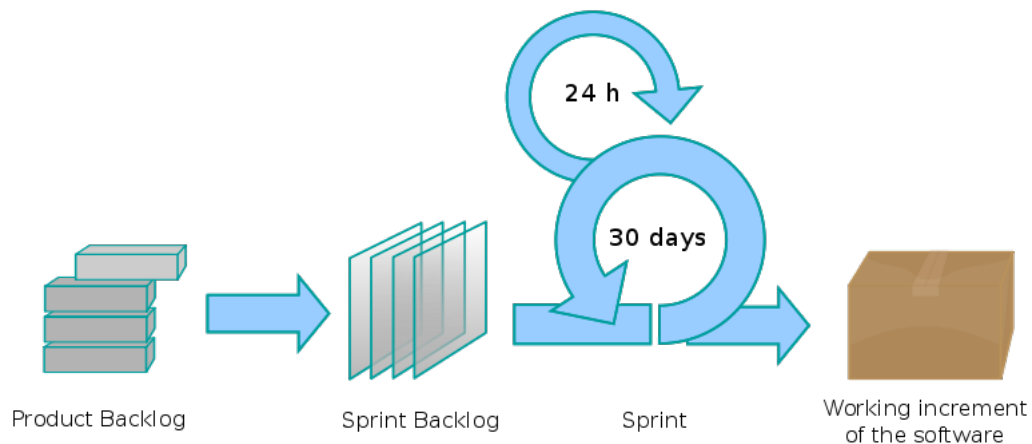


Figura 2: SCRUM Process [SCRUM]

saber o que foi concluído, o que não e também apresenta aos clientes. Diversas outras etapas são definidas, tais quais, reunião diária com duração máxima de uma hora que tem o intuito de se repassar o que foi feito ontem, o que será feito hoje, etc. Antes de se começar uma nova Sprint é realizada uma reunião de planejamento de Sprint, que seleciona o trabalho que será feito na Sprint seguinte e define o tempo estimado para cada problema ser resolvido.

Este trabalho de conclusão de curso, por ser realizado por apenas uma pessoa com ajuda de um vendedor. Simplificará o método SCRUM utilizando somente algumas de suas etapas. O trabalho criará um Product Backlog que definirá os requisitos de sistema iniciais, ou seja, quais funcionalidades se deseja no programa ordenadas por ordem decrescente de risco. Em seguida se inicia um processo iterativo, onde inicialmente será definido quais requisitos serão solucionados na Sprint seguinte. Ao fim de uma sprint, as funcionalidades serão demonstradas ao vendedor, o qual manterá uma utilização contínua do sistema, gerando feedback sobre as funcionalidades criadas e sobre possíveis novos requisitos de sistema.

Este trabalho utiliza algumas técnicas ou valores de desenvolvimento ágil, tais quais:

- Código - É a parte mais importante do desenvolvimento. Sem ele não existe produto funcionando
- Ouvir - Programadores devem conversar com os clientes para saber o que o sistema precisa fazer
- Sprint - Ciclos de desenvolvimento

Então temos que a valorização do código é muito alta, afinal sem o mesmo, é impossível ter o produto funcionando. Como o mercado e por consequência os usuários e clientes estão em constante mudança a comunicação é essencial para o sucesso do mesmo, fazendo que mesmo durante o processo de desenvolvimento o software sofra mudanças e se adapte à necessidade deste. O desenvolvimento é dividido em ciclos com problemas a serem resolvidos (funcionalidades a serem implementadas) definidos, onde após cada ciclo a comunicação com o cliente ou usuário é necessária para averiguar se existe alguma modificação no sistema que é necessária e se as funcionalidades implementadas estão de acordo com a necessidade dele.

## *4 Sistema de Monitoramento de Encomendas e acompanhamento pós venda*

### **4.1 Descrição do sistema**

Este trabalho propõe a criação de um sistema intermediário a ambos os sistemas apresentados e que adiciona novas funcionalidades que permitem que nosso objetivo seja alcançado. O sistema é uma aplicação na internet que consiste de um sistema de pesquisa de status de encomendas. Para usuários cadastrados é possível a adição de códigos de rastreio a sua conta. Os códigos de rastreio são monitorados automaticamente pelo sistema informando ao usuário modificações no seu status. Estas verificações ocorrem de forma automática de 3 a 6 vezes por dia.

Levando em consideração a necessidade de pequenos e médios vendedores este sistema permite o cadastro de um segundo e-mail que, quando o status de um pacote for "Entregue" o sistema contate este e-mail automaticamente para fazer uma pesquisa de satisfação do cliente. Assim possibilita que estes vendedores descubram suas falhas e com isso aperfeiçoem seus métodos de produção para corrigir as falhas descobertas.

Para uma exibição mais amigável o mesmo utiliza do sistema de mapas Google Maps para exibir de maneira mais natural a atual localização do pacote e trecho a ser percorrido. Por exemplo, quando o pacote estiver como "Encaminhado para Cidade B" é desenhada uma linha vermelha entre a cidade A e a cidade B, e quando o status do pacote for alterado, assim garantindo que o mesmo já percorreu o trecho da cidade A para cidade B, esta se tornará verde.

## 4.2 Solução Proposta

### 4.2.1 Análise de Requisitos

A análise de requisitos foi realizada com auxílio de um vendedor que utilizará o sistema. A partir de conversas foi levantada a seguinte lista de requisitos para o sistema:

- Verificação de forma automatizada do status de encomendas;
- Possibilidade de monitoramento de diversas encomendas em paralelo;
- Cadastro de usuários que utilizarão o sistema;
- Notificação via e-mail da mudança de status;
- Sistema de etiquetas para facilitar a compreensão do que cada código de rastreio realmente é;
- Cadastro de e-mails secundários para pesquisa de satisfação do cliente;
- Repassar as informações de satisfação ao usuário.

Com isso uma análise de riscos sobre cada requisito foi feita e a lista foi ordenada de acordo com esses riscos.

### 4.2.2 SPRINT - Desenvolvimento - 1

Na primeira SPRINT o maior risco do sistema foi selecionado para diminuir os riscos de desenvolvimento do projeto analisando a viabilidade do sistema. O maior risco considerado foi a "Verificação de forma automatizada do status de encomendas" já que para atingir este objetivo precisaríamos de um meio para acessar as informações de localização e status da transportadora utilizada e essa funcionalidade poderia conter barreiras que tornariam o desenvolvimento do sistema inviável.

As maiores preocupações foram a impossibilidade de acesso de forma automática, porque o sistema poderia exigir verificação via captcha ou mesmo não possuir uma forma de fazer tal acompanhamento. A transportadora mais comum e também a mais utilizada pelo nosso cliente é a empresa de Correios. Então foi verificada a existência desses empecilhos e constatado que: o acompanhamento via código de rastreio é possível, verificação via captcha não existe e não existe uma API de acesso. Com isso, foi decidido

que uma requisição ao sistema seria feita no endereço e formato padrão para verificação da localização e status da encomenda, o resultado obtido seria analisado por padrões nas estruturas HTML, utilizando expressões regulares.

Com isso, foram definidos 2 diagramas de fluxo em alto nível. Um informa como seria o método geral de busca que pode ser visto na Figura 3 e também o método específico de busca dos correios demonstrado na Figura 4

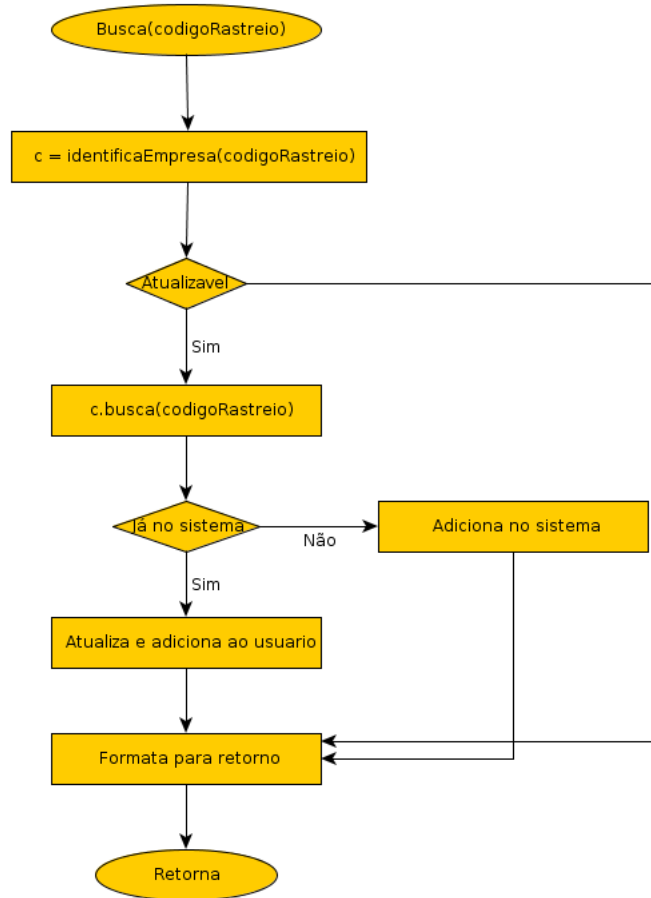


Figura 3: Diagrama demonstrando o método geral de busca.

### 4.2.3 SPRINT - Modelagem do Banco de dados

Uma sprint com foco na modelagem de banco de dados foi realizada. A modelagem neste estágio permite que os requisitos sejam verificados novamente. Para que todos os requisitos pudessem ser atingidos a seguinte modelagem relacional foi proposta na Figura 5 onde podemos visualizar as seguintes tabelas: *Users* trata do armazenamento de informações do usuário, *User\_has\_Tracks* armazena os códigos de rastreo e suas especificações de cada usuário, *Tracks* que possui informações das encomendas monitoradas,

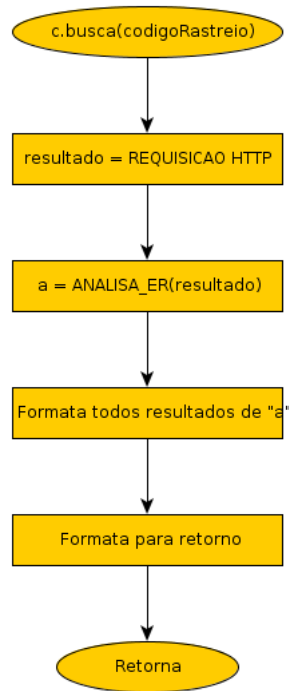


Figura 4: Diagrama demonstrando o método específico de busca dos correios.

*Snippets* guarda informações relacionadas aos trechos de cada encomenda, *SnippetsGT* possui informações extras sobre os trechos e não é utilizado em todo trecho, *Notifications* guarda a quais e-mails serão enviadas as pesquisas de satisfação quando a encomenda for marcada como entregue.

A mesma foi desenvolvida com foco nos requisitos do sistema, então ao seu término uma análise se os requisitos podem ser atingidos com o modelo proposto foi feita, após esta análise constatou-se que as funcionalidades do sistema podem ser implementadas com o modelo proposto.

#### 4.2.4 SPRINT - Desenvolvimento - 2

Os requisitos a serem realizados nesta etapa foram "Cadastro de usuários que utilizarão o sistema" e "Possibilidade de monitoramento de diversas encomendas em paralelo". Estas foram escolhidas pois andam lado a lado, já que encomendas pertencem a Usuários. Já que o sistema não possuía nenhuma interface gráfica definida, esta SPRINT definiu alguns padrões da interface assim como a interface de algumas páginas, como a página inicial, que pode ser vista na Figura 6

Como o banco de dados já possuía as características de um usuário bem definidas a SPRINT seguiu sem maiores problemas, criando o formulário de cadastro, verificação de

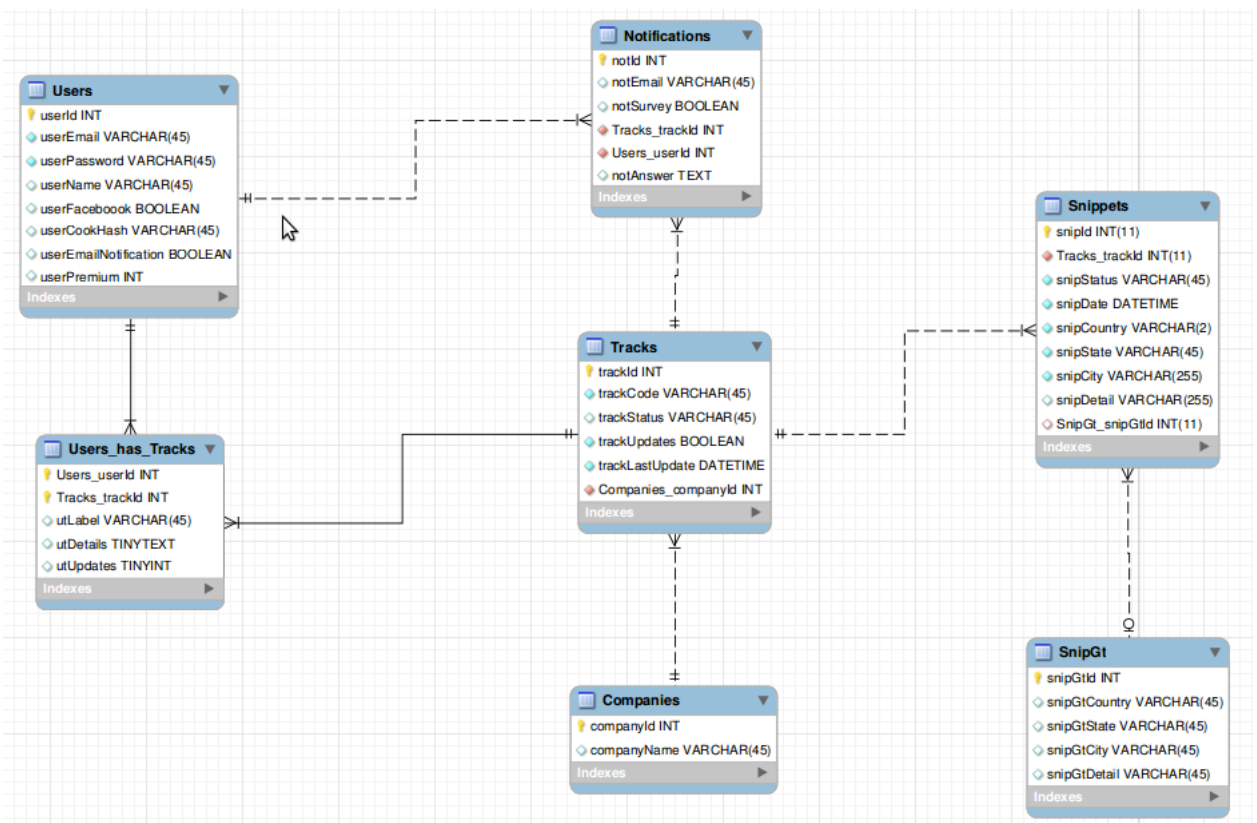


Figura 5: Modelagem do Banco de dados Relacional

usuário já existente, aplicação de hash na senha, inserção no banco de dados. A adoção de sessões foi definida para manter o usuário autenticado.

#### 4.2.5 SPRINT - Otimização de consultas

O MySQL possui um otimizador de consultas, facilitando o desenvolvimento, mas nem sempre as consultas que criamos são otimizadas de maneira eficiente. Para analisar como uma consulta foi otimizada por ele, podemos simplesmente incluir as palavras "explain extended" na frente da consulta que o MySQL vai nos mostrar que índices utilizou, ordem das ações, etc. Para análise de eficiência das consultas foi feita uma pequena modificação no ezSQL (biblioteca que trata as interações com o banco de dados), esta modificação faz com que o sistema, quando em modo debug, execute automaticamente as queries de uma certa página e em seguida execute-a com as palavras "explain extended" na frente. O ezSQL trata de exibir os debugs de maneira amigável, tornando assim a verificação de eficiência mais fácil.

Esta verificação pode concluir que as consultas deste trabalho estão otimizadas especialmente em um caso não trivial onde dois LEFT JOINS são utilizados. O resultado



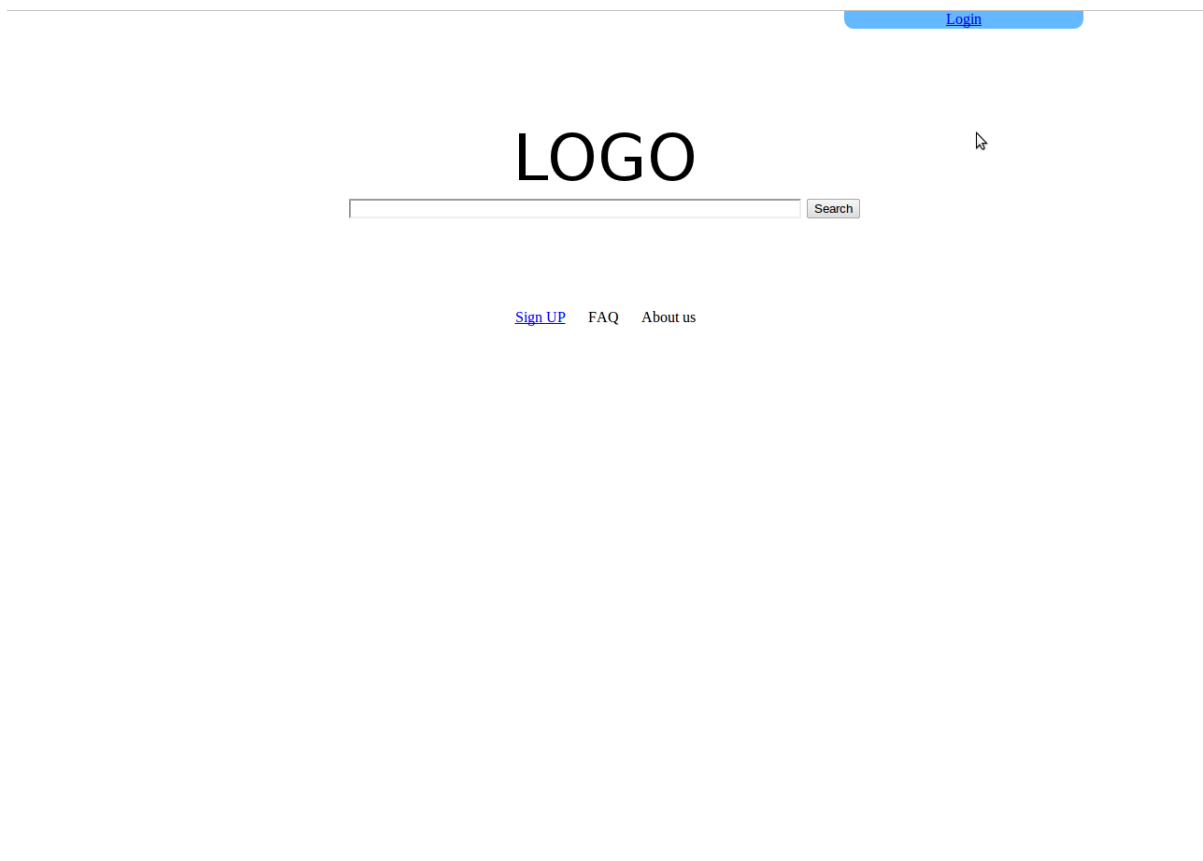


Figura 6: Página inicial do Sistema

pode ser visualizado na Tabela 1 e Tabela 2.

Tabela 1: Tabela demonstrando verificação de eficiência de consulta (parte 1)

ID	select_type	table	type	possible_keys
1	SIMPLE	Users_has_Tracks	ref	PRIMARY, fk_Users_has_Tracks_Users1
2	SIMPLE	Tracks	eq_ref	PRIMARY
3	SIMPLE	Companies	eq_ref	PRIMARY

Nesta tabela cada coluna possui um significado. A coluna "select\_type" representa o tipo de SELECT utilizado. A coluna "table" representa sobre qual tabela os dados são buscados. A coluna "type" define o tipo de join utilizado e é um dos campos com maior influência. A coluna "possible\_keys" representa os índices que podem ser utilizados para a consulta e a coluna "key" qual índice que foi utilizado já a coluna "key\_len" é o tamanho da chave utilizada. A coluna "ref" por sua vez indica quais colunas ou constantes são comparadas para obter o resultado esperado e a coluna "rows" demonstra quantos resultados o MySQL acredita que deve verificar para comparar os resultados. [Mysql Explain Output]

A coluna de maior importância neste caso é a coluna "type" que pode ser visualizada

Tabela 2: Tabela demonstrando verificação de eficiência de consulta (parte2)

ID	key	key_len	ref	rows	key	extra
1	PRIMARY	4	const	4	PRIMARY	temp; filesort
2	PRIMARY	4	Users_has_Tracks.Tracks_trackId	1	PRIMARY	
3	PRIMARY	4	Tracks.Companies_companyId	1	PRIMARY	

na Tabela 1 e pode possuir os seguintes valores:

- System - A tabela possui somente uma linha
- Const - A tabela possui no máximo um item a ser identificado, é muito rápido pois utiliza chave primária ou chave única
- Eq\_ref - Pode ser utilizado em itens indexadas que são comparadas utilizando o operador de igualdade
- Ref - É utilizado quando se trabalha sobre itens que não são chaves primárias ou únicas e trabalham sobre os operadores de igualdade ou desigualdade
- Outros

Os valores demonstrados são os mais eficientes para junção de tabelas. Maiores informações sobre as opções de valores para cada coluna podem ser encontrados em [Mysql Explain Output]

#### 4.2.6 SPRINT - Desenvolvimento - 3

Os requisitos a serem realizados nesta etapa foram "Notificação via e-mail da mudança de status" e "Sistema de etiquetas para facilitar a compreensão do que cada código de rastreio realmente é". A inclusão de notificação via e-mail fez com fosse necessário a criação de atividades cronometradas no servidor, garantindo dessa forma que as verificações de mudanças nos status sejam executadas. A inclusão do sistema de etiquetas não necessitou de grandes modificações na parte gráfica do sistema, um novo botão para alteração da etiqueta encomenda foi incluída ao lado do botão de excluir o código de rastreio. A interface gráfica da página do usuário pode ser vista na Figura 7

Na Figura 7 temos o campo para busca por códigos de rastreio, ao seu lado o menu do usuário e na parte central uma listagem com os códigos de rastreio do usuário. Esses códigos de rastreio podem ser expandidos como demonstrado no primeiro item da tabela. A coluna código de rastreio pode receber uma etiqueta e quando uma encomenda possuir

Código de Rastreio	Status	Última verificação	Compania
<input type="checkbox"/> <a href="#">Teste</a>	Entregue	17/05/2012 17:00	Correios
<p><b>Link Oficial</b> <a href="#">Correios</a></p> <p><b>Status</b> Entregue</p> <p><b>Enviado de</b> CURITIBA - PR - BR</p> <p><b>Data de envio</b> 16/05/12 16:09:00</p> <p><b>Última Localização em</b> FLORIANOPOLIS - SC - BR 17/05/12 15:31:00</p>			
<input checked="" type="checkbox"/> SW393928283BR	Entregue	23/04/2012 19:19	Correios
<input checked="" type="checkbox"/> Enc	Entregue	22/04/2012 17:57	Correios
<input checked="" type="checkbox"/> TESTE	Entregue	11/04/2012 16:26	Correios
<input checked="" type="checkbox"/> SW288083802BR	Entregue	06/03/2012 18:59	Correios
<input checked="" type="checkbox"/> SW260423280BR	Entregue	22/05/2011 16:55	Correios

Figura 7: Página de Usuário logado no Sistema

uma etiqueta, a etiqueta é mostrado ao invés do código de rastreio. O botão na direita com um quadro e um lápis permite a edição da etiqueta e o botão vermelho com um sinal de menos dentro indica a remoção do código de rastreio. Quando um item é expandido os detalhes do mesmo são exibidos, link para o sistema oficial dos correios, último status, quando o mesmo foi enviado e quando foi a última atualização, também é possível visualizar no mapa a representação gráfica de onde a encomenda passou.

#### 4.2.7 SPRINT - Análise de satisfação do cliente

Nesta etapa deixamos de focar um pouco no desenvolvimento para fazer uma análise mais profunda da satisfação do cliente e garantir que todas as funções e funcionalidades estão retornando o esperado (pelo sistema e pelo cliente). O cliente foi questionado sobre as funcionalidades de maior importância como o acompanhamento de status, notificação via e-mail e etiquetas. Todas as funcionalidades estão de acordo com o que o cliente deseja, porém um comportamento estranho foi encontrado quando uma encomenda específica teve seu status atualizado para Conferido. O erro pode ser visualizado na Figura 8.

Verificando o sistema novamente foi constatado que este erro ocorreu porque em nosso sistema de internacionalização não existe a palavra Conferido como status, o que fazia o sistema tentar pesquisar qual o equivalente desse status e não encontrar, retornando

<b>Link Oficial</b>	<a href="#">Correios</a>
<b>Status</b>	Project-Id-Version: Keep-track Report-Msgid-Bugs-To: POT- Creation-Date: 2012-03-06 22:33-0300 PO-Revision-Date: 2012- 04-11 14:35-0300 Last-Translator: Language-Team: Language: MIME-Version: 1.0 Content-Type: text/plain; charset=UTF-8 Content-Transfer-Encoding: 8bit X-Poedit-KeywordsList: _gettext,gettext_noop X-Poedit-SourceCharset: utf-8 X-Poedit- Basepath: . X-Poedit-Language: Portuguese X-Poedit-Country: BRAZIL X-Poedit-SearchPath-0: .
<b>Enviado de</b>	BLUMENAU - SC - BR
<b>Data de envio</b>	04/05/12 16:15:00
<b>Última Localização em</b>	IPORA - PR - BR 11/05/12 12:24:00

Figura 8: Erro durante tentativa de visualização de código de rastreio com status Confe-  
rido

assim o erro demonstrado na figura anterior. Este status nunca havia sido utilizado. Para reduzir as chances de erros como esse ocorram, quando o sistema detecta um Status inexistente os administradores são automaticamente avisados via e-mail.

#### 4.2.8 SPRINT - Desenvolvimento - 4

Os requisitos abordados nesta fase do desenvolvimento foram "Adição de e-mails secundários para pesquisa de satisfação do cliente" e "Repassar as informações de satisfação". A implementação dos mesmos aconteceu sem maiores problemas. Com estes requisitos implementados temos o objetivo geral e os específicos alcançados.

Para tornar o sistema ainda mais funcional, no arquivo de configurações do mesmo é possível editar as perguntas que o sistema vai fazer ao cliente quando este receber a pesquisa de satisfação por e-mail. Não importa o número de perguntas o sistema consegue tratar e armazenar as respostas de forma dinâmica, serializando as respostas de acordo com a sequência de perguntas. Para implementar essa funcionalidade de armazenamento foi adicionado um campo na tabela *Notifications* que permite salvar os resultados de cada pesquisa de satisfação de cada cliente.

Como um extra também foi desenvolvida uma interface simplória de estatísticas que pode ser visualizada na Figura 9. Esta, por sua vez, demonstra de maneira gráfica o

percentual que cada opção de resposta recebeu. A criação de imagens fica por conta do Google Chart Tools.

O desenvolvimento da interface de estatísticas não foi muito a fundo já que este não era o objetivo deste trabalho, porém o sistema simples foi adicionado para dar uma idéia de outras funcionalidades que podem ser acopladas ao sistema. O sistema Google Chart Tools é bastante complexo e permite a criação de diversos tipos de gráficos, porém a explicação do mesmo foge do escopo deste trabalho.

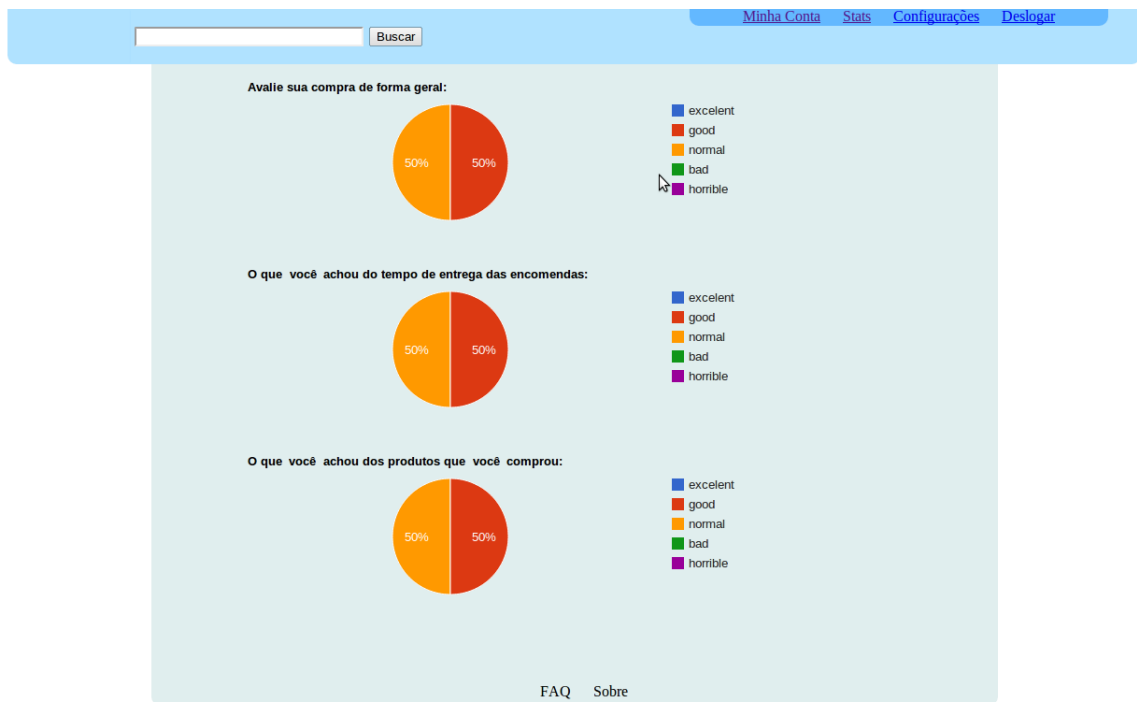


Figura 9: Exemplo de estatísticas.

#### 4.2.9 SPRINT - Ajustes finais

Esta etapa consiste em realizar ajustes finos no sistema. Por exemplo, modificação do sistema de escolha de língua para o arquivo de configurações, possibilitando o usuário definir a língua padrão assim como as línguas que estão presentes no sistema. Os ícones foram definidos no arquivo de configuração, possibilitando que um administrador altere os logotipos de maneira fácil e rápida. Esta também realizou a tradução das palavras faltantes do Inglês para o português.

Diversos outros ajustes foram realizados, como remoção de mensagem de Atenção quando um usuário não possuía nenhum usuário para notificação, adição de novos status

nos sistemas dos Correios como por exemplo "Destinatário Ausente", inclusão de uma configuração para definição do logo do site.

Também foram realizadas modificações na parte de geração de estatísticas para exibir o total de respostas que uma determinada pergunta recebeu assim permitindo que caso o usuário inclua uma nova pergunta depois que o sistema já esta em funcionamento as estatísticas ainda sejam válidas, já que, o total de cada pergunta é exibido.

### 4.3 Visão Geral

Com todas as SPRINTS concluídas temos o sistema pronto para utilização. Na figura 10 podemos verificar como é a página inicial do sistema. A mesma tenta manter um visual limpo e possuir as principais funções do sistema nela, que seria a busca por um código assim como o formulário de autenticação de usuário que pode ser visualizado na figura 11.

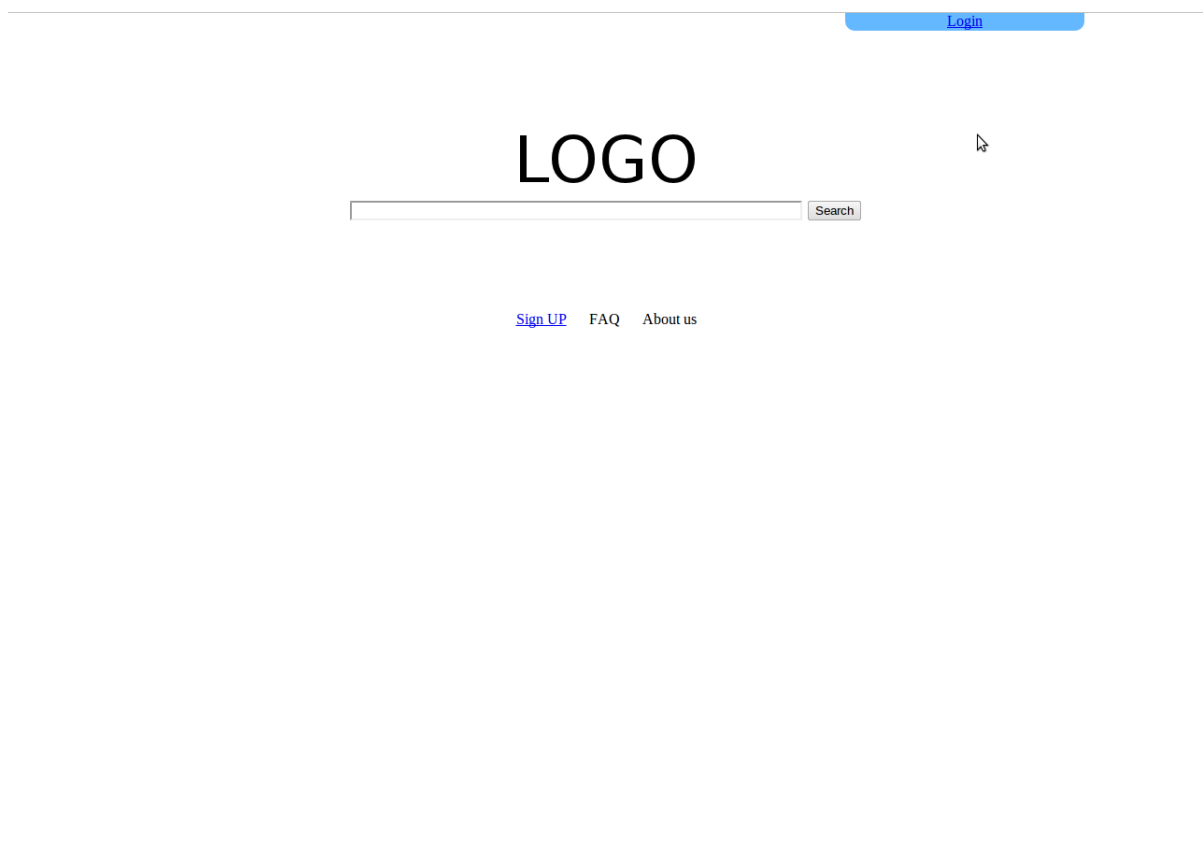


Figura 10: Página inicial do Sistema

Em seguida o usuário autenticado é direcionado a sua página inicial, que pode ser visualizada na Figura 12. Esta página possui os códigos de rastreamento de um usuário que

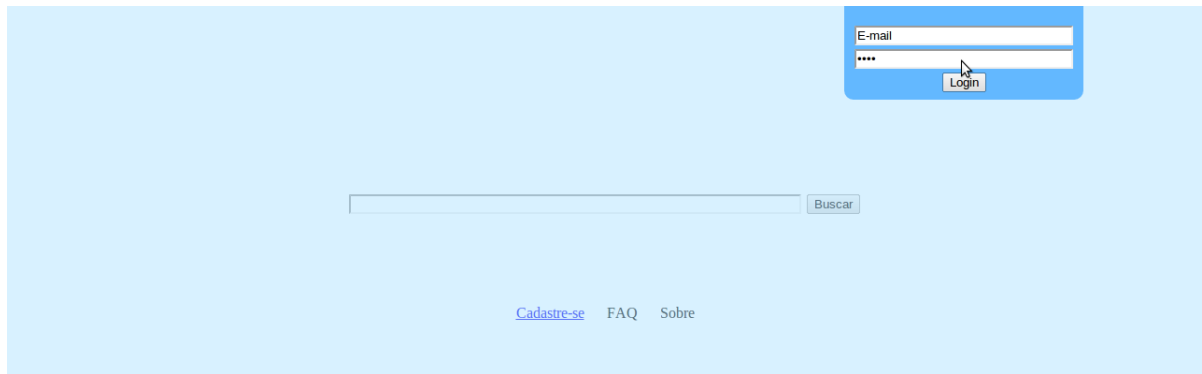


Figura 11: Página com autenticação do usuário

por padrão são exibidos minimizados e podem ser maximizados com um duplo clique na linha do mesmo, ou utilizando a seta de expansão na esquerda. Ela também permite que o usuário remova um código de rastreio no botão vermelho e também edite a etiqueta que está relacionada a um código de rastreio utilizando o icone do lápis.

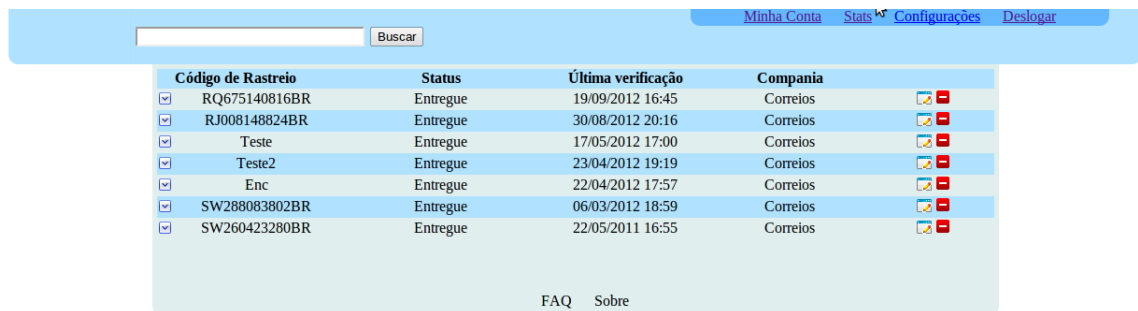


Figura 12: Página de Usuário logado no Sistema

Nesta página o usuário também encontra os Links para sua página inicial, estatísticas de sua conta, suas configurações e também a opção de se fechar sua autenticação. Seguindo para sua página de estatísticas, que pode ser visualizada na Figura 13, ele pode visualizar graficamente as respostas das suas pesquisas de satisfações, e a partir delas, pode elaborar o que deve ser melhorado na sua empresa.

Caso o mesmo deseje modificar suas configurações ele pode utilizar do link configurações e então será redirecionado para a Figura 14 onde ele possui as opções de modificar seu e-mail, sua senha, seu nome de usuário.

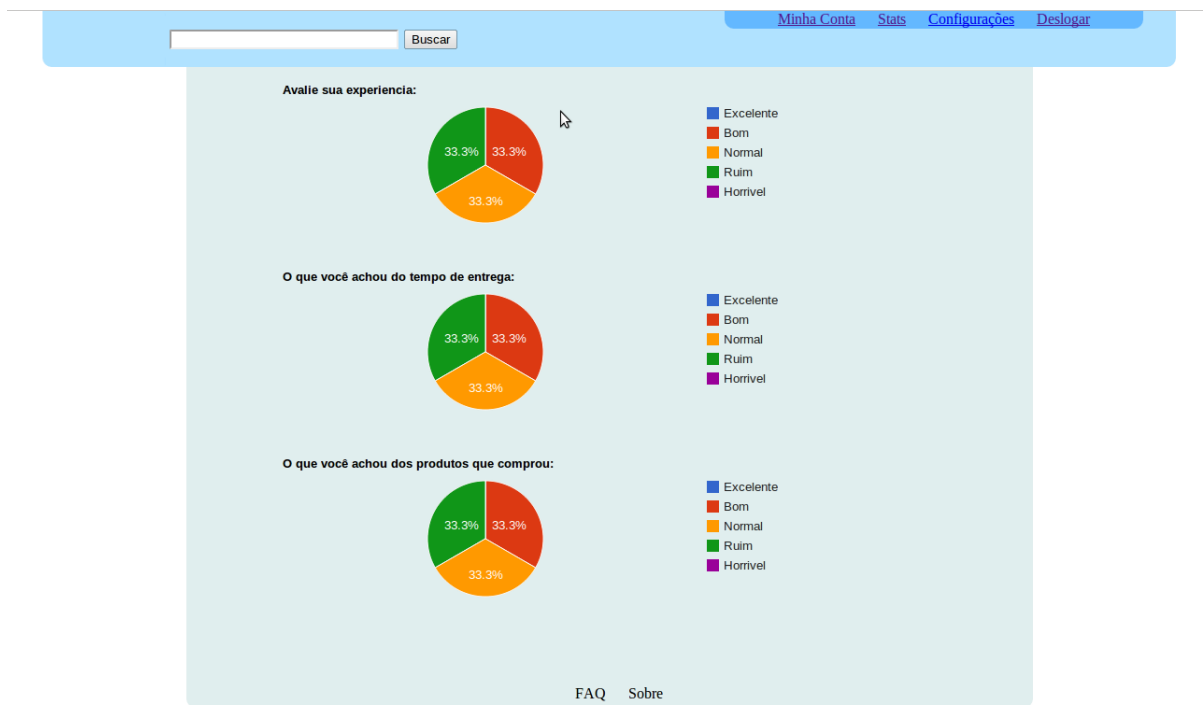


Figura 13: Exemplo de estatísticas.

The configuration page includes the following fields and elements:

- E-mail:** matheusbrat@gmail.com
- Senha:** [Redacted]
- Confirmação da Senha:** [Redacted]
- Nome de Usuário:** matheusbrat
- Senha Anterior:** [Redacted]
- Button:** Atualizar Informações
- Links:** FAQ, Sobre

Figura 14: Página de configurações do usuário

## 4.4 Análise de um caso de uso

### 4.4.1 Informações sobre a empresa

A empresa que está como nosso cliente e é a usuária do sistema é um empresa de pequeno porte que trabalha com Moda Infantil, Feminina e Masculina. O sistema torna-se necessário já que a mesma possui uma loja online onde vende seus produtos e realiza acompanhamento manual de suas vendas.



#### **4.4.2 Análise anterior a utilização do software**

Em contato com um dos responsáveis pelas vendas online foi verificado como o processo de acompanhamento de vendas é feito na empresa antes da utilização do Software proposto neste trabalho. Este contato realiza o cadastro do código de rastreio das encomendas manualmente, adicionando este a uma tabela no Excel que é responsável por manter o registro de vendas da empresa. Os produtos que foram vendidos recentemente são monitorados de maneira manual. Quando o funcionário responsável pelo sistema verifica que a encomenda foi entregue, ele entra em contato com o comprador para obter algumas informações sobre a venda, se os produtos que chegaram estão certos, se o tempo de entrega foi o correto, entre outras perguntas. Por este processo ser feito manualmente, não é um processo rigoroso, então não são todas as vendas que são acompanhadas para analisar a satisfação do cliente. Com isso, interpretações mais precisas sobre a satisfação não são feitas e o processo de venda da empresa pode estar deixando de ser aperfeiçoado.

#### **4.4.3 Análise posterior a utilização do software**

Novamente o responsável pela verificação e manutenção dos códigos de rastreio e acompanhamento pós venda foi consultado para verificar se o sistema supria as necessidades dele e se existiu diferença considerável com a utilização do mesmo. Este relatou que o sistema é muito prático e tem auxiliado bastante, destacou que a funcionalidade de etiquetas é muito eficiente já que dessa maneira ele pode padronizar as etiquetas conseguindo facilmente rastrear o número do pedido. Mesmo com o volume de vendas abaixo da média deles, este constatou que o mesmo tem um saldo positivo na manutenção deste processo interno da empresa. O mesmo também ressaltou que os clientes ficam bastante satisfeitos com a eficácia que o acompanhamento pós venda é feito, o que só é possível por causa do sistema. Durante o período de testes, o servidor ficou fora um dia para manutenção e o responsável não foi notificado previamente, e por consequência passou um dia às cegas. Ele frisou que neste dia sentiu bastante falta do sistema.

## ***5 Conclusão e Trabalhos futuros***

### **5.1 Conclusão**

O sistema foi desenvolvido como planejado, as funcionalidades previstas foram implementadas e também alguns extras como a exibição de gráficos demonstrando as estatísticas de respostas dos clientes. Analisando as diferenças entre o comentário anterior a utilização do sistema e posterior do sistema pelo nosso "cliente", podemos perceber que o mesmo causou impacto positivo e que as funcionalidades estão de acordo com o que ele desejava. A utilidade do mesmo é ressaltada quando o mesmo relata que no dia que o sistema ficou fora, ele sentiu falta da praticidade e das funcionalidades que o sistema proporciona.

### **5.2 Trabalhos Futuros**

Analisando o sistema de maneira geral, podemos visualizar algumas modificações que poderiam aperfeiçoar o sistema, tais quais:

- Criação de itens de estatísticas mais detalhados, com opção de filtros por data, etc.
- Criação de itens de estatísticas que demonstre o tempo de entrega.
- Integração com ERPs clássicos.
- Criação de uma API, permitindo que outros serviços se conectem a este.

## *Referências*

- AGILE Manifesto. Website. Disponível em: <<http://www.agilemanifesto.org/>>.
- ANYWHERE ECT - Integração com os Correios. Website. Disponível em: <<http://www.ledcorp.com.br/category/ECT-Correios.aspx>>.
- COHEN, M. L. D.; COSTA, P. Agile software development.
- DONEY, P. M.; CANNON, J. P. An examination of the nature of trust in buyer-seller relationship. *Journal of Marketing*, v. 61, 1997.
- DWYER, P. H. S. F. R.; OH, S. Developing buyer and seller relationship. *Journal of Marketing*, v. 51, p. 11–27, 1987.
- GETTEXT. Website. Disponível em: <<http://en.wikipedia.org/wiki/Gettext>>.
- JAVASCRIPT. Website. Disponível em: <<http://en.wikipedia.org/wiki/JavaScript>>.
- LEVITT, T. After the sale is over... *Harvard Bussiness Review*, 1983.
- MYSQL. Website. Disponível em: <<http://www.mysql.com/why-mysql/marketshare/>>.
- MYSQL Explain Output. Website. Disponível em: <<http://dev.mysql.com/doc/refman/5.0/en/explain-output.html>>.
- PACKAGETRACKR. Website. Disponível em: <<http://www.packagetrackr.com/about>>.
- PHP. Website. Disponível em: <<http://en.wikipedia.org/wiki/PHP>>.
- SCRUM. Website. Disponível em: <[http://en.wikipedia.org/wiki/Scrum\\_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))>.

## *ANEXO A - Artigo*

# Sistema de Monitoramento de Encomendas

Matheus Bratfisch<sup>1</sup>

<sup>1</sup>Centro Tecnológico – Universidade Federal de Santa Catarina (UFSC)  
Florianópolis – SC – Brazil

matheusbrat@gmail.com

**Abstract.** *Companies sell things over the internet or phone every day, usually this companies believe that the selling process is done when the product is shipped but the market is not like this now. The consumers are becoming more exigent. The internet popularization is creating a new environment where the buyers gives feedbacks about a company, their products and their process. With this in mind we can clearly see that a sell is not ended when the product is shipped, now the companies must listen and keep the consumer close to then. This article brings a solution to help this companies to keep track of the product that they send via mail, help then to get feedback with this information they can improve their intern process and improve their quality.*

**Resumo.** *Empresas vendem coisas pela internet todos os dias, geralmente estas empresas acreditam que o processo de venda esta termiando quando o produto é enviado mas o mercado não é assim agora. Os consumidores estão ficando mais exigente. A popularização da internet esta criando um novo ambiente onde os compradores dão feedback sobre uma empresa. Com isso em mente podemos facilmente verificar que a venda não esta completa quando o produto é pago, agora as empresas devem manter os consumidores próximos. Este artigo traz uma solução para ajudar as empresas a acompanhar os pedidos, ajudando essas empresas a conseguir feedback e com essas informações eles podem melhorar os seus processos internos da empresa.*

## 1. Motivação

A relação entre um vendedor e um comprador não termina quando uma venda é realizada. A relação que se intensifica depois de uma venda vem crescendo e esta determina a escolha do comprador numa próxima oportunidade. A venda, então, meramente consuma o ponto quando o casamento começa. Quão bem este relacionamento é depende em quão bem o vendedor gerencia o seu relacionamento. [Levitt 1983] O que demonstra que o acompanhamento pós venda é de grande importância. Com isso percebemos a necessidade de uma ferramenta que auxilie neste processo novamente tornando o comprador mais satisfeito e possibilitando que o vendedor melhore cada vez mais. Junto a nova relação entre o vendedor/comprador aparece um fator crucial que são as transportadoras, que realizam a entrega do produto comprado. Unindo a nova relação do consumidor e as transportadoras foi constatado a necessidade de um sistema que permita o acompanhamento de encomendas de forma autonoma e que este pode auxiliar a melhorar a relação entre ambos utilizando de uma pesquisa de satisfação.

## 2. Relação Cliente com o Vendedor

As relações entre Cliente e Vendedor, sua qualidade e como evolui-la é um assunto bastante estudado pela área de marketing. Esta muitas vezes propõem modelos a serem seguidos para que a mesma tenha sucesso. Infelizmente maioria das pesquisas e estratégias de marketing tratam a troca entre comprador vendedor como um evento discreto, não como uma relação contínua. [F. Robert Dwyer and Oh 1987] Ou seja esta transação é tratada como um evento único que termina com a concretização da venda. Essas relações contínuas assumem diferentes formas. [F. Robert Dwyer and Oh 1987]

Alguns modelos levantam diversos fatores a serem acompanhados e modificados para que uma relação entre Cliente e Vendedor obtenha sucesso. Um dos fatores citados é a confiança, que também pode ser chamada de credibilidade. Este é formado por diversos fatores como:

- Reputação
- Tamanho
- Características do vendedor
- Tempo do "relacionamento"
- Performance na entrega
- Preço
- Performasse do Produto/Serviço
- Entre outros.

Na Figura 1 é demonstrado a interação entre esses diversos fatores na decisão de escolha do comprador como um diagrama.

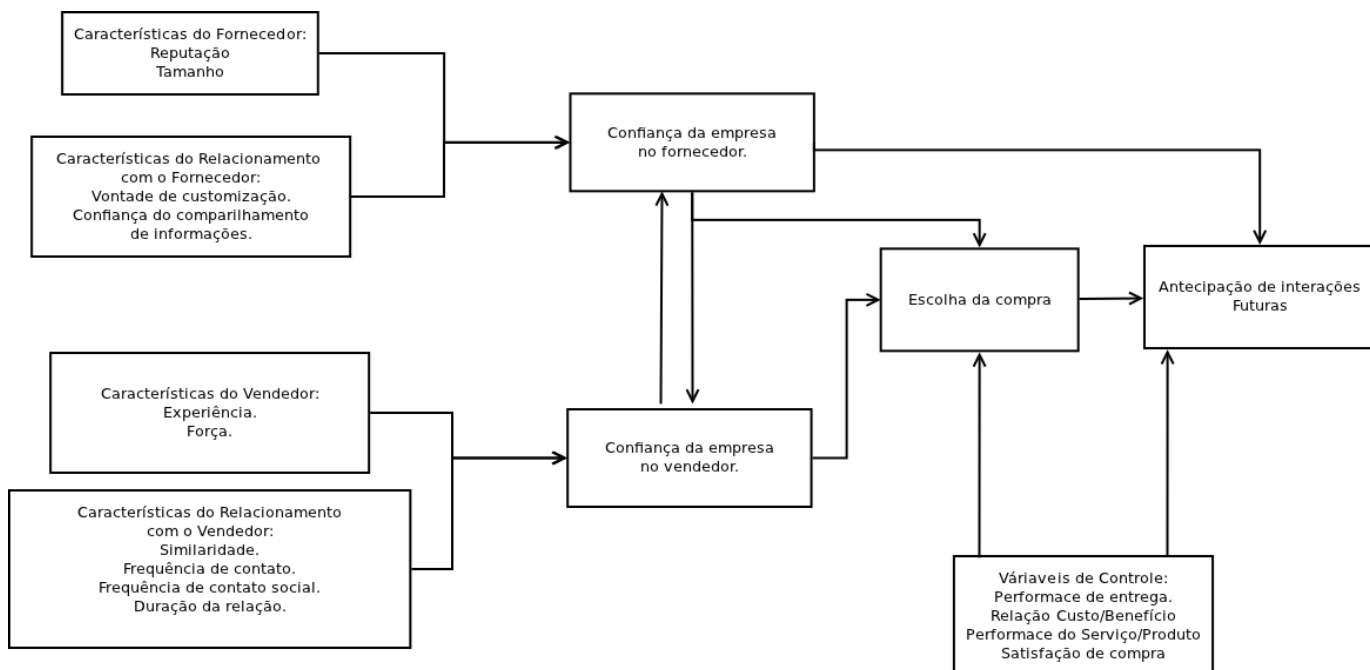


Figure 1. Antecedentes e Conseqüências da confiança numa empresa e no vendedor [Doney and Cannon 1997]

### **3. Análise de soluções existentes**

#### **3.1. PackageTrackr**

Inaugurado em Janeiro de 2008, o PackageTrackr teve como motivação as deficiências das diversas transportadoras. O mesmo é uma solução para monitorar pacotes de maneira automática. Este possui muitas funcionalidades e suporte a diversas transportadoras.

Algumas de suas funcionalidades que assumem mais destaque são:

- Suporte a diversas transportadoras (Fedex, USPS, DHL, Australian Post, Canadian Post, etc)
- Acesso via Smartphone (Android, Iphone)

#### **3.2. Anywhere ECT - Integração com os correios**

A empresa Ledcorp criada a quase 10 anos com especialidade em acessibilidade e comunicação, mobilidade, rastreamento e monitoração de veículos, cargas e pessoas, integração de sistemas e outros. Possui um modulo chamado de Anywhere ECT. Este realiza controle e monitoramento do andamento do processo de entrega de encomendas pelos Correios.

Utilizando este junto a outras ferramentas pode-se obter funcionalidades extras como acompanhar em um mapa a evolução do status das postagens, definir SLA (Service Level Agreement), possibilidade de notificação via SMS, integração com redes sociais.

Ele também pode ser integrado a qualquer ERP de forma direta ou por EDI, atualizando diretamente o status de entrega de forma automatizada.

### **4. Tecnologias utilizadas**

Este sistema foi desenvolvido utilizando de tecnologias para WWW (World Wide Web), algumas delas são, PHP: Hypertext Processor (PHP) que é uma linguagem de servidor para fornecer páginas dinâmicas, HTML (HyperText Markup Language) que é a linguagem de marcação utilizada para definir o local e como serão exibidos os diversos elementos de uma página, JavaScript que é uma linguagem que atua no navegador do cliente que trata de fazer ações sem necessidade de intervenção do servidor, MySQL que é um banco de dados relacional, aberto e atualmente é o banco de dados relacional mais utilizado do mundo [mys ] e para a internacionalização do sistema foi decidido pela utilização da biblioteca gettext que possui suporte junto ao PHP.

### **5. Descrição da Solução.**

Com base nas funcionalidades dos sistemas já existentes e conversas com uma empresa que tem interesse nesse sistema, foi definido uma solução indermediária a ambas e que adicionasse novas funcionalidades. A lista de funcionalidades do sistema é:

- Verificação de forma automatizada do status de encomendas
- Possibilidade de monitoramento de diversas encomendas em paralelo
- Cadastro de usuários que utilizarão o sistema
- Notificação via e-mail da mudança de status
- Sistema de etiquetas para facilitar a compreensão do que cada código de rastreio realmente é.

- Cadastro de e-mails secundários para pesquisa de satisfação do cliente
- Repassar as informações de satisfação ao usuário

A verificação de forma automatizada é realizada pelo sistema de 3 a 6 vezes por dia dependendo dos parâmetros de configuração, as perguntas para pesquisa de satisfação também podem ser configuradas assim como diversas outras opções. O foco do sistema desenvolvido foi nos Correios brasileiros, porém a implementação da busca pelos Correios foi feita de maneira modular, tornando muito simples a adição de novas companhias de rastreamento. Pensando também na diversidade linguística que existe no mundo o sistema é todo localizado, permitindo com que o mesmo seja portado para outros idiomas.

A exibição das informações de localização das encomendas são feitas de forma textual e também de forma gráfica por meio do sistema Google Maps que pode ser visualizado na Figura 2. Permitindo com isso uma fácil compreensão da localização da encomenda por parte do usuário.

Código de Rastreo	Status	Última verificação	Compania
<a href="#">Teste</a>	Entregue	17/05/2012 17:00	Correios
<b>Link Oficial</b>	<a href="#">Correios</a>		
<b>Status</b>	Entregue		
<b>Enviado de</b>	CURITIBA - PR - BR		
<b>Data de envio</b>	16/05/12 16:09:00		
<b>Última Localização em</b>	FLORIANOPOLIS - SC - BR		
	17/05/12 15:31:00		
<input checked="" type="checkbox"/> SW393928283BR	Entregue	23/04/2012 19:19	Correios
<input checked="" type="checkbox"/> Enc	Entregue	22/04/2012 17:57	Correios
<input checked="" type="checkbox"/> TESTE	Entregue	11/04/2012 16:26	Correios
<input checked="" type="checkbox"/> SW288083802BR	Entregue	06/03/2012 18:59	Correios
<input checked="" type="checkbox"/> SW260423280BR	Entregue	22/05/2011 16:55	Correios

FAQ Sobre

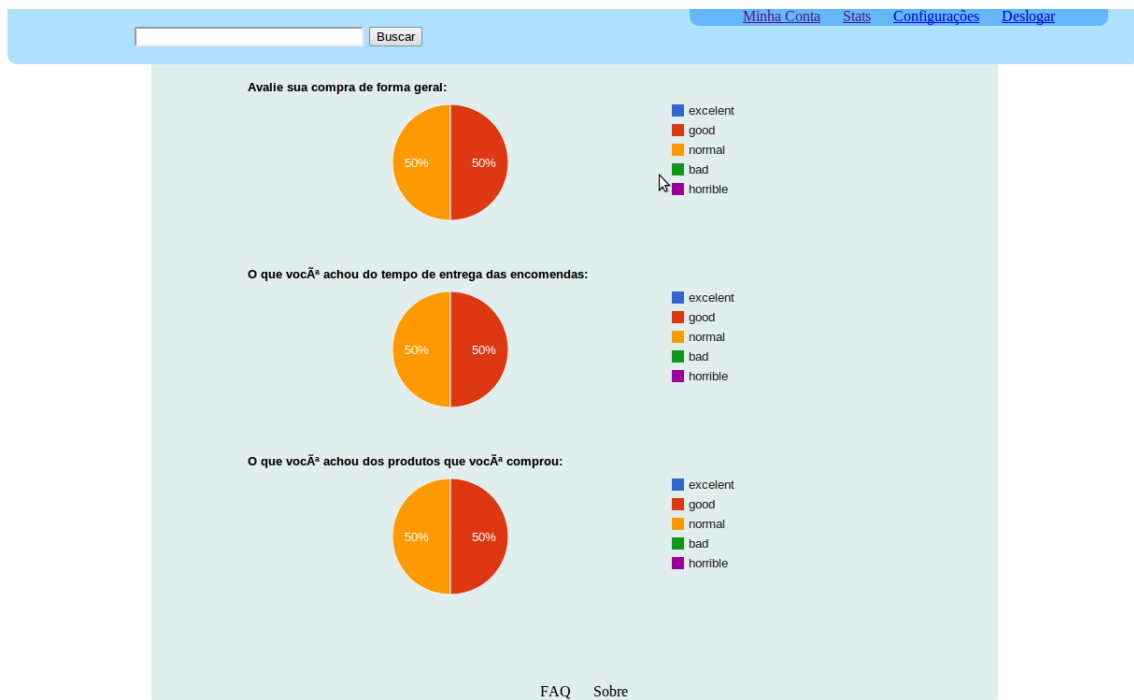
**Figure 2. Página de Usuário logado no Sistema**

As respostas da pesquisa de satisfação são armazenadas utilizando serialização já que as mesmas são configuráveis, o sistema também possui uma interface simplista de exibição das estatísticas que pode ser visualizada na Figura 3 deixando para trabalhos futuros a ampliação das informações filtradas com este sistema.

## 6. Conclusões e Trabalhos Futuros

O sistema aqui descrito foi utilizado por uma empresa que vende online mas não realiza uma grande quantidade de vendas, porém o responsável pela manutenção e acompanhamento dessas vendas sentiu diferença na utilização do sistema, de maneira positiva. Com isso é comprovado que o mesmo possui utilidade e é de fácil uso mesmo para empresas pequenas, provavelmente em cenários onde as empresas possuem volume de venda





**Figure 3. Exemplo de estatísticas.**

maior os ganhos também serão maiores. Como trabalhos futuros existem diversas opções, como desenvolvimento de uma API, integração com ERPs clássicos, análise do sistema em outros cenários comprovando ou negando a eficiência do mesmo, entre outros.

## References

Mysql. Website.

Doney, P. M. and Cannon, J. P. (1997). An examination of the nature of trust in buyer-seller relationship. *Journal of Marketing*, 61.

F. Robert Dwyer, P. H. S. and Oh, S. (1987). Developing buyer and seller relationship. *Journal of Marketing*, 51:11–27.

Levitt, T. (1983). After the sale is over... *Harvard Business Review*.

```
/home/x-warrior/Desktop/Keep-Track/admin/autoupdater.php
```

```
<?php
include("../config.php");
include_once "../includes/ez_sql_core.php";
include_once "../includes/ez_sql_mysql.php";
include_once "../functions.php";
unregisterGLOBALS();

setupLocale("../");

$db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
$db->query("SET CHARACTER SET utf8 ");

$result = $db->get_results("select trackCode from Tracks where trackUpdates = 1");
foreach ($result as $tc) {
    search($tc->trackCode);
}
?>
```

```
/home/x-warrior/Desktop/Keep-Track/config.php
```

```
<?php
define('LOGO', 'images/keeptrack.png');
define('LOGO_TOP', 'images/keeptrack-top.png');
define('DELIVERED_STR', 'Delivered');
define('CHECKED_STR', 'Checked');
define('OUT_FOR_DELIVERY_STR', 'Out for delivery');
define('POSTED_STR', 'Posted');
define('ON_WAY_STR', 'On way');
define('WAITING_STR', 'Ready for Pickup');
define('RECEIVER_NOT_HOME', 'Receiver not at home');
_('Delivered');
_('Out for delivery');
_('Posted');
_('On way');
_('Ready for Pickup');
_('Checked');
_('Receiver not at home');

$dbName = "matbrac_ktrack";
$dbUser = "matbrac_sandbox";
$dbPass = "Nhttpv-%HUapZ";
$dbHost = "localhost";
$salt = "!ad$%"; // NEVER CHANGE PLEASE!

$PERGUNTAS[0] = _("Rate your experience");
$PERGUNTAS[1] = _("What do you think about delivery time:");
$PERGUNTAS[2] = _("What do you think about the products that you bought:");

$locales[0] = Array("name" => "pt_BR", "icon" => "images/icons/br.png");
$locales[1] = Array("name" => "en_US", "icon" => "images/icons/us.png");
```

```
/home/x-warrior/Desktop/Keep-Track/delete.php
```

```
<?php
@session_start();
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";

if($db == null) {
```

```

$db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
$db->query("SET CHARACTER SET utf8 ");
}

if (isLoggedIn()) {
    $userId = secureDataToDB($_SESSION['userId']);
    $trackId = secureDataToDB($_GET['track']);
    if (!empty($trackId) && !empty($userId)) {
        $db->query("DELETE FROM `Users_has_Tracks` WHERE `Users_userId`='$userId'
and `Tracks_trackId`='$trackId'");
        $db->query("DELETE FROM `Notifications` WHERE `Users_userId`='$userId'
and `Tracks_trackId`='$trackId'");
        if ($_GET['direct'] == true) {
            if ($_SERVER[HTTP_REFERER]) {
                header("Location: ".$_SERVER[HTTP_REFERER]);
            }
        }
    }
}
}

```

/home/x-warrior/Desktop/Keep-Track/edit.php

```

<?php
@session_start();
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";

if($db == null) {
    $db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
    $db->query("SET CHARACTER SET utf8 ");
}

if (isLoggedIn()) {
    //$userId = secureDataToDB($_GET['u']);
    $userId = $_SESSION['userId'];
    $trackId = secureDataToDB($_GET['t']);
    $trackCode = secureDataToDB($_GET['tc']);
    $label = trim(secureDataToDB($_GET['l']));
    if (!empty($trackId) && !empty($userId)) {
        if(empty($label) || is_null($label) || $label == "Label" || $label == "") {
            $db->query("update `Users_has_Tracks` set `utLabel` = null where Users_userId = '$userId' and
Tracks_trackId='$trackId'");
            echo $trackCode;
        } else {
            $labeldb = $label;
            $db->query("update `Users_has_Tracks` set `utLabel` = '$labeldb' where Users_userId =
'$userId' and Tracks_trackId='$trackId'");
            if($db->rows_affected == 1) {
                echo "$label";
            } else {
                echo $trackCode;
            }
        }
    }
}
}
}

```

/home/x-warrior/Desktop/Keep-Track/functions.php

```

<?php
include_once 'includes/system/Correios.php';
include_once 'includes/system/Fedex.php';

```

```
// UTIL
include("includes/functions/utilFunctions.php");

// GRAPHIC
include("includes/functions/graphicFunctions.php");

// USER
include("includes/functions/userFunctions.php");

// COMPANY
include("includes/functions/companyFunctions.php");

// TRACKS
include("includes/functions/tracksFunctions.php");

// SEARCHER
include("includes/functions/searchFunctions.php");

// Notification
include("includes/functions/notificationFunctions.php");
?>
```

/home/x-warrior/Desktop/Keep-Track/head.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<title></title>
</head>
<script type="text/javascript" src="includes/js/jquery-1.5.2.js"></script>
<script type="text/javascript" src="includes/js/jquery.spotlight.js"></script>
<script type="text/javascript" src="includes/js/jquery-ui-1.8.12.custom.min.js"></script>
<link href="css/cupertino/jquery-ui-1.8.12.custom.css" rel="stylesheet" type="text/css">
```

/home/x-warrior/Desktop/Keep-Track/includes/ez\_sql\_core.php

<?php

```

/*****
* Author: Justin Vincent (jv@jvmultimedia.com)
* Web...: http://twitter.com/justinvincent
* Name...: ezSQL
* Desc...: ezSQL Core module - database abstraction library to make
*          it very easy to deal with databases.
*
*/

/*****
* ezSQL Constants
*/

define('EZSQL_VERSION','2.03');
define('OBJECT','OBJECT',true);
define('ARRAY_A','ARRAY_A',true);
define('ARRAY_N','ARRAY_N',true);
define('EZSQL_CORE_ERROR','ezSQLcore can not be used by itself (it is designed for use by database
specific modules).');

/*****
```

```
* Core class containg common functions to manipulate query result
* sets once returned
*/

class ezSQLcore
{
    var $trace                = false; // same as $debug_all
    var $debug_all           = false; // same as $trace
    var $debug_called        = false;
    var $vardump_called      = false;
    var $show_errors         = true;
    var $num_queries         = 0;
    var $last_query          = null;
    var $last_error          = null;
    var $col_info            = null;
    var $captured_errors     = array();
    var $cache_dir           = false;
    var $cache_queries       = false;
    var $cache_inserts       = false;
    var $use_disk_cache      = false;
    var $cache_timeout       = 24; // hours

    // == TJH == default now needed for echo of debug function
    var $debug_echo_is_on = true;

    /*****
    * Constructor
    */

    function ezSQLcore()
    {
    }

    /*****
    * Print SQL/DB error - over-ridden by specific DB class
    */

    function register_error($err_str)
    {
        // Keep track of last error
        $this->last_error = $err_str;

        // Capture all errors to an error array no matter what happens
        $this->captured_errors[] = array
        (
            'error_str' => $err_str,
            'query'      => $this->last_query
        );
    }

    /*****
    * Turn error handling on or off..
    */

    function show_errors()
    {
        $this->show_errors = true;
    }

    function hide_errors()
    {
        $this->show_errors = false;
    }

    /*****
    * Kill cached query results
    */
}
```

```

*/

function flush()
{
    // Get rid of these
    $this->last_result = null;
    $this->col_info = null;
    $this->last_query = null;
    $this->from_disk_cache = false;
}

/*****
* Get one variable from the DB - see docs for more detail
*/

function get_var($query=null,$x=0,$y=0)
{
    // Log how the function was called
    $this->func_call = "\$db->get_var(\"$query\",$x,$y)";

    // If there is a query then perform it if not then use cached results..
    if ( $query )
    {
        $this->query($query);
    }

    // Extract var out of cached results based x,y vals
    if ( $this->last_result[$y] )
    {
        $values = array_values(get_object_vars($this->last_result[$y]));
    }

    // If there is a value return it else return null
    return (isset($values[$x]) && $values[$x]!='')?$values[$x]:null;
}

/*****
* Get one row from the DB - see docs for more detail
*/

function get_row($query=null,$output=OBJECT,$y=0)
{
    // Log how the function was called
    $this->func_call = "\$db->get_row(\"$query\",$output,$y)";

    // If there is a query then perform it if not then use cached results..
    if ( $query )
    {
        $this->query($query);
    }

    // If the output is an object then return object using the row offset..
    if ( $output == OBJECT )
    {
        return $this->last_result[$y]?$this->last_result[$y]:null;
    }
    // If the output is an associative array then return row as such..
    elseif ( $output == ARRAY_A )
    {
        return $this->last_result[$y]?get_object_vars($this->last_result[$y]):null;
    }
    // If the output is an numerical array then return row as such..
    elseif ( $output == ARRAY_N )
    {
        return $this->last_result[$y]?array_values(get_object_vars($this->last_result[$y])):null;
    }
}

```

```

    }
    // If invalid output type was specified..
    else
    {
        $this->print_error(" \${db->get_row(string query, output type, int offset) -- Output type
must be one of: OBJECT, ARRAY_A, ARRAY_N");
    }
}

/*****
* Function to get 1 column from the cached result set based in X index
* see docs for usage and info
*/

function get_col($query=null,$x=0)
{
    // If there is a query then perform it if not then use cached results..
    if ( $query )
    {
        $this->query($query);
    }

    // Extract the column values
    for ( $i=0; $i < count($this->last_result); $i++ )
    {
        $new_array[$i] = $this->get_var(null,$x,$i);
    }

    return $new_array;
}

/*****
* Return the the query as a result set - see docs for more details
*/

function get_results($query=null, $output = OBJECT)
{
    // Log how the function was called
    $this->func_call = "\${db->get_results(\"$query\", $output)";

    // If there is a query then perform it if not then use cached results..
    if ( $query )
    {
        $this->query($query);
    }

    // Send back array of objects. Each row is an object
    if ( $output == OBJECT )
    {
        return $this->last_result;
    }
    elseif ( $output == ARRAY_A || $output == ARRAY_N )
    {
        if ( $this->last_result )
        {
            $i=0;
            foreach( $this->last_result as $row )
            {
                $new_array[$i] = get_object_vars($row);

                if ( $output == ARRAY_N )
                {

```

```

        $new_array[$i] = array_values($new_array[$i]);
    }

    $i++;
}

return $new_array;
}
else
{
    return null;
}
}
}

```

```

/*****
* Function to get column meta data info pertaining to the last query
* see docs for more info and usage
*/

```

```

function get_col_info($info_type="name",$col_offset=-1)
{
    if ( $this->col_info )
    {
        if ( $col_offset == -1 )
        {
            $i=0;
            foreach($this->col_info as $col )
            {
                $new_array[$i] = $col->{$info_type};
                $i++;
            }
            return $new_array;
        }
        else
        {
            return $this->col_info[$col_offset]->{$info_type};
        }
    }
}

```

```

/*****
* store_cache
*/

```

```

function store_cache($query,$is_insert)
{
    // The would be cache file for this query
    $cache_file = $this->cache_dir.'/'.md5($query);

    // disk caching of queries
    if ( $this->use_disk_cache && ( $this->cache_queries && ! $is_insert ) || ( $this->
>cache_inserts && $is_insert ) )
    {
        if ( ! is_dir($this->cache_dir) )
        {
            $this->register_error("Could not open cache dir: $this->cache_dir");
            $this->show_errors ? trigger_error("Could not open cache dir: $this-
>cache_dir",E_USER_WARNING) : null;
        }
        else
        {

```



```

        // Cache all result values
        $result_cache = array
        (
            'col_info' => $this->col_info,
            'last_result' => $this->last_result,
            'num_rows' => $this->num_rows,
            'return_value' => $this->num_rows,
        );
        error_log ( serialize($result_cache), 3, $cache_file);
    }
}

/*****
* get_cache
*/

function get_cache($query)
{
    // The would be cache file for this query
    $cache_file = $this->cache_dir.'/'.md5($query);

    // Try to get previously cached version
    if ( $this->use_disk_cache && file_exists($cache_file) )
    {
        // Only use this cache file if less than 'cache_timeout' (hours)
        if ( (time() - filemtime($cache_file)) > ($this->cache_timeout*3600) )
        {
            unlink($cache_file);
        }
        else
        {
            $result_cache = unserialize(file_get_contents($cache_file));

            $this->col_info = $result_cache['col_info'];
            $this->last_result = $result_cache['last_result'];
            $this->num_rows = $result_cache['num_rows'];

            $this->from_disk_cache = true;

            // If debug ALL queries
            $this->trace || $this->debug_all ? $this->debug() : null ;

            return $result_cache['return_value'];
        }
    }
}

/*****
* Dumps the contents of any input variable to screen in a nicely
* formatted and easy to understand way - any type: Object, Var or Array
*/

function vardump($mixed='')
{
    // Start outup buffering
    ob_start();

    echo "<p><table><tr><td bgcolor=ffffff><blockquote><font color=000090>";
    echo "<pre><font face=arial>";

    if ( ! $this->vardump_called )
    {

```

```

        echo "<font color=800080><b>ezSQL</b> (v".EZSQL_VERSION.") <b>Variable Dump..</b></font>\n
\n";
    }

    $var_type = gettype ($mixed);
    print_r(($mixed?$mixed:"<font color=red>No Value / False</font>"));
    echo "\n\n<b>Type:</b> " . ucfirst($var_type) . "\n";
    echo "<b>Last Query</b> [${this->num_queries}<b>:</b> " . ($this->last_query?$this->
>last_query:"NULL")."\n";
    echo "<b>Last Function Call:</b> " . ($this->func_call?$this->func_call:"None")."\n";
    echo "<b>Last Rows Returned:</b> ".count($this->last_result)."\n";
    echo "</font></pre></font></blockquote></td></tr></table>". $this->donation();
    echo "\n<hr size=1 noshade color=dddddd>";

    // Stop output buffering and capture debug HTML
    $html = ob_get_contents();
    ob_end_clean();

    // Only echo output if it is turned on
    if ( $this->debug_echo_is_on )
    {
        echo $html;
    }

    $this->vardump_called = true;

    return $html;
}

/*****
* Alias for the above function
*/

function dumpvar($mixed)
{
    $this->vardump($mixed);
}

/*****
* Displays the last query string that was sent to the database & a
* table listing results (if there were any).
* (abstracted into a seperate file to save server overhead).
*/

function debug()
{
    // Start outup buffering
    ob_start();

    echo "<blockquote>";

    // Only show ezSQL credits once..
    if ( ! $this->debug_called )
    {
        echo "<font color=800080 face=arial size=2><b>ezSQL</b> (v".EZSQL_VERSION.") <b>Debug..</
b></font><p>\n";
    }

    if ( $this->last_error )
    {
        echo "<font face=arial size=2 color=000099><b>Last Error --</b> [<font color=000000><b>
$this->last_error</b></font>]<p>";
    }

    if ( $this->from_disk_cache )

```

```

    {
        echo "<font face=arial size=2 color=000099><b>Results retrieved from disk cache</b></font><p>";
    }

    echo "<font face=arial size=2 color=000099><b>Query</b> [\$this->num_queries] <b>--</b> ";
    echo "[<font color=000000><b>\$this->last_query</b></font>]</font><p>";

    echo "<font face=arial size=2 color=000099><b>Query Result..</b></font>";
    echo "<blockquote>";

    if ( \$this->col_info )
    {

        // =====
        // Results top rows

        echo "<table cellpadding=5 cellspacing=1 bgcolor=555555>";
        echo "<tr bgcolor=eeeeee><td nowrap valign=bottom><font color=555599 face=arial size=2><b>
(row)</b></font></td>";

        for ( \$i=0; \$i < count(\$this->col_info); \$i++ )
        {
            echo "<td nowrap align=left valign=top><font size=1 color=555599 face=arial>{\$this-
>col_info[\$i]->type} {\$this->col_info[\$i]->max_length}</font><br><span style='font-family: arial; font-
size: 10pt; font-weight: bold;'>{\$this->col_info[\$i]->name}</span></td>";
        }

        echo "</tr>";

        // =====
        // print main results

    if ( \$this->last_result )
    {

        \$i=0;
        foreach ( \$this->get_results(null,ARRAY_N) as \$one_row )
        {
            \$i++;
            echo "<tr bgcolor=ffffff><td bgcolor=eeeeee nowrap align=middle><font size=2
color=555599 face=arial>\$i</font></td>";

            foreach ( \$one_row as \$item )
            {
                echo "<td nowrap><font face=arial size=2>\$item</font></td>";
            }

            echo "</tr>";
        }

    } // if last result
    else
    {
        echo "<tr bgcolor=ffffff><td colspan=\".(count(\$this->col_info)+1).\"><font face=arial
size=2>No Results</font></td></tr>";
    }

    echo "</table>";

} // if col_info
else
{
    echo "<font face=arial size=2>No Results</font>";
}

```

```

    echo "</blockquote></blockquote>".$this->donation()."<hr noshade color=dddddd size=1>";

    // Stop output buffering and capture debug HTML
    $html = ob_get_contents();
    ob_end_clean();

    // Only echo output if it is turned on
    if ( $this->debug_echo_is_on )
    {
        echo $html;
    }

    $this->debug_called = true;

    $this->query("explain extended " . $this->last_query);

    return $html;
}

/*****
* Naughty little function to ask for some remuniration!
*/

function donation()
{
    return "<font size=1 face=arial color=000000>If ezSQL has helped <a href=\"https://
www.paypal.com/xclick/business=justin%40justinvincent.com&item_name=ezSQL&no_note=1&tax=0\" style=
\"color: 0000CC;\">make a donation!</a> &nbsp;&nbsp;<!--[ go on! you know you want to! ]--></font>";
}

?>

/home/x-warrior/Desktop/Keep-Track/includes/ez_sql_mysql.php

<?php

/*****
* Author: Justin Vincent (jv@jvmultimedia.com)
* Web...: http://twitter.com/justinvincent
* Name..: ezSQL_mysql
* Desc..: mySQL component (part of ezSQL database abstraction library)
*/

/*****
* ezSQL error strings - mySQL
*/

$ezsql_mysql_str = array
(
    1 => 'Require $dbuser and $dbpassword to connect to a database server',
    2 => 'Error establishing mySQL database connection. Correct user/password? Correct hostname?
Database server running?',
    3 => 'Require $dbname to select a database',
    4 => 'mySQL database connection is not active',
    5 => 'Unexpected error while trying to select database'
);

/*****
* ezSQL Database specific class - mySQL
*/

if ( ! function_exists ( 'mysql_connect' ) ) die('<b>Fatal Error:</b> ezSQL_mysql requires mySQL Lib to

```

```

be compiled and or linked in to the PHP engine');
    if ( ! class_exists ( 'ezSQLcore' ) ) die('<b>Fatal Error:</b> ezSQL_mysql requires ezSQLcore
(ez_sql_core.php) to be included/loaded before it can be used');

class ezSQL_mysql extends ezSQLcore
{

    var $dbuser = false;
    var $dbpassword = false;
    var $dbname = false;
    var $dbhost = false;

    /*****
    * Constructor - allow the user to perform a quick connect at the
    * same time as initialising the ezSQL_mysql class
    */

    function ezSQL_mysql($dbuser='', $dbpassword='', $dbname='', $dbhost='localhost')
    {
        $this->dbuser = $dbuser;
        $this->dbpassword = $dbpassword;
        $this->dbname = $dbname;
        $this->dbhost = $dbhost;
    }

    /*****
    * Short hand way to connect to MySQL database server
    * and select a MySQL database at the same time
    */

    function quick_connect($dbuser='', $dbpassword='', $dbname='', $dbhost='localhost')
    {
        $return_val = false;
        if ( ! $this->connect($dbuser, $dbpassword, $dbhost,true) );
        else if ( ! $this->select($dbname) );
        else $return_val = true;
        return $return_val;
    }

    /*****
    * Try to connect to MySQL database server
    */

    function connect($dbuser='', $dbpassword='', $dbhost='localhost')
    {
        global $ezsql_mysql_str; $return_val = false;

        // Must have a user and a password
        if ( ! $dbuser )
        {
            $this->register_error($ezsql_mysql_str[1]. ' in '.__FILE__.' on line '.__LINE__);
            $this->show_errors ? trigger_error($ezsql_mysql_str[1],E_USER_WARNING) : null;
        }
        // Try to establish the server database handle
        else if ( ! $this->dbh = @mysql_connect($dbhost,$dbuser,$dbpassword,true) )
        {
            $this->register_error($ezsql_mysql_str[2]. ' in '.__FILE__.' on line '.__LINE__);
            $this->show_errors ? trigger_error($ezsql_mysql_str[2],E_USER_WARNING) : null;
        }
        else
        {
            $this->dbuser = $dbuser;
            $this->dbpassword = $dbpassword;
            $this->dbhost = $dbhost;
            $return_val = true;
        }
    }
}

```

```

    return $return_val;
}

/*****
* Try to select a MySQL database
*/

function select($dbname='')
{
    global $ezsql_mysql_str; $return_val = false;

    // Must have a database name
    if ( ! $dbname )
    {
        $this->register_error($ezsql_mysql_str[3]. ' in '.__FILE__.' on line '.__LINE__);
        $this->show_errors ? trigger_error($ezsql_mysql_str[3],E_USER_WARNING) : null;
    }

    // Must have an active database connection
    else if ( ! $this->dbh )
    {
        $this->register_error($ezsql_mysql_str[4]. ' in '.__FILE__.' on line '.__LINE__);
        $this->show_errors ? trigger_error($ezsql_mysql_str[4],E_USER_WARNING) : null;
    }

    // Try to connect to the database
    else if ( !@mysql_select_db($dbname,$this->dbh) )
    {
        // Try to get error supplied by mysql if not use our own
        if ( !$str = @mysql_error($this->dbh) )
            $str = $ezsql_mysql_str[5];

        $this->register_error($str.' in '.__FILE__.' on line '.__LINE__);
        $this->show_errors ? trigger_error($str,E_USER_WARNING) : null;
    }
    else
    {
        $this->dbname = $dbname;
        $return_val = true;
    }

    return $return_val;
}

/*****
* Format a MySQL string correctly for safe MySQL insert
* (no mater if magic quotes are on or not)
*/

function escape($str)
{
    return mysql_escape_string(stripslashes($str));
}

/*****
* Return MySQL specific system date syntax
* i.e. Oracle: SYSDATE Mysql: NOW()
*/

function sysdate()
{
    return 'NOW()';
}

/*****
* Perform MySQL query and try to detirmin result value
*/

```

```
function query($query)
{
    // Initialise return
    $return_val = 0;

    // Flush cached values..
    $this->flush();

    // For reg expressions
    $query = trim($query);

    // Log how the function was called
    $this->func_call = "\$db->query(\"$query\")";

    // Keep track of the last query for debug..
    $this->last_query = $query;

    // Count how many queries there have been
    $this->num_queries++;

    // Use core file cache function
    if ( $cache = $this->get_cache($query) )
    {
        return $cache;
    }

    // If there is no existing database connection then try to connect
    if ( ! isset($this->dbh) || ! $this->dbh )
    {
        $this->connect($this->dbuser, $this->dbpassword, $this->dbhost);
        $this->select($this->dbname);
    }

    // Perform the query via std mysql_query function..
    $this->result = @mysql_query($query,$this->dbh);

    // If there is an error then take note of it..
    if ( $str = @mysql_error($this->dbh) )
    {
        $is_insert = true;
        $this->register_error($str);
        $this->show_errors ? trigger_error($str,E_USER_WARNING) : null;
        return false;
    }

    // Query was an insert, delete, update, replace
    $is_insert = false;
    if ( preg_match("/^(insert|delete|update|replace)\s+/i",$query) )
    {
        $this->rows_affected = @mysql_affected_rows();

        // Take note of the insert_id
        if ( preg_match("/^(insert|replace)\s+/i",$query) )
        {
            $this->insert_id = @mysql_insert_id($this->dbh);
        }

        // Return number fo rows affected
        $return_val = $this->rows_affected;
    }
    // Query was a select
    else
    {
        // Take note of column info
    }
}
```

```

    $i=0;
    while ($i < @mysql_num_fields($this->result))
    {
        $this->col_info[$i] = @mysql_fetch_field($this->result);
        $i++;
    }

    // Store Query Results
    $num_rows=0;
    while ( $row = @mysql_fetch_object($this->result) )
    {
        // Store results as an objects within main array
        $this->last_result[$num_rows] = $row;
        $num_rows++;
    }

    @mysql_free_result($this->result);

    // Log number of rows the query returned
    $this->num_rows = $num_rows;

    // Return number of rows selected
    $return_val = $this->num_rows;
}

// disk caching of queries
$this->store_cache($query,$is_insert);

// If debug ALL queries
$this->trace || $this->debug_all ? $this->debug() : null ;

return $return_val;
}
}

```

```
?>
```

/home/x-warrior/Desktop/Keep-Track/includes/functions/companyFunctions.php

```

<?
function getId($companyName) {
    global $db;
    $company = $db->get_row("select companyId from `Companies` where LOWER(companyName) = LOWER
('$companyName')");
    $companyId = $company->companyId;
    return $companyId;
}

function identifyCompany($trackCode) {
    global $system;
    $trackCode = str_replace(" ", "", $trackCode);

    $size = count($system);
    foreach($system as $sys => $value) {
        $match = preg_match($value["regex"], $trackCode);
        if($match == true) {
            return $sys;
        }
    }
}
}
?>

```



/home/x-warrior/Desktop/Keep-Track/includes/functions/graphicFunctions.php

```
<?
function showMenu() {
    if(isLogged()) {
        include ('loggedMenu.php');
    } else {
        include ('notLoggedMenu.php');
    }
}
?>
```

/home/x-warrior/Desktop/Keep-Track/includes/functions/notificationFunctions.php

```
<?php
function getNotifications($trackId, $userId) {
    global $db;
    $res = $db->get_results("select notEmail from Notifications where `Users_userId` = '$userId' and `Tracks_trackId` = '$trackId'");
    return $res;
}
function getNotificationsTrackCode($trackCode, $userId) {
    return getNotifications(getTrackId($trackCode), $userId);
}
function addNotification($trackId, $userId, $email) {
    global $db;
    if(!existsNotification($trackId, $userId, $email)) {
        $db->query("INSERT into `Notifications` (notEmail, notSurvey, Tracks_TrackId, Users_userId) values ('$email', 0, $trackId, $userId)");
    }
}
function existsNotification($trackId, $userId, $email) {
    global $db;
    $res = $db->get_row("select count(notId) as has from Notifications where `Users_userId` = '$userId' and `Tracks_trackId` = '$trackId' and `notEmail` = '$email'");
    if($res->has == 0) {
        return false;
    }
    return true;
}
function sendNotificationsEmail($trackId) {
    global $db;
    global $PERGUNTAS;
    $res = $db->get_results("select notEmail,Users_userId,notId from Notifications where `notSurvey` = 0 and `Tracks_trackId` = '$trackId'");
    if(count($res) > 0) {
        foreach($res as $not) {
            $message = '<html>
                <body>
                <h3>' . _("Satisfaction Survey") . '</h3>
                (sent to: ' . $not->notEmail . ')
                <form method="post" action="http://sandbox.matbra.com/keeptrack/survey.php">';

            for ($i = 0; $i < count($PERGUNTAS); $i++) {
                $name = "qst" . $i;
                $pergunta = $PERGUNTAS[$i];
                $message .= $pergunta . '
                <input type="radio" name=".' . $name . '" value="excelent"> ' . _("Excelent") . '
                <input type="radio" name=".' . $name . '" value="good"> ' . _("Good") . '
                <input type="radio" name=".' . $name . '" value="normal" checked> ' . _("Normal") . '
                <input type="radio" name=".' . $name . '" value="bad"> ' . _("Bad") . '
                <input type="radio" name=".' . $name . '" value="horrible"> ' . _("Horrible") . ' <br /><br />';
            }
        }
    }
};
```

```

$message .= '<input type="hidden" name="trackid" value="' . $trackId . "'>
          <input type="hidden" name="userId" value="' . $not->Users_userId . "'>
          <input type="hidden" name="notId" value="' . $not->notId . "'>
          <input type="submit" value="Send"></form>
        </body>
      </html>';
$headers = 'From: survey@ktrack.com' . "\r\n" .
          'Reply-To: help@ktrack.com' . "\r\n" .
          'X-Mailer: PHP/' . phpversion() . "\r\n" .
          'Content-type: text/html; charset=iso-8859-1' . "\r\n";
mail("matheusbrat@gmail.com", _("Satisfaction Survey"), $message, $headers);
echo "email enviado para " . $not->notEmail . "<br />";
}
$db->query("update `Notifications` set `notSurvey` = 1 where `Tracks_trackId` = '$trackId'");
}
}

```

/home/x-warrior/Desktop/Keep-Track/includes/functions/searchFunctions.php

```

<?
function search($trackCode) {
    global $system;
    global $db;

    $track = $db->get_row("select trackId, trackUpdates, trackLastUpdate from `Tracks` where trackCode =
'$trackCode'");
    $trackId = $track->trackId;

    $companyName = identifyCompany($trackCode);
    if(!is_null($companyName)) {

        $userId = $_SESSION["userId"];

        $now = new DateTime();
        $lastUpdate = new DateTime($track->trackLastUpdate);
        $minutes = round(abs($now->format('U') - $lastUpdate->format('U')) / (60));

        // IF KEEP UPDATES || NO TRACKS IN THE SYSTEM
        if(($track->trackUpdates == 1 && $minutes >= 30) || empty($track)) {
            if(function_exists($system[$companyName]["method"])) {
                $snip = call_user_func($system[$companyName]["method"], $trackCode);
            }
            // IF NO TRACK ON THE SYSTEM
            if(empty($track)) {
                $total = 0;
                $companyId = getCompanyId($companyName);
                $db->query("INSERT INTO `Tracks` (`trackCode`, `trackUpdates`, `trackLastUpdate`,
`Companies_companyId`) VALUES ('$trackCode', 1, NOW(), $companyId)");
                $trackId = $db->insert_id;
                checkTrackToUser($userId, $trackId);

                // IF TRACK ON THE SYSTEM
            } else {
                $total = totalSnips($trackId);
                $db->query("UPDATE `Tracks` SET `trackLastUpdate`=NOW() WHERE `trackId`='$trackId'");
                checkTrackToUser($userId, $trackId);
            }
            updateSnips($trackId, $total, $snip);

            // IF DO NOT KEEP UPDATES
        } else {
            $snip = generateSnipFromTrackId($trackId);
        }
    }
}

```

```

        checkTrackToUser($userId, $trackId);
    }
    if($snip[0]["status"] == DELIVERED_STR) {
        sendNotificationsEmail($trackId);
    }
    $snip = formatSnip($snip);
    $tracks = createTrackArray($companyName, $trackCode, $snip);
    return $tracks;
}
return;
}
?>

```

/home/x-warrior/Desktop/Keep-Track/includes/functions/tracksFunctions.php

```

<?
function totalSnips($track) {
    global $db;
    $l = $db->get_row("select count(snipStatus) as total from `Snippets` where `Tracks_trackId` =
'$track'");
    $total = $l->total;
    return $total;
}

function formatSnip($snip) {
    date_default_timezone_set ("America/Sao_Paulo");
    for ($i = 0; $i < count($snip); $i++) {
        $snip[$i]["status"] = _($snip[$i]["status"]);
        $snip[$i]["date"] = date_(_('m/d/y h:i:s a'),strtotime($snip[$i]["date"]));
    }
    return $snip;
}

function isDelivered($trackCode) {
}

function isDeliveredTrackId($trackId) {
    global $db;
    $res = $db->get_row("select trackStatus from `Tracks` where trackId = '$trackId'");
    if ($res->trackStatus == DELIVERED_STR)
        return true;
    return false;
}

function createTrackArray($companyName, $trackCode, $snip) {
    global $system;
    return array("name" => $companyName, "link" => str_replace("[:trackCode:]", $trackCode, $system
[$companyName]["link"]), "snips" => $snip);
}

function getTrackId($trackCode) {
    global $db;
    $track = $db->get_row("select trackId from `Tracks` where trackCode = '$trackCode'");
    $trackId = $track->trackId;
    return $trackId;
}

function updateSnips($trackId, $total, $snip) {
    global $db;
    // IF THERE IS SNIPS IN THE TRACKS BUT NOT IN THE SYSTEM
    if($total < count($snip)) {
        $dif = count($snip) - $total - 1;
        for ($i = $dif; $i >= 0; $i--) {
            $status = $snip[$i]["status"];
            $date = $snip[$i]["date"];

```

```

    $city = $snip[$i]["location"]["city"];
    $state = $snip[$i]["location"]["state"];
    $country = $snip[$i]["location"]["country"];
    $details = $snip[$i]["details"];
    $goingTo = $snip[$i]["goingTo"];

    $gtId = 'NULL';
    if(!empty($goingTo)) {
        $gtCity = $goingTo["city"];
        $gtState = $goingTo["state"];
        $gtCountry = $goingTo["country"];
        $gtDetails = $snip[$i]["detailsGoingTo"];
        $db->query("INSERT INTO `SnipGt` (`snipGtCountry`, `snipGtState`, `snipGtCity`,
`snipGtDetail`) VALUES ('$gtCountry', '$gtState', '$gtCity', '$gtDetails')");
        $gtId = $db->insert_id;
    }
    $db->query("INSERT INTO `Snippets` (`snipStatus`, `snipDate`, `snipCountry`, `snipState`,
`snipCity`, `snipDetail`, `Tracks_trackId`, `SnipGt_snipGtId`) VALUES ('$status', '$date', '$country',
'$state', '$city', '$details', $trackId, $gtId)");
    }
    $lastStatus = $snip[0]["status"];
    if($lastStatus == DELIVERED_STR) {
        $db->query("UPDATE `Tracks` SET `trackStatus`='$lastStatus', `trackUpdates`=0,
`trackLastUpdate`=NOW() WHERE `trackId`='$trackId'");
    } else {
        $db->query("UPDATE `Tracks` SET `trackStatus`='$lastStatus', `trackLastUpdate`=NOW() WHERE
`trackId`='$trackId'");
    }

    sendNotifications($trackId);
} else if($total > count($snip)) {
    //TODO: THERE IS SOMETHING WRONG!?
    die("B");
}
}

function generateSnipFromTrackId($trackId) {
    global $db;
    $results = $db->get_results("select snipStatus as status, snipDate as date, snipCountry, snipState,
snipCity, snipDetail as details, snipGtCountry, snipGtState, snipGtCity, snipGtDetail as detailsGoingTo
from `Snippets` LEFT JOIN `SnipGt` on (`Snippets`.SnipGt_snipGtId = `SnipGt`.snipGtId) where
Tracks_trackId = '$trackId' order by snipId desc", ARRAY_A);
    for ($i = 0; $i < count($results); $i++) {
        if($results[$i]["snipGtCountry"] == "") {
            unset($results[$i]["detailsGoingTo"]);
        } else {
            $results[$i]["goingTo"] = array("city" => $results[$i]["snipGtCity"], "state" => $results[$i]
["snipGtState"], "country" => $results[$i]["snipGtCountry"]);
        }

        $results[$i]["location"] = array("city" => $results[$i]["snipCity"], "state" => $results[$i]
["snipState"], "country" => $results[$i]["snipCountry"]);
        unset($results[$i]["snipGtCountry"]);
        unset($results[$i]["snipGtState"]);
        unset($results[$i]["snipGtCity"]);
        unset($results[$i]["snipCountry"]);
        unset($results[$i]["snipState"]);
        unset($results[$i]["snipCity"]);
        $snip = $results;
    }
    return $snip;
}
?>

```

```
<?
function isLoggedIn() {
    if (!empty($_SESSION["user"]) && !empty($_SESSION["userId"])) {
        return true;
    }
    return false;
}

function userCheckCookie() {
    global $db;
    $cook = $_COOKIE['user'];
    if(!empty($cook)) {
        $cook = secureDataToDB($cook);
        $user = $db->get_row("SELECT userId,userEmail from `Users` where userCookHash = '$cook'");
        if (!empty($user)) {
            $_SESSION["userId"] = $user->userId;
            $_SESSION["user"] = $user->userEmail;
            setcookie("user", $cook, time()+604800);
        }
    }
}

function userUpdateHash($userId, $cookHash) {
    global $db;
    $db->query("UPDATE `Users` set userCookHash = '$cookHash' where userId = '$userId'");
}

function userUpdateTrackCodes($userId) {
    global $db;
    $res = $db->get_results("select trackCode from `Users_has_Tracks` as ut LEFT JOIN `Tracks` as t on
(ut.Tracks_trackId = t.trackId) where ut.Users_userId = '$userId' and t.trackUpdates = 1");
    if(sizeof($res) > 0) {
        foreach($res as $s) {
            search($s->trackCode);
        }
    }
}

function userLogin($usera, $password, $keepLogged = false) {
    global $salt;
    $userId = isValid($usera, $password);
    if ($userId !== false) {
        $_SESSION["userId"] = $userId;
        $_SESSION["user"] = $usera;
        if ($keepLogged == true) {
            $cookHash = md5($usera . uniqid($salt, true) . rand(0,10));
            userUpdateHash($userId, $cookHash);
            setcookie("user", $cookHash, time()+604800);
        }
        return true;
    }
    return false;
}

function getUserName($usera) {
    global $db;
    $userEmail = $db->get_var("SELECT userName FROM `Users` where userEmail = '$usera'");
    if(!empty($userEmail)) {
        return $userEmail;
    }
    return false;
}

function isUsed($user) {
```

```

    global $db;
    $userId = $db->get_var("SELECT userId FROM `Users` where userEmail = '$user'");
    if(!empty($userId)) {
        return $userId;
    }
    return false;
}

function isValid($usera, $password) {
    global $db;
    global $salt;
    $password = md5($password . $salt);
    $userId = $db->get_var("SELECT userId FROM `Users` where userEmail = '$usera' AND userPassword = '$password'");
    if(!empty($userId)) {
        return $userId;
    }
    return false;
}

function addTrackToUser($track, $user) {
    global $db;
    $db->query("INSERT INTO `Users_has_Tracks` (`Users_userId`, `Tracks_trackId`) VALUES ($user, $track)");
}

function hasTrack($trackId, $user) {
    global $db;
    $user = $db->get_row("select count(Users_userId) as has from `Users_has_Tracks` where `Tracks_trackId` = '$trackId' AND `Users_userId` = '$user'");
    if ($user->has == 0) {
        return false;
    }
    return true;
}

function hasTrackCode($trackCode, $user) {
    global $db;
    return hasTrack(getTrackId($trackCode), $user);
}

function checkTrackToUser($userId, $trackId) {
    if (!empty($userId)) {
        if(!hasTrack($trackId, $userId)) {
            addTrackToUser($trackId, $userId);
            return true;
        }
    }
    return false;
}
?>

```

/home/x-warrior/Desktop/Keep-Track/includes/functions/utilFunctions.php

```

<?
function unregister_GLOBALS()
{
    if (!ini_get('register_globals')) {
        return;
    }

    // Might want to change this perhaps to a nicer error
    if (isset($_REQUEST['GLOBALS']) || isset($_FILES['GLOBALS'])) {
        die('GLOBALS overwrite attempt detected');
    }
}

```

```

// Variables that shouldn't be unset
$noUnset = array('GLOBALS', '_GET',
                '_POST', '_COOKIE',
                '_REQUEST', '_SERVER',
                '_ENV', '_FILES');

$input = array_merge($_GET, $_POST,
                    $_COOKIE, $_SERVER,
                    $_ENV, $_FILES,
                    isset($_SESSION) && is_array($_SESSION) ? $_SESSION : array());

foreach ($input as $k => $v) {
    if (!in_array($k, $noUnset) && isset($GLOBALS[$k])) {
        unset($GLOBALS[$k]);
    }
}

function secureDataToDB($str) {
    global $db;
    return $db->escape($str);
}

//LOCALES
function show_locales() {
    global $locales;
    foreach($locales as $locale) {
        $n = $locale["name"];
        $i = $locale["icon"];
        echo "<a href=\"locale.php?l=$n\"><img src=\"$i\" /></a> ";
    }
}

function setupLocale($path = '') {
    global $locales;

    // settings you may want to change
    if($_SESSION["locale"] != "") {
        $locale = $_SESSION['locale']; // the locale you want
    } else {
        $locale = $locales[0]["name"];
    }
    $locales_root = $path . "locales"; // locales directory
    $domain = $locale; // the domain you're using, this is the .PO/.MO file name without the extension

    if(is_dir("$locales_root/$locale/")) {
        // activate the locale setting
        setlocale(LC_ALL, $locale);
        //putenv("LANG=$locale");
        // path to the .MO file that we should monitor
        $filename = "$locales_root/$locale/LC_MESSAGES/$domain.mo";
        $mtime = filemtime($filename); // check its modification time
        // our new unique .MO file
        $filename_new = "$locales_root/$locale/LC_MESSAGES/{$domain}_{$mtime}.mo";

        if (!file_exists($filename_new)) { // check if we have created it before
            // if not, create it now, by copying the original
            copy($filename,$filename_new);
        }
        // compute the new domain name
        $domain_new = "{$domain}_{$mtime}";
        // bind it
        bindtextdomain($domain_new,$locales_root);
        // then activate it
        textdomain($domain_new);
        // all done
    }
}

```

```

}

function sendNotifications($trackId) {
    global $db;
    // POSSIVELMENTE MAIS EFICIENTE!
    // select userId, userEmail, userName, utLabel, trackCode, trackStatus, trackLastUpdate from Tracks
    as t LEFT JOIN Users_has_Tracks as ut on (t.trackId = ut.Tracks_trackId) LEFT JOIN Users as u on
    (ut.Users_userId = u.userId) where t.trackId = '$trackId'
    $res = $db->get_results("select userId, userEmail, userName, utLabel, trackCode, trackStatus,
    trackLastUpdate from Users as u, Users_has_Tracks as ut, Tracks as t where `u`.userId = `ut`.Users_userId
    and `ut`.Tracks_trackId = `t`.trackId and `t`.trackId = '$trackId'");
    if(sizeof($res) > 0) {
        foreach($res as $s) {
            if($s->userId != $_SESSION['userId']) {
                $label = "";
                if(strlen($s->utLabel) > 0) {
                    $label = "(" . $s->utLabel . ")";
                }
                $userName = $s->userName;
                $trackCode = $s->trackCode;
                $subject = _("Status update on ") . $s->trackCode . " " . $label;
                $format = _("Hello %s,\nYour package %s %s has been updated.\nFor more information
                access: http://sandbox.matbra.com/keeptrack/?tc=" . "%s" . "&act=search\n\nKtrack Team.");
                $message = sprintf($format, $userName, $trackCode, $label, $trackCode);
                $headers = 'From: notification@ktrack.com' . "\r\n" .
                'Reply-To: help@ktrack.com' . "\r\n" .
                'X-Mailer: PHP/' . phpversion();
                mail($s->userEmail, $subject, $message, $headers);
            }
        }
    }
}
?>

```

/home/x-warrior/Desktop/Keep-Track/includes/system/Correios.php

```

<?php
$system["Correios"] = array(
    // "regex" => "/^[A-Z]{2}[A-Za-z0-9]{9}[A-Z]{2}/",
    "regex" => "/^[A-Z]{2}[A-Za-z0-9]{9}BR$/",
    "method" => "getCorreios",
    "link" => "http://websro.correios.com.br/sro_bin/txect01$.QueryList?P_LINGUA=001&P_TIPO=001&P_COD_UNI=
    [[:trackCode:]]",
    "words" => array("Entregue" => DELIVERED_STR, "Conferido" => CHECKED_STR, "Entrega Efetuada" =>
    DELIVERED_STR, "Encaminhado" => ON_WAY_STR, "Saiu para entrega" => OUT_FOR_DELIVERY_STR, "Distribuição
    Especial Agendada" => OUT_FOR_DELIVERY_STR, "Postado" => POSTED_STR, "Postagem - DH" => POSTED_STR,
    "Aguardando retirada" => WAITING_STR, "Receiver not at home" => RECEIVER_NOT_HOME),
    "timeZone" => "America/Sao_Paulo"
);

function genLocation($l) {
    $city = substr($l, 0, strpos($l, "/"));
    $state = substr($l, strpos($l, "/") + 1, strlen($l));
    $country = "BR";
    return array("city" => $city, "state" => $state, "country" => $country);
}

function convertStatus($status) {
    global $system;
    if(array_key_exists($status, $system["Correios"]["words"])) {
        return $system["Correios"]["words"][$status];
    } else {
        mail("matheusbrat@gmail.com", "NEW STATUS WORDS", $status . " CORREIOS!");
    }
}

```



```

}

function getCorreios($trackCode) {
    $ch = curl_init("http://websro.correios.com.br/sro_bin/txect01$.QueryList?
P_LINGUA=001&P_TIPO=001&P_COD_UNI=" . $trackCode);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, false);
    $resp = curl_exec($ch);
    //$resp = mb_convert_encoding(mb_convert_encoding($resp, 'ISO-8859-1'), 'UTF-8');
    preg_match_all("/^<tr><td.*>(.*?)</td><td>(.*?) - (.*?)</td><td>(.*?)</td></tr>(\n<tr><td colspan=2>
(.*?) - (.*?)</td></tr>)?$/m", $resp, $match, PREG_SET_ORDER);
    foreach ($match as $r) {
        $place = getLocation($r[3]);
        $datetime = explode(" ", $r[1]);
        $date = implode("-",array_reverse(explode("/", $datetime[0])));
        $date = $date . " " . $datetime[1];
        $status = convertStatus(strip_tags($r[4]));
        if(count($r) == 5) {
            $tracks[] = array("date" => $date, "location" => $place, "details" => $r[2], "status" =>
$status);
        } else {
            $goingPlace = getLocation($r[7]);
            $tracks[] = array("date" => $date, "location" => $place, "details" => $r[2], "status" =>
$status, "goingTo" => $goingPlace, "detailsGoingTo" => $r[6]);
        }
    }
    return $tracks;
}

```

/home/x-warrior/Desktop/Keep-Track/includes/system/Fedex.php

```

<?php
$system["Fedex"] = array(
    "regex" => "/^[0-9]{12}$/",
    "method" => "getFedex",
    "link" => "http://www.fedex.com/Tracking?tracknumbers=[[ :trackCode:]]&action=track&language=english",
    "words" => array("Delivered" => DELIVERED_STR, "In transit" => ON_WAY_STR, "On FedEx vehicle for
delivery" => OUT_FOR_DELIVERY_STR, "Picked up" => POSTED_STR),
    "timeZone" => "America/Sao_Paulo"
);
function getArray($node)
{
    $array = false;

    if ($node->hasAttributes())
    {
        foreach ($node->attributes as $attr)
        {
            $array[$attr->nodeName] = $attr->nodeValue;
        }
    }

    if ($node->hasChildNodes())
    {
        if ($node->childNodes->length == 1)
        {
            $array[$node->firstChild->nodeName] = $node->firstChild->nodeValue;
        }
        else
        {
            foreach ($node->childNodes as $childNode)
            {
                if ($childNode->nodeType != XML_TEXT_NODE)
                {
                    $array[$childNode->nodeName][] = $this->getArray($childNode);
                }
            }
        }
    }
}

```

```

    }
  }
}

return $array;
}

function getFedex($trackCode) {
    $ch = curl_init("http://www.fedex.com/Tracking?tracknumbers=" . $trackCode .
"&action=track&language=english");
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_HEADER, false);
    $resp = curl_exec($ch);
    //$resp = mb_convert_encoding(mb_convert_encoding($resp, 'ISO-8859-1'), 'UTF-8');
    $doc = new DOMDocument();
    @$doc->loadHTML($resp);
    $a = $doc->getElementsByTagName("tableRowOdd");
    print_r("Tamanho: " . $a->length);
    for ($i =0; $i < $a->length; $i++) {
        print_r($a->item($i));
    }
    die(print_r("a"));
    $str = '<div style="overflow: visible; height: 399px;" id="table.travelHistory._contents._rows"
class="tableContentsRows">
    <div class="tableRowOdd" id="table.travelHistory._contents._row1"><div style="width: 160px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row1._col1">May
18, 2011 14:00</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col2"><div class="detailfirstscan">Delivered</div></div><div
style="width: 170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col3">FLORIANOPOLIS BR</div><div style="width: 169px; text-
align: left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row1._col4">Payment
received</div></div><div class="tableRowEven" id="table.travelHistory._contents._row2"><div style="width:
160px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col1">May 18, 2011 09:57</div><div style="width: 180px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row2._col2">0n
FedEx vehicle for delivery</div><div style="width: 170px; text-align: left; height: 17px;"
class="tableNotLastCell" id="table.travelHistory._contents._row2._col3">BLUMENAU BR</div><div
style="width: 169px; text-align: left; height: 17px;" class="tableLastCell"
id="table.travelHistory._contents._row2._col4"></div></div>
    </div>';
    $str = '<div style="overflow: visible; height: 399px;" id="table.travelHistory._contents._rows"
class="tableContentsRows"><div class="tableRowOdd" id="table.travelHistory._contents._row1"><div
style="width: 160px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col1">May 18, 2011 14:00</div><div style="width: 180px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row1._col2"><div
class="detailfirstscan">Delivered</div></div><div style="width: 170px; text-align: left; height: 17px;"
class="tableNotLastCell" id="table.travelHistory._contents._row1._col3">FLORIANOPOLIS BR</div><div
style="width: 169px; text-align: left; height: 17px;" class="tableLastCell"
id="table.travelHistory._contents._row1._col4">Payment received</div></div><div class="tableRowEven"
id="table.travelHistory._contents._row2"><div style="width: 160px; text-align: left; height: 17px;"
class="tableNotLastCell" id="table.travelHistory._contents._row2._col1">May 18, 2011 09:57</div><div
style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col2">0n FedEx vehicle for delivery</div><div style="width:
170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col3">BLUMENAU BR</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row2._col4"></div></div></
div>';
    preg_match_all('#<div.*class="tableContentsRows">.*</div>#',$str,$match,PREG_SET_ORDER);
    //preg_match_all('#(<div.*id="table\.travelHistory\._contents\._row[0-9]{0,2}">(.*?)</div>)+#',$match
[0][0],$mtc,PREG_SET_ORDER);
    //preg_match_all('/<div.*><div.*>.*</div><div.*>.*</div><div.*>.*</div><div.*>.*</div></div>/',
$match[0][0], $test);
    //print_r($test);
    //preg_match_all('/(<div.*><div.*>(.*?)</div><div.*>(.*?)</div><div.*>(.*?)</div><div.*>.*</div></
div>)*/', $str,$match);
    //preg_match_all("/(<div.*><div.*>(.*?)</div><div.*>(.*?)</div><div.*>(.*?)</div><div.*>(.*?)</div></

```

```

div>)+/m", $resp, $match, PREG_SET_ORDER);
    die (print_r($match));
    foreach ($match as $r) {
        $place = getLocation($r[3]);
        $datetime = explode(" ", $r[1]);
        $date = implode("-",array_reverse(explode("/", $datetime[0])));
        $date = $date . " " . $datetime[1];
        $status = convertStatus(strip_tags($r[4]));
        if(count($r) == 5) {
            $tracks[] = array("date" => $date, "location" => $place, "details" => $r[2], "status" =>
$status);
        } else {
            $goingPlace = getLocation($r[7]);
            $tracks[] = array("date" => $date, "location" => $place, "details" => $r[2], "status" =>
$status, "goingTo" => $goingPlace, "detailsGoingTo" => $r[6]);
        }
    }
    return $tracks;
}
}
/*<div class="tableRowOdd" id="table.travelHistory._contents._row1">
    <div style="width: 160px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col1">May 18, 2011 14:00</div>
    <div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col2"><div class="detailfirstscan">Delivered</div></div>
    <div style="width: 170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row1._col3">FLORIANOPOLIS BR</div>
    <div style="width: 169px; text-align: left; height: 17px;" class="tableLastCell"
id="table.travelHistory._contents._row1._col4">Payment received</div>
</div>
<div class="tableRowEven" id="table.travelHistory._contents._row2">
    <div style="width: 160px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col1">May 18, 2011 09:57</div>
    <div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col2">On FedEx vehicle for delivery</div>
    <div style="width: 170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row2._col3">BLUMENAU BR</div>
    <div style="width: 169px; text-align: left; height: 17px;" class="tableLastCell"
id="table.travelHistory._contents._row2._col4"></div>
</div>
<div class="tableRowOdd" id="table.travelHistory._contents._row3"><div style="width: 160px; text-align:
left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row3._col1">May 17, 2011
18:49</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row3._col2">At local FedEx facility</div><div style="width: 170px;
text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row3._col3">BLUMENAU BR</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row3._col4"></div></div>
<div class="tableRowEven" id="table.travelHistory._contents._row4"><div style="width: 160px; text-align:
left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row4._col1">May 17, 2011
13:25</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row4._col2">Delivery exception</div><div style="width: 170px; text-
align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row4._col3">BLUMENAU BR</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row4._col4">Held, unable to
collect payment</div></div>
<div class="tableRowOdd" id="table.travelHistory._contents._row5"><div style="width: 160px; text-align:
left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row5._col1">May 17, 2011
10:23</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row5._col2">On FedEx vehicle for delivery</div><div style="width:
170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row5._col3">BLUMENAU BR</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row5._col4"></div></
div><div class="tableRowEven" id="table.travelHistory._contents._row6"><div style="width: 160px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row6._col1">May
13, 2011 17:37</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row6._col2">At local FedEx facility</div><div style="width: 170px;
text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row6._col3">BLUMENAU BR</div><div style="width: 169px; text-align:

```



```

id="table.travelHistory._contents._row16._col3">DENVER, CO</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row16._col4"></div></
div><div class="tableRowOdd" id="table.travelHistory._contents._row17"><div style="width: 160px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row17._col1">May
10, 2011 19:10</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row17._col2">Left FedEx origin facility</div><div style="width: 170px;
text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row17._col3">BOULDER, CO</div><div style="width: 169px; text-align:
left; height: 17px;" class="tableLastCell" id="table.travelHistory._contents._row17._col4"></div></
div><div class="tableRowEven" id="table.travelHistory._contents._row18"><div style="width: 160px; text-
align: left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row18._col1">May
10, 2011 16:28</div><div style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row18._col2">Picked up</div><div style="width: 170px; text-align:
left; height: 17px;" class="tableNotLastCell" id="table.travelHistory._contents._row18._col3">BOULDER,
CO</div><div style="width: 169px; text-align: left; height: 17px;" class="tableLastCell"
id="table.travelHistory._contents._row18._col4"></div></div><div class="tableRowOdd"
id="table.travelHistory._contents._row19"><div style="width: 160px; text-align: left; height: 17px;"
class="tableNotLastCell" id="table.travelHistory._contents._row19._col1">May 10, 2011 10:50</div><div
style="width: 180px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row19._col2">Shipment information sent to FedEx</div><div
style="width: 170px; text-align: left; height: 17px;" class="tableNotLastCell"
id="table.travelHistory._contents._row19._col3"></div><div style="width: 169px; text-align: left; height:
17px;" class="tableLastCell" id="table.travelHistory._contents._row19._col4"></div></div></div>*/

```

```
/home/x-warrior/Desktop/Keep-Track/index.php
```

```

<?
//error_reporting(E_ERROR | E_WARNING | E_PARSE | E_NOTICE);
session_start();
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";
unregisterGLOBALS();

setupLocale();

$db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
$db->query("SET CHARACTER SET utf8 ");

if(!isLoggedIn()) {
    userCheckCookie();
}

//$db->debug_all = true;

$act = $_GET['act'];
switch ($act) {
    case "search":
        $content = "infos.php";
        include ("page.php");
        break;
    case "stats":
        $content = "stats.php";
        include ("page.php");
        break;
    case "login":
        if(isLoggedIn()) {
            $content = "logged.php";
            include("page.php");
        } else {
            $user = secureDataToDB($_POST['email']);
            $password = $_POST['password'];
            $keeplogged = false;
            if(userLogin($user, $password, true)) {
                $content = "logged.php";
            }
        }
    }

```

```
        include("page.php");
    } else {

        include("main.php");
    }
}
break;
case "logout":
$_SESSION["user"] = "";
$_SESSION["userId"] = "";
setcookie("user", '', time()-100000);
include("main.php");
break;
case "config":
if (isLoggedIn()) {
    $content = "userConfig.php";
    include ("page.php");
} else {
    include("main.php");
}
break;

case "setconfig":
if (isLoggedIn()) {
    $email = $_SESSION['user'];
    $oldpassword = $_POST["old_password"];
    $password = $_POST['password'];
    if(isValid($email,$oldpassword)) {
        $username = secureDataToDB($_POST['username']);
        if(!empty($password)) {
            $password_conf = $_POST['password_conf'];
            if($password == $password_conf) {
                $password_encode = md5($password . $salt);
                $db->query("update `Users` set userName = '$username', userPassword =
'$password_encode' where userEmail = '$email'");
            }
        } else {
            $db->query("update `Users` set `userName` = '$username' where `userEmail` =
'$email'");
        }
    }
    $content = "userConfig.php";
    include ("page.php");
} else {
    include("main.php");
}

break;

case "signup":
$_SESSION["user"] = "";
$_SESSION["userId"] = "";
setcookie("user", '', time()-100000);
$content = "userSignUp.php";
include ("page.php");
break;

case "signupuser":
$email = secureDataToDB($_POST['email']);
if(!isUsed($email)) {
    $password = $_POST['password'];
    if(!empty($password)) {
        $password_conf = $_POST['password_conf'];
        $username = secureDataToDB($_POST['username']);
        if($password == $password_conf) {

            $password_encode = md5($password . $salt);
```



```

        $db->query("insert into `Users` (userEmail, userPassword, userName,
userEmailNotification) values ('$email', '$password_encode', '$username', 0)");
        $error[0] = _("Signed up!");

        if(userLogin($email, $password, true)) {
            $content = "logged.php";
            include("page.php");
        } else {

            include("main.php");
        }

        break;
    } else {
        $error[0] = _("Password doesn't match.");
    }
} else {
    $error[0] = _("Empty password");
}
} else {
    $error[0] = _("User e-mail already in use.");
}
if(!empty($error[0])) {
    $content = "userSignUp.php";
    include ("page.php");
}
break;
default:
    if (isLoggedIn()) {
        $content = "logged.php";
        include("page.php");
    } else {
        include("main.php");
    }
}
?>

```

/home/x-warrior/Desktop/Keep-Track/infos.php

```

<?php
@session_start();
@header('Content-Type: text/html; charset=ISO-8859-1');
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";

setupLocale();

if($db == null) {
    $db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
    $db->query("SET CHARACTER SET utf8 ");
}

$trackCode = secureDataToDB($_GET['tc']);
if(empty($trackCode)) {
    include ("main.php");
    die();
}
$tracks = search($trackCode);
$snips = $tracks["snips"];

$ts = count($snips)-1;

```

```

$trackLink = $tracks["link"];
$trackCompany = $tracks["name"];

$trackStatus = $snips[0]["status"];
$trackFrom = $snips[$ts]["location"]["city"] . " - " . $snips[$ts]["location"]["state"] . " - " . $snips
[$ts]["location"]["country"];
$trackFromDate = $snips[$ts]["date"];
$trackLastLocation = $snips[0]["location"]["city"] . " - " . $snips[0]["location"]["state"] . " - " .
$snips[0]["location"]["country"];
$trackLastLocationDate = $snips[0]["date"];
?>
<table width="100%" height="100%" border="0" cellspacing="0">
  <tr>
    <td width="50%" align="left">

      <table width="100%" border="0" cellspacing="0">
        <tr>
          <td><b><?php echo _('Official link'); ?></b></td>
          <td><?php echo "<a href=\"\$trackLink\" target=\"_blank\">\$trackCompany</a>\"?>
          </td>
        </tr>
        <tr>
          <td><b><?php echo _('Status'); ?></b></td>
          <td><?php echo \$trackStatus; ?></td>
        </tr>
        <tr>
          <td><b><?php echo _('Shipped from'); ?></b></td>
          <td><?php echo \$trackFrom; ?></td>
        </tr>
        <tr>
          <td><b><?php echo _('Shipped date'); ?></b></td>
          <td><?php echo \$trackFromDate; ?></td>
        </tr>
        <tr>
          <td><b><?php echo _('Last Location'); ?></b></td>
          <td><?php echo \$trackLastLocation; ?></td>
        </tr>
        <tr>
          <td><b><?php echo _('on'); ?></b></td>
          <td><?php echo \$trackLastLocationDate; ?></td>
        </tr>
      </table></td>
    <td width="50%" height="100%"><div
      id="map_canvas<?php echo "-" . \$trackCode; ?>"
      style="width: 100%; height: 100%"></div></td>
  </tr>
  <tr>
    <td colspan="2">
      <?php
        if ($_SESSION['userId'] != "") {
          $trackId = getTrackId($trackCode);
          if(hasTrack($trackId, $_SESSION['userId'])) {
            if(!isDeliveredTrackId($trackId)) {
              ?>
              <table border="0" cellspacing="0">
                <tr>
                  <td><?php echo _('Notifications:'); ?></td>
                </tr>
              </table>
              <?php
                $nots = getNotifications($trackId, $_SESSION['userId']);
                if(count($nots) > 0) {
                  foreach($nots as $not) {
                    echo "<tr><td>". $not->notEmail . "</td></tr>";
                  }
                }
              ?>
            <tr class="not-empty"><td></td></tr>
          }
        }
      </td>
    </tr>
  </tr>

```



```

        <tr>
            <td><input type="text" class="note-<?php echo $trackCode; ?>"><a href="#" onClick="addNot
('<?php echo $trackCode; ?>', '<?php echo $trackId; ?>')"></a></td>
        </tr>
    </table>
    <?php
        }
    }
    }
    ?>
</td>
</tr>
</table>

<script>
<?php echo "var map_v" . $trackCode . " = []"; ?>

<?php
if(count($snips) > 0) {
    echo "var_map_json" . $trackCode . " = eval('" . json_encode($snips) . "')";
    echo "initialize('map_canvas-$trackCode', var_map_json" . $trackCode . ", map_v" . $trackCode . ")";
}
?>

</script>

```

/home/x-warrior/Desktop/Keep-Track/locale.php

```

<?php
session_start();
$_SESSION['locale'] = $_GET['l'];
echo "<script>history.back();</script>";

```

/home/x-warrior/Desktop/Keep-Track/logged.php

```

<table class="tracks" cellspacing="0" width="100%">
    <tr class="track" bgcolor="#B0E2FF">
        <td></td>
        <td width="20%" align="left"><b><?php echo _('Tracking Code'); ?></b></td>
        <td width="25%"><b><?php echo _('Status'); ?></b></td>
        <td width="20%"><b><?php echo _('Last Check'); ?></b></td>
        <td width="20%"><b><?php echo _('Company'); ?></b></td>
        <td width="15%"></td>
    </tr>
    <?php
    $userId = secureDataToDB($_SESSION["userId"]);
    userUpdateTrackCodes($userId);
    $tracks = $db->get_results("select
trackId,trackCode,trackStatus,trackUpdates,trackLastUpdate,companyName,utLabel from `Tracks` LEFT JOIN
`Companies` on (`Companies`.companyId = `Tracks`.companies_companyId) LEFT JOIN `Users_has_Tracks` on
(`Users_has_Tracks`.Tracks_trackId = `Tracks`.trackId) where `Users_has_Tracks`.Users_userId = '$userId'
order by trackLastUpdate desc, utLabel");
    $bgcolor = "#E0EEEE";
    if(!empty($tracks)) {
        foreach($tracks as $track) {
            $ti = $track->trackId;
            $tc = $track->trackCode;
            $ts = $track->trackStatus;
            $tu = $track->trackUpdates;
            $cn = $track->companyName;
            $label = $track->utLabel;
            $tlu = new DateTime($track->trackLastUpdate);

```



```
</body>
</html>
```

/home/x-warrior/Desktop/Keep-Track/notadd.php

```
<?php
@session_start();
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";

if($db == null) {
    $db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
    $db->query("SET CHARACTER SET utf8 ");
}

if (isLoggedIn()) {
    $userId = $_SESSION['userId'];
    $trackId = secureDataToDB($_GET['track']);
    $email = secureDataToDB($_GET['e']);
    addNotification($trackId, $userId, $email);
}
```

/home/x-warrior/Desktop/Keep-Track/notLoggedMenu.php

```
<style>
.loginForm {
    position: absolute;
    top: 0px;
    left: 70%;
    height: 100px;
    width: 20%;
    text-align: center;
    background-color: #63B8FF;
    -webkit-border-bottom-right-radius: 10px;
    -webkit-border-bottom-left-radius: 10px;
    -moz-border-radius-bottomright: 10px;
    -moz-border-radius-bottomleft: 10px;
    border-bottom-right-radius: 10px;
    border-bottom-left-radius: 10px;
    -moz-border-radius-bottomright: 10px;
    -moz-border-radius-bottomleft: 10px;
    border-bottom-right-radius: 10px;
    border-bottom-left-radius: 10px;
    z-index: 10;
}

.loginButton {
    position: absolute;
    background-color: #63B8FF;
    top: 0px;
    width: 20%;
    left: 70%;
    text-align: center;
    -webkit-border-bottom-right-radius: 10px;
    -webkit-border-bottom-left-radius: 10px;
    -moz-border-radius-bottomright: 10px;
    -moz-border-radius-bottomleft: 10px;
    border-bottom-right-radius: 10px;
    border-bottom-left-radius: 10px;
    z-index: 10;
}

input {
    width: 90%;
}
```

```
}

</style>
<script>
function closeLoginForm(key) {
    if ($("#loginForm").is(":visible")) {
        showLoginButton();
        $("#spotlight").remove();
    }
}

$(document).ready(function() {
    $("#loginForm").hide();
    $("#login").click(function() {
        showLoginForm();
        return false;
    });
    $("#loginFormClose").click(function() {
        closeLoginForm();
    });

    $("#email").focus(function() {
        if ($("#email").val() == "<?php echo _('E-mail'); ?>") {
            $("#email").val("");
        }
    });

    $("#email").focusout(function() {
        if ($("#email").val() == "") {
            $("#email").val("<?php echo _('E-mail'); ?>");
        }
    });

    $(document).keypress(function(event) {
        buttonHandler(event.keyCode);
    });

function showLoginButton() {
    $("#loginForm").hide();
    $("#login").show();
    $("#loginButton").focus();
}

function showLoginForm() {
    $("#login").hide();
    $("#loginForm").show();
    $("#loginForm").focus();
    $("#loginForm").spotlight({color: '#B0E2FF', exitEvent: 'click', animate:false});
    $("#spotlight").click(function() {
        closeLoginForm();
    });
}

function buttonHandler(key) {
    if (key == 27) {
        closeLoginForm();
    }
}
</script>
<div id="login" class="loginButton">
    <a href="#"><?php echo _('Login'); ?></a>
</div>
```

```

<div id="loginForm" class="loginForm">
  <br />
  <form name="log" action="?act=login" method="post">
    <input type="text" name="email" id="email" value="<?php echo _('E-mail'); ?>" /><br /> <input
      type="password" name="password" id="password" value="1234" /><br />
    <button id="loginButton" type="submit" value="Submit"><?php echo _('Login'); ?></button><br />
  </form>
</div>

```

/home/x-warrior/Desktop/Keep-Track/page.php

```

<?php include_once "head.php"; ?>
<style>
.loginMenu {
  position: absolute;
  background-color: #63B8FF;
  top: 0px;
  width: 35%;
  left: 60%;
  text-align: center;
  -webkit-border-bottom-right-radius: 10px;
  -webkit-border-bottom-left-radius: 10px;
  -moz-border-radius-bottomright: 10px;
  -moz-border-radius-bottomleft: 10px;
  border-bottom-right-radius: 10px;
  border-bottom-left-radius: 10px;
}

.content {
  position: absolute;
  background-color: #E0EEEE;
  top: 60px;
  width: 70%;
  left: 15%;
  text-align: center;
  -webkit-border-bottom-right-radius: 10px;
  -webkit-border-bottom-left-radius: 10px;
  -moz-border-radius-bottomright: 10px;
  -moz-border-radius-bottomleft: 10px;
  border-bottom-right-radius: 10px;
  border-bottom-left-radius: 10px;
  padding: 5px;
}

.top {
  position: absolute;
  width: 94%;
  background-color: #B0E2FF;
  top: 0px;
  left: 3%;
  padding-left: 10px;
  -webkit-border-bottom-right-radius: 10px;
  -webkit-border-bottom-left-radius: 10px;
  -moz-border-radius-bottomright: 10px;
  -moz-border-radius-bottomleft: 10px;
  border-bottom-right-radius: 10px;
  border-bottom-left-radius: 10px;
  height: 60px;
  z-index: -1;
}
</style>
<script
  type="text/javascript"
  src="http://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
function extraInfo(t, ti, tc) {

```

```

    if(! $(t).hasClass("clicked")) {
        if (! $(t).next().hasClass("extra-" + ti + "-child")) {
            $(t).after("<tr class=\"extra-\"+ ti +\"-child\"><td colspan='6'><hr /><img src='images/icons/ajax-loader.gif' /> <?php echo _('Loading'); ?><hr /></td></tr>");
            $.ajax({
                data: "tc=" + tc,
                url: "infos.php",
                cache: false,
                contentType: "charset=utf-8",
                scriptCharset: "charset=utf-8",
                dataType: "html",
                success: function(html){
                    if($(t).next().hasClass("extra-" + ti + "-child")) {
                        $(t).next().remove();
                    }
                    $(t).after("<tr class=\"extra-\"+ ti +\"-child\"><td colspan='6'><hr /> " +
html + "<hr /></td></tr>");
                }
            });

            $(t).addClass("clicked");
            $(t + "-img").attr("src", "images/icons/collapse.png");
            $(t + "-img").attr("alt", "<?php echo _('Collapse'); ?>");
        }
    } else {
        $(t + "-child").remove();
        $(t).removeClass("clicked");
        $(t + "-img").attr("src", "images/icons/expand.png");
        $(t + "-img").attr("alt", "<?php echo _('Expand'); ?>");
    }
    return false;
    //$('#extrainfo').dialog('open');
}

function del(c, ti, tc) {
    $.ajax({
        data: "track=" + ti,
        url: "delete.php",
        cache: false,
        success: function(html){
            $(c + '-child').css('background-color', '#CD5555');
            $(c).css('background-color', '#CD5555');
            $(c + '-child').animate({ opacity: "hide" }, "slow", "linear", function() { $(c + '-child').remove(); });
            $(c).animate({ opacity: "hide" }, "1000", "linear", function() {
                $(c).remove();
                $('.tracks').find("tr").each(function(index,obj) {
                    if($(obj).hasClass("track")) {
                        if(index > 0) {
                            cor = (lastColor == "#E0EEEE") ? "#B0E2FF" : "#E0EEEE";
                            $(obj).attr('bgcolor', cor);
                        }
                        lastColor = $(obj).attr('bgcolor');
                    }
                });
            });
        }
    });
}

return false;
}

//GMAPS
var geocoder;
function initialize(canvas, snips, coordinates) {
    geocoder = new google.maps.Geocoder();
    addLocation(snips, 0, coordinates, canvas);
}
function addNot(trackCode, trackId) {

```

```

email = $(".note-" + trackCode).val();
$.ajax({
  data: "track=" + trackId + "&e=" + email,
  url: "notadd.php",
  cache: false,
  success: function(html) {
    $(".not-empty").before("<tr><td>" + email + "</td></tr>");
  }
});
}
function edit_label(trackId, trackCode, user) {
  eval("label = $editLabel" + trackId + ";");
  if(label == trackCode) {
    label = "";
  }
  if($(".track-" + trackId).hasClass('edit-mode')) {
    if(label.length > 0) {
      val = label;
    } else {
      val = trackCode;
    }
    $(".track-" + trackId).html(val);
    $(".track-" + trackId).removeClass('edit-mode');
  } else {
    if (label.length > 0) {
      val = label;
    } else {
      val = "<? echo _("Label"); ?>";
    }
    $(".track-" + trackId).addClass('edit-mode');
    $(".track-" + trackId).html("<input class='track-edit-" + trackId + "' alt=\"<? echo _("Label") ?>
\" maxLength='45' size='15' type='text' name='label' onkeyup='save_label(event.keyCode,
this.value, '" + trackId + "', '" + user + "', '" + trackCode + "');\" value=\"" + val + "\"/>");
  }
}
}

function save_label(key, value, trackId, user, trackCode) {
  if(key == 13) {
    value = encodeURI(value);
    $.ajax({
      contentType: "charset=utf-8",
      scriptCharset: "charset=utf-8",
      data: "t=" + trackId + "&u=" + user + "&l=" + value + "&tc=" + trackCode,
      url: "edit.php",
      cache: false,
      success: function(html){
        $(".track-" + trackId).html(html);
        eval("$editLabel" + trackId + " = \"" + html + "\"");
        $(".track-" + trackId).removeClass('edit-mode');
      }
    });
  }
}

function unique(array) {
  var a = [];
  var l = array.length;
  for(var i=0; i<l; i++) {
    for(var j=i+1; j<l; j++) {
      if (array[i].lat() == array[j].lat() && array[i].lng() == array[j].lng())
        j = ++i;
    }
    a.push(array[i]);
  }
  return a;
}
}

```

```
function createMap(canvas, lat, lng) {
  var myLatLng = new google.maps.LatLng(posLat, posLng);
  var myOptions = {
    zoom:5,
    center: myLatLng,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  }

  map = new google.maps.Map(document.getElementById(canvas), myOptions);
  return map;
}

function drawLine(map, coordinates, color) {
  var trackPath = new google.maps.Polyline({
    path: coordinates,
    strokeColor: color,
    strokeOpacity: 1.0,
    strokeWeight: 5
  });

  trackPath.setMap(map);
}

function createDonePath(map, coordinates) {
  drawLine(map, coordinates, "#66FF00");
}

function createNotDonePath(map, coordinates) {
  drawLine(map, coordinates, "#FF0000");
}

function postinit(snips, canvas, coordinates) {

  coordinates = unique(coordinates);
  posLat = 0;
  posLng = 0;
  size = coordinates.length;
  for (i=0; i < size; i++) {
    posLat = posLat + coordinates[i].lat();
    posLng = posLng + coordinates[i].lng();
  }
  posLat = posLat / size;
  posLng = posLng / size;

  map = createMap(canvas, posLat, posLng);

  createDonePath(map, coordinates);

  coordinatesGoingTo = [];

  if(typeof snips[0]["goingTo"] != "undefined") {
    coordinatesGoingTo.push(coordinates[0]);
    address = snips[0]["goingTo"]["city"] + " - " + snips[0]["goingTo"]["state"] + " - " + snips[0]
["goingTo"]["country"];
    geocoder.geocode( { 'address': address}, function(results, status) {
      if (status == google.maps.GeocoderStatus.OK) {
        coordinatesGoingTo.push(results[0].geometry.location);
        createNotDonePath(map, coordinatesGoingTo);
      }
    });
  }
}

function addLocation(snips, index, coordinates, canvas) {
  if(coordinates.length == snips.length) {
```





```

}

?>
<script
  type="text/javascript" src="https://www.google.com/jsapi"></script>
<script type="text/javascript">

  // Load the Visualization API and the piechart package.
  google.load('visualization', '1.0', {'packages':['corechart']});

  // Set a callback to run when the Google Visualization API is loaded.
  google.setOnLoadCallback(drawChart);

  // Callback that creates and populates a data table,
  // instantiates the pie chart, passes in the data and
  // draws it.

  function drawChart() {
    <?php
      for ($i = 0; $i < count($PERGUNTAS); $i++) {
        ?>
          var data = google.visualization.arrayToDataTable([
            ['Task', 'Clients',
            <?php
              $total = 0;
              foreach ($perguntas[$i] as $name => $value) {
                echo " ['".gettext(ucfirst($name))."', $value],";
                $total = $total + $value;
              }
            ?>
              ]);
          new google.visualization.PieChart(document.getElementById('visualization<?php echo
$i; ?>')).
            draw(data, {title:"<?php echo _($PERGUNTAS[$i]) . " (" . $total . ")"; ?>",
backgroundColor:'#E0EEEE', sliceVisibilityThreshold:0});
          <?php
            }
          ?>

          // Create and draw the visualization.

        }
      }
    </script>
    <?php
      for($i = 0; $i < count($PERGUNTAS); $i++) {
        echo '<div id="visualization' . $i . '"></div>';
      }
    ?>

/home/x-warrior/Desktop/Keep-Track/survey.php

<?php
@session_start();
include_once "config.php";
include_once "includes/ez_sql_core.php";
include_once "includes/ez_sql_mysql.php";
include_once "functions.php";

if($db == null) {
  $db = new ezSQL_mysql($dbUser,$dbPass,$dbName,$dbHost);
  $db->query("SET CHARACTER SET utf8 ");
}

$trackId = intval($_POST['trackid']);
$nid = intval($_POST['notId']);

```

```

$userId = intval($_POST['userId']);
if (is_integer($trackId) && is_integer($nid) && is_integer($userId)) {
    //$res = $db->get_row("select Users_userId from Notifications where notId = $nid and Tracks_trackId =
$trackid and Users_userId = $userId");
    //$userId = $res->Users_userId;
    //$res = $db->get_row("select userEmail from Users where userId = '$userId'");
    //$email = $res->userEmail;

    $array = Array();
    for ($i = 0; $i < count($PERGUNTAS); $i++) {
        $str = "qst" . $i;
        $array[$i] = $_POST[$str];
    }
    $data = serialize($array);
    $db->query("update Notifications set notAnswer = '$data' where notId = $nid and Tracks_trackId =
$trackid and Users_userId = $userId");

    echo _("Survey Sent");
}
?>

```

/home/x-warrior/Desktop/Keep-Track/userConfig.php

```

<div align="center">
<form method="post" action="?act=setconfig">
<table>
<tr>
<td><? echo _("E-mail"); ?></td><td><input type="text" name="login" value="<? echo $_SESSION
["user"]; ?>" disabled maxlength="45"/></td>
</tr>
<tr>
<td><? echo _("Password"); ?></td><td><input type="password" name="password" /></td>
</tr>
<tr>
<td><? echo _("Password Confirmation"); ?></td><td><input type="password" name="password_conf" /></td>
</tr>
<tr>
<td><? echo _("User Name"); ?></td><td><input type="text" value="<? echo getUsername($_SESSION
["user"]); ?>" maxlength="45" name="username" /></td>
</tr>
<tr>
<td><? echo _("Old Password"); ?></td><td><input type="password" name="old_password" /></td>
</tr>
<tr>
<td colspan="2" align="center"><input type="submit" value="<? echo _("Update information"); ?>" /></td>
</tr>
</table>
</form>
</div>

```

/home/x-warrior/Desktop/Keep-Track/userSignUp.php

```

<div align="center">
<form method="post" action="?act=signupuser">
<table>
<tr>
<td><? echo _("E-mail"); ?></td><td><input type="text" value="<? echo $_POST['email']; ?>"
name="email" maxlength="45"/></td>
</tr>
<tr>
<td><? echo _("Password"); ?></td><td><input type="password" name="password" /></td>
</tr>
<tr>

```

```
<td><? echo _("Password Confirmation"); ?></td><td><input type="password" name="password_conf" /></td>
</tr>
</tr>
<tr>
  <td><? echo _("User Name"); ?></td><td><input type="text" value="<? echo $_POST['username']; ?>"
maxlength="45" name="username" /></td>
</tr>
<tr>
  <td colspan="2" align="center"><input type="submit" value="<? echo _("Sign up"); ?>" /></td>
</tr>
</table>
</form>
</div>
```