

TIAGO VELLOSO SILVA

Sistema de Georreferenciamento para Auxílio aos Usuários de Ônibus

Monografia apresentada à Banca examinadora do curso de Engenharia da Computação da FATECS – Faculdade de Tecnologia e Ciências Sociais Aplicadas – Centro Universitário de Brasília – UniCEUB, como requisito para obtenção do título de Engenheiro da Computação, sob a orientação do Professor Miguel Archanjo.

**Brasília
2009**

*Dedico o presente trabalho à
minha mais que amada família.*

Agradeço meus amigos, minha mãe, meu pai, meus avôs e minha namorada pelo auxílio nas horas difíceis.

RESUMO

O transporte público coletivo constitui-se em uma das preocupações primordiais em qualquer cidade.

A dependência pela espera de ônibus, em pontos de parada, sem que o cidadão disponha de qualquer informação ou expectativa a respeito do provável tempo de demora é um dos fatores que direta ou indiretamente contribui para maior complexidade no trânsito.

Utilizando-se dos recursos da moderna tecnologia da informação é possível implementar um sistema que forneça dados precisos a respeito do tempo e da distância que qualquer ônibus se encontra de determinado ponto, em cada instante.

Este projeto tem como objetivo calcular a distância e o tempo aproximados para que determinado ônibus chegue a um ponto informado, apresentando-os, aos usuários, em forma de mapas interativos ou textos estáticos.

A solução consiste no uso de um receptor GPS que registra a posição do ônibus a cada segundo e, a transmissão a cada dez segundos, para a base de dados. É possível que os usuários do transporte público realizem pesquisas, por intermédio de um navegador web.

A solução proposta foi desenvolvida na linguagem de programação PHP fazendo o uso de um banco de dados MySQL.

Palavras-chave: Sistema de Posicionamento Global. GPS. Transporte público Coletivo. Sistema Web. Google Maps. Goops.

ABSTRACT

The public transportation is one of the prime concerns in any city all over the world. Submitting person to wait for a bus, in a bus stop, without having any kind of information or expectation about the probable time of the bus arrival, is one of the factors that direct or indirectly contributes to the urban traffic's complexity.

Using computer modern technologies resources its is possible to develop a system which provides accurate data about buses schedule and distance, time by time, regarding a certain point.

The objective of the present project is to calculate the distance and the approximate time that a certain bus will arrive in a known point, presenting the information to the user by using interactive maps or static texts.

The solution consists in using a GPS receptor which registers the bus position each second and transmits the data each ten seconds to the database. It is possible that public transportation's users do researchers by using a web navigator.

The proposed solution was developed using PHP programming language and using MySQL database.

Palavras-chave: Global Positioning System. GPS. Public Transportation. Web System. Google Maps. Goops.

LISTA DE ILUSTRAÇÕES

Figura 1 - Constelação de Satélites GPS. Fonte: Posicionamento pelo NAVSTAR-GPS, 2000.	26
Figura 2 - Segmentos do GPS. Modificado de: Posicionamento pelo NAVSTAR-GPS, 2000.	27
Figura 3 - Exemplificação da Equação. Fonte: Electromagnetics for Engineers, 2005.	30
Figura 4 - Exemplo de um arquivo .kml. Fonte: Web 2.0 e Mashups, 2007.....	32
Figura 5 - Arquivo .kml executado no Google Maps. Fonte: Web 2.0 e Mashups, 2007.....	33
Figura 6 - Estrutura do projeto. Fonte: Autor.	40
Figura 7 - Modelo de Dados. Fonte: Autor.	41
Figura 8 - Estrutura de Pacotes do Sistema. Fonte: Autor.	43
Figura 9 - Tela inicial do sistema. Fonte: Autor.	44
Figura 10 - Diagrama de Caso de Uso. Fonte: Autor.	45
Figura 11 - Tela de pesquisa da funcionalidade "Pesquisar via Texto". Fonte: Autor.	46
Figura 12 - Tela de Resultado. Fonte: Autor.	47
Figura 13 - Fragmento do código para chamar a função distTempo.js. Fonte: Autor.	48
Figura 14 - Função distTempo que calcula o tempo e a distância aproximados. Fonte: Autor.	48
Figura 15 - Tela do mapa interativo. Fonte: Autor.	49
Figura 16 - Tela de pesquisa da funcionalidade "Pesquisar via Mapa". Fonte: Autor.	50
Figura 17 - Fragmento do código que disponibilizará o mapa interativo. Fonte: Autor.	51
Figura 18 - Comunicação Processamento Automático. Fonte: Autor.	52
Figura 19 – Gráfico de latitudes das avenidas W3 e L2. Fonte: Autor.	56
Figura 20 - Gráfico de longitudes das avenidas W3 e L2. Fonte: Autor.	57
Figura 21 - Gráfico de latitudes das avenidas W3 e L2 particionadas. Fonte: Autor.	58
Figura 22 - Gráfico de longitudes das avenidas W3 e L2 particionadas. Fonte: Autor.	58
Figura 23 - Dados Cadastrados no Banco de Dados. Fonte: Autor.	59

Figura 24 - Dados da tabela referente a posição dos ônibus. Fonte: Autor.	60
Figura 25 - Mapa com a marcação dos pontos e ônibus. Fonte: Autor.	61
Figura 26 - Dispositivos para obtenção das Coordenadas. Fonte: Autor.	63

LISTA DE TABELAS

Tabela 1 - Exemplos de transporte privado e público.....	17
Tabela 2 - Tabela de custos.....	39
Tabela 3 - Arquitetura do projeto.....	53
Tabela 4 - Testes e Resultados com dados fixos.	62
Tabela 5 - Coordenadas obtidas através do GPS.	64
Tabela 6 - Resultado das pesquisas com o ônibus em movimento.	65

LISTA DE ABREVIATURAS E SIGLAS

ABREVIATURAS

Etc. por et cetera

Obs. por observação

SIGLAS

PHP – Hypertext PreProcessor

GPS – Global Positioning System

URL – Uniform Resource Locator

HTTP – Hypertext Transfer Protocol

MVC – Model-View-Controller

JS – JavaScript

CSS – Cascading Style Sheets

KML – Keyhole Markup Language

XML – Extensible Markup Language

API – Application Programming Interface

FTP – File Transfer Protocol

SUMÁRIO

INTRODUÇÃO	13
<i>Preposto</i>	13
<i>Motivação</i>	13
<i>Objetivos</i>	13
<i>Estrutura da Monografia</i>	14
O TRANSPORTE COLETIVO	16
<i>Preposto</i>	16
1.1 História do transporte	16
1.2 Modalidades de transporte	17
1.3 Transporte coletivo	18
1.3.1 Definições	19
1.4 Serviços	19
1.4.1 Serviços de ônibus	19
1.4.2 Serviços de micro-ônibus.....	20
1.5 O transporte coletivo em Brasília	21
FUNDAMENTOS SOBRE GPS	22
<i>Preposto</i>	22
2.1 Sistemas de Navegação por Satélites	22
2.2 Glonass - Global Navigation Satellite System	23
2.3 Galileo	24
2.4 GPS - Global Position System	25
2.4.1 Segmentos GPS	26
2.5 Latitude e Longitude	27
2.6 Obtendo a Posição	28
2.7 Navegação	30

2.8 NMEA	31
2.9 Padrão KML - Keyhole Markup Language	32
APLICAÇÃO PARA A WEB	34
<i>Preposto</i>	34
3.1 Linguagem de programação PHP.....	34
3.2 Servidor Apache.....	35
3.3 MySQL.....	35
3.4 Arquitetura de um sítio da WEB.....	36
3.4.1 Camada Lógica de Apresentação – Visão.....	36
3.4.2 Camada Lógica de Negócio - Controle.....	37
3.4.3 Camada Lógica de Integração - Modelo	37
SISTEMA DE GEORREFENCIAMENTO PARA USUÁRIOS DE ÔNIBUS.....	38
<i>Preposto</i>	38
4.1 Sobre o Sistema SAOU	38
4.2 Receptor	39
4.3 Tratamento das informações.....	40
4.3.1 Transmissão	41
4.3.2 Cadastramento	42
4.3.3 Organização Estrutural	42
4.4 Funcionalidades.....	44
4.4.1 Pesquisa Via Texto	45
4.4.2 Pesquisa Via Mapa	49
4.4.3 Processamento Automático.....	52
4.5 Arquitetura	53
4.6 Acesso pelos Usuários	54
4.7 Software Goops	54
4.8 API Google Maps	55
4.9 Problemas encontrados.....	55
4.10 Testes e Resultados.....	59

4.10.1 Teste Local.....	60
4.10.2 Teste na rua.....	63
4.11 Limitações do Sistema.....	65
CONCLUSÃO.....	66
Sugestão para projetos futuros	67
REFERÊNCIAS	68
APÊNDICE A – Testes e Resultados com dados fixos.....	70
APÊNDICE B – Coordenadas obtidas através do GPS	72
APÊNDICE C – Resultado das pesquisas com o ônibus em movimento	78
APÊNDICE D – Códigos do programa	79

INTRODUÇÃO

PREPOSTO

Este trabalho de conclusão do curso de Engenharia de Computação propõe um Sistema de Georreferenciamento para Usuários de Ônibus. A motivação e os objetivos gerais e específicos deste projeto serão apresentados a seguir, juntamente com a estrutura do trabalho.

MOTIVAÇÃO

A falta de informações a respeito do correto itinerário dos ônibus é um problema para os usuários do transporte coletivo em Brasília. Os usuários desconhecem, de uma forma precisa, em que horário determinada linha de ônibus passará por um ponto específico.

Hoje, os usuários de ônibus, ao chegarem aos pontos, ficam esperando vários minutos e, em certas circunstâncias, até horas, por determinada linha.

Esse problema e outras situações já vivenciadas – como não saber com exatidão o horário de determinado ônibus ou chegar poucos minutos após o ônibus ter passado – resulta, quase sempre, em perdas significativas de precioso tempo.

Desta maneira, propõe-se desenvolver um sistema de informações baseado, em tecnologia de georreferenciamento, que identifique as melhores opções de deslocamento entre um lugar e outro, utilizando o ônibus como meio de transporte.

OBJETIVOS

Este projeto tem como objetivo desenvolver uma solução para o transporte público coletivo, o ônibus, utilizando as seguintes tecnologias: sistema GPS, banco

de dados MySQL, linguagem de programação PHP, servidor Apache, API do Google Maps e Goops. Sua implementação possibilitará, ao usuário, realizar pesquisas, através de mapas ou listas de pontos, para obtenção de informações a respeito da localização de ônibus, suas distâncias e tempos aproximados para chegar aos pontos informados.

Como objetivos específicos podem-se citar:

- a) selecionar um receptor GPS com características que se adéquem ao projeto;
- b) adquirir os dados do receptor GPS e tratá-los;
- c) criar um sistema, utilizando o serviço web e disponibilizá-lo na internet;
- d) utilizar mapas do Google Maps;
- e) criar um banco de dados com informações como: posição dos ônibus, posição dos pontos de ônibus, nome das avenidas, nome dos bairros, linhas de ônibus e seus itinerários.

ESTRUTURA DA MONOGRAFIA

Esta monografia está estruturada em introdução, quatro capítulos de desenvolvimento e conclusão, da seguinte forma:

Introdução

Aqui apresentam-se a motivação, os objetivos gerais e os específicos e a estrutura da monografia.

Capítulo 1 – O Transporte coletivo

Abordam-se a história do transporte coletivo - desde o período colonial até os dias atuais - as diferentes modalidades de transporte, definições de termos técnicos utilizados no transporte e o transporte coletivo em Brasília.

Capítulo 2 – Fundamentos sobre GPS

Apresenta-se o sistema GPS: um pouco de sua história, as partes que o compõem, seu funcionamento e o padrão de arquivos KML que armazenam as

coordenadas do dispositivo. Demonstra-se, também, como se realizam os cálculos para obtenção da posição de um receptor.

Capítulo 3 – Aplicação para Web

O capítulo apresenta a solução tecnológica sugerida, o papel dos softwares utilizados para a aplicação da Web, a linguagem de programação, o servidor web e o banco de dados e, ainda, como deve ser composta uma arquitetura de software para sistemas web.

Capítulo 4 – Sistema de Georreferenciamento para Usuários de Ônibus

Nesta parte, é descrito como foi realizado o projeto, os padrões utilizados, as formas de implementação, suas funcionalidades, as ferramentas utilizadas, como o Goops e o Google Maps, e a estrutura do projeto.

Conclusão

Na conclusão é apresentado o desfecho do trabalho: se os objetivos definidos foram alcançados; comentam-se as dificuldades enfrentadas e acena-se com indicações para projetos futuros.

O TRANSPORTE COLETIVO PREPOSTO

A necessidade de ir e vir e a liberdade de se locomover - e com segurança -, como preconiza o Código de Trânsito Brasileiro, são duas características fundamentais de nossa civilização.

1.1 História do transporte

Durante o período colonial, a Coroa de Portugal solicitava às autoridades coloniais no Brasil a construção de uma estrada ligando o Sul ao Norte do país com o objetivo de oferecer um meio alternativo de comunicação entre as províncias brasileiras em caso de um bloqueio marítimo por alguma potência estrangeira.

A segunda metade do século XIX foi marcada pela introdução de ferrovias no Brasil, e uma sucessão de planos de viação eram apresentados aos governos, todos eles focando as vias férreas e a navegação fluvial e marítima como a solução para os problemas do isolamento a que se viam submetidas as regiões brasileiras.

Em 1920 os planejadores começaram a pensar em rodovias. Uma vez já serem conhecidas as deficiências estruturais pelas quais padecia o setor ferroviário, tornou-se praticamente inevitável a busca por modos alternativos de transportes de longa distância.

O reconhecimento oficial das rodovias como modalidade prioritária de transporte no Brasil teve de esperar ainda até o início dos anos 50, quando da aprovação de um novo plano nacional de viação, em 1951. O texto do plano claramente especificava que as rodovias, a partir de então, passariam a assumir a "função pioneira, outrora exclusiva das estradas de ferro".

A concepção das rodovias federais brasileiras confere a Brasília o status de ser o centro que irradia as Rodovias Federais. Da BR-010 a BR-080, todas nascem da DF-001 - Estrada Parque do Contorno (EPCT).

Atualmente, o órgão gestor competente pela outorga de permissão e autorização para operacionalização dos serviços de transporte rodoviário interestadual e internacional de passageiros é a Agência Nacional de Transportes Terrestres - ANTT (Agência Nacional de Transportes Terrestres, 2009).

1.2 Modalidades de transporte

O transporte pode ser classificado inicialmente em transporte privado e transporte público. A Tabela 1 exemplifica os tipos de cada transporte.

Tabela 1 - Exemplos de transporte privado e público.

Transporte Privado	Transporte Público	
	Público Privado	Público Coletivo
A pé	Táxis	Ônibus
Bicicleta	Moto táxi e moto frete	Micro-ônibus
Automóvel	Transporte escolar	Metrô
Motocicleta		Barcas
Carroças e veículos com tração animal		

Fonte: A cidade, o transporte e o trânsito, 2005.

O transporte privado é usado apenas pelo proprietário ou por quem ele permitir; é o caso da bicicleta, da motocicleta e do automóvel e seu uso está submetido a regras gerais do Código de Trânsito, com a habilitação dos condutores, as vias que podem usar, onde podem estacionar etc.

O transporte público está à disposição do público, mediante pagamento. Sua principal característica legal é que está submetido a algum tipo de controle por parte do governo, referente, por exemplo, ao tipo de veículo que pode ser usado e à tarifa que pode ser cobrada dos usuários. O caso mais comum é aquele em que muitas pessoas, independentemente de se conhecerem ou não, podem usá-lo, quando ele

se chama coletivo, como nos ônibus, trens e metrô; em outros casos, ele é exclusivo das pessoas que chamam por seus serviços – caso do táxi e do ônibus fretado por empresas para transporte de seus funcionários (Vasconcellos, 2005).

1.3 Transporte coletivo

A Constituição Federal define o transporte coletivo urbano como um serviço público essencial que, como tal, deve ser provido diretamente pelo Estado, ou por particulares sob delegação do poder público responsável (União, estados ou municípios). O ônibus é o principal meio de transporte nas cidades brasileiras. Segundo o sistema de informações desenvolvido pela Associação Nacional de Transportes Públicos (ANTP/2005), uma frota de aproximadamente 93.000 ônibus circula nas cidades brasileiras com mais de 60 mil habitantes, transportando, em média, 1 bilhão de passageiros por mês. Além deste modo predominante, algumas cidades contam com metrô e ferrovias participando do transporte coletivo de passageiros.

O serviço de transporte coletivo é parte fundamental da estrutura de funcionamento das cidades e essencial para a vida da população; como tal precisa ser organizado e gerido pelo poder público, não admitindo riscos de descontinuidade, devendo ser prestado de forma a oferecer melhor atendimento à população com conforto, fluidez e segurança.

O sistema de transporte coletivo deve ser organizado na forma de uma única rede, com os diversos modos de transporte integrados física, operacional e tarifariamente, independentemente de quem os opere, inclusive considerando os serviços sob gestão de outros níveis de governo (municipal, estadual e federal).

O serviço de transporte coletivo deve ser prestado de forma profissional e com uma adequada organização dos processos de trabalho necessários: manutenção da frota, operação de tráfego, controle e administração, segundo condições mínimas estabelecidas na regulamentação.

No Distrito Federal, o DFTRANS é o órgão gestor do STPC – Serviço de Transporte Público Coletivo (Vasconcellos, 2005).

1.3.1 Definições

Serviço – refere-se às diferentes formas de atendimento às necessidades do deslocamento da população;

Linha – refere-se ao serviço de um determinado meio de transporte, com regras operacionais próprias;

Itinerário – refere-se ao trajeto predeterminado a ser percorrido pelos veículos de uma linha;

Terminal – construção para a operação dos ônibus; eles servem para as pessoas trocarem de ônibus e atingirem seu destino final. Os terminais são mais conhecidos popularmente como “Ponto de Ônibus”;

Frota – conjunto de veículos de um mesmo tipo à disposição dos serviços;

Intervalo (headway) – tempo decorrido entre a passagem de dois veículos sucessivos de uma mesma linha;

Tarifa – valor a ser cobrado do usuário para garantir o direito ao transporte em uma determinada itinerário.

Carro – como é conhecido o ônibus.

(Velloso e Vaz, 2005)

1.4 Serviços

1.4.1 Serviços de ônibus

A produção da indústria brasileira, medida pela fabricação de carrocerias para ônibus, situa-se na faixa de 18 mil unidades por ano, sendo que pouco mais da metade é destinada ao transporte urbano.

Há uma grande variedade de modelos no mercado que, entretanto, podem ser agrupados em alguns tipos básicos:

- Ônibus convencionais, os mais utilizados, com capacidade para transportar 80 a 95 passageiros, com nível de conforto de 6 passageiros em pé/m²;
- Ônibus articulados, que comportam entre 125 e 160 passageiros, com o mesmo padrão de serviço.

Em uma faixa intermediária, há modelos que transportam aproximadamente 105 passageiros por carro.

Em algumas cidades circulam modelos especiais, como ônibus articulados e bi-articulados (Velloso e Vaz, 2005).

1.4.2 Serviços de micro-ônibus

Recentemente, tem crescido a utilização de microônibus. Menores, mais versáteis e com custo operacional inferior ao dos ônibus, estes veículos têm se apresentado como uma alternativa para reduzir custos operacionais, atuando em regiões de menor demanda de passageiros, ou para estender a rede de linhas para locais de difícil acesso, onde, pelas condições do sistema viário (ruas estreitas, curvas acentuadas, topografia muito irregular ou pavimento de baixa qualidade), os ônibus não conseguem entrar.

Os microônibus também têm sido empregados na operação de serviços diferenciados, geralmente com tarifa superior a dos ônibus, na tentativa de atrair usuários do transporte individual para o sistema público. Nestes casos, podem apresentar itens de conforto e segurança adicionais (bancos individuais e estofados, ar condicionado, por exemplo) e não permitir o transporte de passageiros em pé, além de permitir uma relativa flexibilidade de itinerário ao longo do percurso (Velloso e Vaz, 2005).

1.5 O transporte coletivo em Brasília

O DFTRANS - Transporte Urbano do Distrito Federal - foi criado no dia 11 de Julho de 2003 pela transformação do Departamento Metropolitano de Transportes Urbanos - DMTU.

O trabalho do DFTRANS é garantir à população um transporte eficiente e seguro, colocando em prática o direito de ir e vir do cidadão, com conforto e no menor tempo possível. O enfoque é o usuário e o objetivo é aumentar a oferta de transporte para que a população seja atendida satisfatoriamente, sempre tendo uma atenção particular com os idosos e portadores de deficiência ou necessidades especiais.

O DFTRANS também tem como atribuições o planejamento das linhas, a avaliação de desempenho, a caracterização da demanda e da oferta de serviços, a elaboração dos estudos dos custos de serviços e dos níveis tarifários, a gestão, o controle, a fiscalização dos serviços públicos de passageiros, a administração e a operação de terminais.

Algumas das propostas do GDF possuem foco na implantação de uma nova concepção de operação do sistema de transporte público coletivo, fundamentada na ideia de integração entre itinerários de ônibus e metrô. As intervenções físicas contextualizam-se como um meio de viabilizar o modelo de gestão.

Estão previstas construções e melhorias no sistema viário urbano e rodoviário do Distrito Federal, estabelecimento de vias exclusivas para ônibus, construção de terminais de passageiros, implementação da bilhetagem automática e um amplo conjunto de ações no âmbito da melhoria da gestão e do fortalecimento institucional do Sistema de Transporte Urbano do DF (Governo do Distrito Federal, 2009).

FUNDAMENTOS SOBRE GPS PREPOSTO

O GPS é um Sistema de Posicionamento Global para obtenção de localizações de pontos na Terra e suas distâncias em relação a outros pontos. Embora ele seja o mais conhecido, também existem outros semelhantes como: GALILEO projeto iniciado pela União Européia e o GLONASS, desenvolvido pela União Soviética (El-Rabbany, 2002).

2.1 Sistemas de Navegação por Satélites

Global Navigation Satellite Systems (GNSS) é termo genérico em inglês para expressar sistemas de navegação por satélites que permitem a determinação de posicionamento geo-espacial com cobertura global.

Os GNSS permitem que receptores eletrônicos determinem longitude, latitude e altitude de um determinado ponto usando sinais transmitidos por ondas de rádio através de satélites. Receptores na terra em posições fixas também podem ser usados para calcular tempo preciso como referência para experimentos científicos.

O homem sempre buscou diferentes e refinados meios para se localizar na superfície terrestre. Durante muito tempo, a humanidade utilizou-se de recursos naturais como o sol, as estrelas, acidentes topográficos ou a direção predominante dos ventos para se orientar. No entanto, a evolução científica e tecnológica influenciou o desenvolvimento da Geodésia e da Cartografia.

Desenvolvida há cerca de trinta anos, de início para fins essencialmente militares, a navegação por satélites permite captar os sinais emitidos por uma constelação de satélites para determinar com maior exatidão, e a qualquer momento, posições no tempo e no espaço.

O princípio de funcionamento é simples: os satélites são equipados com um relógio atômico que mede o tempo com extrema precisão. Esses satélites emitem sinais personalizados que indicam a hora de partida do satélite. O receptor no solo,

integrado, por exemplo, num telefone móvel, possui, por sua vez, em memória as coordenadas exatas das órbitas de todos os satélites da constelação. Lendo o sinal que lhe chega, o receptor pode, assim, reconhecer o satélite emissor, determinar o tempo que o sinal demorou a chegar até si e calcular a distância que o separa do satélite. Se um receptor no solo receber os sinais de, pelo menos, quatro satélites simultaneamente, pode calcular a sua posição exata.

As observações de satélites artificiais iniciaram-se aproximadamente na década de sessenta e revolucionaram as técnicas de posicionamento. O primeiro sistema de satélites artificiais utilizado foi o TRANSIT, que teve seus fundamentos idealizados com o objetivo de navegação. Teve como idéia básica a localização sobre a Terra de uma antena receptora de sinais, emitidos por satélites artificiais em órbita terrestre, de posições perfeitamente conhecidas.

Os satélites geodésicos são observados com propósitos de navegação e de posicionamento, tendo como objetivo conhecer a posição de um determinado ponto na superfície terrestre. Atualmente, o sistema de posicionamento global por satélite mais conhecido é o sistema americano denominado Global Position System –GPS.

Além do sistema GPS, existem outros: o sistema russo em operação, GLONASS; o sistema europeu GALILEO - que estará em operação a partir de 2010 e o sistema COMPASS de origem chinesa (El-Rabbany, 2002).

2.2 Glonass - Global Navigation Satellite System

Trata-se do Sistema de Posicionamento Global por Satélite russo, e é equivalente ao GPS americano. Conta com uma constelação de 24 satélites divididos em três órbitas.

O GLONASS usa três níveis orbitais com oito satélites em cada nível. Os satélites GLONASS ainda tem um índice maior de falhas do que o GPS devido a estar incompleto, o que faz o sistema menos atrativo para os usuários.

Em 25 de dezembro de 2007 foram lançados a bordo de um foguete os últimos 3 satélites que comporão o sistema. O GLONASS pretende tornar-se um

sistema alternativo ao GPS, pois o governo dos EUA, em algumas ocasiões, o “desliga” para o usuário civil, como aconteceu nas operações militares no Iraque.

O primeiro satélite GLONASS foi lançado em 12 de Outubro de 1982, e tinha apenas objetivos militares. A versão comercial do sistema é de 1993.

Uma constelação completa do GLONASS será composta de 24 satélites em 3 planos orbitais – 8 satélites por plano. Os planos tem a inclinação de 64.8° que é maior que os planos orbitais do GPS (55°) – isto é um benefício para os usuários localizados em latitudes altas (ou baixas) já que os satélites GLONASS viajam muito mais ao norte (ou sul) que os satélites GPS.

Os satélites GLONASS orbitam à uma altitude de 19,100km – mais baixo que a órbita do GPS de 20,200km. Esta órbita mais baixa significa que os satélites GLONASS completam uma em volta da Terra em 11 horas e 15 minutos – comparado às 11 horas e 58 minutos para a órbita do GPS.

Segundo Parkinson e Spilker (2006) o propósito do sistema global de navegação por satélite GLONASS é prover a um número ilimitado de usuários com serviços de posicionamento tridimensional, medida de velocidade e de tempo em qualquer lugar do globo terrestre e em qualquer condição climática (El-Rabbany, 2002).

2.3 Galileo

Trata-se do Sistema de Posicionamento Global por Satélite europeu. Concebido desde o início como um projeto civil, em oposição ao GPS americano e ao GLONASS russo, todos de origem militar.

O Galileo apresenta várias vantagens, entre as quais a de proporcionar maior precisão (ainda a ser confirmado em testes reais), maior segurança (possibilidade de transmitir e confirmar pedidos de ajuda em caso emergência) e menos sujeito a problemas (o sistema tem a capacidade de testar a sua integridade automaticamente). Além disso, o sistema será inter-operável com os outros dois sistemas já existentes, permitindo uma maior cobertura de satélites.

O GALILEO será composto por uma constelação de 30 satélites em órbita a uma altitude de aproximadamente 24.000 quilômetros, dos quais três ficarão em reserva como suplentes caso sejam necessários, e prevê-se a sua entrada em funcionamento em 2013.

Os primeiros sinais Galileo foram transmitidos no dia 12 de Janeiro de 2006 pelo satélite GIOVE-A que tinha sido colocado em órbita a 28 de Dezembro de 2005. O segundo satélite experimental GIOVE-B foi lançado a 27 de Abril de 2008 por um foguetão Soyuz, da base de Baikonur, no Cazaquistão.

A sua gestão e o seu controle serão assegurados por estações terrestres. Operacional a partir de 2008, o GALILEO ajudará, nomeadamente, a resolver os problemas de mobilidade e de transporte com que se confrontam atualmente as várias zonas do planeta. Além disso, graças à compatibilidade e à interoperabilidade dos dois sistemas, GALILEO e GPS, os utilizadores de todo o mundo terão mais facilmente acesso aos sinais emitidos pelos satélites de navegação e se beneficiarão de um desempenho de muito maior qualidade (El-Rabbany, 2002).

2.4 GPS - Global Position System

O Sistema de Posicionamento Global é popularmente conhecido pelo acrónimo GPS e, por intermédio de um conjunto de satélites, fornece a um aparelho receptor a posição do mesmo com referência às coordenadas terrestres.

Mas, antes de ser utilizado em larga escala, o GPS era uma ferramenta de navegação do Departamento de Defesa dos Estados Unidos da América – DoD (Departmento of Defense). Foi desenvolvido pelas forças armadas nos anos 70, e era aplicado unicamente para uso militar.

Atualmente, sua função de navegação é aplicada em diversas áreas civis, como o rastreamento de veículos, aeronaves, o uso de mapas em automóveis e mapeamento topográfico. (Monico, 2000)

O GPS providencia informações contínuas da posição do receptor em qualquer lugar do mundo em que se esteja, sob qualquer condição climática. Por questões de segurança, pois um número ilimitado de usuários utilizam os serviços

de posicionamento, torna o GPS um sistema passivo, isto é, os usuários só podem receber os sinais dos satélites. (El-Rabbany, 2002)

A manipulação dos dados obtidos por meio do receptor GPS permitiu aumentar a produtividade em vários segmentos que necessitam de informações sobre posicionamento em comparação aos métodos convencionais antes utilizados (Monico, 2000).

2.4.1 Segmentos GPS

O sistema GPS é composto por três segmentos principais: espacial, controle e usuário.

O segmento espacial é composto por 24 satélites organizados de modo que quatro satélites fiquem dispostos em seis planos orbitais. Através dessa geometria, de quatro a dez satélites poderão ficar visíveis em qualquer lugar do mundo (El-Rabbany, 2002).

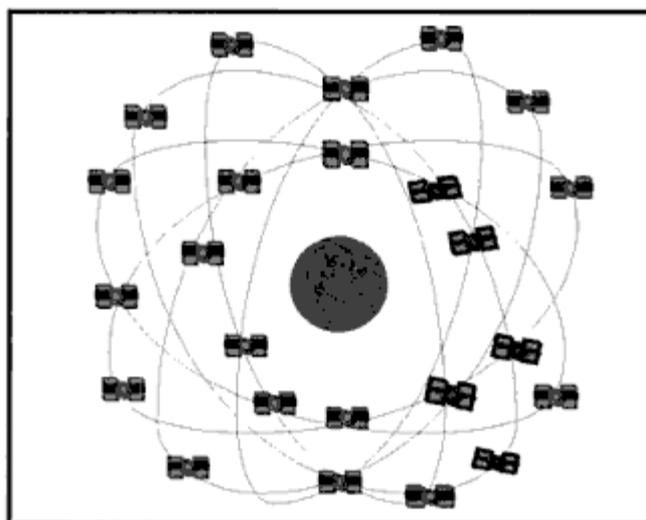


Figura 1 - Constelação de Satélites GPS. Fonte: Posicionamento pelo NAVSTAR-GPS, 2000.

O segmento de controle é composto por cinco estações monitoras espalhadas pelo globo. Essas estações monitoram e controlam o sistema de satélites, calculam as correções dos relógios dos satélites e atualizam periodicamente as mensagens de navegação dos satélites (El-Rabbany, 2002).

E, por último, o segmento do usuário que é composto pelo receptor GPS, que deverá ser adquirido pelo usuário de acordo com a necessidade de uso ou projeto. Hoje existe uma grande quantidade de aparelhos no mercado para as mais diversas aplicações (El-Rabbany, 2002).

A Figura 2 mostra os três segmentos que compõem o GPS.

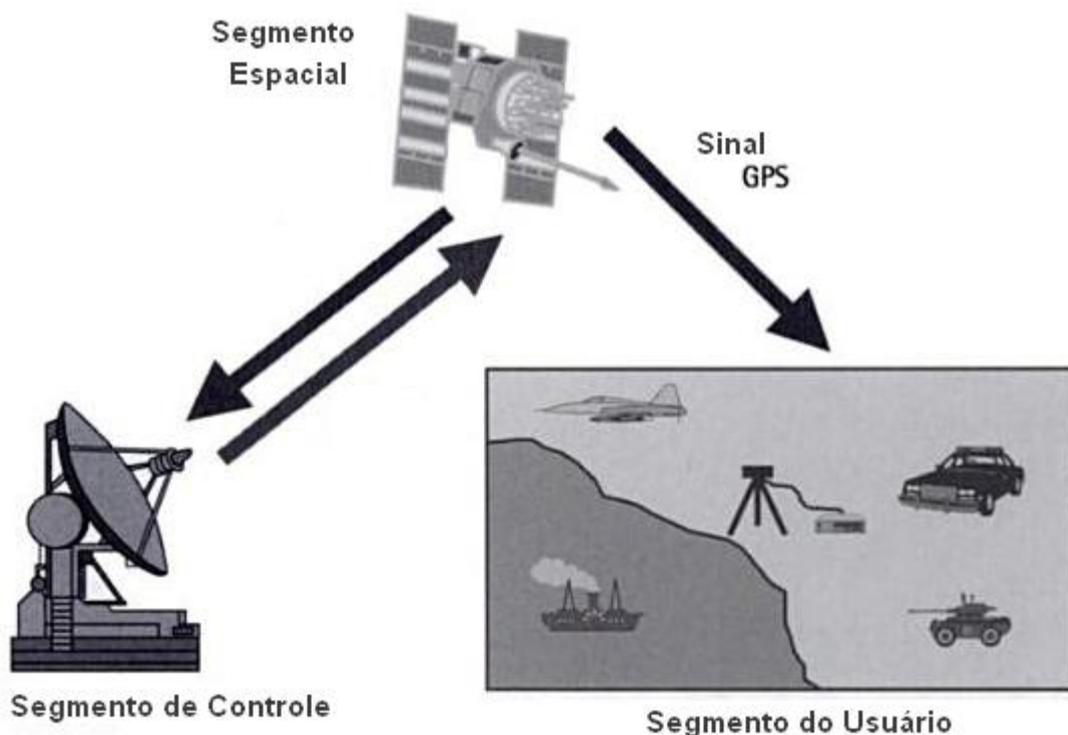


Figura 2 - Segmentos do GPS. Modificado de: Posicionamento pelo NAVSTAR-GPS, 2000.

2.5 Latitude e Longitude

Latitude é o ângulo de um ponto sobre a superfície da Terra para o plano equatorial, medido a partir do centro da esfera. Linhas que unem pontos da mesma latitude são chamadas paralelos, essas linhas traçam círculos concêntricos na superfície da Terra, paralelo ao equador. O Pólo Norte é de 90° N; o pólo sul é de 90° S. O paralelo 0° de latitude é o “equador”, o plano fundamental de todos os sistemas de coordenadas geográficas. O equador divide o globo em hemisférios Norte e Sul.

Longitude é o ângulo leste ou oeste de um meridiano de referência entre os dois pólos geográficos para outro meridiano que passa por um ponto arbitrário.

Todos os meridianos são metades de grandes círculos, e não são paralelas. Eles convergem para os pólos norte e sul. Dessa forma, descreve a localização de um lugar na Terra medido em graus, de zero a 180 para leste ou para oeste, a partir do Meridiano de Greenwich. Diferentemente da latitude, que tem a linha do Equador como um marco inicial natural, não há uma posição inicial natural para marcar a longitude. O Meridiano de Greenwich foi adotado como o primeiro meridiano de conhecimento mundial em 1884 na International Meridian Conference.

Cada grau de longitude é subdividido em 60 minutos, e estes sequencialmente em 60 segundos. Uma longitude é especificada no formato graus^o minutos' segundos". Caso a localidade esteja no oeste coloca-se um sinal negativo (-) na frente da longitude, ou pode-se utilizar as letras "E" e "W" para indicações de Leste e Oeste.

A combinação de latitude e longitude se resulta em uma precisa posição no globo terrestre.

Cada ponto que se expressa em coordenadas esféricas pode ser expressa como uma xyz coordenada cartesiana. Este não é um método útil para a gravação de uma posição nos mapas, mas é usado para calcular as distâncias e realizar outras operações matemáticas. A origem é normalmente o centro da esfera, um ponto próximo ao centro da Terra.

2.6 Obtendo a Posição

Para saber a localização de um corpo, mais especificamente o receptor GPS, é preciso saber quais são suas coordenadas geográficas. Para obter essas informações, o sistema GPS realiza a técnica de triangulação da localização, isto é, conhecendo as coordenadas dos satélites, podemos calcular as três distâncias. Essas distâncias são estabelecidas medindo os tempos que os sinais levam para se deslocar dos satélites até os receptores GPS multiplicados pela velocidade da luz. Esses tempos são determinados de acordo com o tempo de deslocamento ao longo da distância, e é possível identificar comparando o código do receptor com o sinal recebido pelo satélite.

Uma quarta medida é utilizada para corrigir o erro de tempo do receptor, isto é, devido ao não-sincronismo dos relógios dos satélites e o do usuário.

Dessa forma, são geradas as seguintes quatro equações, que ao serem solucionadas, resulta na informação da localização do receptor (x_0, y_0, z_0) e o t_0 que é o desvio de tempo (Ulaby, 2005).

$$d_1^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2 + (z_1 - z_0)^2 = c[(t_1 + t_0)]^2$$

$$d_2^2 = (x_2 - x_0)^2 + (y_2 - y_0)^2 + (z_2 - z_0)^2 = c[(t_2 + t_0)]^2$$

$$d_3^2 = (x_3 - x_0)^2 + (y_3 - y_0)^2 + (z_3 - z_0)^2 = c[(t_3 + t_0)]^2$$

$$d_4^2 = (x_4 - x_0)^2 + (y_4 - y_0)^2 + (z_4 - z_0)^2 = c[(t_4 + t_0)]^2$$

Equação 1 - Equação para obtenção das distâncias entre o receptor e o satélite. Fonte: Electromagnetics for Engineers, 2005

Onde:

x_0, y_0, z_0 = A localização do receptor

$(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$ = A localização dos satélites

d_1, d_2, d_3 = Distância entre o receptor e os satélites

t_1, t_2, t_3 = Atrasos de tempo

$3 \times 10^8 \frac{\text{m}}{\text{s}}$ = Velocidade da luz

A Figura 3 exemplifica, em forma de desenho, de onde surgiram as variáveis das quatro equações.

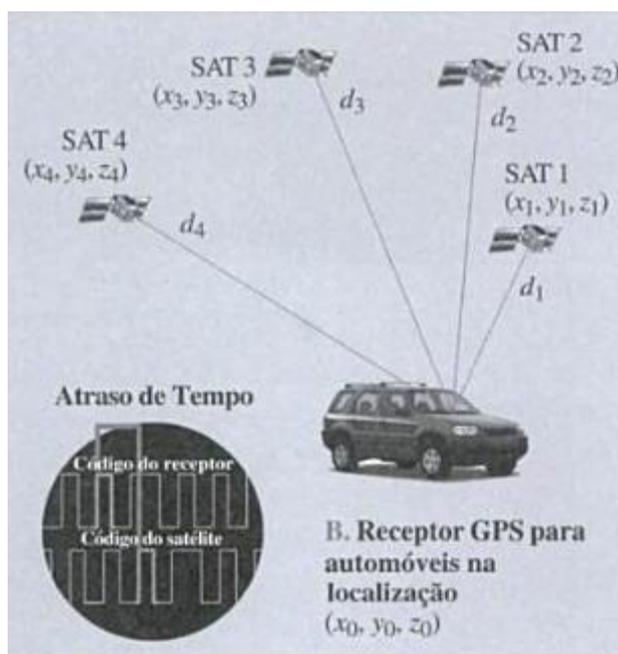


Figura 3 - Exemplificação da Equação. Fonte: Electromagnetics for Engineers, 2005.

2.7 Navegação

A partir do momento que o receptor GPS é ligado, ele começa a receber, de forma ininterrupta, o sinal de pelo menos dois, dos 24 satélites que compõe o sistema GPS e, a partir desse sinal, o receptor combina a latitude para calcular o ponto exato em que o receptor está.

Assim que o usuário determina o ponto a que quer chegar, o software de navegação do dispositivo encontra a posição exata do destino novamente usando a latitude e longitude dele, mas desta vez, com as informações que já estão armazenadas no mapa da cidade em que o usuário está.

Neste momento, o software de navegação combina três informações – as coordenadas do ponto inicial; as informações do mapa roteirizado da cidade, que contem a mão das ruas e a ordem de importância delas; e as coordenadas do ponto final – para criar a rota entre os dois pontos.

Como a comunicação entre o receptor e os satélites GPS é ininterrupta, o sistema tem a todo o momento a posição exata do dispositivo e registra seus deslocamentos. Dessa maneira, ele “sabe” onde o receptor passou e a sua localização atual, possibilitando assim, a navegação até o ponto de destino.

2.8 NMEA

A maior parte dos fabricantes de dispositivos GPS suportam uma interface protocolo padrão denominada "NMEA 0183 interface padrão." Este protocolo foi definido por NMEA (National Marine Electronics Association).

O protocolo NMEA 0183 é um controlador que permite que um programa de software se comunique com um dispositivo GPS. O protocolo NMEA 0183 contém informações sobre a posição, o estado de satélite e qualquer outro estado de navegação. No entanto, nem todos os dispositivos GPS utilizam o protocolo NMEA 0183.

A interface de hardware para GPS é projetado para atender as exigências NMEA. Eles também são compatíveis com a maioria dos portos de série do computador usando protocolos RS232, porém estritamente o padrão NMEA não é RS232. É recomendada a conformidade do EIA-422.

A velocidade da interface pode ser ajustada em alguns modelos, mas o padrão NMEA é de 4800 bits/s (taxa de bits por segundo), com 8 bits de dados, sem paridade, e um bit de parada. Todas as unidades que oferecem suporte NMEA deve apoiar esta velocidade.

Note que, na taxa de 4800 bits/s é possível enviar dados suficientes para preencher mais de uma segunda faixa de tempo. Por esta razão, algumas unidades de apenas enviam atualizações a cada dois segundos ou podem enviar alguns dados a cada segundo, reservando os outros dados a serem enviados com menos frequência.

O padrão NMEA já existe há muitos anos (1983) e passou por várias revisões. O protocolo foi alterado eo número e tipos de frases podem ser diferentes, dependendo da revisão. Algumas unidades Garmin e outras marcas podem ser definidas como sendo de 9600 para a saída NMEA ou mesmo superiores.

2.9 Padrão KML - Keyhole Markup Language

KML é um padrão aberto, oficialmente conhecido como OpenGIS KML Encoding Standard (OGC KML) e é mantido pela Open Geospatial Consortium, Inc. (OGC).

KML é usado para exibir dados geográficos em um navegador da Terra, como Google Earth. O padrão utiliza uma estrutura de tags, que consistem em instruções, tendo uma marca de início e outra de fim, com elementos e atributos aninhados, se baseando no padrão XML. Todas as tags diferenciam maiúsculas e minúsculas e devem aparecer exatamente como listadas na Referência do KML.

O Google Earth e o Google Maps possuem um recurso prático que facilita a visualização do arquivo KML de qualquer arquivo e o software Goops armazena as informações recebidas do receptor GPS nesse padrão de arquivo (Junior, 2007).

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.google.com/earth/kml/2">
<Document>
  <name>toronto.kml</name>
  <Placemark>
    <name>CN Tower</name>
    <description>The CN Tower (Canada's National Tower, Canadian National Tower),
at 553.33 metres (1,815 ft., 5 inches) is the tallest freestanding structure on land.
It is located in the city of Toronto, Ontario, Canada, and is considered the
signature icon of the city. The CN Tower attracts close to two million visitors
annually.

http://en.wikipedia.org/wiki/CN\_Tower</description>
    <Point>
      <coordinates>-79.386864,43.642426</coordinates>
    </Point>
  </Placemark>
</Document>
</kml>
```

Figura 4 - Exemplo de um arquivo .kml. Fonte: Web 2.0 e Mashups, 2007.

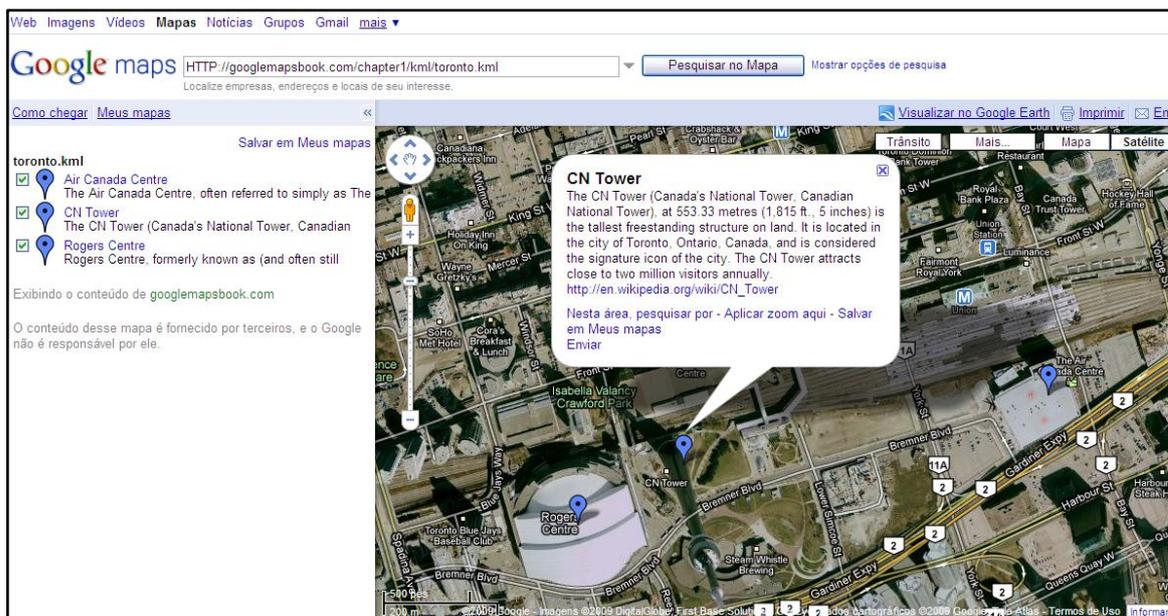


Figura 5 - Arquivo .kml executado no Google Maps. Fonte: Web 2.0 e Mashups, 2007.

APLICAÇÃO PARA A WEB PREPOSTO

Uma aplicação web é uma aplicação de software que utiliza a internet, por intermédio de um browser como ambiente de execução, isto é, trata-se de um conjunto de programas projetados inicialmente para serem utilizados através de um navegador como o Internet Explorer ou o Mozilla Firefox. Esses aplicativos são utilizados pela internet ou em redes privadas, pois são executados em um servidor de HTTP. (Laurie e Laurie, 2002)

3.1 Linguagem de programação PHP

Tecnologia para desenvolvimento de aplicações WEB semelhante ao Microsoft Active Server Pages (ASP), porém tem a vantagem da portabilidade de plataforma podendo ser executado em outros Sistemas Operacionais além dos da Microsoft. Pode ser facilmente codificado, facilitando, assim, a elaboração e a manutenção de uma aplicação.

Podemos dizer que o PHP é a linguagem de programação mais utilizada na Web para o desenvolvimento de páginas avançadas. Com a chegada da quinta versão do PHP, aspectos não utilizados nas versões anteriores podem ser observados, como a programação orientada a objetos (POO).

A programação orientada a objetos é uma maneira de programar que traz os processos de programação para próximo da realidade. Isto é, trata cada componente de um programa como objeto com características e funcionalidades próprias.

O PHP pode atender milhões de acessos por dia através de um servidor como o Apache, sua linguagem é baseada em outras como o C, o Java e o Perl, pode ser portado para diferentes sistemas operacionais como o Linux, Windows e o FreeBSD e é gratuito. (Luke Welling, 2005)

O PHP foi escolhido por possuir licença gratuita e ser uma linguagem muito utilizada na internet, facilitando buscas por soluções em fóruns e artigos e, acima de tudo, ser uma linguagem de fácil aprendizado.

3.2 Servidor Apache

É um software de código aberto, inicialmente desenvolvido por um grupo de 20 programadores voluntários, chamado de Apache Group. Esse programa possui livre distribuição e é muito prestigiado por seu desempenho, segurança e flexibilidade.

Mesmo sendo inicialmente projetado para a plataforma UNIX, hoje, ele funciona em várias plataformas como o Windows e OS2. Dessa forma, ele é o servidor mais utilizado na internet atualmente (Scrimger, 2002).

O objetivo geral do servidor web é traduzir a URL em um arquivo e mandar de volta através da internet ou traduzir em um programa e retornar seu resultado pela internet, recebendo e processando as requisições via protocolo HTTP, e se comunicando através das redes pelo protocolo TCP/IP (Laurie e Laurie, 2002).

O apache foi escolhido, pois é um servidor gratuito, de fácil instalação e que possui suporte ao PHP.

3.3 MySQL

O MySQL é um servidor de banco de dados relacional, SGBD – Sistema de Gerenciamento de Banco de Dados, escrito na linguagem de programação C e C++ que possui dupla licença, isto é, possui uma licença de software livre que é o caso da maioria dos projetos, e outra para atender aplicações de grande porte.

Esse banco de dados foi inicialmente projetado para trabalhar com aplicações que necessitassem no máximo 100 milhões de registros por tabela, contudo, as últimas versões, segundo o autor, já ultrapassaram essa fronteira inúmeras vezes.

Hoje, o MySQL é um banco de dados que permite executar, em um sistema, mais de um bilhão de consultas por dia, ou até mesmo processar milhares de transações por minuto.

Esse SGBD é suportado pela maioria dos sistemas operacionais, é portátil também entre os diferentes sistemas, plataformas, compiladores e fornece uma API para linguagens como Java, PHP, Python, C, C++ entre outras.

A linguagem utilizada é o SQL, suportando execuções de scripts com até 61 uniões de tabelas e várias linhas de código (Milani, 2006).

O MySQL foi escolhido por ser gratuito, de fácil integração com o PHP, possuir inúmeros softwares que fazem a administração desse SGBD e ser de fácil instalação e manutenção.

3.4 Arquitetura de um sítio da WEB

A arquitetura de software refere-se ao projeto e à implementação das estruturas de mais alto nível de abstração do projeto. É o resultado da união de uma série de elementos arquiteturais organizados de modo a satisfazer as principais funcionalidades, requisitos de desempenho e tecnológicos do sistema, assim como outras necessidades como confiabilidade, escalabilidade, portabilidade e disponibilidade.

Um sistema em camadas é organizado hierarquicamente, cada camada fornecendo serviços para camada acima e usando serviços da camada abaixo. Porém, ainda que imperceptível, este sistema reduz o desempenho da aplicação, pois adiciona um overhead e latência ao processamento dos dados.

Podemos dizer que o padrão de modelo mais citado é o MVC – Model View Controller, que consiste em decompor uma aplicação em três partes: o modelo, a visão e o controle. É originalmente aplicado em modelos de interação gráfica com o usuário para entrada, processamento e saída de dados (Fowler, 2006).

3.4.1 Camada Lógica de Apresentação – Visão

A camada de apresentação ou visão é desenvolvida para permitir acesso através da WEB e, quando esta camada está sendo desenvolvida, a preocupação principal é com os mecanismos de interface com o usuário.

Essa camada é utilizada para a apresentação das informações, isto é, com a camada de apresentação, é possível que um mesmo dado recebido do modelo possa ser exibido de formas diferentes ao usuário (Fowler, 2006).

3.4.2 Camada Lógica de Negócio - Controle

A camada de negócio ou controle do sistema é construída pensando nas ações realizadas pela camada de apresentação.

Quaisquer alterações nas informações apresentadas são manipuladas pelo controlador. Ele recebe a entrada do usuário, manipula a camada modelo e atualiza a camada de apresentação.

Essa camada invoca um método da camada de integração e esta retorna uma instância de uma classe apropriada. Esse tipo de estratégia deverá facilitar a incorporação de outras tecnologias e reduzir o acoplamento entre as camadas (Fowler, 2006).

3.4.3 Camada Lógica de Integração - Modelo

A camada de integração ou modelo é construída pensando nas políticas de negócio do projeto. Essa camada é utilizada, basicamente, para interações com o banco de dados; ela define o modo de acesso a dados e é formada por classes que implementam o padrão DAO (Data Access Object), objeto de acesso aos dados. A motivação deste padrão é tornar o código de persistência mais organizado, reutilizável e desacoplado da lógica de negócio (Fowler, 2006).

SISTEMA DE GEORREFENCIAMENTO PARA USUÁRIOS DE ÔNIBUS PREPOSTO

Para o projeto ser acessado via WEB, foi desenvolvido um sistema de pesquisas batizado de SAOU cujo significado é: Sistema de Auxílio a Usuários de Ônibus.

4.1 Sobre o Sistema SAOU

O projeto foi desenvolvido para atender os usuários do transporte público coletivo. Uma pesquisa realizada pela FLC Comunicações, mostra que apenas 4% dos usuários do transporte público são pertencentes a classe D, desta forma, é possível concluir que a grande maioria dos usuários possuem conexão com a internet, seja em casa, no trabalho ou aonde estuda, pois segundo a 20ª Pesquisa Anual da Fundação Getúlio Vargas, o Brasil possui um computador para casa três habitantes, o que equivale 60 milhões de unidades em uso (O Repórter, 2009).

Hoje existem no mercado alguns sites que fornecem um serviço parecido com o desenvolvido pelo projeto, como o site “Vá de Ônibus”, que disponibiliza as informações, através de pesquisas interativas, a respeito de todas as linhas das 37 cidades do Rio de Janeiro (Siqueira, 2009).

O projeto foi pesquisado e desenvolvido utilizando apenas pesquisas bibliográficas, faltando assim um contato direto com as empresas de ônibus para um melhor entendimento de seus pontos de vista a respeito do mesmo.

Portanto o projeto tende a ser apenas um protótipo, evitando atender os interesses das empresas de ônibus. Os custos para a criação do protótipo pode ser visto na Tabela 2.

Tabela 2 - Tabela de custos

Ferramentas	Fonte	Custo
Receptor GPS HOLUX M1000C	Trilha21	R\$ 289,99
Notebook	Submarino	R\$ 1.699,00
Licença do Apache	-	Gratuito
Licença do PHP	-	Gratuito
Licença do MySQL	-	Gratuito
Licença do Goops	-	Gratuito
Licença da API do Google Maps	-	Gratuito
Plano Vivo Internet 3G Ilimitado	Vivo	R\$ 119,90 por mês
Assinatura de um servidor web	Miarroba.com	Gratuito
Total:		R\$ 2108,89

Fonte: Autor

4.2 Receptor

Foi utilizado no projeto o receptor Wireless Bluetooth M-1000C da Holux, que possui uma tecnologia avançada: ele permite armazenar mais de 100 mil coordenadas de latitude e longitude em sua memória interna e sua conexão pode ser feita por USB ou Bluetooth.

Sua tecnologia possibilita procurar por 66 satélites ao mesmo tempo, obtendo atualizações da posição a cada segundo (Manual do Holux M1000C).

4.3 Tratamento das informações

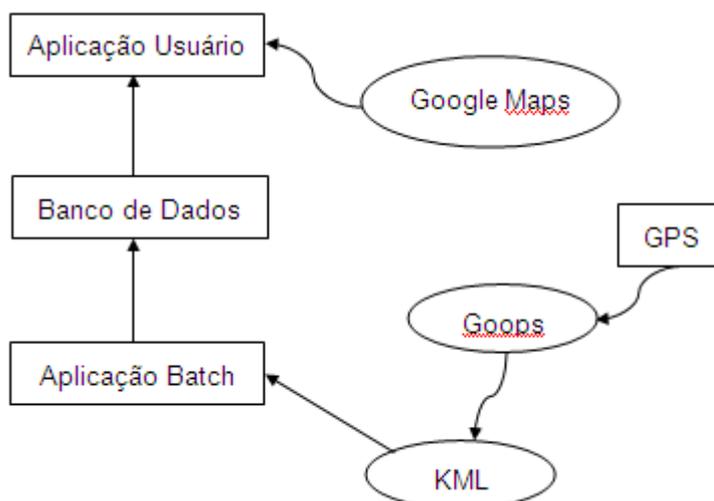


Figura 6 - Estrutura do projeto. Fonte: Autor.

A Figura 6 demonstra como as informações irão trafegar pelo sistema. Podemos notar que projeto possui a aplicação, que terá uma interface amigável para o usuário e se comunicará com as funções do Google Maps para realizar a exibição de mapas interativos.

A aplicação se comunicará com o banco de dados que possuirá dados pré-cadastrados a respeito dos bairros que o sistema irá englobar, avenidas, pontos de ônibus, linha dos ônibus, seus itinerários e posição atual dos mesmos. O modelo relacional pode ser visto na Figura 7.

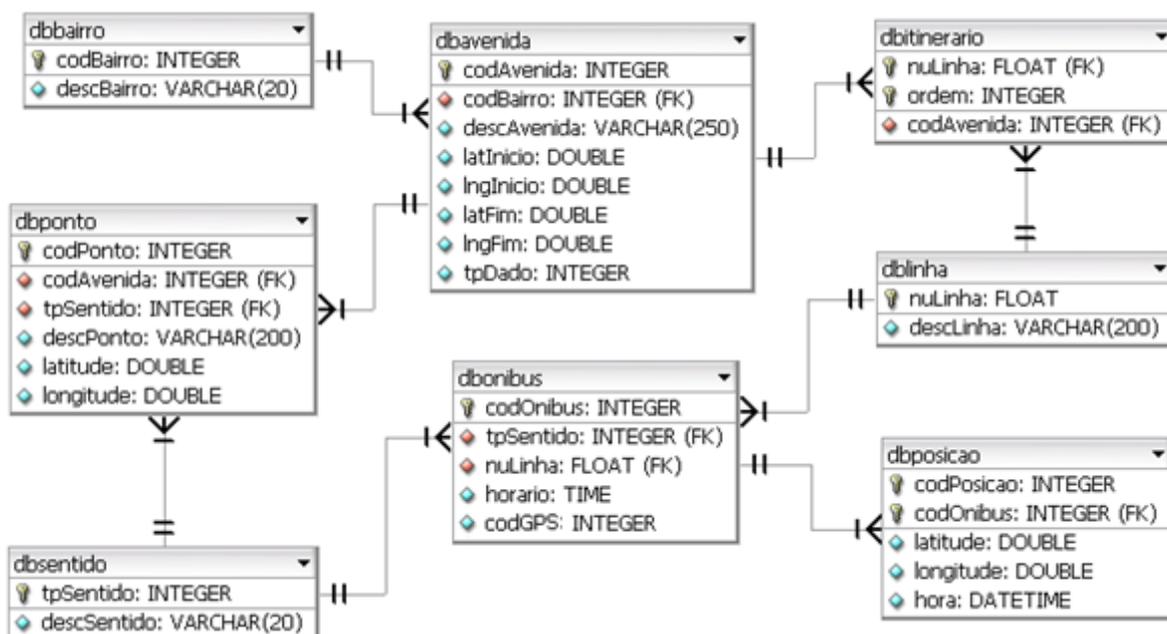


Figura 7 - Modelo de Dados. Fonte: Autor.

A Aplicação Automática, nome recebido por ser a parte automática do sistema, receberá informações do GPS através dos arquivos KML gerados pelo software Goops e as armazenará no banco de dados.

4.3.1 Transmissão

A transmissão das posições marcadas pelo GPS ocorrerá através da comunicação pela porta USB com o notebook.

Um software de distribuição gratuita, chamado Goops, fará a leitura desses dados a cada segundo e realizará a gravação em um arquivo com extensão KML. Esse arquivo será gravado automaticamente no servidor através de um FTP, configurado previamente pelo software.

Para realizar essa transferência via FTP, o notebook possuirá uma conexão com a internet via 3g.

Uma aplicação desenvolvida em PHP realizará a leitura desse arquivo, buscará neste apenas as informações úteis ao sistema e as gravará na tabela referente à posição, no banco de dados MySQL.

O usuário através de uma página web poderá realizar pesquisas, através de mapas interativos ou textos estáticos, e o sistema retornará o resultado de acordo com os parâmetros informados pelo mesmo.

4.3.2 Cadastramento

O cadastramento das informações é feito em dois passos: o primeiro realizado na implantação do sistema, quando obtivemos as primeiras informações a respeito das linhas dos ônibus, seus horários, a localização das paradas; o segundo passo é realizado continuamente através do recebimento das informações de posição dos ônibus que possuem o kit composto pelo notebook, internet 3g, o sistema automático executando e o software Goops.

Esses dados são mantidos em um banco de dados cujo seu modelo de dados foi projetado especificamente para esse projeto e apresentado no tópico 4.2 deste documento.

4.3.3 Organização Estrutural

Na Figura 8 é apresentada a estrutura de pacotes do sistema SAUO.

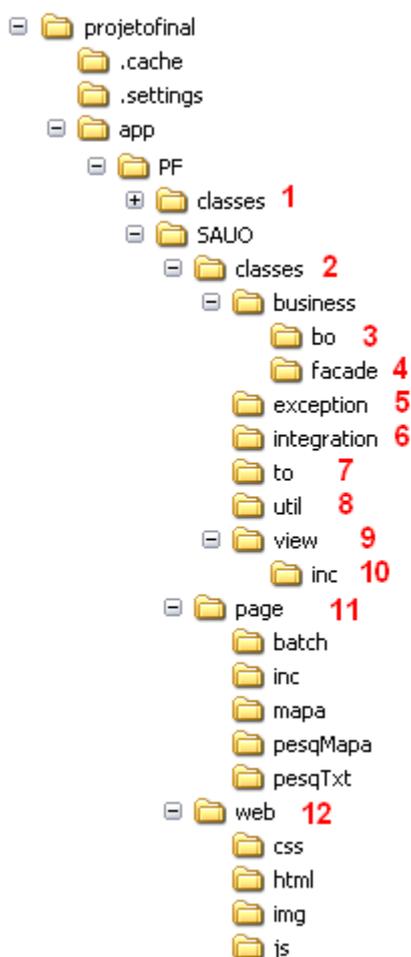


Figura 8 - Estrutura de Pacotes do Sistema. Fonte: Autor.

Foi definida essa estrutura de pacotes para facilitar futuras novas implementações de funcionalidades dentro do sistema, bem como novos sistemas comunicando com o SAUO. A seguir, segue a explicação de cada pasta utilizada pelo sistema demonstrado na Figura 8.

[1] A pasta “classes” poderá armazenar futuras classes caso a estrutura venha possuir outro sistema se comunicando com o SAUO, e estes possuam classes em comum.

[2] Pasta “classes”, porém, classes específicas para o projeto, mas comum a todas as funcionalidades.

[3] Pasta contendo as classes de negócio.

[4] Pasta contendo as classes de fachada.

[5] Pasta contendo as classes exceção do negócio e do sistema, isto é, responsável por tratar as exceções de banco de dados, infra-estrutura e as exceções de negócio do sistema.

[6] Pasta contendo as classes que integram com o banco de dados.

[7] Pasta contendo as classes de objetos.

[8] Pasta que contém funções e classes úteis que facilitem a programação.

[9] Pasta que contém as classes de Action, isto é, as que realizam as ações requeridas pelo sistema ou pelo usuário.

[10] Pasta para armazenamentos dos arquivos de includes.

[11] Pasta com os arquivos PHP de interface com o usuário.

[12] Pasta com arquivos web como imagens, arquivos de estilos (css), funções Java Script (js).

4.4 Funcionalidades

O sistema implementa três funcionalidades, duas para uso do usuário e uma para uso do sistema. As funcionalidades referentes ao usuário poderão ser acessadas através do menu superior.

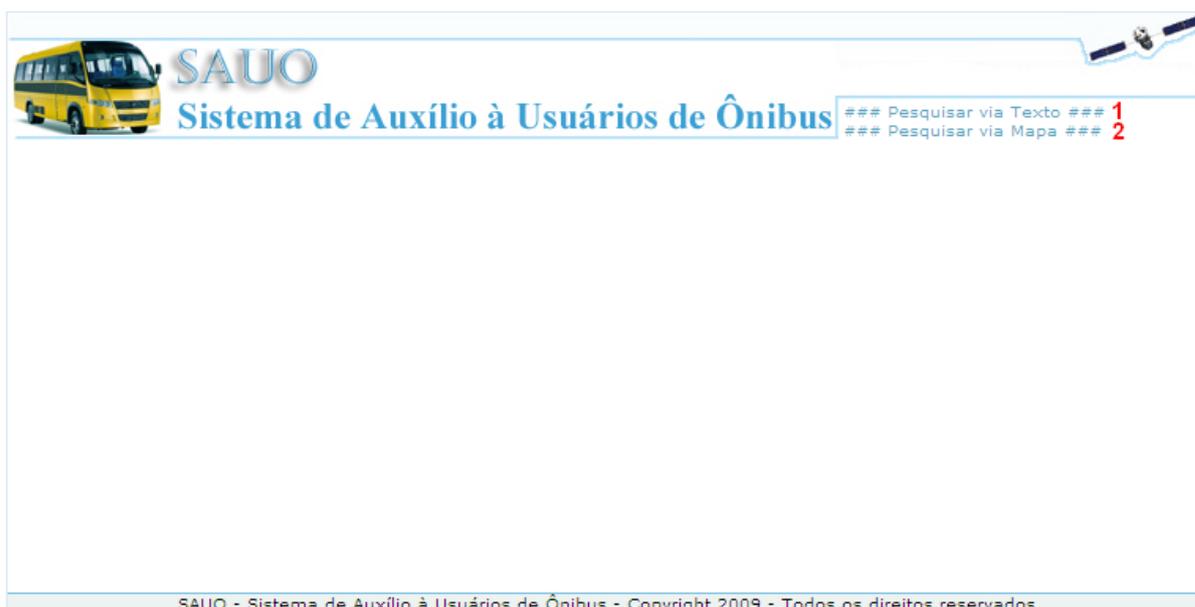


Figura 9 - Tela inicial do sistema. Fonte: Autor.

A figura acima exibe a tela inicial do sistema que possui o link para as duas funcionalidades que poderão ser acessadas pelo usuário. A indicação de número 1 na figura mostra o link da funcionalidade “Pesquisar via Texto” e a indicação de número 2 mostra o link da funcionalidade “Pesquisar via Mapa”.

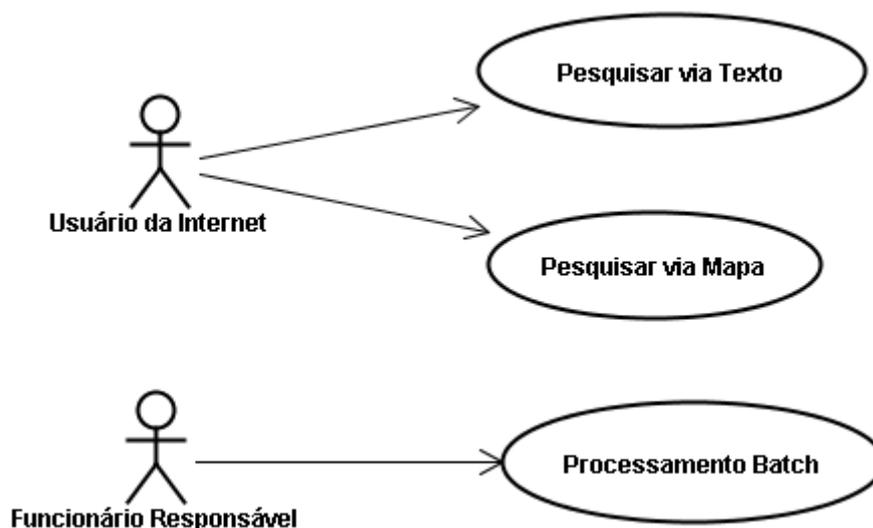


Figura 10 - Diagrama de Caso de Uso. Fonte: Autor.

4.4.1 Pesquisa Via Texto

Essa funcionalidade foi implementada para o usuário realizar a pesquisa da localização de todos os ônibus que farão o trajeto entre o ponto de origem e o ponto de destino informado pelo usuário.

Ao acessar essa funcionalidade pelo link “Pesquisa via Texto” no menu superior, será disponibilizada uma tela com as opções:

- Bairro de Origem;
- Avenida de Origem;
- Ponto de Origem;
- Bairro de Destino;
- Avenida de Destino;
- Ponto de Destino.

SAUO
Sistema de Auxílio à Usuários de Ônibus

Pesquisar via Texto ###
Pesquisar via Mapa

Pesquisar via Texto

Ponto de Origem:

Bairro*: Avenida*: Ponto*:

Ponto de Destino:

Bairro*: Avenida*: Ponto*:

*Campos Obrigatórios.

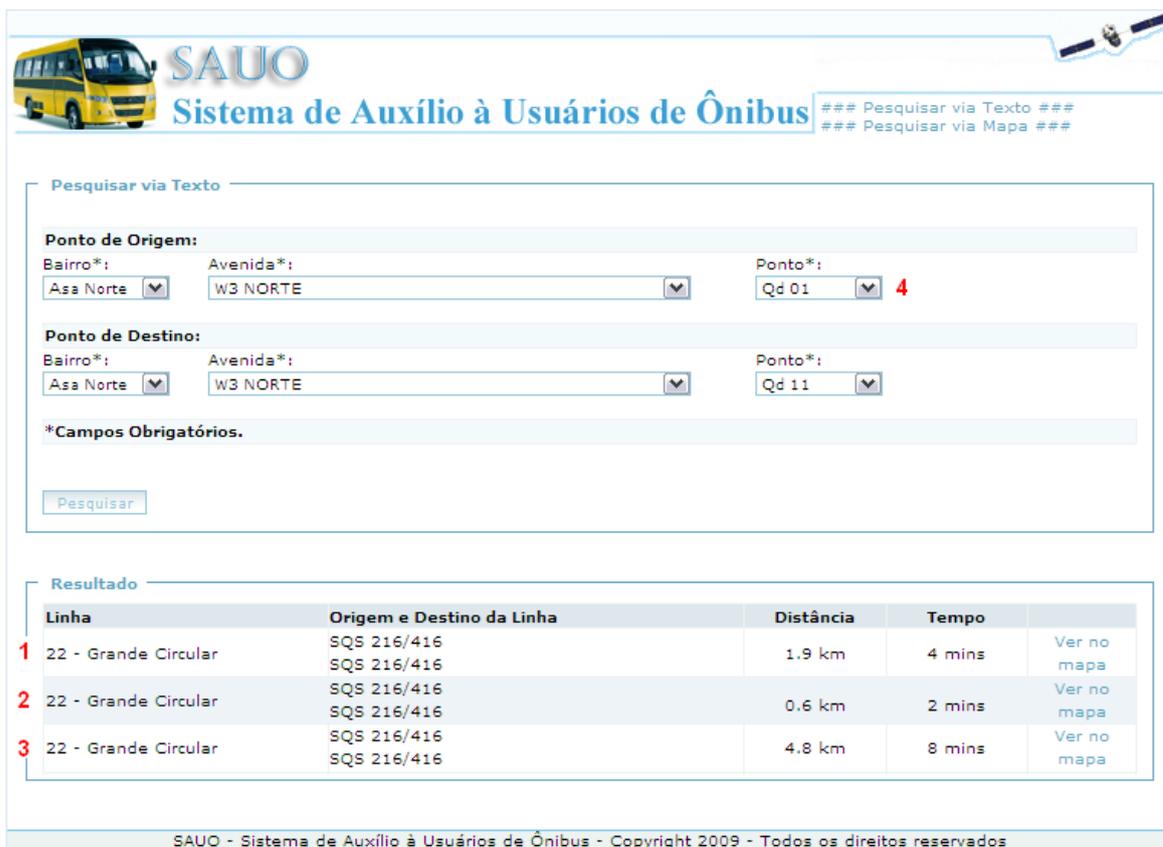
Pesquisar

SAUO - Sistema de Auxílio à Usuários de Ônibus - Copyright 2009 - Todos os direitos reservados

Figura 11 - Tela de pesquisa da funcionalidade "Pesquisar via Texto". Fonte: Autor.

Ao informar todos os campos obrigatórios para a pesquisa, o sistema irá buscar na base de dados todas as informações a respeito dos parâmetros informados e as disponibilizará em uma tela organizada em forma de tabela.

A resposta do sistema será composta pelas linhas de ônibus que farão aquele trajeto, isto é, os ônibus que passarão naquele terminal de origem e farão o trajeto passando pelo terminal de destino informado, sua distância e tempo aproximados de chegada ao terminal de origem informado.



SAUO
Sistema de Auxílio à Usuários de Ônibus

Pesquisar via Texto ###
Pesquisar via Mapa

Pesquisar via Texto

Ponto de Origem:
 Bairro*: Avenida*: Ponto*: **4**

Ponto de Destino:
 Bairro*: Avenida*: Ponto*:

*Campos Obrigatórios.

Resultado

Linha	Origem e Destino da Linha	Distância	Tempo	
1 22 - Grande Circular	SQS 216/416 SQS 216/416	1.9 km	4 mins	Ver no mapa
2 22 - Grande Circular	SQS 216/416 SQS 216/416	0.6 km	2 mins	Ver no mapa
3 22 - Grande Circular	SQS 216/416 SQS 216/416	4.8 km	8 mins	Ver no mapa

SAUO - Sistema de Auxílio à Usuários de Ônibus - Copyright 2009 - Todos os direitos reservados

Figura 12 - Tela de Resultado. Fonte: Autor.

Podemos notar na figura que o sistema retornou três resultados de uma mesma linha, isso acontece porque uma linha não possui apenas um ônibus, e sim, vários. Analisando o resultado da pesquisa, podemos dizer que o número 1 destacado em vermelho é um ônibus pertencente à linha 22, chamado Grande Circular, que tem sua origem no ponto da SQS 216/416 e seu destino final nessa mesma quadra. Sua distância para o ponto informado do usuário, Qd 01 da W3 Norte (vide número 4 destacado) é de 1,9 km tendo um tempo aproximado de chegada de 4 minutos.

A busca foi realizada por meio de consultas na base de dados, tendo como origem todos os ônibus que possuem latitudes e longitudes anteriores aos do ponto de partida.

Dessa maneira, após a consulta ter sido realizada, os cálculos sobre a distância e tempo aproximados foram feitos através da função da API do Google maps. A Figura 13 é um fragmento de código que mostra a chamada da função

“distTempo” e os parâmetros passados para a mesma, e a Figura 14 mostra a função javascript que realizará os cálculos.

```

...
<div id="map_canvas" style="display:none;" ></div>
<script language="JavaScript" src="../../web/js/distTempo.js"></script>
<?php
    $bgcolor = '#EDF3F6';
    while(list($key, $to) = each($listaResultado)) {
        if ($key % 2 == 0) {
            $bgcolor = "#FFFFFF";
        } else {
            $bgcolor = "#EDF3F6";
        }
        $latOnibus = $to->getLatitudeOrigem();
        $lonOnibus = $to->getLongitudeOrigem();
        $latPonto = $to->getLatitudeDestino();
        $lonPonto = $to->getLongitudeDestino();
        echo"<script> distTempo ($key, '$latOnibus,$lonOnibus',
'$latPonto,$lonPonto');</script>";
    ?>
...

```

Figura 13 - Fragmento do código para chamar a função distTempo.js. Fonte: Autor

```

function distTempo(num, from, to){
    var map;
    var gdir;
    map = new GMap2(document.getElementById("map_canvas"));
    gdir = new GDirections(map, document.getElementById("directions"));
    GEvent.addListener(gdir, "load", function() {
        document.getElementById("getDistance"+num).innerHTML = gdir.getDistance().html;
        document.getElementById("getDuration"+num).innerHTML = gdir.getDuration().html;
    });
    gdir.load("from: " + from + " to: " + to, { "locale": 'en_US' });
}

```

Figura 14 - Função distTempo que calcula o tempo e a distância aproximados. Fonte: Autor

O sistema também disponibilizará a localização aproximada do ônibus em um mapa interativo, através da API Google Maps, indicando o ponto de origem através de um balão e o ônibus através de uma imagem de um ônibus.

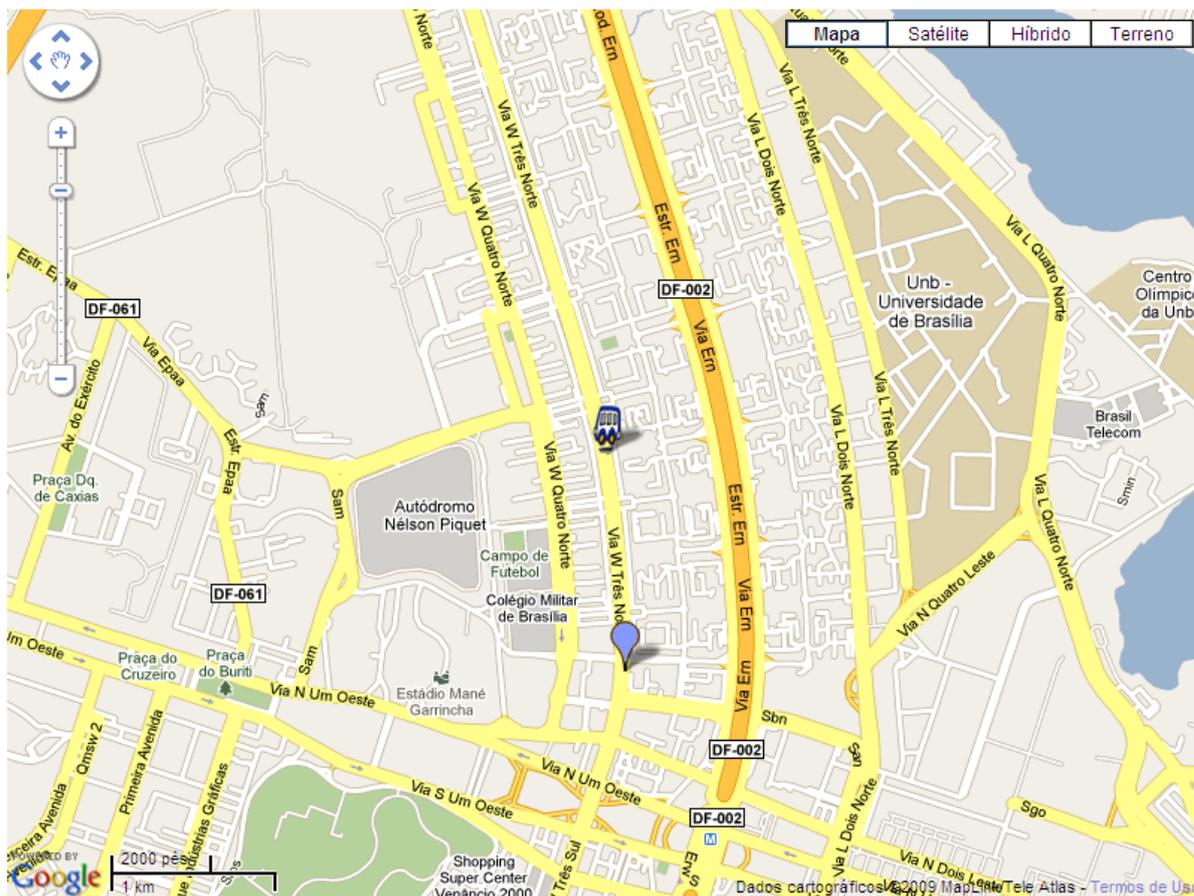


Figura 15 - Tela do mapa interativo. Fonte: Autor.

4.4.2 Pesquisa Via Mapa

Essa funcionalidade foi implementada para o usuário realizar a pesquisa da localização de todos os ônibus que farão o trajeto entre o ponto de origem e o ponto de destino informado pelo usuário.

Ao acessar essa funcionalidade pelo link “Pesquisa via Mapa” no menu superior, o sistema disponibilizará uma tela com o mapa do centro de Brasília.

Todas as marcações em vermelho representam os pontos de ônibus localizados nas avenidas W3 Norte e Sul e L2 Norte e Sul.



Figura 16 - Tela de pesquisa da funcionalidade "Pesquisar via Mapa". Fonte: Autor.

O usuário deverá clicar no mapa para indicar o ponto de origem e clicar novamente para indicar o ponto de destino desejado. Como se pode ver na Figura 16, o sistema fará as marcações da origem e do destino, na cor azul e verde, respectivamente.

Ao acionamento do botão "pesquisar", o sistema irá buscar na base de dados todas as informações a respeito dos parâmetros informados e as disponibilizará em uma tela organizada em forma de tabela.

A resposta do sistema será a mesma da funcionalidade de pesquisa via texto (vide figura 9) e será composta pelas linhas de ônibus que farão aquele trajeto, isto é, os ônibus que passarão naquele ponto de origem e farão o trajeto passando pelo ponto de destino informado, sua distância e tempo aproximados de chegada ao ponto de origem informado. Os cálculos de tempo e distância aproximados serão os mesmos já demonstrados na Figura 13 e na Figura 14.

O sistema também disponibilizará a localização aproximada do ônibus em um mapa interativo, indicando o ponto de origem através de um balão e o ônibus através de uma imagem de um ônibus. Esse mapa será disponibilizado, utilizando as funcionalidades da API do Google Maps, passando como parâmetros os dados sobre a latitude e a longitude referentes ao ponto de origem e do ônibus. O fragmento do código está evidenciado na Figura 17.

```

...
    $olat = $_SESSION['olat'] ;
    $olon = $_SESSION['olon'];
    $dlat = $_SESSION['dlat'] ;
    $dlon = $_SESSION['dlon'];

?>
<html><head> <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAzE16s71_pxo34DuUrx_JGhSLWnp9S-
9KMjylpQfCwmMVH5WUehTRiFxrNaJ9G1XvqfcCkJt86BRI2A" type="text/javascript"></script>
<script type="text/javascript">
function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.setCenter(new GLatLng(<?php echo "$olat,$olon" ?>), 14);
        map.setUIToDefault();
        var blueIcon = new GIcon(G_DEFAULT_ICON);
        blueIcon.image = "http://gmaps-samples.googlecode.com/svn/trunk/markers/blue/blank.png";
        var onibus = new GIcon(G_DEFAULT_ICON);
        onibus.image = "../web/img/onibus.gif";
        markerOptionsPonto = { icon:blueIcon };
        markerOptionsOnibus = { icon:onibus };
        var latLng = new GLatLng(<?php echo "$olat,$olon" ?>);
        map.addOverlay(new GMarker(latLng, markerOptionsOnibus));
        var latLng2 = new GLatLng(<?php echo "$dlat,$dlon" ?>);
        map.addOverlay(new GMarker(latLng2, markerOptionsPonto));
    }
}
</script>
</head>
<body onload="initialize()" onunload="GUnload()">
<div id="container"><table width="760" border="0" cellspacing="0" cellpadding="0">
<tr> <td colspan="2" class="rodape"></td> <div id="map_canvas" style="width: 800px; height: 600px"></div>
</tr>
</table>
</div></body></html>

```

Figura 17 - Fragmento do código que disponibilizará o mapa interativo. Fonte: Autor.

4.4.3 Processamento Automático

Essa parte do sistema recebeu esse nome por causa de seu processamento automático, pois ao ser iniciado, será executado em um loop, isto é, sua execução será até atingir a capacidade máxima de armazenamento do receptor GPS, que é de 200.000 (duzentos mil pontos).

Essa funcionalidade tem como objetivo buscar os dados do GPS por meio do software Goops e os armazenar no banco de dados.

O Software Goops irá obter os dados do receptor Holux a cada segundo e os gravará em um arquivo com extensão KML em um diretório pré-configurado via FTP.

O script PHP irá, a cada 10 segundos, abrir esse arquivo, ler o último dado sobre posicionamento gravado (latitude e longitude) e o inserir no banco de dados juntamente com as informações sobre aquele ônibus e a hora da gravação.

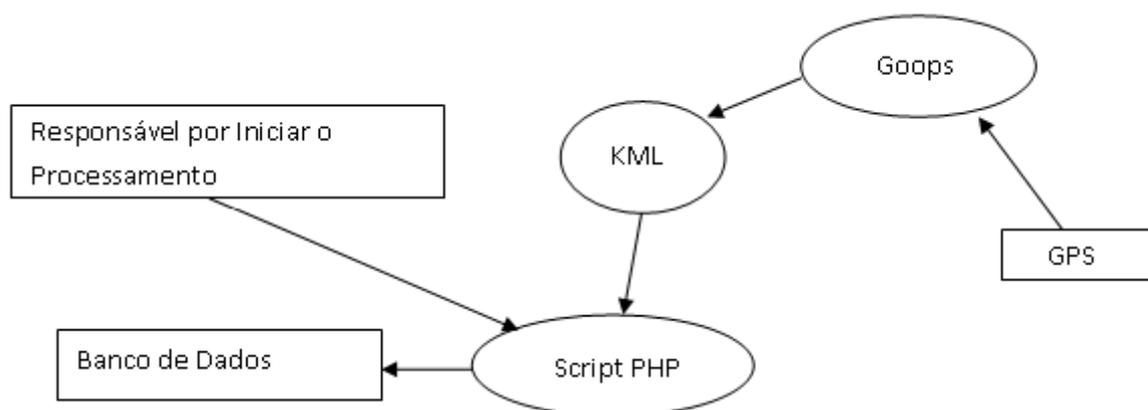


Figura 18 - Comunicação Processamento Automático. Fonte: Autor.

O endereço de acesso a esta funcionalidade é:
<http://tiagoveloso.webcindario.com/projetofinal/app/PF/SAUO/classes/view/batchAction.php?acao=montarBatch>

4.5 Arquitetura

Este projeto seguirá padrões para construção de componentes de software, baseados em camadas de responsabilidades e características específicas seguindo o padrão MVC. Dessa forma, o projeto foi dividido nas seguintes três camadas:

Tabela 3 - Arquitetura do projeto.

Camada Lógica - Modelo	Localização Física	Descrição
Apresentação - View	Plataforma baixa	Disponibiliza a interface com o usuário utilizando recursos como páginas em PHP.
Negócio - Controller	Plataforma baixa	Camada na qual serão implementadas todas as regras de negócio da aplicação.
Integração - Model	Plataforma baixa	Fornece acesso dos dados à camada de negócio.

Fonte: Autor.

A camada visão ou apresentação é que possuirá as páginas PHP e é nelas que o usuário irá navegar e fazer as solicitações de pesquisa. Nela, também, que ficarão as funções JavaScript responsáveis por manipular os mapas do Google Maps.

A classe Action, que tem como objetivo processar a requisição e gerar resposta para o componente de visão, isto é, irá fazer o papel de Controller.

A camada de controle é composta pelas classes Facades e Business Objects. As Facades obtêm a conexão com o banco de dados e realiza o controle de transação. Os Objetos de Negócio ou Business Object são entidades de negócio responsáveis por prover o modelo de negocio e definem quais os serviços prestados pelo domínio de informação e delegam o acesso aos dados para camada de integração. Esses objetos são responsáveis por executar regras de negócio reutilizáveis entre casos de uso.

E, por último, a camada de integração ou modelo será responsável pelo acesso ao banco de dados. No caso do projeto haverá uma classe DAO por domínio

de informação que será responsável por abstrair e encapsular todo acesso ao armazenamento persistente na fonte de dados, diminuindo o acoplamento entre a lógica de persistência e a lógica de aplicação.

Para transferir diversos elementos de dados entre as camadas, serão utilizados Transfer Objects. Em vez de enviar ou receber dados individuais, um Transfer Object contém todos os elementos de dados em uma única estrutura requerida pela solicitação ou resposta. Uma classe Transfer Object pode expor todos os seus membros declarando-os como públicos e fornecer um construtor que aceite todos os atributos necessários ou tornar seus atributos privados e acessados pelos métodos getter e setter.

4.6 Acesso pelos Usuários

O acesso ao sistema pelos usuários poderá ser feito através de uma conexão com a internet. Ao acessar, através dos navegadores Microsoft internet Explorer, o endereço: <http://tiagovelloso.webcindario.com/projetofinal/app/PF/SAUO/index.php>, serão disponibilizadas ao usuário as funcionalidades do sistema SAUO.

E os usuários responsáveis pelo processamento automático, deverão conhecer especificamente o endereço do script PHP, pois esse não será publicado, e a funcionalidade só será iniciada ao digitá-lo no navegador. O endereço é: <http://tiagovelloso.webcindario.com/projetofinal/app/PF/SAUO/batch/batch.php>

4.7 Software Goops

Goops é um plu-in compacto que habilita o rastreamento em tempo real e navegação pelo Google Earth. Esse software permite que se veja a posição atual no Google Earth e mantém o histórico de onde passou, incluindo data, hora e velocidade.

Ele possui tecnologia e performance do mesmo nível que os softwares que necessitam de licença para funcionar.

O Goops está disponível em várias versões, mas a utilizada no projeto foi a versão Free, que é gratuita e atende às necessidades. O recurso disponível nessa versão mais utilizado no projeto é o percurso em tempo real, que é a habilidade de salvá-los em formato kml via FTP (Goops, 2009).

4.8 API Google Maps

A API do Google Maps permite usar JavaScript para incorporar o Google Maps na página da web. A API fornece diversos utilitários para manipular mapas e adicionar conteúdo ao mapa através de diversos serviços; dessa forma, é possível criar aplicativos de mapas robustos no site.

Essa API é um serviço gratuito em fase beta de teste, disponível para qualquer site que seja gratuito para os consumidores (Google, 2009).

4.9 Problemas encontrados

O primeiro problema encontrado na realização do projeto aconteceu ao se instalar o servidor Apache 2.2.11, o PHP 5.3.0 e o MySQL 5.1.36: tudo parecia ter funcionado perfeitamente, até a funcionalidade chegar na camada de persistência e tentar realizar a conexão com o MySQL. Verifiquei, então, que o banco de dados estava instalado corretamente, porém o módulo que o conecta ao PHP não é automaticamente instalado como acontecia com versões anteriores do PHP.

Após várias buscas por soluções, a conclusão de que a melhor delas seria a instalação de um pacote pronto chamado WAMP, que é disponibilizado na Web. Esse pacote já está na versão 2.0 e permite a instalação e a configuração padrão do servidor Apache, a base de dados MySQL e o módulo de programação em PHP versão 5.

Assim, o módulo de conexão do PHP com a base de dados foi instalada corretamente.

O segundo problema foi a forma de obtenção dos dados do receptor Holux M-1000: após alguns dias de estudos e algumas possibilidades descartadas, como,

utilizar uma API do Java ou do C++ para ler os dados na porta USB ou utilizar de alguma forma o hyperterminal do Windows para fazer essa leitura, a solução encontrada foi utilizar um software opensource chamado Goops. Ele é um pequeno executável, que não necessita de instalação e que gera arquivos com extensão KML periodicamente, exatamente da forma que o projeto necessita.

O terceiro problema foi encontrar um servidor publicado na internet que aceitasse a estrutura Apache+PHP+Mysql, barato ou de graça. A solução encontrada foi um portal chamado Miarroba que pode ser acessado pelo endereço <http://hosting.miarroba.com/>.

O problema encontrado ao realizar a implementação foi que as avenidas percorridas pela linha do “Grande Circular” possuem intervalos de latitudes e longitudes em comum. É impossível que possuam duas coordenadas em duas avenidas diferentes, mas, como o código não trata de pontos específicos, e sim, de intervalos de coordenadas, as avenidas ficaram com intervalos coincidentes, como se pode ver na Figura 19 e na Figura 20.

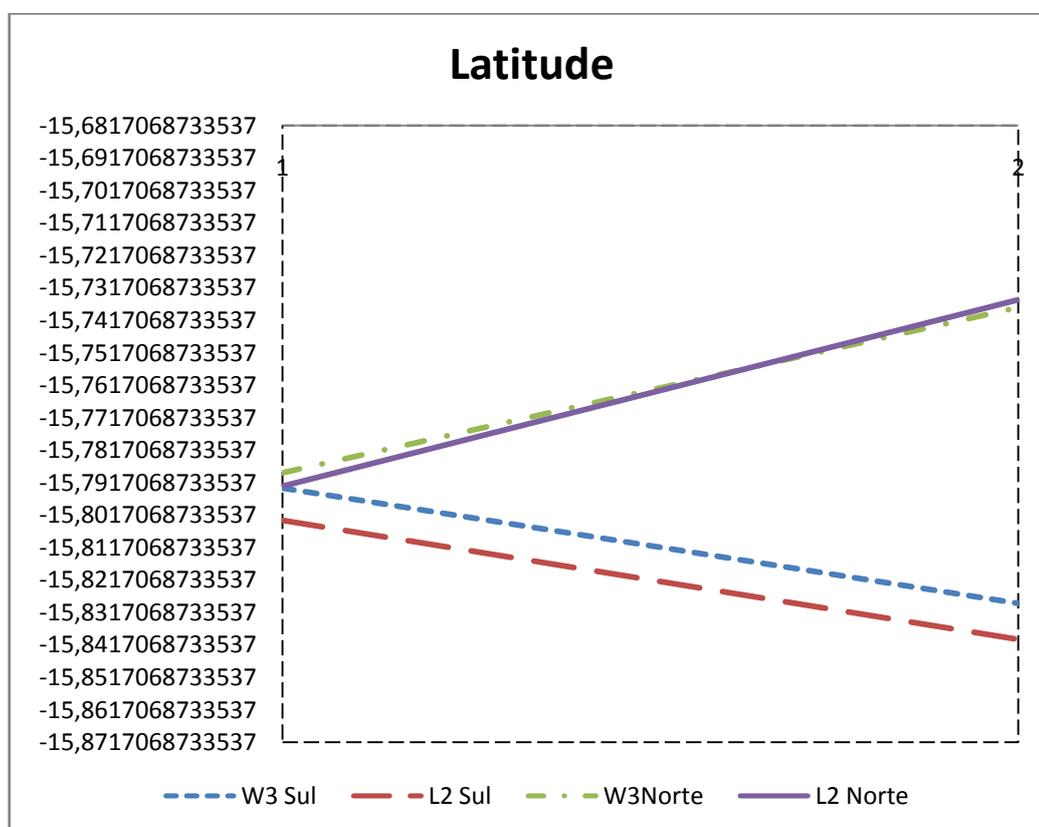


Figura 19 – Gráfico de latitudes das avenidas W3 e L2. Fonte: Autor.

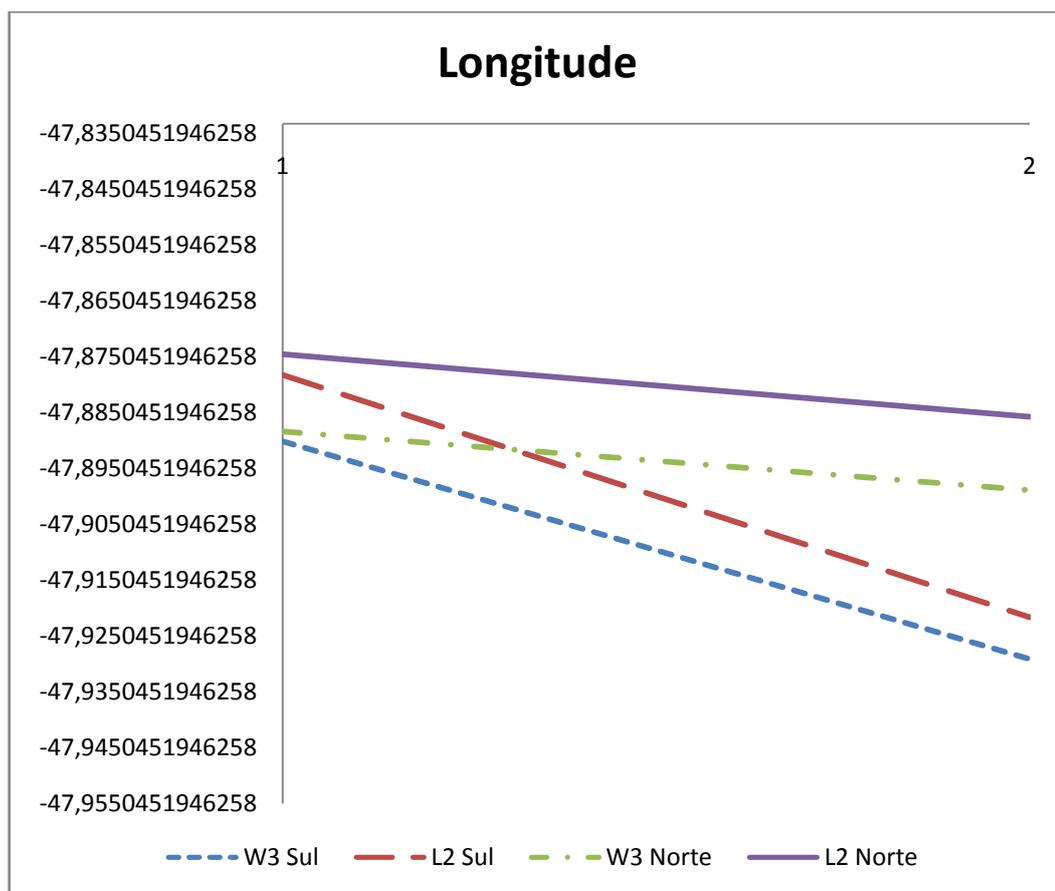


Figura 20 - Gráfico de longitudes das avenidas W3 e L2. Fonte: Autor.

É possível identificar, nos gráficos, que as avenidas W3 Sul e L2 Sul possuem coordenadas de latitude e longitude em comum. Para resolver esse problema, essas duas avenidas foram divididas em quatro partes cada, para reduzir ao máximo a quantidade de coordenadas idênticas. As novas coordenadas foram representadas na Figura 21 e na Figura 22.

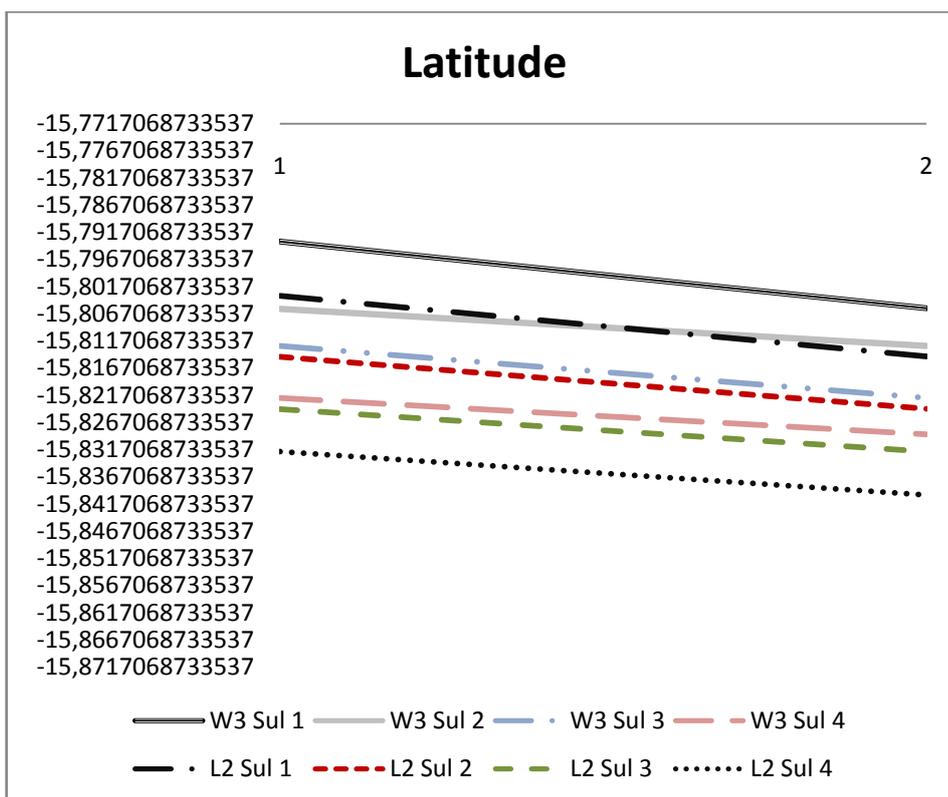


Figura 21 - Gráfico de latitudes das avenidas W3 e L2 particionadas. Fonte: Autor.

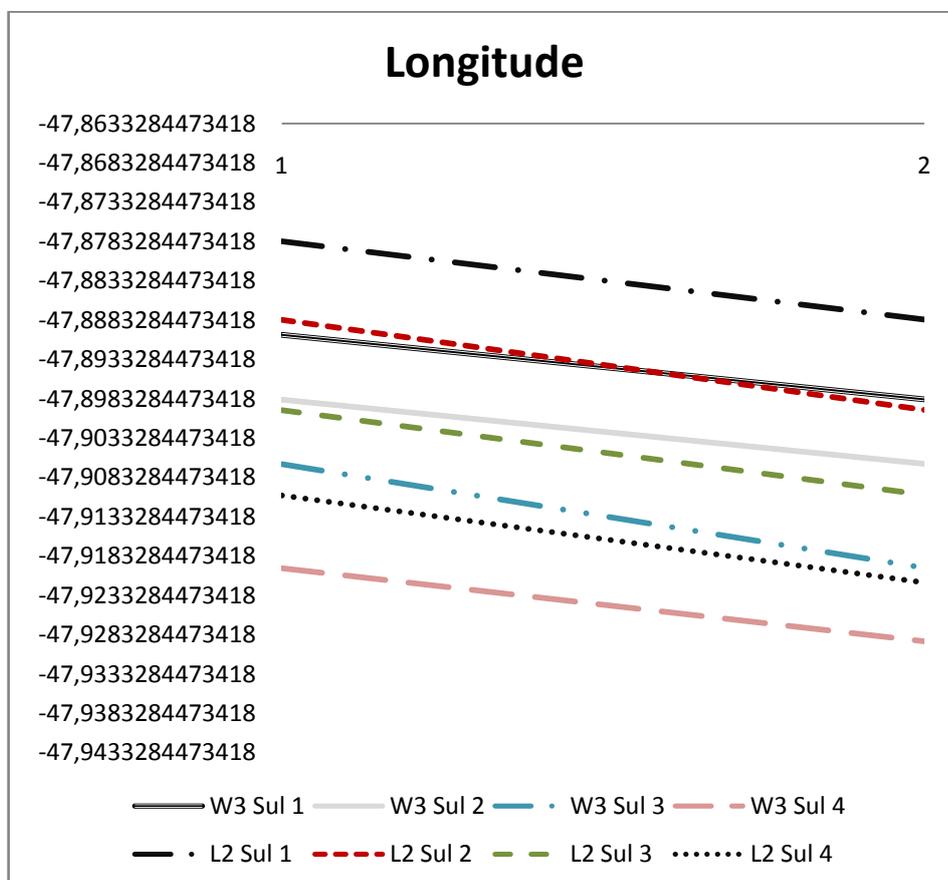


Figura 22 - Gráfico de longitudes das avenidas W3 e L2 particionadas. Fonte: Autor.

4.10 Testes e Resultados

Os testes abrangem as avenidas W3 Norte e Sul e L2 Norte e Sul, simulando a linha “Grande Circular”, que realiza o percurso entre essas avenidas em dois sentidos.

Os testes foram divididos em duas etapas, sendo a primeira um teste local com dados inseridos manualmente e a segunda um teste na rua, obtendo os dados dinamicamente através do GPS.

A técnica utilizada para realizar ambos os testes foi a chamada “caixa preta”, que é um teste orientado a entrada e saída de dados, isto é, os dados de entrada são fornecidos, o teste é executado e o resultado obtido é comparado a um resultado previamente conhecido.

Para realizar os testes, é necessário que o sistema possua inicialmente os dados referentes aos bairros, avenidas, linhas dos ônibus, itinerários das linhas, ônibus referentes àquelas linhas, os sentidos em que eles trafegam e os pontos.

A Figura 23 mostra os dados cadastrados para ambos os testes.

dbavenida							
codAvenida *	descAvenida	codBairro	latInicio	lngInicio	latFim	lngFim	tpDado
10	W3 SUL	2	-15,7934098409468	-47,8901982307434	-15,8288793614498	-47,9291868209838	1
17	L2 SUL	2	-15,8033617541246	-47,8783106803894	-15,8400268197142	-47,9216980934143	1
19	L2 NORTE	1	-15,7928110587895	-47,8746199607849	-15,735340393406	-47,8858208656311	1
26	W3 NORTE	1	-15,7886401822516	-47,8884387016296	-15,7376742298588	-47,8989315032958	1
39	W3 Sul Part 1	2	-15,7934098409468	-47,8901982307434	-15,8056948082171	-47,8983950614929	2
40	W3 Sul Part 2	2	-15,8057773937099	-47,8984379768371	-15,8126112264558	-47,9066133499145	2
41	W3 Sul Part 3	2	-15,8126318721265	-47,9066562652587	-15,8221493019046	-47,9198312759399	2
42	W3 Sul Part 4	2	-15,8221905912966	-47,9198741912841	-15,8288793614498	-47,9291868209838	2
43	L2 Sul Part 1	2	-15,8033617541246	-47,8783106803894	-15,8145519102903	-47,8882455825805	2
44	L2 Sul Part 2	2	-15,8146138467018	-47,8882884979248	-15,8241518277014	-47,8997468948364	2
45	L2 Sul Part 3	2	-15,8242137611727	-47,8997898101806	-15,8320172267189	-47,9105615615844	2
46	L2 Sul part 4	2	-15,832037870408	-47,9106259346008	-15,8400268197142	-47,9216980934143	2

dbitinerario			
nuLinha *	ordem *	codAvenida	
22	1	26	
22	2	10	
22	3	17	
22	4	19	

dblinha	
nuLinha *	descLinha
11	Qd. 716 N/Espanada (Aeroporto - CNB)
22	Grande Circular
31	SQS 216-416 / W3 Sul - Norte (SQN 212/213)/ L2 N-S/ ESPLANADA
32	SQS 216-416/ L2 Sul-Norte (Espanada)/ W3 Norte-Sul
1012	Paranoá/L2 Norte (Rodov. do Plano Piloto)
1014	Paranoá/Centro de Ensino Especial de Brasília
1015	CEE-02/Paranoá/CEE-02
105,4	Circular W3 Norte-Sul/L2 Sul (Câmara Legislativa)
106,2	Circular L2 Norte-Sul/W3 Sul-Norte (Câmara Legislativa)
110	Rodov. do Plano Piloto/UnB
1102	Rodov. do Plano Piloto/UnB (CAESB)
115,1	Rodov. do Plano Piloto/L2-W3 Norte
1152	Rodov. do Plano Piloto/L2-W3 Norte/Espanada
116,1	Circular Rodov. do Plano Piloto/W3-L2 Norte/Espanada
116	Rodov. do Plano Piloto/Avenidas W3-L2 Norte
1283	Rodov. do Plano Piloto/L2 Norte/Granja do Torto
1361	Rodov. do Plano Piloto/Clube do Congresso (L2 Norte)

dbonibus			
codOnibus *	tpSentido *	nuLinha	horario
1	1	22	06:20:00
2	1	22	06:40:00
3	1	22	07:00:00
4	1	22	07:20:00
5	2	22	07:40:00
6	2	22	08:00:00
7	2	22	08:20:00
8	2	22	08:40:00
9	1	22	09:00:00
10	1	22	09:20:00
11	2	22	09:40:00
12	2	22	10:00:00
13	2	22	10:20:00
14	1	22	10:40:00
15	1	22	11:00:00
16	2	22	11:20:00

dbsentido	
tpSentido *	descSentido
1	Norte - Sul
2	Sul - Norte

dbbairro	
codBairro *	descBairro
1	Asa Norte
2	Asa Sul

Figura 23 - Dados Cadastrados no Banco de Dados. Fonte: Autor.

4.10.1 Teste Local

Este teste tem como objetivo observar se as consultas no banco de dados e os cálculos realizados pela API do Google Maps estão corretos. Desta maneira, o resultado esperado é a busca correta no banco, retornando todos os ônibus que possuem latitudes e longitudes anteriores ao do ponto de origem selecionado pelo usuário, isto é, todos os ônibus que ainda vão passar por aquele ponto.

Este teste foi realizado utilizando os dados pré-inseridos referentes ao posicionamento do ônibus. Dessa forma, 20 posições foram inseridas para diferentes ônibus de uma mesma linha.

Observam-se, na Figura 24, as posições inseridas a respeito dos ônibus da linha número 22. As posições utilizadas para realizar as pesquisas serão sempre as últimas de cada ônibus. Exemplo: A posição utilizada para o ônibus cujo código é 4 é: latitude=-14,7691278 e longitude= -47,8891931.

dbposicao				
codPosicao *	codOnibus *	latitude	longitude	hora
1	4	-15,7719867	-47,8885475	16/9/2009 10:07:45
2	4	-15,7691278	-47,8891931	16/9/2009 10:09:33
4	5	-15,7840934	-47,8875558	16/9/2009 10:09:37
5	5	-15,7808208	-47,887492	16/9/2009 10:13:11
6	6	-15,758411	-47,892798	23/9/2009 23:57:02
7	6	-15,766071	-47,890355	24/9/2009 00:09:21
8	6	-15,761439	-47,891628	24/9/2009 01:13:01
69845	11	-15,769018	-47,8754353	19/11/2009 16:46:54
69844	9	-15,8255504	-47,9013615	19/11/2009 16:43:16
69843	8	-15,8327913	-47,9273253	19/11/2009 14:36:09
69842	3	-15,829607	-47,9296535	19/11/2009 14:34:42
69841	2	-15,81299317	-47,90713369	19/11/2009 14:33:23
69840	7	-15,80575158	-47,8982394	19/11/2009 14:32:09
69839	1	-15,79530425	-47,89094388	19/11/2009 14:29:03
69838	1	-15,79334273	-47,89023578	19/11/2009 14:28:39
69837	1	-15,790126841	-47,88915216	19/11/2009 14:28:11
69836	1	-15,786466956	-47,88805246	19/11/2009 14:27:47
69835	1	-15,78280705	-47,88767695	19/11/2009 14:27:20
69834	1	-15,78065952	-47,88773596	19/11/2009 14:26:54
69833	1	-15,77912117	-47,88791298	19/11/2009 14:26:24

Figura 24 - Dados da tabela referente a posição dos ônibus. Fonte: Autor.

Na Figura 25, é possível observar os pontos de ônibus, nos marcadores em vermelho, e a localização dos ônibus, nos marcadores em azul.



Figura 25 - Mapa com a marcação dos pontos e ônibus. Fonte: Autor.

Esse mapa foi construído com informações do banco de dados que estão listadas na Figura 23 e Figura 24.

Após serem inseridos esses dados na base de dados, a aplicação web foi iniciada, e acionada a funcionalidade “Pesquisar via Texto”. Para visualizar a tela da funcionalidade, vide tópico 4.4.

Os parâmetros Origem e Destino da tela “Pesquisar via Texto” preenchidos e alguns dos resultados gerados pelo sistema estão descritos na Tabela 4, a tabela completa pode ser vista no apêndice A.

As mesmas pesquisas foram realizadas pela funcionalidade “Pesquisar via Mapa” e os resultados gerados pelo sistema foram os mesmos. Isso ocorreu devido a ser o mesmo código utilizado pelas duas funcionalidades, a partir da camada de controle.

Tabela 4 - Testes e Resultados com dados fixos.

Origem			Destino			Resultado	
Bairro	Avenida	Ponto	Bairro	Avenida	Ponto	Distância	Tempo
Asa Norte	W3	703/704	Asa Sul	W3	713	1.6 km	3 mins
Asa Norte	W3	702	Asa Sul	L2	604	1.9 km	3 mins
Asa Sul	W3	705/706	Asa Sul	L2	615	1.7 km	3 mins

Fonte: Autor.

Aos analisar esses dados, notamos que o sistema realiza corretamente as pesquisas em qualquer sentido, retornando a distância e o tempo aproximados em quilômetros e minutos ou segundos respectivamente.

A mensagem de erro “Não existem ônibus que realizem esse trajeto” é exibida quando os parâmetros selecionados não correspondem ao trajeto de nenhuma linha.

O sistema não retorna resultado quando não existem ônibus que correspondem aos parâmetros de origem e destino informados que estejam trafegando na mesma avenida do ponto de origem.

4.10.2 Teste na rua

Este teste possui o objetivo de testar a comunicação entre o dispositivo GPS com o software Goops, a comunicação do Goops com o script PHP, utilizando a comunicação 3g e a gravação dos dados no banco de dados MySQL. Durante esse teste, consultas são realizadas para testar o sistema utilizando dados dinâmicos.

Com os objetivos traçados, o resultado esperado é o sucesso na comunicação e a busca correta no banco de dados utilizando a última posição adquirida do GPS, retornando os ônibus que ainda vão passar pelo ponto informado como parâmetro.

Esse teste foi realizado utilizando um carro para simular um ônibus pertencente à linha do “Grande Circular”.

O carro percorreu a W3 Norte e Sul marcando sua posição através do GPS, e o arquivo gerado pelo software Goops era enviado via FTP utilizando a conexão do modem 3g.



Figura 26 - Dispositivos para obtenção das Coordenadas. Fonte: Autor.

O primeiro ônibus simulado foi de código 14 no sentido norte sul, seguindo pelos pontos das 700 e o segundo foi o de código 16 realizando o sentido sul norte, seguindo pelos pontos das 500.

Alguns dos dados gravados sobre as coordenadas podem ser vistos na Tabela 5. A tabela completa está evidenciada no apêndice B.

Tabela 5 - Coordenadas obtidas através do GPS.

Código da Posição	Código do Ônibus	Latitude	Longitude	Hora de inserção no Banco de Dados
1	4	-15,7719867	-47,8885475	16/9/09 10:07:45
2	4	-15,7691278	-47,8891931	16/9/09 10:09:33
6	6	-15,758411	-47,892798	23/9/09 23:57:02
7	6	-15,766071	-47,890355	24/9/09 0:09:21
69844	9	-15,8255504	-47,9013615	19/11/09 16:43:16
69845	11	-15,769018	-47,8754353	19/11/09 16:46:54
69854	14	-15,76309	-47,89124	22/11/09 22:44:53
69855	14	-15,76364	-47,89109	22/11/09 22:45:03

Fonte: Autor.

Através da tabela completa, é possível observar que a simulação foi realizada entre os horários 22:44:53 e 23:05:44 do dia 22 de novembro de 2009.

Analisando a tabela completa, observamos o sucesso na comunicação entre o GPS, o software Goops, o script PHP e o banco de dados MySQL, pois enquanto o carro realizava o percurso, os dados recebidos eram corretamente trafegados até serem gravados no banco de dados.

E algumas pesquisas realizadas durante o percurso podem ser vistas na Tabela 6. E a tabela completa está no apêndice C.

Tabela 6 - Resultado das pesquisas com o ônibus em movimento.

Hora da Pesquisa	Origem			Destino			Resultado	
	Bairro	Avenida	Ponto	Bairro	Avenida	Ponto	Distância	Tempo
22:40	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
22:45	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
							2.2 Km	3 mins
22:49	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
							1.1 Km	2 mins

Fonte: Autor.

Observando a Tabela 6 e os objetivos traçados, é possível notar que o sistema, corretamente, considera a última posição armazenada para realizar as consultas e cálculos de tempo e distância.

Para ambos os testes, é possível perceber resultados satisfatórios, baseados nos objetivos traçados e metodologia utilizada, a respeito das consultas realizadas no banco de dados, dos cálculos das distâncias e tempos aproximados e a comunicação.

4.11 Limitações do Sistema

O sistema foi implementado utilizando apenas a linha do “Grande Circular”; dessa forma os cálculos são realizados baseados em latitudes e longitudes. Sendo assim, o sistema, na atual implementação, não possui em seu código a realização de rotas, impossibilitando, por exemplo, o tratamento de desvios de itinerários e linhas que não trafegam unicamente nas avenidas principais do plano piloto (W3 e L2).

A segunda limitação do sistema é a não retirada de um ônibus da lista de resultados em caso de sinistro, isto é, a impossibilidade de ele prever quando um ônibus, por ter quebrado, vai deixar de passar por aquele ponto informado.

CONCLUSÃO

Esse projeto apresenta o desenvolvimento de uma nova forma de realizar consultas sobre a posição de um ônibus de uma linha em relação a determinado terminal escolhido.

O sistema proposto faz uso de um receptor GPS em cada veículo da frota e registra sua posição no dispositivo a cada segundo e no banco de dados a cada 10 segundos. Desta forma, é possível que o usuário, através de um navegador da internet, possa realizar consultas e obtenha respostas sobre distâncias e tempos aproximados das linhas que farão determinado trajeto.

O projeto foi testado através de um carro simulando um veículo da linha responsável por passar pelas avenidas W3 Norte e Sul e L2 norte e sul, popularmente conhecida como “Grande Circular”. Esse teste comprovou a eficácia do sistema em receber e armazenar os dados sobre as coordenadas e o bom funcionamento das ferramentas e do receptor utilizado.

O segundo teste foi realizado pela página armazenada no servidor da internet, onde consultas foram feitas, verificando-se que o código, através da API do Google Maps, calcula de forma satisfatória a distância e o tempo do carro, simulando o ônibus, em relação aos terminais informados.

Desta forma, nota-se que os objetivos traçados no primeiro capítulo foram atingidos, concretizando um sistema web muito amigável e um sistema automático funcional.

Nota-se que o sistema sem o ônibus como plano de fundo poderia ter seu funcionamento baseado em qualquer referência móvel que fizesse o uso do receptor GPS, tornando o projeto um sistema de rastreamento que realiza cálculos de tempo e distância a qualquer ponto previamente cadastrado.

É lícito supor-se que, se o projeto vier a ser implantado em larga escala, de acordo com as pesquisas realizadas, será possível observarem-se mudanças, como a melhoria na qualidade de vida dos usuários de transporte público e, até, a migração dos usuários do transporte privado para o uso dos ônibus, melhorando assim a engenharia de tráfego nos locais atingidos.

Sugestão para projetos futuros

Como sugestão para trabalhos subseqüentes, sugere-se o desenvolvimento de hardware para a transmissão dos dados sobre o posicionamento, substituindo assim o protótipo composto pela internet 3g, notebook e GPS, utilizando apenas um hardware microcontrolado.

Outra sugestão seria a exibição das informações de distância e tempo relativos às linhas de ônibus em cada terminal, apresentando em tempo real esses dados aos usuários sem a necessidade de uma conexão, pelos usuários, com a internet.

REFERÊNCIAS

- AKITA, F.** Repensando a Web com Rails. Rio de Janeiro: Brasport, 2006.
- EL-RABBANY, A. **Introducion to GPS:** the global positioning system. Norwood: Artech House, 2002.
- FOWLER, M. **Padrões de arquitetura de aplicações corporativas.** Tradução de Acauan Fernandes. Porto Alegre: Bookman, 2006.
- GOODMAN, D. **JavaScript, a bíblia.** Tradução de Daniel Vieira. Rio de Janeiro: Campus, 2001.
- GOOGLE. API do Google Maps - Google Code. **API do Google Maps - Google Code**, 2009. Disponível em: <<http://code.google.com/intl/pt-BR/apis/maps/>>. Acesso em: 6 Novembro 2009.
- GOOPS. Goops. **Goops**, 2009. Disponível em: <<http://goopstechnologies.com/goops.html>>. Acesso em: 6 Novembro 2009.
- GOVERNO DO DISTRITO FEDERAL. GDF - Portal do Cidadão. **GDF - Portal do Cidadão**, 2009. Disponível em: <<http://www.gdf.df.gov.br/>>. Acesso em: 03 out. 2009.
- JUNIOR, C. S. D. M. **Web 2.0 e Mashups:** reinventando a internet. Rio de Janeiro: Brasport, 2007.
- LAURIE, B.; LAURIE, P. **Apache:** the definitive guide. 3ª Edição. ed. Sebastopol: O'Reilly Media, 2002.
- LUKE WELLING, L. T. **PHP e MySQL Desenvolvimento Web.** 3ª Edição. ed. Rio de Janeiro: Campus, 2005.
- MILANI, A. **MySQL:** guia do programador. São Paulo: Novatec Editora, 2006.
- MONICO, J. F. G. **Posicionamento pelo NAVSTAR-GPS:** descrição, fundamentos e aplicações. São Paulo: UNESP, 2000.
- SCRIMGER, R. **TCP/IP:** a bíblia. Tradução de Edson Furmankievicz. Rio de Janeiro: Campus, 2002.

ULABY, F. **Electromagnetics for Engineers**. Porto Alegre: Bookman, 2005.

VASCONCELLOS, E. A. D. **A cidade, o transporte e o trânsito**. São Paulo: Fenaseg, 2005.

VELLOSO, M. S.; VAZ, A. M. **Estudo da Operação do Serviço de Transporte de Vizinhança - Linhas 31 e 35**. Brasília. 2005.

APÊNDICE A – Testes e Resultados com dados fixos

Origem			Destino			Resultado	
Bairro	Avenida	Ponto	Bairro	Avenida	Ponto	Distância	Tempo
Asa Norte	W3	703/704	Asa Sul	W3	713	1.6 km	3 mins
Asa Norte	W3	702	Asa Sul	L2	604	1.9 km	3 mins
Asa Sul	W3	705/706	Asa Sul	L2	615	1.7 km	3 mins
Asa Sul	W3	713	Asa Sul	L2	602	4.6 km	8 mins
						2.0 km	3 mins
Asa Sul	W3	710	Asa Sul	W3	715	3.1 Km	8 mins
						0.4 Km	3 mins
Asa Norte	W3	708/709	Asa Norte	W3	702	Nenhum Resultado	
Asa Sul	W3	506/507	Asa Norte	L2	611	Erro: "Não existem ônibus que realizem esse trajeto."	
Asa Norte	L2	608	Asa Sul	L2	607	2.2 Km	3 mins
Asa Norte	L2	612	Asa Norte	W3	713/714	3.5 Km	5 mins
Asa Norte	L2	604	Asa Norte	L2	609	Nenhum Resultado	
Asa Sul	L2	601	Asa Norte	L2	606	3.3 Km	4 mins
Asa Sul	L2	613	Asa Sul	L2	604	Nenhum Resultado	
Asa Sul	L2	607	Asa Norte	W3	716	0.8 Km	1 mins
Asa Norte	W3	502	Asa Sul	L2	403	Nenhum Resultado	
Asa Norte	W3	510/511	Asa Sul	W3	715	0.3 Km	2 mins
Asa Norte	L2	404	Asa Norte	L2	402	0.6 Km	57 secs
Asa Norte	L2	411	Asa Norte	L2	404	Nenhum Resultado	
Asa Sul	W3	502/503	Asa Sul	L2	604	1.2 Km	1 min

Asa Sul	W3	509/510	Asa Norte	W3	502	Nenhum Resultado	
Asa Sul	W3	504/505	Asa Norte	L2	407	0.5 Km	49 secs
Asa Sul	L2	408	Asa Norte	L2	709	Erro: "Não existem ônibus que realizem esse trajeto."	

APÊNDICE B – Coordenadas obtidas através do GPS

Código da Posição	Código do Ônibus	Latitude	Longitude	Hora de inserção no Banco de Dados
1	4	-15,7719867	-47,8885475	16/9/09 10:07:45
2	4	-15,7691278	-47,8891931	16/9/09 10:09:33
6	6	-15,758411	-47,892798	23/9/09 23:57:02
7	6	-15,766071	-47,890355	24/9/09 0:09:21
8	6	-15,761439	-47,891628	24/9/09 1:13:01
69833	1	-15,77912117	-47,88791298	19/11/09 14:26:24
69834	1	-15,78065952	-47,88773596	19/11/09 14:26:54
69837	1	-15,79012684	-47,88915216	19/11/09 14:28:11
69838	1	-15,79334273	-47,89023578	19/11/09 14:28:39
69839	1	-15,79530425	-47,89094388	19/11/09 14:29:03
69840	7	-15,80897755	-47,90167809	19/11/09 14:32:09
69841	2	-15,81299317	-47,90713369	19/11/09 14:33:23
69844	9	-15,8255504	-47,9013615	19/11/09 16:43:16
69845	11	-15,769018	-47,8754353	19/11/09 16:46:54
69854	14	-15,76309	-47,89124	22/11/09 22:44:53
69855	14	-15,76364	-47,89109	22/11/09 22:45:03
69856	14	-15,76448	-47,89081	22/11/09 22:45:13
69857	14	-15,76593	-47,89035	22/11/09 22:45:23
69858	14	-15,76713	-47,88998	22/11/09 22:45:33
69859	14	-15,76804	-47,88973	22/11/09 22:45:43
69860	14	-15,76922	-47,88942	22/11/09 22:45:53

69861	14	-15,77032	-47,88916	22/11/09 22:46:03
69862	14	-15,77166	-47,88889	22/11/09 22:46:13
69863	14	-15,77225	-47,88883	22/11/09 22:46:23
69864	14	-15,77251	-47,88881	22/11/09 22:46:33
69865	14	-15,77281	-47,88875	22/11/09 22:46:43
69866	14	-15,77302	-47,88872	22/11/09 22:46:53
69867	14	-15,7732	-47,88868	22/11/09 22:47:03
69868	14	-15,77363	-47,88859	22/11/09 22:47:13
69869	14	-15,77427	-47,88842	22/11/09 22:47:23
69870	14	-15,77535	-47,88828	22/11/09 22:47:33
69871	14	-15,77621	-47,88817	22/11/09 22:47:43
69872	14	-15,77768	-47,88801	22/11/09 22:47:53
69873	14	-15,77862	-47,88793	22/11/09 22:48:03
69874	14	-15,77994	-47,88783	22/11/09 22:48:13
69875	14	-15,78032	-47,88779	22/11/09 22:48:23
69876	14	-15,78032	-47,88779	22/11/09 22:48:33
69877	14	-15,78032	-47,88779	22/11/09 22:48:43
69878	14	-15,78083	-47,88775	22/11/09 22:48:53
69879	14	-15,78152	-47,88774	22/11/09 22:49:03
69880	14	-15,7829	-47,88772	22/11/09 22:49:13
69881	14	-15,78417	-47,88778	22/11/09 22:49:23
69882	14	-15,78459	-47,88782	22/11/09 22:49:33
69883	14	-15,78607	-47,88796	22/11/09 22:49:43
69884	14	-15,78675	-47,88813	22/11/09 22:49:53

69885	14	-15,78705	-47,88822	22/11/09 22:50:03
69886	14	-15,78748	-47,88833	22/11/09 22:50:13
69887	14	-15,78752	-47,88834	22/11/09 22:50:23
69888	14	-15,78773	-47,88839	22/11/09 22:50:33
69889	14	-15,78851	-47,88862	22/11/09 22:50:43
69890	14	-15,78999	-47,88914	22/11/09 22:50:53
69891	14	-15,79099	-47,88946	22/11/09 22:51:03
69892	14	-15,79241	-47,88989	22/11/09 22:51:13
69893	14	-15,79358	-47,89025	22/11/09 22:51:23
69894	14	-15,79459	-47,89067	22/11/09 22:51:33
69895	14	-15,79562	-47,89104	22/11/09 22:51:43
69896	14	-15,79672	-47,89153	22/11/09 22:51:53
69897	14	-15,79758	-47,89192	22/11/09 22:52:03
69898	14	-15,7988	-47,89265	22/11/09 22:52:13
69899	14	-15,79969	-47,89323	22/11/09 22:52:23
69900	14	-15,80007	-47,89352	22/11/09 22:52:33
69901	14	-15,80142	-47,89458	22/11/09 22:52:43
69902	14	-15,80238	-47,89546	22/11/09 22:52:53
69903	14	-15,80316	-47,89617	22/11/09 22:53:03
69904	14	-15,80346	-47,89641	22/11/09 22:53:13
69905	14	-15,80352	-47,89647	22/11/09 22:53:23
69906	14	-15,80385	-47,89676	22/11/09 22:53:33
69907	14	-15,80442	-47,89731	22/11/09 22:53:43
69908	14	-15,80511	-47,89795	22/11/09 22:53:54

69909	14	-15,80611	-47,89887	22/11/09 22:54:04
69910	14	-15,80684	-47,89963	22/11/09 22:54:14
69911	14	-15,80778	-47,90059	22/11/09 22:54:24
69912	14	-15,80848	-47,90137	22/11/09 22:54:34
69913	14	-15,80919	-47,90218	22/11/09 22:54:44
69914	14	-15,80994	-47,90305	22/11/09 22:54:54
69915	14	-15,81067	-47,90396	22/11/09 22:55:04
69916	14	-15,81121	-47,90465	22/11/09 22:55:14
69917	14	-15,81146	-47,90497	22/11/09 22:55:24
69918	14	-15,81146	-47,90497	22/11/09 22:55:34
69919	14	-15,81145	-47,90495	22/11/09 22:55:44
69920	14	-15,81171	-47,90529	22/11/09 22:55:54
69921	14	-15,8125	-47,90636	22/11/09 22:56:04
69922	14	-15,81311	-47,90722	22/11/09 22:56:14
69923	14	-15,81364	-47,90795	22/11/09 22:56:24
69924	14	-15,81434	-47,90888	22/11/09 22:56:34
69925	14	-15,81434	-47,90874	22/11/09 22:56:44
69926	14	-15,81387	-47,90811	22/11/09 22:56:54
69927	14	-15,81316	-47,90713	22/11/09 22:57:04
69928	14	-15,8125	-47,90619	22/11/09 22:57:14
69929	14	-15,81226	-47,90587	22/11/09 22:57:24
69930	14	-15,81226	-47,90587	22/11/09 22:57:34
69931	14	-15,81222	-47,90582	22/11/09 22:57:44
69932	14	-15,81181	-47,90523	22/11/09 22:57:54

69933	16	-15,81123	-47,90446	22/11/09 22:58:04
69934	16	-15,81043	-47,9034	22/11/09 22:58:14
69935	16	-15,80969	-47,9025	22/11/09 22:58:24
69936	16	-15,80886	-47,90152	22/11/09 22:58:34
69937	16	-15,80792	-47,90049	22/11/09 22:58:44
69938	16	-15,80729	-47,89983	22/11/09 22:58:54
69939	16	-15,80616	-47,89863	22/11/09 22:59:04
69940	16	-15,80526	-47,89778	22/11/09 22:59:14
69941	16	-15,80435	-47,89693	22/11/09 22:59:24
69942	16	-15,80398	-47,89661	22/11/09 22:59:34
69943	16	-15,80384	-47,89646	22/11/09 22:59:44
69944	16	-15,80322	-47,89591	22/11/09 22:59:54
69945	16	-15,80229	-47,8951	22/11/09 23:00:04
69946	16	-15,80138	-47,89433	22/11/09 23:00:14
69947	16	-15,80045	-47,89358	22/11/09 23:00:24
69948	16	-15,79918	-47,89261	22/11/09 23:00:34
69949	16	-15,79811	-47,89196	22/11/09 23:00:44
69950	16	-15,79717	-47,89143	22/11/09 23:00:54
69951	16	-15,79621	-47,89101	22/11/09 23:01:04
69952	16	-15,79582	-47,89087	22/11/09 23:01:14
69953	16	-15,79582	-47,89087	22/11/09 23:01:24
69954	16	-15,79582	-47,89087	22/11/09 23:01:34
69955	16	-15,79578	-47,89085	22/11/09 23:01:44
69956	16	-15,79499	-47,89055	22/11/09 23:01:54

69957	16	-15,79372	-47,89013	22/11/09 23:02:04
69958	16	-15,7923	-47,88967	22/11/09 23:02:14
69959	16	-15,79069	-47,88908	22/11/09 23:02:24
69960	16	-15,78918	-47,8886	22/11/09 23:02:34
69961	16	-15,78841	-47,88837	22/11/09 23:02:44
69962	16	-15,78709	-47,88804	22/11/09 23:02:54
69963	16	-15,78626	-47,88781	22/11/09 23:03:04
69964	16	-15,78624	-47,8878	22/11/09 23:03:14
69965	16	-15,78618	-47,88779	22/11/09 23:03:24
69966	16	-15,78514	-47,88763	22/11/09 23:03:34
69967	16	-15,78412	-47,88756	22/11/09 23:03:44
69968	16	-15,78315	-47,88751	22/11/09 23:03:54
69969	16	-15,78178	-47,88752	22/11/09 23:04:04
69970	16	-15,78077	-47,88755	22/11/09 23:04:14
69971	16	-15,7793	-47,88765	22/11/09 23:04:24
69972	16	-15,77822	-47,88774	22/11/09 23:04:34
69973	16	-15,77787	-47,88777	22/11/09 23:04:44
69974	16	-15,777	-47,8879	22/11/09 23:04:54
69975	14	-15,77596	-47,88798	22/11/09 23:05:04
69976	16	-15,77542	-47,88803	22/11/09 23:05:14
69977	16	-15,77535	-47,88804	22/11/09 23:05:24
69978	16	-15,77495	-47,88809	22/11/09 23:05:34
69979	16	-15,77382	-47,88828	22/11/09 23:05:44

APÊNDICE C – Resultado das pesquisas com o ônibus em movimento

Hora da Pesquisa	Origem			Destino			Resultado	
	Bairro	Avenida	Ponto	Bairro	Avenida	Ponto	Distância	Tempo
22:40	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
22:45	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
							2.2 Km	3 mins
22:49	Asa Norte	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
							1.1 Km	2 mins
22:52	Asa Sul	W3	703/704	Asa Sul	W3	710	1.6 km	3 mins
22:55	Asa Sul	W3	707/708	Asa Sul	W3	710	2.4 km	4 mins
							0.6 Km	58 secs
22:56	Asa Sul	W3	707/708	Asa Sul	W3	710	2.4 km	4 mins
22:57	Asa Sul	W3	503/504	Asa Norte	W3	510	0.8 Km	1 min
22:58	Asa Sul	W3	503/504	Asa Norte	W3	510	0.8 Km	1 min
							1.2 Km	2 mins
23:01	Asa Sul	W3	503/504	Asa Norte	W3	510	0.8 Km	1 min
23:05	Asa Norte	W3	510	Asa Norte	W3	511	1.3 Km	3 mins

APÊNDICE D – Códigos do programa

O código implementado nesse projeto está demonstrado a seguir:

Código 1 - BatchBO.class.php

```
<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetofinal/app/PF/classes/util/constantess.inc.php');
require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(INTEGRATION_PATH_PF . '/DAOFactory.class.php');

/**
 * Classe Business Object do caso de uso 'Processamento Automático.
 * @author Tiago Velloso Silva 16.11.2009
 * @version 1.0
 */
class BatchBO {
    public function __construct() {
    }

    public function processar($codOnibus, $array) {
        $listaOrigemTO = NULL;
        try {
            //Recupera a instância da classe DAO através da Factory.
            $batchDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'BatchDAO');

            $batchDAO->processar($codOnibus, $array);

        } catch(SystemException $se) {
            throw $se;
        }
    }
}
?>
```

Código 2 - ParametrosBO.class.php

```
<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetofinal/app/PF/classes/util/constantess.inc.php');
require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(INTEGRATION_PATH_PF . '/DAOFactory.class.php');
```

```

/**
 * Classe Business Object do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ParametrosBO {
    public function __construct() {
    }

    public function consultarAvenida($parametrosTO) {
        $listaAvenidaTO = NULL;
        try {
            //Recupera a instância da classe DAO através da Factory.
            $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');

            $listaAvenidaTO = $parametrosDAO->consultarAvenida($parametrosTO
        } catch(SystemException $se) {
            throw $se;
        }
        return $listaAvenidaTO;
    }

    public function consultarPonto($parametrosTO) {
        $listaPontoTO = NULL;
        try {
            $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');

            $listaPontoTO = $parametrosDAO->consultarPonto($parametrosTO);
        } catch(SystemException $se) {
            throw $se;
        }
        return $listaPontoTO;
    }

    public function consultarSentido($to) {
        $listaPontoTO = NULL;
        try {
            $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');

            $to = $parametrosDAO->consultarSentido($to);
        } catch(SystemException $se) {
            throw $se;
        }
        return $to;
    }
}

```

```

        public function consultarPontos($to) {
            $listaPontoTO = NULL;
            try {
                $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');
                $to = $parametrosDAO->consultarPontos($to);
            } catch(SystemException $se) {
                throw $se;
            }
            return $to;
        }

        public function listarLinhas() {
            $listaLinhaTO = NULL;
            try {
                $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');
                $listaLinhaTO = $parametrosDAO->listarLinhas();
            } catch(SystemException $se) {
                throw $se;
            }
            return $listaLinhaTO;
        }

        public function consultarOnibus($nuLinha) {
            $listaOnibusTO = NULL;
            try {
                $parametrosDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ParametrosDAO');
                $listaOnibusTO = $parametrosDAO->consultarOnibus($nuLinha);
            } catch(SystemException $se) {
                throw $se;
            }
            return $listaOnibusTO;
        }
    }
}
?>

```

Código 3 - PesqTxtBO.class.php

```

<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
    require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');
    require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
    require_once(INTEGRATION_PATH_PF . '/DAOFactory.class.php');

```

```

/**
 * Classe Business Object do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class PesqTxtBO {
    public function __construct() {
    }
    public function listarOrigem() {
        $listaOrigemTO = NULL;
        try {
            //Recupera a instância da classe DAO através da Factory.
            $pesqTxtDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'PesqTxtDAO');
            $listaOrigemTO = $pesqTxtDAO->listarOrigem();
        } catch(SystemException $se) {
            throw $se;
        }
        return $listaOrigemTO;
    }
    public function consultarAvenida() {
        $listaOrigemTO = NULL;
        try {
            $pesqTxtDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'PesqTxtDAO');
            $listaOrigemTO = $pesqTxtDAO->listarOrigem();
        } catch(SystemException $se) {
            throw $se;
        }
        return $listaOrigemTO;
    }
}
?>

```

Código 4 - ResultPesqBO.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(INTEGRATION_PATH_PF . '/DAOFactory.class.php');
require_once(TO_PATH . '/ResultadoTO.class.php');

```

```

/**
 * Classe Business Object do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ResultPesqBO {
    public function __construct() {
    }
    public function pesquisar($to, $tipo) {
        $listaResultado = NULL;
        try {
            $resultPesqDAO = DAOFactory::getInstanceClass(DAOFactory::$SAUO,
'ResultPesqDAO');
            if($tipo == 1){ //pesquisar via texto
                $listaResultado = $resultPesqDAO->pesquisar($to);
            }
            if($tipo == 2){ //pesquisar via mapa
                $listaResultado = $resultPesqDAO->pesquisarMapa($to);
            }
        } catch(SystemException $se) {
            throw $se;
        }
        return $listaResultado;
    }
}
?>

```

Código 5 - AbstractFacade.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/app/PF/SAUO/classes/util/constantes.inc.php');
require_once(UTIL_PATH . '/connection/ConnectionUtil.class.php');

/**
 * Classe responsável por abstrair particularidades das classes facade.
 * @author Tiago Velloso Silva - 18.08.2009
 * @version 1.0
 */
abstract class AbstractFacade {

    /**
     * Cria uma conexão com o banco de dados.
     * @return connection
     */
}

```

```

protected function getConnection() {
    try {
        $connectionUtil = ConnectionUtil::getInstance();
        return $connectionUtil->getConnection();
    } catch (SystemException $se) {
        throw $se;
    }
}

/**
 * Abre uma transação setando a propriedade auto commit do banco de dados como false.
 * @param Connection $connection
 */
protected function openTransaction($connection) {
    try {
        $connectionUtil = ConnectionUtil::getInstance();
        $connectionUtil->openTransaction($connection);
    } catch (SystemException $se) {
        throw $se;
    }
}

/**
 * Realiza um commit na transação.
 * @param Connection $connection
 */
protected function commitTransaction($connection) {
    try {
        $connectionUtil = ConnectionUtil::getInstance();
        $connectionUtil->commitTransaction($connection);
    } catch (SystemException $se) {
        throw $se;
    }
}

/**
 * Realiza um rollback na transação.
 * @param Connection $connection
 */
protected function rollbackTransaction($connection) {
    try {
        $connectionUtil = ConnectionUtil::getInstance();
        $connectionUtil->rollbackTransaction($connection);
    } catch (SystemException $se) {

```

```

        throw $se;
    }
}

/**
 * Fecha a conexão liberando os recursos do banco de dados.
 * @param Connection $connection
 */
protected function closeConnection($connection) {
    try {
        $connectionUtil = ConnectionUtil::getInstance();
        $connectionUtil->closeConnection($connection);
    } catch (SystemException $se) {
        throw $se;
    }
}
}
?>

```

Código 6 - BatchFacade.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
require_once(BUSINESS_PATH_PF . '/facade/AbstractFacade.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(BUSINESS_PATH . '/bo/BatchBO.class.php');

/**
 * Classe facade do caso de uso 'Processamento Batch'.
 * @author Tiago Velloso Silva 16.11.2009
 * @version 1.0
 */
class BatchFacade extends AbstractFacade {

    public function __construct() {
    }

    public function processar($codOnibus, $array) {
        $listaOrigemTO = NULL;
        try {
            //Cria uma instância da classe BO.
            $batchBO = new BatchBO();
            $batchBO->processar($codOnibus, $array);
        } catch (SystemException $se) {
            parent::closeConnection($connection);
            throw $se;
        }
    }
}

```

```

        } catch(BusinessException $be) {
            parent::closeConnection($connection);
            throw $be;
        }
    }
}
?>

```

Código 7 - ParametrosFacade.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
require_once(BUSINESS_PATH_PF . '/facade/AbstractFacade.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(BUSINESS_PATH . '/bo/ParametrosBO.class.php');

/**
 * Classe facade do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ParametrosFacade extends AbstractFacade {
    public function __construct() {
    }
    public function consultarAvenida($parametrosTO) {
        try {
            //Cria uma instância da classe BO.
            $parametrosBO = new ParametrosBO();
            $listaAvenidaTO = $parametrosBO->consultarAvenida($parametrosTO);
        } catch(SystemException $se) {
            parent::closeConnection($connection);
            throw $se;
        } catch(BusinessException $be) {
            parent::closeConnection($connection);
            throw $be;
        }
        return $listaAvenidaTO;
    }
    public function consultarPonto($parametrosTO) {
        try {
            //Cria uma instância da classe BO.
            $parametrosBO = new ParametrosBO();
            $listaPontoTO = $parametrosBO->consultarPonto($parametrosTO);
        } catch(SystemException $se) {
            parent::closeConnection($connection);

```

```

        throw $se;
    } catch(BusinessException $be) {
        parent::closeConnection($connection);
        throw $be;
    }
    return $listaPontoTO;
}
public function consultarSentido($to) {
    try {
        //Cria uma instância da classe BO.
        $parametrosBO = new ParametrosBO();
        $to = $parametrosBO->consultarSentido($to);
    } catch(SystemException $se) {
        parent::closeConnection($connection);
        throw $se;
    } catch(BusinessException $be) {
        parent::closeConnection($connection);
        throw $be;
    }
    return $to;
}
public function consultarPontos($to) {
    try {
        //Cria uma instância da classe BO.
        $parametrosBO = new ParametrosBO();
        $to = $parametrosBO->consultarPontos($to);
    } catch(SystemException $se) {
        parent::closeConnection($connection);
        throw $se;
    } catch(BusinessException $be) {
        parent::closeConnection($connection);
        throw $be;
    }
    return $to;
}
public function listarLinhas() {
    try {
        //Cria uma instância da classe BO.
        $parametrosBO = new ParametrosBO();
        $listaLinhaTO = $parametrosBO->listarLinhas();
    } catch(SystemException $se) {
        parent::closeConnection($connection);
        throw $se;
    } catch(BusinessException $be) {

```

```

        parent::closeConnection($connection);
        throw $be;
    }
    return $listaLinhaTO;
}
public function consultarOnibus($nuLinha) {
    try {
        //Cria uma instância da classe BO.
        $parametrosBO = new ParametrosBO();
        $listaOnibusTO = $parametrosBO->consultarOnibus($nuLinha);
    } catch(SystemException $se) {
        parent::closeConnection($connection);
        throw $se;
    } catch(BusinessException $be) {
        parent::closeConnection($connection);
        throw $be;
    }
    return $listaOnibusTO;
}
}
?>

```

Código 8 - PesqTxtFacade.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantas.inc.php');
require_once(BUSINESS_PATH_PF . '/facade/AbstractFacade.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(BUSINESS_PATH . '/bo/PesqTxtBO.class.php');

/**
 * Classe facade do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class PesqTxtFacade extends AbstractFacade {
    public function __construct() {
    }
    public function listarOrigem() {
        $listaOrigemTO = NULL;
        try {
            //Cria uma instância da classe BO.
            $pesqTxtBO = new PesqTxtBO();
            $listaOrigemTO = $pesqTxtBO->listarOrigem();
        } catch(SystemException $se) {

```

```

        parent::closeConnection($connection);
        throw $se;
    } catch(BusinessException $be) {
        parent::closeConnection($connection);
        throw $be;
    }
    return $listaOrigemTO;
}
}
?>

```

Código 9 - ResultPesqFacade.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantas.inc.php');
require_once(BUSINESS_PATH_PF . '/facade/AbstractFacade.class.php');
require_once(EXCEPTION_PATH_PF . '/BusinessException.class.php');
require_once(BUSINESS_PATH . '/bo/ResultPesqBO.class.php');

/**
 * Classe facade do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ResultPesqFacade extends AbstractFacade {
    public function __construct() {
    }
    public function pesquisar($to, $tipo) {
        $listaResultado = NULL;
        try {
            //Cria uma instância da classe BO.
            $resultPesqBO = new ResultPesqBO();
            $listaResultado = $resultPesqBO->pesquisar($to, $tipo);
        } catch(SystemException $se) {
            parent::closeConnection($connection);
            throw $se;
        } catch(BusinessException $be) {
            parent::closeConnection($connection);
            throw $be;
        }
        return $listaResultado;
    }
}
?>

```

Código 10 - BusinessException.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/app/PF/SAUO/classes/util/constantes.inc.php');
require_once(UTIL_PATH . '/PropertiesUtil.class.php');

/**
 * Classe responsável por tratar as exceções de negócio do sistema.
 * @author Tiago Velloso Silva - 17.08.2009
 * @version 1.0
 */
class BusinessException extends Exception {
    private $mensagem = NULL;
    /**
 * Construtor personalizado que receberá uma código de erro e argumentos para o arquivo de
properties.
 * @param string $message Mensagem de erro ou código do arquivo de properties (parâmetro
obrigatório).
 * @param string $arg0 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param string $arg1 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param string $arg2 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param int $code Código de erro (parâmetro não-obrigatório).
 * @author Tiago Velloso Silva
 */
    public function __construct($message, $arg0 = NULL, $arg1 = NULL, $arg2 = NULL, $code = 0) {
        $propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
        if (!is_null($message) && !(substr($message, 0, 3) === 'MSG')) {
            parent::__construct($message, $code);
            $this->mensagem = $message->toString();
        } else if (is_null($message)) {
            $message = 'MSG0000';
            $this->mensagem = $propertiesUtil->getPropertyMessage($message, $arg0, $arg1,
$arg2);
        } else {
            $this->mensagem = $propertiesUtil->getPropertyMessage($message, $arg0, $arg1,
$arg2);
        }
    }

    public function __toString() {
        return __CLASS__ . ':' . '[' . $this->code . ']' . ':' . '[' . $this->message . '];
    }

    /**
 * Retorna a mensagem do arquivo de properties ou do sistema.
 */
}

```

```

        public function getMensagem() {
            return $this->mensagem;
        }
    }
?>

```

Código 11 - SystemException.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/app/PF/SAUO/classes/util/constantes.inc.php');
require_once(UTIL_PATH . '/PropertiesUtil.class.php');
/**
 * Classe responsável por tratar as exceções de banco de dados e infra-estrutura do sistema.
 * @author Tiago Velloso Silva - 17.08.2009
 * @version 1.0
 */
class SystemException extends Exception {
    private $mensagem = NULL;
    /**
 * Construtor personalizado que receberá uma código de erro e argumentos para o arquivo de
properties.
 * @param string $message Mensagem de erro ou código do arquivo de properties (parâmetro
obrigatório).
 * @param string $arg0 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param string $arg1 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param string $arg2 Argumento para o arquivo de properties (parâmetro não-obrigatório).
 * @param int $code Código de erro (parâmetro não-obrigatório).
 * @author Tiago Velloso Silva
 */
    public function __construct($message, $arg0 = NULL, $arg1 = NULL, $arg2 = NULL, $code = 0) {
        $propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
        if (!is_null($message) && !(substr($message, 0, 3) === 'MSG')) {
            parent::__construct($message, $code);
            $this->mensagem = $message->toString();
        } else if (is_null($message)) {
            $message = 'MSG0000';
            $this->mensagem = $propertiesUtil->getPropertyMessage($message, $arg0, $arg1,
$arg2);
        } else {
            $this->mensagem = $propertiesUtil->getPropertyMessage($message, $arg0, $arg1, $arg2);
        }
    }
    public function __toString() {
        return __CLASS__ . ':' . '[' . $this->code . ']' . ':' . '[' . $this->message . '];
    }
}
/**

```

```

        * Retorna a mensagem do arquivo de properties ou do sistema.
        */
        public function getMensagem() {
            return $this->mensagem;
        }
    }
?>

```

Código 12 - BatchDAO.class.php

```

<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantas.inc.php');
    require_once($_SERVER['DOCUMENT_ROOT'] .
'/projeto-final/app/PF/SAUO/classes/integration/Conexao_Mysql_Batch.class.php');
    require_once(TO_PATH . '/BairroTO.class.php');

    /**
     * Classe DAO do caso de uso 'Processamento Batch'.
     * @author Tiago Velloso Silva 16.11.2009
     * @version 1.0
     */
    class BatchDAO{

        public function __construct() {

        }

        /**
         * Retorna um TO de origem.
         */
        public function processar($codOnibus, $array) {
            $callStatement = NULL;
            try {

                $conexao = new Conexao_Mysql_Batch();
                $conexao->connectBatch();
                $conexao->useDatabaseBatch();
                $query = sprintf("INSERT INTO dbposicao

(codOnibus, latitude, longitude, hora)

VALUES ('%s', '%s', '%s', now());",

mysql_real_escape_string($codOnibus),

mysql_real_escape_string($array[1]),

```

```

mysql_real_escape_string($array[0])
);

        $callStatement = $conexao->queryBatch($query);

    } catch(SQLException $se) {
        throw new SystemException(RESOURCE_PATH, 'MSG0000');
    }
}
}
?>

```

Código 13 - Conexao_Mysql.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantes.inc.php');
require_once($_SERVER['DOCUMENT_ROOT'] .
'projeto-final/app/PF/SAUO/classes/integration/Rowset_Mysql.class.php');
require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');

class Conexao_Mysql {
    private $dbh;

    public function connect() {
        $this->dbh = mysql_connect('localhost', 'root', 'root')
        or die(mysql_error());
    }

    public function useDatabase() {
        mysql_select_db('projeto-final', $this->dbh);
    }

    public function query($sql) {
        $res = mysql_query($sql, $this->dbh);
        return new Rowset_Mysql( $res );
    }
}
?>

```

Código 14 - ParametrosDAO.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantes.inc.php');
require_once($_SERVER['DOCUMENT_ROOT'] .

```

```

'/projetofinal/app/PF/SAUO/classes/integration/Conexao_Mysql.class.php');
require_once(TO_PATH . '/BairroTO.class.php');
require_once(TO_PATH . '/ParametrosTO.class.php');
require_once(TO_PATH . '/PontoTO.class.php');
require_once(TO_PATH . '/LinhaTO.class.php');
require_once(TO_PATH . '/OnibusTO.class.php');

/**
 * Classe DAO para Pesquisas dos Parametros a serem informados.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ParametrosDAO{

    public function __construct() {

    }

    /**
     * Retorna uma lista de AvenidaTO.
     */
    public function consultarAvenida($parametrosTO) {
        $resultado = NULL;
        $avenidaTO = NULL;
        $callStatement = NULL;
        $listaAvenidaTO = array();

        try {
            $conexao = new Conexao_Mysql();
            $conexao->connect();
            $conexao->useDatabase();
            $query = sprintf("SELECT codAvenida, descAvenida FROM dbavenida
WHERE codBairro='%s' AND tpDado=1",
                                mysql_real_escape_string($parametrosTO-
>getBairroOrigem()));

            $callStatement = $conexao->query($query);

            while ($resultado = $callStatement->fetchAssoc()) {
                $avenidaTO = new AvenidaTO();

                $avenidaTO->setCodBairro($codbairro);
                $avenidaTO->setCodAvenida($resultado['codAvenida']);
                $avenidaTO->setDescAvenida($resultado['descAvenida']);
            }
        }
    }
}

```

```

        $listaAvenidaTO[] = $avenidaTO;
    }

    } catch(SQLException $se) {
        throw new SystemException(RESOURCE_PATH, 'MSG0000');
    }
    return $listaAvenidaTO;
}

/**
 * Retorna uma lista de PontoTO.
 */
public function consultarPonto($parametrosTO) {
    $resultado = NULL;
    $pontoTO = NULL;
    $callStatement = NULL;
    $listaPontoTO = array();

    try {
        $conexao = new Conexao_Mysql();
        $conexao->connect();
        $conexao->useDatabase();
        $query = sprintf("SELECT codPonto, descPonto, latitude, longitude,
tpSentido
                                FROM dbponto
                                WHERE codAvenida='%s'
                                ORDER BY descPonto",
                                mysql_real_escape_string($parametrosTO->getLocalOrigem()));

        $callStatement = $conexao->query($query);

        while ($resultado = $callStatement->fetchAssoc()) {
            $pontoTO = new PontoTO();

            $pontoTO->setCodAvenida($codbairro);
            $pontoTO->setCodPonto($resultado['codPonto']);
            $pontoTO->setDescPonto($resultado['descPonto']);
            $pontoTO->setLatitude($resultado['latitude']);
            $pontoTO->setLongitude($resultado['longitude']);
            $pontoTO->setTpSentido($resultado['tpSentido']);
            $listaPontoTO[] = $pontoTO;
        }
    } catch(SQLException $se) {

```

```

        throw new SystemException(RESOURCE_PATH, 'MSG0000');
    }
    return $listaPontoTO;
}

/**
 * Consultar o sentido dos pontos selecionados pelo usuário
 * Retorna o ParametroTO.
 */
public function consultarSentido($parametrosTO) {
    $resultado = NULL;
    $callStatement = NULL;

    try {
        $conexao = new Conexao_Mysql();
        $conexao->connect();
        $conexao->useDatabase();

        $query = sprintf("SELECT tpSentido FROM dbponto WHERE
codPonto='%s'",
        mysql_real_escape_string($parametrosTO->getPontoOrigem()));
        $callStatement = $conexao->query($query);
        $resultado = $callStatement->fetchAssoc();
        $parametrosTO->setTpSentidoOrigem($resultado['tpSentido']);

        $query = sprintf("SELECT tpSentido FROM dbPonto WHERE
codPonto='%s'",
        mysql_real_escape_string($parametrosTO->getPontoDestino()));
        $callStatement = $conexao->query($query);
        $resultado = $callStatement->fetchAssoc();
        $parametrosTO->setTpSentidoDestino($resultado['tpSentido']);

    } catch(SQLException $se) {
        throw new SystemException(RESOURCE_PATH, 'MSG0000');
    }
    return $parametrosTO;
}

/**
 * Retorna o mesmo TO enviado com a lista de PontoTO.
 */
public function consultarPontos($to) {
    $resultado = NULL;

```

```

        $pontoTO = NULL;
        $callStatement = NULL;
        $listaPontoTO = array();

        try {
            $conexao = new Conexao_Mysql();
            $conexao->connect();
            $conexao->useDatabase();
            $query = "SELECT codPonto, descPonto, latitude, longitude, tpSentido,
codAvenida FROM dbponto";

            $callStatement = $conexao->query($query);

            while ($resultado = $callStatement->fetchAssoc()) {
                $pontoTO = new PontoTO();

                $pontoTO->setCodPonto($resultado['codPonto']);
                $pontoTO->setDescPonto($resultado['descPonto']);
                $pontoTO->setLatitude($resultado['latitude']);
                $pontoTO->setLongitude($resultado['longitude']);
                $pontoTO->setTpSentido($resultado['tpSentido']);
                $pontoTO->setCodAvenida($resultado['codAvenida']);
                $listaPontoTO[] = $pontoTO;
            }

            $to->setListaPontoTO($listaPontoTO);

        } catch(SQLException $se) {
            throw new SystemException(RESOURCE_PATH, 'MSG0000');
        }
        return $to;
    }

    /**
     * Retorna a lista de linhaTO.
     */
    public function listarLinhas() {
        $resultado = NULL;
        $linhaTO = NULL;
        $callStatement = NULL;
        $listaLinhaTO = array();

        try {
            $conexao = new Conexao_Mysql();
            $conexao->connect();

```

```

$conexao->useDatabase();
$query = "SELECT nuLinha, descLinha FROM dblinha";

$callStatement = $conexao->query($query);

while ($resultado = $callStatement->fetchAssoc()) {
    $linhaTO = new linhaTO();

    $linhaTO->setNuLinha($resultado['nuLinha']);
    $linhaTO->setDescLinha($resultado['descLinha']);

    $listaLinhaTO[] = $linhaTO;
}

} catch(SQLException $se) {
    throw new SystemException(RESOURCE_PATH, 'MSG0000');
}
return $listaLinhaTO;
}

/**
 * Retorna a lista de onibusTO.
 */
public function consultarOnibus($nuLinha) {
    $resultado = NULL;
    $onibusTO = NULL;
    $callStatement = NULL;
    $listaOnibusTO = array();

    try {
        $conexao = new Conexao_Mysql();
        $conexao->connect();
        $conexao->useDatabase();
        $query = sprintf("SELECT codOnibus, horario FROM dbonibus WHERE
nuLinha='%s'",
mysql_real_escape_string($nuLinha));

        $callStatement = $conexao->query($query);

        while ($resultado = $callStatement->fetchAssoc()) {
            $onibusTO = new onibusTO();

            $onibusTO->setCodOnibus($resultado['codOnibus']);
            $onibusTO->setHorario($resultado['horario']);

```

```

                $listaOnibusTO[] = $onibusTO;
            }

        } catch(SQLException $se) {
            throw new SystemException(RESOURCE_PATH, 'MSG0000');
        }
        return $listaOnibusTO;
    }
}
?>

```

Código 15 - PesqTxtDAO.class.php

```

<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantas.inc.php');
    require_once($_SERVER['DOCUMENT_ROOT'] .
'/projetoFinal/app/PF/SAUO/classes/integration/Conexao_Mysql.class.php');
    require_once(TO_PATH . '/BairroTO.class.php');

    /**
     * Classe DAO do caso de uso 'Pesquisar por Texto'.
     * @author Tiago Velloso Silva 19.08.2009
     * @version 1.0
     */
    class PesqTxtDAO{

        public function __construct() {

        }

        /**
         * Retorna um TO de origem.
         */
        public function listarOrigem() {
            $resultado = NULL;
            $bairroTO = NULL;
            $callStatement = NULL;
            $listaOrigemTO = array();

            try {
                $conexao = new Conexao_Mysql();
                $conexao->connect();
                $conexao->useDatabase();
                $callStatement = $conexao->query('SELECT codBairro, descBairro

```

```

FROM projetofinal.dbBairro');

        while ($resultado = $callStatement->fetchAssoc()) {
            $bairroTO = new BairroTO();

            $bairroTO->setCodBairro($resultado['codBairro']);
            $bairroTO->setDescBairro($resultado['descBairro']);
            $listabairroTO[] = $bairroTO;
        }
        $listaOrigemTO[0] = $listabairroTO;

    } catch(SQLException $se) {
        throw new SystemException(RESOURCE_PATH, 'MSG0000');
    }
    return $listaOrigemTO;
}
}
?>

```

Código 16 - ResultPesqDAO.class.php

```

<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetofinal/app/PF/classes/util/constantes.inc.php');
    require_once($_SERVER['DOCUMENT_ROOT'] .
'/projetofinal/app/PF/SAUO/classes/integration/Conexao_Mysql.class.php');
    require_once(TO_PATH . '/OrigemTO.class.php');
    require_once(TO_PATH . '/ParametrosTO.class.php');
    require_once(TO_PATH . '/ParametrosMapaTO.class.php');
    require_once(TO_PATH . '/resultadoTO.class.php');
    require_once(TO_PATH . '/linhaTO.class.php');

    /**
     * Classe DAO para o Resultado das Pesquisas.
     * @author Tiago Velloso Silva 19.08.2009
     * @version 1.0
     */
    class ResultPesqDAO {

        public function __construct() {

        }

        /**
         * Retorna uma listaResultado com os TOs .

```

```

*/
public function pesquisar($parametrosTO) {
    try {
        $resultado = NULL;
        $resultadoTO = NULL;
        $callStatement = NULL;
        $tipoConsulta = NULL;
        $listaResultado = array();

        $localOrigem = $parametrosTO->getLocalOrigem();
        $localDestino = $parametrosTO->getLocalDestino();
        $sentidoOrigem = $parametrosTO->getTpSentidoOrigem();
        $sentidoDestino = $parametrosTO->getTpSentidoDestino();

        //Norte para sul -> Aumenta Latitude e Longitude (26 para 10) = 1
        //Sul para norte -> Diminui Latitude e Longitude (10 para 26) = 2

        if($sentidoOrigem == $sentidoDestino && $sentidoOrigem == 1){
            $tipoConsulta = 1;
        }
        if($sentidoOrigem == $sentidoDestino && $sentidoOrigem == 2){
            $tipoConsulta = 2;
        }

        $conexao = new Conexao_Mysql();
        $conexao->connect();
        $conexao->useDatabase();

        //Buscar pela latitude e longitude dos ponto de origem informado
        $query = sprintf("SELECT latitude, longitude, codAvenida FROM
projetofinal.dbponto WHERE codPonto='%s'",

mysql_real_escape_string($parametrosTO->getPontoOrigem()));
        $callStatement = $conexao->query($query);
        while ($resultado1 = $callStatement->fetchAssoc()) {
            $latitudePonto = $resultado1['latitude'];
            $longitudePonto = $resultado1['longitude'];
            $codAvenida = $resultado1['codAvenida'];
        }

        if($codAvenida == 10 || $codAvenida == 26){
            if($tipoConsulta == 1){
                $sinal = '>=';
            }else{

```

```

        $sinal = '<=';
    }
    $avenidaParcial[] = 39;
    $avenidaParcial[] = 40;
    $avenidaParcial[] = 41;
    $avenidaParcial[] = 42;
}
if($codAvenida == 17 || $codAvenida == 19){
    if($tipoConsulta == 1){
        $sinal = '<=';
    }else{
        $sinal = '>=';
    }
    $avenidaParcial[] = 43;
    $avenidaParcial[] = 44;
    $avenidaParcial[] = 45;
    $avenidaParcial[] = 46;
}

if($codAvenida == 19 || $codAvenida == 26){
    $query1 = sprintf("SELECT dblinha.nuLinha,
dblinha.descLinha, dbonibus.codOnibus, dbposicao.latitude, dbposicao.longitude, MAX(dbposicao.hora) AS
Hora
FROM
dbavenida, dblinha, dbonibus, dbposicao
WHERE
dblinha.nuLinha = dbonibus.nuLinha AND
dbonibus.codOnibus = dbposicao.codOnibus AND
dbposicao.latitude between latInicio AND latFim AND
dbposicao.longitude between lngFim AND lngInicio AND
dbposicao.latitude %s '%s' AND
dbavenida.codAvenida = '%s' AND
dbonibus.tpSentido = '%s'
dbonibus.codOnibus",
GROUP BY
mysql_real_escape_string($sinal),
mysql_real_escape_string($latitudePonto),
mysql_real_escape_string($codAvenida),
mysql_real_escape_string($tipoConsulta));

```

```

}else if($codAvenida == 10 || $codAvenida == 17){

                                $query1 = sprintf(" SELECT dblinha.nuLinha,
dblinha.descLinha, dbonibus.codOnibus, dbposicao.latitude, dbposicao.longitude, MAX(dbposicao.hora) AS
Hora

                                                                    FROM
dbavenida, dblinha, dbonibus, dbposicao

                                                                    WHERE
dblinha.nuLinha = dbonibus.nuLinha AND

                                dbonibus.codOnibus = dbposicao.codOnibus AND

                                dbposicao.latitude between latFim AND latInicio AND

                                dbposicao.longitude between lngFim AND lngInicio AND

                                dbposicao.latitude %s '%s' AND

                                (codAvenida = '%s' OR codAvenida = '%s' OR codAvenida = '%s' OR codAvenida = '%s') AND

                                dbonibus.tpSentido = %s

                                                                    GROUP BY
dbonibus.codOnibus",
                                mysql_real_escape_string($sinal),
                                mysql_real_escape_string($latitudePonto),
                                mysql_real_escape_string($avenidaParcial[0]),
                                mysql_real_escape_string($avenidaParcial[1]),
                                mysql_real_escape_string($avenidaParcial[2]),
                                mysql_real_escape_string($avenidaParcial[3]),
                                mysql_real_escape_string($tipoConsulta));
                                }
                                $callStatement = $conexao->query($query1);
                                while ($resultado = $callStatement->fetchAssoc()) {
                                    $resultadoTO = new resultadoTO();
                                    $linhaTO = new linhaTO();
                                    $linhaTO->setNuLinha($resultado['nuLinha']);
                                    $linhaTO->setDescLinha($resultado['descLinha']);
                                    $resultadoTO->setLinhaTO($linhaTO);

                                    //Posiçao do Onibus
                                    $resultadoTO->setLatitudeOrigem($resultado['latitude']);
                                    $resultadoTO->setLongitudeOrigem($resultado['longitude']);

                                    //Posicao do Ponto que o usuario ira pegar o onibus
                                    $resultadoTO->setLatitudeDestino($latitudePonto);
                                    $resultadoTO->setLongitudeDestino($longitudePonto);

                                    //Buscar terminal de origem do onibus
                                    $query2 = sprintf("SELECT          dbAvenida.codAvenida,

```

```

descAvenida AS AvenidaOrigem
dbAvenida
FROM dbItinerario,
WHERE
(dbItinerario.ordem=1 and dbItinerario.codAvenida = dbAvenida.codAvenida) and nuLinha='%s',
mysql_real_escape_string($resultado['nuLinha']);
$callStatement1 = $conexao->query($query2);
while ($resultado2 = $callStatement1->fetchAssoc()) {
    $linhaTO->setCodOrigemLinha($resultado2['codAvenida']);
    $linhaTO->setDescOrigemLinha($resultado2['AvenidaOrigem']);
}

//Buscar terminal de destino do onibus
$query3 = sprintf("SELECT dbAvenida.codAvenida,
descAvenida AS AvenidaDestino
dbAvenida
FROM dbItinerario,
WHERE
(dbItinerario.codAvenida = dbAvenida.codAvenida) and nuLinha='%s'
ORDER BY
dbItinerario.ordem DESC
LIMIT 0,1",
mysql_real_escape_string($resultado['nuLinha']));
$callStatement2 = $conexao->query($query3);
while ($resultado3 = $callStatement2->fetchAssoc()) {
    $linhaTO->setCodDestinoLinha($resultado3['codAvenida']);
    $linhaTO->setDescDestinoLinha($resultado3['AvenidaDestino']);
}
$listaResultado[] = $resultadoTO;
}
} catch(SQLException $se) {
    throw new SystemException(RESOURCE_PATH, 'MSG0000');
}
return $listaResultado;
}
}
?>

```

Código 17 - Rowset_Mysql.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
require_once(EXCEPTION_PATH_PF . '/SystemException.class.php');

```

```

class Rowset_Mysql {
    private $res;
    public function __construct($res) {
        $this->res = $res;
    }

    public function count() {
        return mysql_num_rows($this->res);
    }
    public function fetchAssoc() {
        return mysql_fetch_assoc($this->res);
    }
}
?>

```

Código 18 - AvenidaTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class AvenidaTO {
    private $codBairro = NULL;
    private $codAvenida = NULL;
    private $descAvenida = NULL;

    public function getCodBairro() {
        return $this->codBairro;
    }
    public function setCodBairro($codBairro) {
        $this->codBairro = $codBairro;
    }
    public function getCodAvenida() {
        return $this->codAvenida;
    }
    public function setCodAvenida($codAvenida) {
        $this->codAvenida = $codAvenida;
    }
    public function getDescAvenida() {
        return $this->descAvenida;
    }
    public function setDescAvenida($descAvenida) {
        $this->descAvenida = $descAvenida;
    }
}

```

```

    }
}
?>

```

Código 19 - BairroTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class BairroTO {
    private $codBairro = NULL;
    private $descBairro = NULL;

    public function getCodBairro() {
        return $this->codBairro;
    }
    public function setCodBairro($codBairro) {
        $this->codBairro = $codBairro;
    }
    public function getDescBairro() {
        return $this->descBairro;
    }
    public function setDescBairro($descBairro) {
        $this->descBairro = $descBairro;
    }
}
?>

```

Código 20 - LinhaTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class LinhaTO {
    private $nuLinha = NULL;
    private $descLinha = NULL;
    private $codOrigemLinha = NULL;
    private $descOrigemLinha = NULL;
    private $codDestinoLinha = NULL;
}

```

```
private $descDestinoLinha = NULL;

public function getNuLinha() {
    return $this->nuLinha;
}

public function setNuLinha($nuLinha) {
    $this->nuLinha = $nuLinha;
}

public function getDescLinha() {
    return $this->descLinha;
}

public function setDescLinha($descLinha) {
    $this->descLinha = $descLinha;
}

public function getCodOrigemLinha() {
    return $this->codOrigemLinha;
}

public function setCodOrigemLinha($codOrigemLinha) {
    $this->codOrigemLinha = $codOrigemLinha;
}

public function getDescOrigemLinha() {
    return $this->descOrigemLinha;
}

public function setDescOrigemLinha($descOrigemLinha) {
    $this->descOrigemLinha = $descOrigemLinha;
}

public function getCodDestinoLinha() {
    return $this->codDestinoLinha;
}

public function setCodDestinoLinha($codDestinoLinha) {
    $this->codDestinoLinha = $codDestinoLinha;
}

public function getDescDestinoLinha() {
    return $this->descDestinoLinha;
}

public function setDescDestinoLinha($descDestinoLinha) {
    $this->descDestinoLinha = $descDestinoLinha;
}

}

?>
```

Código 21 - OnibusTO.class.php

<?php

```

/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class OnibusTO {
    private $codOnibus = NULL;
    private $tpSentido = NULL;
    private $nuLinha = NULL;
    private $horario = NULL;

    public function getCodOnibus() {
        return $this->codOnibus;
    }
    public function setCodOnibus($codOnibus) {
        $this->codOnibus = $codOnibus;
    }
    public function getTpSentido() {
        return $this->tpSentido;
    }
    public function setTpSentido($tpSentido) {
        $this->tpSentido = $tpSentido;
    }
    public function getNuLinha() {
        return $this->nuLinha;
    }
    public function setNuLinha($nuLinha) {
        $this->nuLinha = $nuLinha;
    }
    public function getHorario() {
        return $this->horario;
    }
    public function setHorario($horario) {
        $this->horario = $horario;
    }
}
?>

```

Código 22 - OrigemTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0

```

```
*/  
class OrigemTO {  
    private $codBairro = NULL;  
    private $descBairro = NULL;  
    private $codPonto = NULL;  
    private $descPonto = NULL;  
    private $codAvenida = NULL;  
    private $descAvenida = NULL;  
  
    public function getCodBairro() {  
        return $this->codBairro;  
    }  
    public function setCodBairro($codBairro) {  
        $this->codBairro = $codBairro;  
    }  
    public function getDescBairro() {  
        return $this->descBairro;  
    }  
    public function setDescBairro($descBairro) {  
        $this->descBairro = $descBairro;  
    }  
  
    public function getCodPonto() {  
        return $this->codPonto;  
    }  
    public function setCodPonto($codPonto) {  
        $this->codPonto = $codPonto;  
    }  
    public function getDescPonto() {  
        return $this->descPonto;  
    }  
    public function setDescPonto($descPonto) {  
        $this->descPonto = $descPonto;  
    }  
    public function getCodAvenida() {  
        return $this->codAvenida;  
    }  
    public function setCodAvenida($codAvenida) {  
        $this->codAvenida = $codAvenida;  
    }  
    public function getDescAvenida() {  
        return $this->descAvenida;  
    }  
    public function setDescAvenida($descAvenida) {
```

```

        $this->descAvenida = $descAvenida;
    }
}
?>

```

Código 23 - ParametrosMapaTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os parametros de pesquisa.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */

class ParametrosMapaTO {
    private $origemLat = NULL;
    private $origemLon = NULL;
    private $destinoLat = NULL;
    private $destinoLon = NULL;
    private $listaPontoTO = NULL;

    public function getOrigemLat() {
        return $this->origemLat;
    }
    public function setOrigemLat($origemLat) {
        $this->origemLat = $origemLat;
    }
    public function getOrigemLon() {
        return $this->origemLon;
    }
    public function setOrigemLon($origemLon) {
        $this->origemLon = $origemLon;
    }
    public function getDestinoLat() {
        return $this->destinoLat;
    }
    public function setDestinoLat($destinoLat) {
        $this->destinoLat = $destinoLat;
    }
    public function getDestinoLon() {
        return $this->destinoLon;
    }
    public function setDestinoLon($destinoLon) {
        $this->destinoLon = $destinoLon;
    }
}

```

```

        public function getListaPontoTO() {
            return $this->listaPontoTO;
        }
        public function setListaPontoTO($listaPontoTO) {
            $this->listaPontoTO = $listaPontoTO;
        }
    }
?>

```

Código 24 - ParametrosTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os parametros de pesquisa.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */

class ParametrosTO {
    private $codLinha = NULL;
    private $bairroOrigem = NULL;
    private $localOrigem = NULL;
    private $pontoOrigem = NULL;
    private $tpSentidoOrigem = NULL;
    private $bairroDestino = NULL;
    private $localDestino = NULL;
    private $pontoDestino = NULL;
    private $tpSentidoDestino = NULL;

    public function getCodLinha() {
        return $this->codLinha;
    }
    public function setCodLinha($codLinha) {
        $this->codLinha = $codLinha;
    }
    public function getBairroOrigem() {
        return $this->bairroOrigem;
    }
    public function setBairroOrigem($bairroOrigem) {
        $this->bairroOrigem = $bairroOrigem;
    }
    public function getLocalOrigem() {
        return $this->localOrigem;
    }
    public function setLocalOrigem($localOrigem) {

```

```

        $this->localOrigem = $localOrigem;
    }
    public function getPontoOrigem() {
        return $this->pontoOrigem;
    }
    public function setPontoOrigem($pontoOrigem) {
        $this->pontoOrigem = $pontoOrigem;
    }
    public function getTpSentidoOrigem() {
        return $this->tpSentidoOrigem;
    }
    public function setTpSentidoOrigem($tpSentidoOrigem) {
        $this->tpSentidoOrigem = $tpSentidoOrigem;
    }
    public function getBairroDestino() {
        return $this->bairroDestino;
    }
    public function setBairroDestino($bairroDestino) {
        $this->bairroDestino = $bairroDestino;
    }
    public function getLocalDestino() {
        return $this->localDestino;
    }
    public function setLocalDestino($localDestino) {
        $this->localDestino = $localDestino;
    }
    public function getPontoDestino() {
        return $this->pontoDestino;
    }
    public function setPontoDestino($pontoDestino) {
        $this->pontoDestino = $pontoDestino;
    }
    public function getTpSentidoDestino() {
        return $this->tpSentidoDestino;
    }
    public function setTpSentidoDestino($tpSentidoDestino) {
        $this->tpSentidoDestino = $tpSentidoDestino;
    }
}
?>

```

Código 25 - PontoTO.class.php

```

<?php
/**

```

```
* Classe que implementa o pattern Transfer Object para os dados de Origem.  
* @author Tiago Velloso Silva 19.08.2009  
* @version 1.0  
*/  
class PontoTO {  
    private $codPonto = NULL;  
    private $descPonto = NULL;  
    private $codAvenida = NULL;  
    private $latitude = NULL;  
    private $longitude = NULL;  
    private $tpSentido = NULL;  
  
    public function getCodPonto() {  
        return $this->codPonto;  
    }  
    public function setCodPonto($codPonto) {  
        $this->codPonto = $codPonto;  
    }  
    public function getDescPonto() {  
        return $this->descPonto;  
    }  
    public function setDescPonto($descPonto) {  
        $this->descPonto = $descPonto;  
    }  
    public function getCodAvenida() {  
        return $this->codAvenida;  
    }  
    public function setCodAvenida($codAvenida) {  
        $this->codAvenida = $codAvenida;  
    }  
    public function getLatitude() {  
        return $this->latitude;  
    }  
    public function setLatitude($latitude) {  
        $this->latitude = $latitude;  
    }  
    public function getLongitude() {  
        return $this->longitude;  
    }  
    public function setLongitude($longitude) {  
        $this->longitude = $longitude;  
    }  
    public function getTpSentido() {  
        return $this->tpSentido;  
    }  
}
```

```

    }
    public function setTpSentido($tpSentido) {
        $this->tpSentido = $tpSentido;
    }
}
?>

```

Código 26 - ResultadoTO.class.php

```

<?php
/**
 * Classe que implementa o pattern Transfer Object para os dados de Origem.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
class ResultadoTO {
    private $linhaTO = NULL;
    private $distancia = NULL;
    private $tempo = NULL;
    private $latitudeOrigem = NULL;
    private $longitudeOrigem = NULL;
    private $latitudeDestino = NULL;
    private $longitudeDestino = NULL;

    public function getLinhaTO() {
        return $this->linhaTO;
    }
    public function setLinhaTO($linhaTO) {
        $this->linhaTO = $linhaTO;
    }
    public function getDistancia() {
        return $this->distancia;
    }
    public function setDistancia($distancia) {
        $this->distancia = $distancia;
    }
    public function getTempo() {
        return $this->tempo;
    }
    public function setTempo($tempo) {
        $this->tempo = $tempo;
    }
    public function getLatitudeOrigem() {
        return $this->latitudeOrigem;
    }
}

```

```

public function setLatitudeOrigem($latitudeOrigem) {
    $this->latitudeOrigem = $latitudeOrigem;
}
public function getLongitudeOrigem() {
    return $this->longitudeOrigem;
}
public function setLongitudeOrigem($longitudeOrigem) {
    $this->longitudeOrigem = $longitudeOrigem;
}
public function getLatitudeDestino() {
    return $this->latitudeDestino;
}
public function setLatitudeDestino($latitudeDestino) {
    $this->latitudeDestino = $latitudeDestino;
}
public function getLongitudeDestino() {
    return $this->longitudeDestino;
}
public function setLongitudeDestino($longitudeDestino) {
    $this->longitudeDestino = $longitudeDestino;
}
}

```

?>

Código 27 - constantes.inc.php

```

<?php
#####
# Arquivo include de constantes do sistema. #
# Autor: Tiago Velloso Silva - 17.08.2009 #
# Version: 1.0 #
#####

define('INTEGRATION_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/classes/integration');
define('BUSINESS_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/classes/business');
define('EXCEPTION_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/classes/exception');
define('TO_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/classes/to);
define('UTIL_PATH', $_SERVER['DOCUMENT_ROOT'] . /app/PF/SAUO/classes/util);
define('VIEW_PATH', $_SERVER['DOCUMENT_ROOT'] . /app/PF/SAUO/classes/view);
define('VIEW_INC_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/classes/view/inc);
define('INC_PATH', $_SERVER['DOCUMENT_ROOT'] .
/app/PF/SAUO/page/inc);

```

```

define('RESOURCE_PATH',      $_SERVER['DOCUMENT_ROOT'] . '/app/PF/SAUO/classes');
define('KML_PATH',           $_SERVER['DOCUMENT_ROOT'] . "");
define('JS_PATH',            $_SERVER['DOCUMENT_ROOT'] .
'/app/PF/SAUO/web/js');
define('ERROR_PAGE',        '/app/PF/SAUO/page/errorPage.php');
?>

```

Código 28 - HtmlUtil.class.php

```

<?php
require_once($_SERVER['DOCUMENT_ROOT'] . '/app/PF/SAUO/classes/util/constantes.inc.php');

/**
 * Classe util para gerar HTML.
 * @author Tiago Velloso Silva - 17.08.2009
 * @version 1.0
 */
final class HtmlUtil {
    private static $selectTag = "";
    private static $alertScript = "";

    /**
     * Não será permitido criar instâncias dessa classe.
     */
    private function __construct() {

    }

    /**
     * Método responsável por gerar um combobox.
     * @param $name Nome do combobox.
     * @param $toName Nome do objeto TO que está no array que gerará os options.
     * @param $valueProperty Nome do atributo do objeto TO que representará o elemento 'value'
da tag <option>.
     * @param $labelProperty Nome do atributo do objeto TO que representará a descrição da tag
<option>.
     * @param $list Array com os TOs que irão compor os options.
     * @param $codigo Usado para gerar um combobox com um option pré-selecionado de
acordo com o código (opcional).
     * @param $javaScript Usado para chamar funções JavaScript (opcional). Ex:
'onChange=\alert();'
     * @param $size Usado para gerar combobox que permite seleção múltipla (opcional).
     */
    public static function gerarCombobox($name, $toName, $valueProperty, $labelProperty, array
$list, $codigo = NULL, $javaScript = NULL, $size = 0) {
        self::$selectTag = '<select name=\'. $name . \' ';
        self::$selectTag .= ((!is_null($javaScript)) ? $javaScript : "");
    }
}

```

```

self::$selectTag .= (($size != 0) ? ' size="' . $size . '\ multiple' : '') . '>';
self::$selectTag .= '<option value=\'-1\'>Selecione...</option>';

if(isset($stoName)) {
    require_once(TO_PATH . '/' . $stoName . '.class.php');
    if(isset($list) && count($list) > 0) {
        while(list($key, $sto) = each($list)) {
            $value = call_user_method('get' . ucfirst($valueProperty),
$sto);
            $label = call_user_method('get' . ucfirst($labelProperty),
$sto);
            $selected = "";
            if(!is_null($codigo) && ($codigo == $label)) {
                $selected = 'selected';
            }
            self::$selectTag .= '<option value="' . $value . "\" .
$selected . '>';
            self::$selectTag .= $label . '</option>';

            $selected = "";
        }
    }
}
self::$selectTag .= '</select>';
echo(self::$selectTag);
}

/**
 * Método responsável por gerar um alert para mensagens de sucesso.
 * @param $mensagem Mensagem a ser exibida no alert.
 */
public static function alert($mensagem) {
    if(!is_null($mensagem) && (is_string($mensagem) && $mensagem !== "")) {
        self::$alertScript = '<script language=\'JavaScript\'>';
        self::$alertScript .= 'alert(\' . $mensagem . '\');';
        self::$alertScript .= '</script>';
    }
    echo(self::$alertScript);
}
}
?>

```

Código 29 - destroySession.inc.php

<?php

/*

```

* Utilizada para excluir varias sessões de uma vez
*/
session_unregister('listaOrigemTO');
session_unregister('listaResultado');
session_unregister('parametrosArray');
session_unregister('nuLinha');

?>

```

Código 30 - batchAction.php

```

<?php
    ob_start();
    require_once($_SERVER['DOCUMENT_ROOT'] .
'/projeto/final/app/PF/classes/util/constantes.inc.php');
    require_once(BUSINESS_PATH_PF . '/facade/FacadeFactory.class.php');
    require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');

    session_start();

    $acao = NULL;
    if($_SERVER['REQUEST_METHOD'] == 'GET') {
        $acao = $_GET['acao'];
    } else if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $acao = $_POST['acao'];
    }

    if(isset($acao) && $acao == 'processar') {
        BatchAction::processar();
    }
    else if(isset($acao) && $acao == 'montarBatch') {
        BatchAction::montarBatch();
    }
    else if(isset($acao) && $acao == 'consultarOnibus') {
        BatchAction::consultarOnibus();
    }

    /**
     * Classe Action do caso de uso 'Processamento Batch'.
     * @author Tiago Velloso Silva 16.11.2009
     * @version 1.0
     */
    final class BatchAction {

        private static $propertiesUtil;

```

```

public static function montarBatch() {
    try {
        include(VIEW_INC_PATH . '/destroySession.inc.php');
        //Recupera a instância da classe Facade através da Factory.
        $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $listaLinhaTO = $parametrosFacade->listarLinhas();

        //Poe a lista de TO na sessão
        $_SESSION['listaLinhaTO'] = $listaLinhaTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/batch/batch.php');
}

/**
 * Realiza o processamento.
 */
public static function processar() {
    try {
        //1 dia = 86400 segundos
        ini_set("max_execution_time", 86400);
        $codOnibus = $_POST['codOnibus'];
        //Quantidade maxima que o receptor GPS pode armazenar os dados
        for($i=1; $i<200000; $i++){
            //Tempo em microsegundos para inciar o processo
            $tempo = microtime(true);

            //parse pro arquivo .KML
            $kml = simplexml_load_file(KML_PATH . '/me.kml');

            //Ler apenas a parte de coordenadas
            $str_coordinates = $kml[0]->Document->Folder->Folder-
>Placemark->LineString->coordinates[0];

            //jogar as coordenadas em um array usando REGEX
            $coordinates = preg_split('#\n#', $str_coordinates,
PREG_SPLIT_NO_EMPTY);

            //jogar as coordenadas em uma string

```

```

$string = implode("#", $coordinates);

//jogar cada linha em um elemento no array e contar as quantidades
$posicao = explode("\n", $string);

//elemento que será gravado no banco de dados
$array = explode(",", $posicao[1]);

//Recupera a instância da classe Facade através da Factory.
$batchFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'BatchFacade');
$batchFacade->processar($codOnibus, $array);

sleep(10);
}

} catch(SystemException $se) {
    $_SESSION['mensagem'] = $se->getMensagem();
    self::goToPage(ERROR_PAGE);
} catch(BusinessException $be) {
    $_SESSION['mensagem'] = $be->getMensagem();
    self::goToPage(ERROR_PAGE);
}
self::goToPage('/projetoFinal/app/PF/SAUO/page/batch/batch.php');
}

public static function consultarOnibus() {
    try {
        $nuLinha = $_POST['linhaOnibus'];
        $_SESSION['nuLinha'] = $nuLinha;
        $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $listaOnibusTO = $parametrosFacade->consultarOnibus($nuLinha);
        //Poe a lista de TO na sessão
        $_SESSION['listaOnibusTO'] = $listaOnibusTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/batch/batch.php');
}
}

```

```

/**
 * Redireciona o fluxo da aplicação para outra página.
 * @param $url URL de destino.
 */
private static function goToPage($url) {
    header('Location: ' . $url);
    exit();
}
/**
 * Carrega o arquivo de properties.
 */
private static function loadPropertiesFile() {
    self::$propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
}
}
?>

```

Código 31 - mapaAction.php

```

<?php
    ob_start();
    require_once($_SERVER['DOCUMENT_ROOT']
'/projetoFinal/app/PF/classes/util/constantes.inc.php');
    require_once(BUSINESS_PATH_PF
        . '/facade/FacadeFactory.class.php');
    require_once(UTIL_PATH_PF
        . '/PropertiesUtil.class.php');
    session_start();
    $acao = NULL;
    if($_SERVER['REQUEST_METHOD'] == 'GET') {
        $acao = $_GET['acao'];
    } else if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $acao = $_POST['acao'];
    }

    if(isset($acao) && $acao == 'visuMapa') {
        MapaAction::visuMapa();
    }

/**
 * Classe Action paa Visualizar o mapa.
 * @author Tiago Velloso Silva 17.11.2009
 * @version 1.0
 */
final class MapaAction {

```

```

private static $propertiesUtil;

public static function visuMapa() {
    try {
        session_unregister('olat');
        session_unregister('olon');
        session_unregister('dlat');
        session_unregister('dlon');
        $_SESSION['olat']      = $_GET['olat'];
        $_SESSION['olon']      = $_GET['olon'];
        $_SESSION['dlat']      = $_GET['dlat'];
        $_SESSION['dlon']      = $_GET['dlon'];

    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/mapa/mapa.php');
}

/**
 * Redireciona o fluxo da aplicação para outra página.
 * @param $url URL de destino.
 */
private static function goToPage($url) {
    header('Location: ' . $url);
    exit();
}

/**
 * Carrega o arquivo de properties.
 */
private static function loadPropertiesFile() {
    self::$propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
}
}
?>

```

Código 32 - pesqMapaAction.php

```

<?php
    ob_start();

```

```

require_once($_SERVER['DOCUMENT_ROOT'] .
'/projetofinal/app/PF/classes/util/constantes.inc.php');
require_once(BUSINESS_PATH_PF . '/facade/FacadeFactory.class.php');
require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');
require_once(TO_PATH . '/ParametrosTO.class.php');
require_once(TO_PATH . '/ParametrosMapaTO.class.php');
require_once(TO_PATH . '/PontoTO.class.php');
session_start();
$acao = NULL;
if($_SERVER['REQUEST_METHOD'] == 'GET') {
    $acao = $_GET['acao'];

} else if($_SERVER['REQUEST_METHOD'] == 'POST') {
    $acao = $_POST['acao'];
}

if(isset($acao) && $acao == 'montarPesqMapa') {
    PesqMapaAction::montarPesqMapa();
}
else if(isset($acao) && $acao == 'pesquisar') {
    PesqMapaAction::pesquisar();
}

/**
 * Classe Action do caso de uso 'Pesquisar por Texto'.
 * @author Tiago Velloso Silva 19.08.2009
 * @version 1.0
 */
final class PesqMapaAction {

    private static $propertiesUtil;

    /**
     * Monta a página de Pesquisa por Mapa.
     */
    public static function montarPesqMapa() {
        include(VIEW_INC_PATH . '/destroySession.inc.php');
        session_unregister('parametrosTO');
        try {
            $parametrosMapaTO = new ParametrosMapaTO();
            $parametrosMapaTO->setOrigemLat    ();
            $parametrosMapaTO->setOrigemLon    ();
            $parametrosMapaTO->setDestinoLat();
            $parametrosMapaTO->setDestinoLon();
            $parametrosFacade =

```

```

FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $parametrosMapaTO = $parametrosFacade-
>consultarPontos($parametrosMapaTO);
        $listaPontoTO = $parametrosMapaTO->getListaPontoTO();

        //Transformar Array em String para ser lido no JavaScript
        while(list($key, $to) = each($listaPontoTO)) {
            $latIng = $latIng."|".$to->getLatitude().",".$to->getLongitude();
        }
        $_SESSION['latIng'] = $latIng;
        $_SESSION['parametrosMapaTO'] = $parametrosMapaTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqMapa/pesqMapa.php');
}
/**
 * Retorna com o resultado da Pesquisa por Mapa.
 */
public static function pesquisar() {
    try {
        $listaOrigemTO = $_SESSION['listaOrigemTO'];
        $listaPontoTO = $_SESSION['parametrosMapaTO']->getListaPontoTO
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;
        $parametrosMapaTO = new ParametrosMapaTO();
        $parametrosMapaTO->setOrigemLat ($ _GET['origemLat']);
        $parametrosMapaTO->setDestinoLat($ _GET['destinoLat']);
        $_SESSION['parametrosMapaTO'] = $parametrosMapaTO;
        $parametrosTO = new ParametrosTO();

//Separar a Latitude e Longitude recebidas do Javascript
        $latIng = explode(",", $ _GET['origemLat']);
        while(list($key, $to) = each($listaPontoTO)) {
            if(str_replace(' ', "", $latIng[0]) == $to->getLatitude()){
                $parametrosTO->setLocalOrigem ($to->getCodAvenida());
                $parametrosTO->setPontoOrigem ($to->getCodPonto());
                $parametrosTO->setTpSentidoOrigem($to->getTpSentido());
            }
        }
        reset($listaPontoTO);
        $latIng = explode(",", $ _GET['destinoLat']);

```

```

        while(list($key, $to) = each($listaPontoTO)) {
            if(str_replace(',', '', $latlng[0]) == $to->getLatitude()){
                $parametrosTO->setLocalDestino ($to->getCodAvenida());
                $parametrosTO->setPontoDestino ($to->getCodPonto());
                $parametrosTO->setTpSentidoDestino($to->getTpSentido());
            }
        }
        if($parametrosTO->getTpSentidoOrigem() != $parametrosTO-
>getTpSentidoDestino()){
            $_SESSION['mensagem'] = 'Não existem ônibus que realizam esse trejeito.';

            self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqMapa/pesqMapa.php');
        }
        //Recupera a instância da classe Facade através da Factory.
        $resultPesqFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ResultPesqFacade');
        $listaResultado = $resultPesqFacade->pesquisar($parametrosTO, 1);
        $_SESSION['listaResultado'] = $listaResultado;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqMapa/pesqMapa.php');
}

/**
 * Redireciona o fluxo da aplicação para outra página.
 * @param $url URL de destino.
 */
private static function goToPage($url) {
    header('Location: ' . $url);
    exit();
}

/**
 * Carrega o arquivo de properties.
 */
private static function loadPropertiesFile() {
    self::$propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
}
}

?>

```

Código 33 - pesqTxtAction.php

```

<?php
    ob_start();
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
    require_once(BUSINESS_PATH_PF . '/facade/FacadeFactory.class.php');
    require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');
    require_once(TO_PATH . '/ParametrosTO.class.php');
    require_once(TO_PATH . '/BairroTO.class.php');
    require_once(TO_PATH . '/AvenidaTO.class.php');
    require_once(TO_PATH . '/PontoTO.class.php');
    session_start();

    $sacao = NULL;
    if($_SERVER['REQUEST_METHOD'] == 'GET') {
        $sacao = $_GET['acao'];
    } else if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $sacao = $_POST['acao'];
    }

    if(isset($sacao) && $sacao == 'montarPesqTxt') {
        PesqTxtAction::montarPesqTxt();
    } else if(isset($sacao) && $sacao == 'pesquisar') {
        PesqTxtAction::pesquisar();
    } else if(isset($sacao) && $sacao == 'consultarAvenida') {
        PesqTxtAction::consultarAvenida();
    } else if(isset($sacao) && $sacao == 'consultarAvenidaDest') {
        PesqTxtAction::consultarAvenidaDest();
    } else if(isset($sacao) && $sacao == 'consultarPontoOrig') {
        PesqTxtAction::consultarPontoOrig();
    } else if(isset($sacao) && $sacao == 'consultarPontoDest') {
        PesqTxtAction::consultarPontoDest();
    }

    /**
     * Classe Action do caso de uso 'Pesquisar por Texto'.
     * @author Tiago Velloso Silva 19.08.2009
     * @version 1.0
     */
    final class PesqTxtAction {

        private static $propertiesUtil;

```

```

/**
 * Monta a página de Pesquisa por Texto.
 */
public static function montarPesqTxt() {
    include(VIEW_INC_PATH . '/destroySession.inc.php');
    session_unregister('parametrosTO');
    try {
        $parametrosTO = new ParametrosTO();
        $parametrosTO->setBairroOrigem ('-1');
        $parametrosTO->setLocalOrigem ('-1');
        $parametrosTO->setPontoOrigem ('-1');
        $parametrosTO->setBairroDestino ('-1');
        $parametrosTO->setLocalDestino ('-1');
        $parametrosTO->setPontoDestino ('-1');
        $_SESSION['parametrosTO'] = $parametrosTO;
        //Recupera a instância da classe Facade através da Factory.
        $pesqTxtFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'PesqTxtFacade');
        $listaOrigemTO = $pesqTxtFacade->listarOrigem();

        //Poe a lista de TO na sessão
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
}

/**
 * Realiza a Pesquisa de acordo com os parametros informados.
 */
public static function pesquisar() {
    try {
        $listaOrigemTO = $_SESSION['listaOrigemTO'];
        include(VIEW_INC_PATH . '/destroySession.inc.php');
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;
        $parametrosTO = new ParametrosTO();
        $parametrosTO->setBairroOrigem ($_POST['bairroOrigem']);
        $parametrosTO->setLocalOrigem ($_POST['localOrigem']);
        $parametrosTO->setPontoOrigem ($_POST['pontoOrigem']);
        $parametrosTO->setBairroDestino ($_POST['bairroDestino']);
    }
}

```

```

        $parametrosTO->setLocalDestino ($_POST['localDestino']);
        $parametrosTO->setPontoDestino ($_POST['pontoDestino']);

        $resultParametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $parametrosTO = $resultParametrosFacade-
>consultarSentido($parametrosTO);
        if($parametrosTO->getTpSentidoOrigem() != $parametrosTO-
>getTpSentidoDestino()){
            $_SESSION['mensagem'] = 'Não existem ônibus que realizam esse
trejeto.';

            self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
        }
        $_SESSION['parametrosTO'] = $parametrosTO;
        $resultPesqFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ResultPesqFacade');
        $listaResultado = $resultPesqFacade->pesquisar($parametrosTO, 1);
        $_SESSION['listaResultado'] = $listaResultado;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
}

/**
 * Realiza a consulta das avenidas da origem.
 */
public static function consultarAvenida() {
    try {
        $listaOrigemTO = $_SESSION['listaOrigemTO'];
        include(VIEW_INC_PATH . '/destroySession.inc.php');
        $parametrosTO = new ParametrosTO();
        $parametrosTO->setBairroOrigem ($_POST['bairroOrigem']);
        $parametrosTO->setLocalOrigem ('-1');
        $parametrosTO->setPontoOrigem ('-1');
        $parametrosTO->setBairroDestino ($_POST['bairroDestino']);
        $parametrosTO->setLocalDestino ($_POST['localDestino']);
        $parametrosTO->setPontoDestino ($_POST['pontoDestino']);
        $_SESSION['parametrosTO'] = $parametrosTO;
        $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $listaAvenidaTO = $parametrosFacade->consultarAvenida($parametrosTO);
    }
}

```

```

        $listaOrigemTO[1] = $listaAvenidaTO;
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
}

/**
 * Realiza a consulta das avenidas dao destino.
 */
public static function consultarAvenidaDest() {
    try {
        $listaOrigemTO = $_SESSION['listaOrigemTO'];
        include(VIEW_INC_PATH . '/destroySession.inc.php');
        $parametrosTO = new ParametrosTO();
        $parametrosTO->setCodLinha ($ _POST['codLinha']);
        $parametrosTO->setBairroOrigem ($ _POST['bairroOrigem']);
        $parametrosTO->setLocalOrigem ($ _POST['localOrigem']);
        $parametrosTO->setPontoOrigem ($ _POST['pontoOrigem']);
        $parametrosTO->setBairroDestino ($ _POST['bairroDestino!']);
        $parametrosTO->setLocalDestino ('-1');
        $parametrosTO->setPontoDestino ('-1');
        $_SESSION['parametrosTO'] = $parametrosTO;
        $temp = $parametrosTO->getBairroOrigem();
        $parametrosTO->setBairroOrigem($parametrosTO->getBairroDestino());
        $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $listaAvenidaTO = $parametrosFacade->consultarAvenida($parametrosTO);
        $listaOrigemTO[2] = $listaAvenidaTO;
        $parametrosTO->setBairroOrigem($temp);
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;

    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
}

```

```

    }

    /**
     * Realiza a consulta dos Pontos da origem.
     */
    public static function consultarPontoOrig() {
        try {
            $listaOrigemTO = $_SESSION['listaOrigemTO'];
            include(VIEW_INC_PATH . '/destroySession.inc.php');
            $parametrosTO = new ParametrosTO();
            $parametrosTO->setCodLinha      ($_POST['codLinha']);
            $parametrosTO->setBairroOrigem  ($_POST['bairroOrigem']);
            $parametrosTO->setLocalOrigem   ($_POST['localOrigem']);
            $parametrosTO->setPontoOrigem   ('-1');
            $parametrosTO->setBairroDestino ($_POST['bairroDestino']);
            $parametrosTO->setLocalDestino  ($_POST['localDestino']);
            $parametrosTO->setPontoDestino  ($_POST['pontoDestino']);
            $_SESSION['parametrosTO']      = $parametrosTO;
            $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
            $listaPontoTO = $parametrosFacade->consultarPonto($parametrosTO);
            $listaOrigemTO[3] = $listaPontoTO;
            $_SESSION['listaOrigemTO'] = $listaOrigemTO;

        } catch(SystemException $se) {
            $_SESSION['mensagem'] = $se->getMensagem();
            self::goToPage(ERROR_PAGE);
        } catch(BusinessException $be) {
            $_SESSION['mensagem'] = $be->getMensagem();
            self::goToPage(ERROR_PAGE);
        }
        self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
    }

    /**
     * Realiza a consulta dos Pontos do destino.
     */
    public static function consultarPontoDest() {
        try {
            $listaOrigemTO = $_SESSION['listaOrigemTO'];
            include(VIEW_INC_PATH . '/destroySession.inc.php');
            $parametrosTO = new ParametrosTO();
            $parametrosTO->setCodLinha      ($_POST['codLinha']);
            $parametrosTO->setBairroOrigem  ($_POST['bairroOrigem']);
            $parametrosTO->setLocalOrigem   ($_POST['localOrigem']);

```

```

        $parametrosTO->setPontoOrigem ($_POST['pontoOrigem']);
        $parametrosTO->setBairroDestino ($_POST['bairroDestino']);
        $parametrosTO->setLocalDestino ($_POST['localDestino']);
        $parametrosTO->setPontoDestino ('-1');
        $_SESSION['parametrosTO'] = $parametrosTO;
        $temp = $parametrosTO->getLocalOrigem();
        $parametrosTO->setLocalOrigem($parametrosTO->getLocalDestino());

        $parametrosFacade =
FacadeFactory::getInstanceClass(FacadeFactory::$SAUO, 'ParametrosFacade');
        $listaPontoTO = $parametrosFacade->consultarPonto($parametrosTO);
        $listaOrigemTO[4] = $listaPontoTO;

        $parametrosTO->setLocalOrigem($temp);
        $_SESSION['listaOrigemTO'] = $listaOrigemTO;
    } catch(SystemException $se) {
        $_SESSION['mensagem'] = $se->getMensagem();
        self::goToPage(ERROR_PAGE);
    } catch(BusinessException $be) {
        $_SESSION['mensagem'] = $be->getMensagem();
        self::goToPage(ERROR_PAGE);
    }
    self::goToPage('/projetoFinal/app/PF/SAUO/page/pesqTxt/pesqTxt.php');
}

/**
 * Redireciona o fluxo da aplicação para outra página.
 * @param $url URL de destino.
 */
private static function goToPage($url) {
    header('Location: ' . $url);
    exit();
}

/**
 * Carrega o arquivo de properties.
 */
private static function loadPropertiesFile() {
    self::$propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
}
}

?>

```

Código 34 - batch.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantes.inc.php');
    require_once(UTIL_PATH_PF . '/HtmlUtil.class.php');
    include(INC_PATH . '/head.inc.php');
    require_once(TO_PATH . '/LinhaTO.class.php');
    require_once(TO_PATH . '/OnibusTO.class.php');
    @session_start();
    $listaLinhaTO = $_SESSION['listaLinhaTO'];
    $listaOnibusTO = $_SESSION['listaOnibusTO'];
    $nuLinha = $_SESSION['nuLinha'];
?>
<html>
<head>
    <script language="javascript">
        function processar() {
            document.batch.action =
"/projeto-final/app/PF/SAUO/classes/view/batchAction.php";
            document.batch.acao.value = "processar";
            document.batch.submit();
        }
        function parar() {
            document.batch.action =
"/projeto-final/app/PF/SAUO/classes/view/batchAction.php";
            document.batch.acao.value = "parar";
            document.batch.submit();
        }
        function consultarOnibus() {
            document.batch.action =
"/projeto-final/app/PF/SAUO/classes/view/batchAction.php";
            document.batch.acao.value = "consultarOnibus";
            document.batch.submit();
        }
    </script>
</head>
<body>
<form name="batch" method="post">
<input type="hidden" name="acao"/>
<div id="container">
<?php
    include(INC_PATH . '/cabecalho.inc.php');
?>
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td colspan="2" class="rodape">
<fieldset>

```

```

<legend>Processamento Batch</legend>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
  <tr> <td colspan="2"><br></td></tr>
  <tr> <td colspan="2" class="td1"><b>Linha e Código do ônibus:</b></td></tr>
  <tr> <td width="15%">Linha*:<br> <?php HtmlUtil::gerarCombobox('linhaOnibus',
'LinhaTO', 'nuLinha', 'descLinha', $listaLinhaTO, $nuLinha, 'onChange=consultarOnibus()'); ?></td></tr>
  <tr><td width="50%"><br>Código*:<br><?php HtmlUtil::gerarCombobox('codOnibus',
'LinhaTO', 'codOnibus', 'horario', $listaOnibusTO, 'NULL'); ?><br><br></td></tr>
<td colspan="2" class="td1"><b>*Campos Obrigatórios.</b></td></tr>
</table>
<br><br>
<div align="left">
  <input type="button" name="Processar" value="Processar" class="botao"
onclick="processar();"/>
  <input type="button" name="Parar" value="Parar" class="botao" onclick="parar();"/>
</div>
</fieldset>
<br>
</td>
</tr>
</table>
<?php
  include(INC_PATH . '/rodape.inc.php');
?>
</div>
</form>
</body>
</html>

```

Código 35 - cabecalho.inc.php

```

<table width="760" border="0" cellpadding="0" cellspacing="0">
  <tr>
    <td width="123"></td>
    <td width="292" valign="top">
      <br />
      <div style="padding-top: 2px; margin-left: 3px;">
        <a href="/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php?acao=montarPesqTxt">###
Pesquisar via Texto ###</a><br>
        <a href="/projetoFinal/app/PF/SAUO/classes/view/pesqMapaAction.php?acao=montarPesqMapa">###
Pesquisar via Mapa ###</a>
      </div>
    </td>
  </tr>
</table>

```

Código 36 - head.inc.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
    <title>SAUO - Projeto Final</title>
    <script type="text/javascript"
src="/projetofinal/app/PF/SAUO/web/js/intratec_ploneScripts4861.js"></script>
    <link href="/projetofinal/app/PF/SAUO/web/css/estilo.css" rel="stylesheet" type="text/css" />
    <style type="text/css" media="all">@import url(/projetofinal/app/PF/SAUO/web/css/estilo.css);</style>

```

Código 37 - rodape.inc.php

```

<table width="760" border="0" cellpadding="0" cellspacing="0" bgcolor="#EDF3F0">
    <tr><td colspan="3"><div align="center" class="poweredBy">SAUO - Sistema de Auxílio à Usuários de
Ônibus - Copyright 2009 - Todos os direitos reservados</div></td></tr>
</table>

```

Código 38 - mapa.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetofinal/app/PF/classes/util/constantes.inc.php');
    require_once(UTIL_PATH_PF . '/HtmlUtil.class.php');
    require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');

    @session_start();

    $olat = $_SESSION['olat'] ;
    $olon = $_SESSION['olon'];
    $dlat = $_SESSION['dlat'] ;
    $dlon = $_SESSION['dlon'];

?>
<html>
<head>
    <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAzE16s71_pxo34DuUrx_JGhSLWnp9S-
9KMjylpQfCwmMVH5WUehTRiFxrNaJ9G1XvqfcCkJt86BRI2A" type="text/javascript"></script>
    <script type="text/javascript">
        function initialize() {
            if (GBrowserIsCompatible()) {
                var map = new GMap2(document.getElementById("map_canvas"));
                map.setCenter(new GLatLng(<?php echo "$olat,$olon" ?>), 14);
                map.setUIToDefault();
                var blueIcon = new GIcon(G_DEFAULT_ICON);
                blueIcon.image = "http://gmaps-samples.googlecode.com/svn/trunk/markers/blue/blank.png";
                var onibus = new GIcon(G_DEFAULT_ICON);
                onibus.image = "../web/img/onibus.gif";
                markerOptionsPonto = { icon:blueIcon };
                markerOptionsOnibus = { icon:onibus };
                var latlng = new GLatLng(<?php echo "$olat,$olon" ?>);
            }
        }
    </script>

```

```

        map.addOverlay(new GMarker(latlng, markerOptionsOnibus));
        var latlng2 = new GLatLng(<?php echo"$dlat,$dlon" ?>);
        map.addOverlay(new GMarker(latlng2, markerOptionsPonto));

    }
}

</script>
</head>
<body onload="initialize()" onunload="GUnload()">
<div id="container">
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td colspan="2" class="rodape"></td>
<div id="map_canvas" style="width: 800px; height: 600px"></div>
</tr>
</table>
</div></body></html>

```

Código 39 - pesqMapa.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projetoFinal/app/PF/classes/util/constantes.inc.php');
    require_once(UTIL_PATH_PF . '/HtmlUtil.class.php');
    require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');
    require_once(TO_PATH . '/ParametrosMapaTO.class.php');
    require_once(TO_PATH . '/ResultadoTO.class.php');
    require_once(TO_PATH . '/LinhaTO.class.php');
    @session_start();
    $listaResultado = $_SESSION['listaResultado'];
    $parametrosMapaTO = $_SESSION['parametrosMapaTO'];
    $latlng = $_SESSION['latlng'];
    $div = "none";
    if(session_is_registered('listaResultado')) {
        $div = "block";
    }
    if(session_is_registered('mensagem')) {
        $mensagem = $_SESSION['mensagem'];
        HtmlUtil::alert($mensagem);
    }
    $propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
    include(INC_PATH . '/head.inc.php');
?>
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:v="urn:schemas-microsoft-com:vml">
<head>
<script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAE16s71_pxo34DuUrx_JGhSUEj4HioMNg0uG

```

```

W8rfEtYcEny5QRTILQkXa4RNrIT87d66VutFZCPkDw" type="text/javascript"></script>
<script type="text/javascript">
    var map;
    function createPolyMarker(latlng, addListener, makePolyline, isClickable) {
        var polyLatLngs = getPolyLatLngs(latlng);
        if (makePolyline) {
            var poly = new GPolyline(polyLatLngs, '#0000ff', 8, .7, {clickable:
isClickable});
        } else {
            var poly = new GPolygon(polyLatLngs, '#0000ff', 1, 1.0, '#ff0000', .5,
{clickable: isClickable});
        }
        var marker = new GMarker(latlng, {clickable: isClickable});
        if (addListener) {
            GEvent.addListener(marker, "click", function(overlaylatlng) {
                if(document.posicao.origemLat.value == "" ||
document.posicao.z.value=="2"){
                    marker.setImage('../web/img/origem.gif');
                }
                if(document.posicao.z.value=="1"){
                    marker.setImage('../web/img/destino.gif');
                }
            });
        }
        map.addOverlay(marker);
    }

    function getPolyLatLngs(latlng) {
        var mapNormalProj = G_NORMAL_MAP.getProjection();
        var mapZoom = map.getZoom();
        var centerPixel = mapNormalProj.fromLatLngToPixel(latlng, mapZoom);
        var polyNumSides = 6;
        var polySideLength = 6;
        var polyRadius = 4;
        var polyLatLngs = [];
        for (var a = 0; a < (polyNumSides+1); a++) {
            var aRad = polySideLength*a*(Math.PI/180);
            var pixelX = centerPixel.x + polyRadius * Math.cos(aRad);
            var pixelY = centerPixel.y + polyRadius * Math.sin(aRad);
            var polyPixel = new GPoint(pixelX,pixelY);
            var polyLatLng =
mapNormalProj.fromPixelToLatLng(polyPixel,mapZoom);
            polyLatLngs.push(polyLatLng);
        }
        return polyLatLngs;
    }
    function load() {

```

```

if (GBrowserIsCompatible()) {
    map = new GMap2(document.getElementById("map"));

    map.setCenter(new GLatLng(-15.7570408,-47.8929799),15);

    map.addControl(new GLargeMapControl());

    map.setMapType(G_NORMAL_MAP);

    var glatLng, i, array_pontos, string_array;

    string_array = "<?php echo $latlng; ?>";

    array_pontos = string_array.split("|");

    for (i in array_pontos){
        if(i !=0 ){
            array = array_pontos[i].split(",");

            glatLng = new GLatLng(array[0],array[1]);

            createPolyMarker(glatLng, true, false, true);

        }
    }

    GEvent.addListener(map, "click", function(overlay, latlng,
overlaylatlng) {

        if (overlaylatlng == undefined){

            alert("Você deve selecionar um ponto de ônibus");

            return false;

        }

        if(document.posicao.origemLat.value == "" ||

document.posicao.z.value=="2"){

            document.posicao.origemLat.value=overlaylatlng

            document.posicao.z.value="1"

        }

        else if(document.posicao.z.value=="1"){

            document.posicao.destinoLat.value=overlaylatlng

            document.posicao.z.value="2"

        }

    });
}

```

```

    }
    function pesquisar() {
        var z = document.posicao.z;
        if(z.value == "") {
            alert('Selecione no mapa o Ponto de Origem e o Ponto de Destino.');
```

return false;

```
        }
        if(z.value == "1") {
            alert('Selecione no mapa o Ponto de Destino.');
```

return false;

```
        }

        document.posicao.action =
"/projeto/final/app/PF/SAUO/classes/view/pesqMapaAction.php";
        document.posicao.acao.value = "pesquisar";

        document.posicao.submit();

    }
    window.onload=load
</script>
</head>
<body onload="load()">
<div id="container">
<?php
    include(INC_PATH . '/cabecalho.inc.php');
?>
<table width="760" border="0" cellspacing="0" cellpadding="0">
<tr>
<td colspan="2" class="rodape">
    <fieldset>
        <legend>Pesquisar via Mapa</legend>
        <div id="map" style="width: 710px; height: 600px"></div>
        <div id="formulario" style="margin: 10px">
            <form action="#" id="posicao" name="posicao">
                <input type="hidden" name="acao"/>
                <input type="hidden" name="z" value="" />
                <br><br>
                Origem Latitude e Longitude: <input type="text" name="origemLat"
size="50" value="" class="texto" readonly="true" />
                <br>
                Destino Latitude e Longitude: <input type="text" name="destinoLat"
size="50" value="" class="texto" readonly="true" />
                <br>
                <input type="button" name="Pesquisar" value="Pesquisar"
class="botao" onclick="pesquisar();"/>
            </form>
        </div>
    </fieldset>
</td>
</tr>
</table>

```

```

        </fieldset>
        <div id="resultado" style="display:<?php print_r( $div ); ?>">
            <fieldset>
                <legend>Resultado</legend>
                <table class="zebrado" id="zebrado">
                    <thead>
                        <tr>
                            <th width="26%" align="left">Linha</th>
                            <th width="38%" align="left">Origem e Destino da Linha</th>
                            <th width="13%">Distância</th>
                            <th width="13%">Tempo</th>
                            <th width="10%">
                                </th>
                        </tr>
                    </thead>
                    <div id="map_canvas" style="display:none;" ></div>
                    <script language="JavaScript" src="../../web/js/distTempo.js"></script>
                    <?php
                        $bgcolor = '#EDF3F6';
                        while(list($key, $to) = each($listaResultado)) {
                            if ($key % 2 == 0) {
                                $bgcolor = "#FFFFFF";
                            } else {
                                $bgcolor = "#EDF3F6";
                            }
                            $latOnibus = $to->getLatitudeOrigem();
                            $lonOnibus = $to->getLongitudeOrigem();
                            $latPonto = $to->getLatitudeDestino();
                            $lonPonto = $to->getLongitudeDestino();
                            echo"<script>teste($key, '$latOnibus,$lonOnibus',
'<script>teste($key, '$latOnibus,$lonPonto');</script>";
                            ?>
                            <tr bgcolor="<?php print_r( $bgcolor ); ?>" class="odd">
                                <td align="left"><?php print_r($to->getLinhaTO()->getNuLinha()); ?>
                                - <?php print_r($to->getLinhaTO()->getDescLinha()); ?></td>
                                <td align="left"><?php print_r($to->getLinhaTO()-
>getDescOrigemLinha());?><br>
                                <?php print_r($to-
>getLinhaTO()->getDescDestinoLinha());?></td>
                                <td align="center"><div id="getDistance"<?php print_r( $key );
?>></div></td>
                                <td align="center"><div id="getDuration"<?php print_r( $key );
?>></div></td>
                                <td align="center"><a
href="/projetoFinal/app/PF/SAUO/classes/view/mapaAction.php?acao=visuMapa&

```

```

                                olat=<?php print_r($latOnibus); ?>&

                                olon=<?php print_r($lonOnibus); ?>&

                                dlat=<?php print_r($latPonto); ?>&

                                dlon=<?php print_r($lonPonto); ?>" target="_blank">Ver no
mapa</a></td>

                                </tr>
                                <?php
                                        }
                                ?>

                                </table>
                                </fieldset>
                                </div>

                                </td>
                                </tr>
                                </table>
                                <?php
                                        include(INC_PATH . '/rodape.inc.php');
                                ?>
                                </div>
                                </body>
                                </html>
                                <?php
                                        session_unregister('mensagem');
                                ?>

```

Código 40 - pesqTxt.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<?php
    require_once($_SERVER['DOCUMENT_ROOT'] . '/projeto-final/app/PF/classes/util/constantes.inc.php');
    require_once(UTIL_PATH_PF . '/HtmlUtil.class.php');
    require_once(UTIL_PATH_PF . '/PropertiesUtil.class.php');
    require_once(TO_PATH . '/OrigemTO.class.php');
    require_once(TO_PATH . '/ParametrosTO.class.php');
    require_once(TO_PATH . '/BairroTO.class.php');
    require_once(TO_PATH . '/AvenidaTO.class.php');
    require_once(TO_PATH . '/PontoTO.class.php');
    require_once(TO_PATH . '/ResultadoTO.class.php');
    require_once(TO_PATH . '/LinhaTO.class.php');
    @session_start();
    $listaOrigemTO = $_SESSION['listaOrigemTO'];
    $listaResultado = $_SESSION['listaResultado'];

```

```

$parametrosTO = $_SESSION['parametrosTO'];
$div = "none";
if(session_is_registered('listaResultado')) {
    $div = "block";
}
if(session_is_registered('mensagem')) {
    $mensagem = $_SESSION['mensagem'];
    HtmlUtil::alert($mensagem);
}
$propertiesUtil = new PropertiesUtil(RESOURCE_PATH, 'resources.properties');
include(INC_PATH . '/head.inc.php');
?>

<html>
<head>
    <script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAAzE16s71_pxo34DuUrx_JGhTncRII33HKXuOY
EzufEooCg2bDRQ0l87QhGhJ7WI-jbK8QQ4HJ8fUQw" type="text/javascript"></script>
    <script language="javascript">
        function pesquisar() {
            var bairroOrigem = document.formPesq.bairroOrigem;
            var localOrigem = document.formPesq.localOrigem;
            var pontoOrigem = document.formPesq.pontoOrigem;
            var bairroDestino = document.formPesq.bairroDestino;
            var localDestino = document.formPesq.localDestino;
            var pontoDestino = document.formPesq.pontoDestino;
            if(bairroOrigem.value == "-1") {
                alert('Campos de preenchimento obrigatório. ');
                bairroOrigem.focus();
                return false;
            }
            if(localOrigem.value == "-1") {
                alert('Campos de preenchimento obrigatório. ');
                localOrigem.focus();
                return false;
            }
            if(pontoOrigem.value == "-1") {
                alert('Campos de preenchimento obrigatório. ');
                pontoOrigem.focus();
                return false;
            }
            if(bairroDestino.value == "-1") {
                alert('Campos de preenchimento obrigatório. ');
                bairroDestino.focus();
                return false;
            }
        }
    </script>

```

```

        if(localDestino.value == "-1") {
            alert('Campos de preenchimento obrigatório. ');
            localDestino.focus();
            return false;
        }
        if(pontoDestino.value == "-1") {
            alert('Campos de preenchimento obrigatório. ');
            pontoDestino.focus();
            return false;
        }

        document.formPesq.action =
"/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php";
        document.formPesq.acao.value = "pesquisar";
        document.formPesq.submit();
    }
    function consultarAvenida() {
        document.formPesq.action =
"/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php";
        document.formPesq.acao.value = "consultarAvenida";
        document.formPesq.submit();
    }
    function consultarAvenidaDest() {
        document.formPesq.action =
"/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php";
        document.formPesq.acao.value = "consultarAvenidaDest";
        document.formPesq.submit();
    }
    function consultarPontoOrig() {
        document.formPesq.action =
"/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php";
        document.formPesq.acao.value = "consultarPontoOrig";
        document.formPesq.submit();
    }
    function consultarPontoDest() {
        document.formPesq.action =
"/projetoFinal/app/PF/SAUO/classes/view/pesqTxtAction.php";
        document.formPesq.acao.value = "consultarPontoDest";
        document.formPesq.submit();
    }
}
</script>
</head>
<body>
<form name="formPesq" method="post">
<input type="hidden" name="acao"/>
<div id="container">

```

```

<?php
    include(INC_PATH . '/cabecalho.inc.php');
?>
<table width="760" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td colspan="2" class="rodape">
            <fieldset>
                <legend>Pesquisar via Texto</legend>
                <table width="100%" border="0" cellspacing="0" cellpadding="0">
                    <tr> <td colspan="3"><br></td></tr>
                    <tr> <td colspan="3" class="td1"><b>Ponto de Origem:</b></td></tr>
                    <tr> <td width="15%">Bairro*:<br> <?php HtmlUtil::gerarCombobox('bairroOrigem',
'BairroTO', 'codBairro', 'descBairro', $listaOrigemTO[0], $parametrosTO->getBairroOrigem(),
'onChange=consultarAvenida()); ?></td>
                        <td width="50%">Avenida*:<br><?php
HtmlUtil::gerarCombobox('localOrigem', 'AvenidaTO', 'codAvenida', 'descAvenida', $listaOrigemTO[1],
$parametrosTO->getLocalOrigem(), 'onChange=consultarPontoOrig()); ?></td>
                        <td width="35%">Ponto*:<br> <?php
HtmlUtil::gerarCombobox('pontoOrigem', 'pontoTO', 'codPonto', 'descPonto', $listaOrigemTO[3], $parametrosTO-
>getPontoOrigem()); ?></td>
                    </tr><td colspan="3"><br></td></tr>
                    <tr> <td colspan="3" class="td1"><b>Ponto de Destino:</b></td></tr>
                    <tr> <td width="15%">Bairro*:<br> <?php HtmlUtil::gerarCombobox('bairroDestino',
'OrigemTO', 'codBairro', 'descBairro', $listaOrigemTO[0], $parametrosTO->getBairroDestino(),
'onChange=consultarAvenidaDest()); ?></td>
                        <td width="50%">Avenida*:<br><?php
HtmlUtil::gerarCombobox('localDestino', 'OrigemTO', 'codAvenida', 'descAvenida', $listaOrigemTO[2],
$parametrosTO->getLocalDestino(), 'onChange=consultarPontoDest()); ?></td>
                        <td width="35%">Ponto*:<br> <?php
HtmlUtil::gerarCombobox('pontoDestino', 'pontoTO', 'codPonto', 'descPonto', $listaOrigemTO[4], $parametrosTO-
>getPontoDestino()); ?></td>
                    </tr>
                    </tr>
                    <td colspan="3"><br></td>
                    </tr>
                    <tr>
                    <td colspan="3" class="td1"><b>*Campos Obrigatórios.</b></td>
                    </tr>
                </table>
                <br><br>
                <div align="left">
                    <input type="button" name="Pesquisar" value="Pesquisar" class="botao"
onclick="pesquisar();"/>
                </div>
            </fieldset>
            <div id="resultado" style="display:<?php print_r( $div ); ?>">
                <fieldset>
                    <legend>Resultado</legend>
                    <table class="zebrado" id="zebrado">
                        <thead>
                            <tr>
                                <th width="26%" align="left">Linha</th>

```

```

<th width="38%" align="left">Origem e Destino da
Linha</th>
<th width="13%">Distância</th>
<th width="13%">Tempo</th>
<th width="10%"></th>
</tr>
</thead>
<div id="map_canvas" style="display:none;" ></div>
<script language="JavaScript" src="../../web/js/distTempo.js"></script>
<?php
    $bgcolor = '#EDF3F6';
    while(list($key, $to) = each($listaResultado)) {
        if ($key % 2 == 0) {
            $bgcolor = '#FFFFFF';
        } else {
            $bgcolor = '#EDF3F6';
        }

        $latOnibus = $to->getLatitudeOrigem();
        $lonOnibus = $to->getLongitudeOrigem();
        $latPonto = $to->getLatitudeDestino();
        $lonPonto = $to->getLongitudeDestino();
        echo"<script>teste($key, '$latOnibus,$lonOnibus', '$latPonto,$lonPonto')</script>";
    }
?>
<tr bgcolor="<?php print_r( $bgcolor ); ?>" class="odd">
<td align="left">
<?php print_r($to->getLinhaTO()->getNuLinha()); ?> - <?php print_r($to->getLinhaTO()-
>getDescLinha()); ?>
</td>
<td align="left">
<?php print_r($to->getLinhaTO()->getDescOrigemLinha());?><br>
<?php print_r($to->getLinhaTO()->getDescDestinoLinha());?>
</td>
<td align="center">
<div id="getDistance"<?php print_r( $key ); ?>></div>
</td>
<td align="center">
<div id="getDuration"<?php print_r( $key ); ?>></div>
</td>
<td align="center"><a href="/projetoFinal/app/PF/SAUO/classes/view/mapaAction.php?acao=visuMapa&
olat=<?php print_r($latOnibus); ?>&
olon=<?php print_r($lonOnibus); ?>&
dlat=<?php print_r($latPonto); ?>&
dlon=<?php print_r($lonPonto); ?>" target="_blank">Ver no
mapa</a></td>
</tr>

```

```

<?php
    }
?>

</table>

</fieldset>

</div>

<br>

</td>
</tr>
</table>
<?php
    include(INC_PATH . '/rodape.inc.php');
?>
</div>
</form>
</body>
</html>
<?php
    session_unregister('mensagem');
?>

```

Código 41 - estilo.css

```

/* CSS Document */
@media all
{
/* GERAL */
body {margin: 0px;
    background-color: #ffffff;
    text-decoration:none;
    font-size:small;
    font-family: Verdana, Arial, Helvetica, sans-serif;
    padding: 0px;
    color: #000000;}

img {border: medium none;
    vertical-align:middle;}

hr {border: 0px;
    margin: 0.5em 0px 1em;
    color: #8cacbb;
    height: 1px;
    background-color: #8cacbb;}

/* TEXTO */

```

```
h1, h2, h3, h4, div, p, ul, ol, li, select, form, label, input, textarea, legend, button, table, tr, td, th, thead, tbody
{font-family: Verdana, Arial, Helvetica, sans-serif;}
```

```
h1 {font-size:160%;
      font-weight: normal;
      margin: 0px 0px 0.25em;
      color: #5ea4c0;
      font-family: elegagarmnd bt, georgia, times;
      background-color: transparent;}
```

```
h2 {font-size:2em;}
```

```
p {margin-bottom:10px;
    font-size:0.9em;}
```

```
a {color: #5EA4C0;
    background-color: transparent;
    text-decoration: none;}
```

```
a:link {color: #5ea4c0;
        background-color: transparent;
        text-decoration: none;}
```

```
a:hover {color: #5ea4c0;
         background-color: transparent;
         text-decoration:none;}
```

```
a:visited, a:active {color: #5ea4c0;
                    background-color: transparent;
                    text-decoration: none;}
```

```
fieldset {border:1px solid #5ea4c0;
          padding:10px;
          margin-bottom:10px;
          padding: 1em;
          width: auto;
          line-height: 1.5em;}
```

```
legend {font-weight: bold;
        padding: 0.5em;
        font-size: 10px;
        color: #5ea4c0;
        background: #ffffff;}
```

```
/* FORMS */
```

```
textfield {border:1px solid #cccccc;
           background-color:#FFFFFF;
           margin:0;
           padding:0;}
```

```
        font-size:0.9em}
input {color: #5ea4c0;
       text-decoration:none;
       margin-bottom: 1px;
       border: #92c6d6 1px solid;
       visibility: visible;
       vertical-align: middle;
       color: #000000;
       font-size: 0.9em;}
input.botao {border: #92c6d6 1px solid;
            background: url(..img/input_fundo.jpg);
            visibility: visible;
            vertical-align: middle;
            color: #92c6d6;
            font-size: 9px;}
input.texto { border: #92c6d6 0px solid;
            visibility: visible;
            vertical-align: middle;
            color: #F70707;
            font-size: 14px;}
input.check {border: 0px;
            visibility: visible;
            vertical-align: middle;}
.combobox {font-size: 9px;
          font-style: normal;
          background-color: #ffffff;
          border: 1px solid #5ea4c0;}
select {border: #92c6d6 1px solid;
        vertical-align: top;
        color: black;
        font-size: 9px;
        background-color: white;}
button {font-family: "Lucida Grande", Verdana, Lucida, Helvetica, Arial, sans-serif;
        visibility: visible;
        border: 1px solid #8cacbb;
        color: #000000;
        vertical-align: middle;
        background-color: #dee7ec;
        padding: 1px;
        cursor: pointer;
        font-size: 0.9em;
        text-transform: lowercase;}
form {border: medium none;
      margin: 0px;}
```

```

/* TABELAS */

td.menu {border-right: #d6e3e7 1px solid;
        background: url(..img/fundo_menu.jpg);}
td.rodape {border-bottom: #7bc3e7 1px solid;
        border-collapse: separate;
        padding: 10px;
        height:270px;
        vertical-align: top;}
table.zebrado {border: 1pt solid #edf3f6;
        width: 100%;
        font-size: 10px;
        border-collapse: collapse;
        padding: 0;}
.zebrado td {border: 1pt solid #edf3f6;
        font-size: 10px;
        border-collapse: collapse;
        padding: 0;}
.zebrado th {border: 1pt solid #edf3f6;
        border-top: #edf3f6 1px solid;
        font-size: 10px;
        background: #dee7ec;
        border-bottom: #edf3f6 1px solid;
        border-collapse: collapse;
        padding: 0;}
td.td1 {border: 1px solid #edf3f6;
        font-size: 10px;
        background: #f4fafc;
        border-collapse: collapse;
        padding: 0;}
tr.td1 {font-size: 10px;
        background: #f4fafc;
        border-collapse: collapse;
        padding: 0;}
.poweredby {font-size: 9px;
        color: #000000;
        text-align: center;
        font-family: "verdana, lucida, helvetica, arial, sans-serif"}

/* EXTRAS */

.bold {font-weight:bold;}
.borda_central {border:2px solid #6eb6cf;}
.textocentralizado {text-align:center;}

```

```
.textodireita {text-align:right;}
.textoesquerda {text-align:left;}
.textojustificado {text-align:justify;}
@media screen {
table {font-size: 10px;}
#container {width: 760px;
margin-left: auto;
margin-right: auto;
border: 1px solid #dce8f2;}
```

Código 42 - distTempo.js

```
function teste(num, from, to){
var map;
var gdir;
map = new GMap2(document.getElementById("map_canvas"));
gdir = new GDirections(map, document.getElementById("directions"));
GEvent.addListener(gdir, "load", function() {
document.getElementById("getDistance"+num).innerHTML = gdir.getDistance().html;
document.getElementById("getDuration"+num).innerHTML = gdir.getDuration().html;
});
gdir.load("from: " + from + " to: " + to, { "locale": 'en_US' });
}
```