

GUSTAVO CARNEVALE COUTO
VITOR DA SILVA GATTAI

MONITORAMENTO DO TRÂNSITO VIA DISPOSITIVO MÓVEL

SÃO PAULO
2008

GUSTAVO CARNEVALE COUTO

VITOR DA SILVA GATTAI

MONITORAMENTO DO TRÂNSITO VIA DISPOSITIVO MÓVEL

Monografia apresentada à Escola
Politécnica da Universidade de São Paulo
para obtenção do título de Engenheiro

Área de concentração:
Engenharia Mecatrônica

Orientador: Professor Doutor Marcos
Ribeiro Pereira Barretto

SÃO PAULO

2008

DEDICATÓRIA

Dedicamos este trabalho às nossas
famílias.

AGRADECIMENTOS

Gustavo gostaria de agradecer:

Aos meus pais Octavio e Rosangela e à minha irmã Camila, assim como toda a minha família, pelo apoio dado durante a realização do curso e do desenvolvimento deste trabalho.

Ao Prof. Dr. Marcos Ribeiro Pereira Barretto pela orientação deste trabalho e pelos ensinamentos dados durante o curso e ao meu colega de trabalho Vitor.

A todos os amigos que me deram força para a realização deste trabalho.

A toda equipe da MapLink pelo fornecimento do serviço e pelo suporte dado a equipe.

Vitor gostaria de agradecer:

A toda minha família, em especial, meus pais Ricardo e Olga, meu irmão Caio e minha namorada Camila por todo o apoio durante o curso.

Ao Prof. Dr. Marcos Ribeiro Pereira Barretto pelo auxílio durante o desenvolvimento deste trabalho e ao meu colega de trabalho Gustavo.

Aos meus amigos do Clube Ipê e do Colégio Augusto Laranja e meus colegas de faculdade que sempre me incentivaram.

A toda equipe da MapLink pelo fornecimento dos *Web Services* e pelo suporte dado.

RESUMO

O principal objetivo do presente trabalho é o desenvolvimento de uma aplicação para dispositivos móveis, com a motivação de estudar métodos de programação para esse tipo de aparelho, além de se conhecer novas ferramentas de desenvolvimento e formas de estruturação de *softwares*. Para esse estudo, ficou decidida a programação de um sistema de monitoramento do trânsito, por se tratar de um tema amplamente discutido e que possui diversos problemas a serem resolvidos, podendo ser bastante explorado do ponto de vista da programação.

Palavras-chave: Programação, Dispositivos Móveis, PDA, *Smartphone*, Web Service, Windows Mobile, Banco de Dados, Monitoramento e Trânsito.

ABSTRACT

The aim of this project is to develop an application for mobile devices, motivated in studying programming methods for this type of device, besides learning new tools and software's structures. For this study, a traffic monitoring system was chosen to be programmed, because it is a much discussed subject which has several problems to be solved and can be highly explored in the programming area.

Keywords: Programming, Mobile Devices, PDA, Smartphone, Web Service, Windows Mobile, Database, Monitoring and Traffic.

LISTA DE FIGURAS

Figura 1 - Aumento do número de veículos..	12
Figura 2 - Cronograma do primeiro semestre.....	15
Figura 3 - Cronograma do segundo semestre	16
Figura 4 - Matriz de Decisão.....	17
Figura 5 - Esquema de comunicação do sistema.....	20
Figura 6 - Comunicação entre as classes do sistema	26
Figura 7 - Situação do trânsito esquematizada em uma rua	28
Figura 8 - Região mapeada na tabela Trânsito.....	29
Figura 9 - Diagrama de Entidade-Relacionamento.....	37
Figura 10 - Login no sistema	39
Figura 11 - Pesquisar Trecho.....	40
Figura 12 - Consultar Trecho	42
Figura 13 - Calcular rota	44
Figura 14 - Opções de rota detalhada	44
Figura 15 - Voltar às opções de rota.....	45
Figura 16 - Consultar rota.....	46

LISTA DE TABELAS

Tabela 1 - Dados na tabela Trânsito	29
Tabela 2 - Dados na tabela Usuário.....	31
Tabela 3 - Dados na tabela Rua	33
Tabela 4 - Dados na tabela Rota	35

LISTA DE ABREVIATURAS E SIGLAS

ASP	Active Server Pages
BD	Banco de Dados
CEP	Código de Endereçamento Postal
CET	Companhia de Engenharia de Tráfego
CPF	Cadastro de Pessoas Físicas
EDGE	Enhanced Data rates for Global Evolution
FHWA	Federal Highway Administration
FIA/USP	Fundação Instituto de Administração da Universidade de São Paulo
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System Mobile
IIS	Internet Information Services
IP	Internet Protocol
IPEA	Instituto de Pesquisa Econômica Aplicada
PC	Personal Computer
PDA	Personal Digital Assistant
PNG	Portable Network Graphics
SQL	Structured Query Language
USB	Universal Serial Bus
Wi Fi	Wireless Fidelity

SUMÁRIO

1 INTRODUÇÃO	10
1.1 Considerações Iniciais	10
1.2 Motivação	10
1.3 O Trânsito em São Paulo	11
2 PLANEJAMENTO DO PROJETO	15
2.1 Atividades Previstas	15
2.2 Escolha do tipo de dispositivo móvel	16
2.3 Escolha da Linguagem e Ambiente de Programação	17
2.4 Utilização do Web Service da MapLink	18
3 ESTRUTURA DO SISTEMA	20
3.1 Organização funcional	20
3.2 Detalhamento dos componentes	21
3.2.1 <i>Smartphone</i>	21
3.2.2 <i>Servidor Local</i>	22
3.2.3 <i>Web Services MapLink</i>	24
3.3 Estrutura do Banco de Dados	26
3.3.1 <i>Tabela Trânsito</i>	27
3.3.2 <i>Tabela Usuário</i>	31
3.3.3 <i>Tabela Rua</i>	32
3.3.4 <i>Tabela Rota</i>	34
3.3.5 <i>Diagrama de Entidade-Relacionamento</i>	36
4 CASOS DE USO	38
5 CONCLUSÕES	47
6 REFERÊNCIAS	50
APÊNDICE A	52
APÊNDICE B	53

1 INTRODUÇÃO

1.1 Considerações Iniciais

Para quem vive em grandes centros urbanos como São Paulo, os congestionamentos no trânsito já se tornaram parte de sua rotina. Muitas soluções, sejam elas estruturais ou não, vêm sendo desenvolvidas, mas o problema parece estar longe de ser resolvido.

Além disso, outro hábito se difundiu rapidamente nas grandes cidades, atingindo pessoas de todas as idades e características: o uso de dispositivos móveis (celulares, *notebooks*, PDAs).

Aliando esses dois conceitos, percebeu-se que a idéia de se desenvolver um sistema para dispositivos móveis, no qual o usuário pudesse visualizar a condição do trânsito nas ruas e avenidas da cidade, em tempo real, seria de grande utilidade. Uma pesquisa de campo, realizada pela equipe desse projeto, revelou forte interesse da população em um sistema como esse.

O objetivo deste relatório é mostrar ao leitor o planejamento e estruturação de um *software*, que foi desenvolvido e testado pela equipe, com as características supracitadas.

1.2 Motivação

A principal motivação para a realização desse trabalho foi a oportunidade que a equipe teve em aprender a utilizar novas ferramentas de programação. Através do desenvolvimento do Traffic Monitor (nome dado pelo grupo ao sistema de monitoramento do trânsito) foi possível aprofundar os conhecimentos em programação orientada a objetos e aprender como criar *softwares* próprios para dispositivos móveis.

Outro importante aprendizado foi a utilização de *Web Services*. Através dessa

poderosa ferramenta consegue-se fazer programas que utilizam métodos existentes em outros servidores na Internet. Isto é, consegue-se desenvolver uma aplicação que se aproveita de funções já disponíveis na rede e apenas usar seus resultados, não sendo necessário reprogramar suas lógicas. Esse tipo de serviço é largamente utilizado em buscas de CEPs, CPFs e endereços. Todo o processamento é realizado no servidor onde está o *Web Service* e este retorna a resposta ao programa “cliente”.

Outra motivação deste trabalho foi o aprofundamento na programação através do Microsoft Visual Studio 2005, um *software* que pode ser utilizado tanto para criar *Web Services* como para desenvolver aplicativos para dispositivos móveis. O Visual Studio é um programa extremamente potente e amigável, que facilita a criação de interfaces gráficas, integração com banco de dados, compilação de programas e publicação de *Web Services*.

Também serviu de motivação a oportunidade de a equipe desenvolver ainda mais seus conhecimentos sobre Banco de Dados e linguagem SQL.

Por fim, além das motivações técnicas já citadas, o fato de se desenvolver um projeto que pudesse ser útil aos usuários, os auxiliando a resolver um problema recorrente de suas vidas, como é o trânsito, serviu de grande inspiração para esse trabalho.

O fato de não existirem soluções definitivas para esse problema também motivou a equipe. Enquanto as soluções estruturais, como criação de metrô e novas vias na cidade, ainda não surtem efeitos significativos, o grupo pensou que uma forma de escapar, ou pelo menos estar informado sobre a condição do trânsito, de forma rápida e fácil, seria bastante útil aos usuários.

1.3 O Trânsito em São Paulo

Os congestionamentos em São Paulo se tornam cada vez mais insuportáveis, principalmente nos horários de pico. Neste ano foram registrados inúmeros recordes de congestionamentos na cidade e o pior é que agora o trânsito não se concentra apenas no período da tarde, ocorrem muitos congestionamentos também no período da manhã nas principais vias de São Paulo.

O número de carros novos que começam a circular nas ruas da metrópole é impressionante, e esses números vêm crescendo cada vez mais. Isso pode ser observado na Figura 1, a qual mostra que no mês de fevereiro deste ano a frota de carros de São Paulo ultrapassou o incrível número de 6 milhões. A recente crise de crédito afetou negativamente a venda de carros, mas a perspectiva é que, apesar de ocorrer com menor intensidade, o número de veículos circulando na cidade continue aumentando. Existem aproximadamente 2,4 habitantes por carro na região metropolitana.

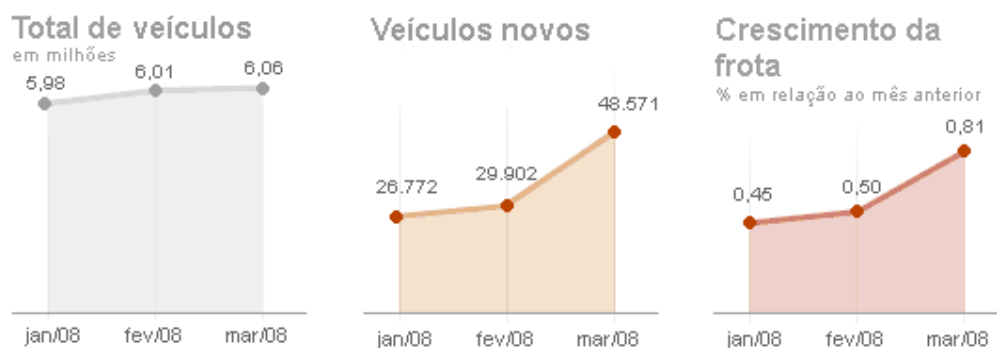


Figura 1 - Aumento do número de veículos. Fonte: UOL Notícias

A situação do trânsito é tão caótica na cidade que já existem registros de congestionamentos dentro de estacionamentos de prédios comerciais da zona sul da cidade. Engarrafamentos nas avenidas principais impedem a vazão dos carros que tentam sair dos edifícios.

Especialistas dizem que caso a frota de carros continue a crescer no mesmo ritmo o trânsito da cidade entraria em colapso em apenas quatro anos. As vias da cidade de São Paulo ficariam paradas durante o dia inteiro.

Além de estresse aos motoristas, o trânsito de São Paulo traz também prejuízo aos cofres públicos. Segundo a Secretaria de Estado dos Transportes Metropolitanos, o trânsito da região metropolitana de São Paulo gera um custo de R\$ 4,1 bilhões por ano.

A constatação tem como base estudos da FIA/USP (Fundação Instituto de Administração da Universidade de São Paulo), do IPEA (Instituto de Pesquisa Econômica Aplicada) e da FHWA (Federal Highway Administration), que convertem

em dinheiro o tempo gasto pelas pessoas nos seus deslocamentos (R\$ 3,6 bilhões), além do prejuízo causado pela poluição atmosférica (R\$ 112 milhões) e com os acidentes de trânsito (R\$ 312 milhões), anualmente.

Obviamente, existe um problema estrutural com relação ao trânsito da cidade. O transporte público deixa muito a desejar. É evidente que os 61,3 km de extensão das linhas de metrô de São Paulo são muito poucos comparados ao volume de pessoas que precisam se deslocar pelas cidades no dia-a-dia. Essas linhas não atingem regiões importantes da metrópole, obrigando os trabalhadores a usarem os ônibus, que, por sua vez pecam nos quesitos eficiência e segurança, levando àqueles que têm condição de comprar um veículo próprio a partirem para essa opção.

O problema é tão grave que a CET (Companhia de Engenharia de Tráfego) criou o rodízio municipal de veículos que proíbe a circulação de automóveis no Centro Expandido (região delimitada pelas Marginais Tietê e Pinheiros, a Avenida dos Bandeirantes, o Complexo Viário Maria Maluf, as Avenidas Tancredo Neves e Juntas Provisórias, o Viaduto Grande São Paulo e as Avenidas Professor Luís Inácio de Anhaia Melo e Salim Farah Maluf) da cidade uma vez por semana nos horários de pico.

Todos esses fatores agindo em conjunto fazem um ciclo vicioso que parece estar longe do fim. Obras estão realizando obras com a intenção de quadruplicar a extensão das linhas de metrô, mas até lá, o paulistano ainda sofrerá muito no trânsito e é difícil saber ao certo se isso resolverá de vez o problema.

Recentemente, algumas soluções, que não resolvem esses problemas estruturais, mas apenas buscam oferecer maneiras do motorista desviar das vias congestionadas, vêm sendo implementadas. Podemos citar como exemplo a Rádio Sulamérica Trânsito, que presta informações das condições das vias aos ouvintes durante todo o dia. Outro exemplo é o mapa de fluidez mostrado no site da CET, que revela o trânsito nas principais vias da cidade em tempo real, mas deixa muito a desejar.

Alguns modelos de GPS estão começando a ser fabricados com recursos anti-trânsito, que possibilitariam ao usuário receber uma rota com as vias de melhor trânsito no momento, mas esse recurso está para ser ativado apenas nos próximos meses e não se sabe ao certo se será eficiente.

A melhor solução, na opinião da equipe, foi feita pela MapLink. A empresa

possui um site no qual o usuário pode acessar a condição o trânsito em diversas vias da cidade. As informações são atualizadas a cada 15 minutos e é possível inclusive calcular uma rota entre dois pontos da cidade e obter um mapa com a indicação da rota que também mostra as condições do trânsito nas vias a serem utilizadas.

2 PLANEJAMENTO DO PROJETO

2.1 Atividades Previstas

O projeto do Traffic Monitor foi desenvolvido durante o ano de 2008.

A Figura 2 e Figura 3 mostram os cronogramas utilizados para o primeiro e segundo semestre, respectivamente.

atividades	mês	Abril					Maio					Junho		
	semana	1a	2a	3a	4a	5a	6a	7a	8a	9a	10a	11a	12a	13a
seleção do ambiente de programação a ser utilizado (Java ou .NET)														
estudo do ambiente de programação para celular														
seleção do ambiente de programação do servidor (Java ou .NET)														
estudo da interface com GPS para celular														
programação de rede com celular														
estudo da programação para rede do servidor														
preparação dos casos de uso do sistema a ser desenvolvido														
preparação do projeto do software a ser desenvolvido (parte do celular+parte do servidor)														

Figura 2 - Cronograma do primeiro semestre

Atividades	mês	Agosto				Setembro					Outubro				Novembro			
	semana	1a	2a	3a	4a	5a	6a	7a	8a	9a	10a	11a	12a	13a	14a	15a	16a	17a
Estruturação do Servidor e Testes de comunicação																		
Estudo do Sistema da CET																		
Familiarização com o ambiente de desenvolvimento em celular																		
Revisão dos Casos de Uso																		
Revisão do Design e da estrutura do software																		
Modelagem do Banco de Dados com informações do trânsito																		
Programação do Software																		
Testes e Ajustes Finais																		

Figura 3 - Cronograma do segundo semestre

Durante a execução do projeto, algumas decisões foram tomadas e as mais importantes estão descritas a seguir.

2.2 Escolha do tipo de dispositivo móvel

Inicialmente, havia uma dúvida entre o desenvolvimento da aplicação para dispositivos mais simples, como celulares, ou mais complexos, como PDAs.

Para essa definição, utilizou-se a matriz de decisão (Figura 4) com os parâmetros julgados relevantes para o trabalho. Os pesos foram dados considerando-se que, com o desenvolvimento do sistema, o objetivo era buscar soluções mais didáticas, familiares e atraentes ao público-alvo.

Matriz de Decisão		Celular		PDA	
Crítérios	Peso	Nota	Subtotal	Nota	Subtotal
Resolução Gráfica	4	6	24	9	36
Programação	4	7	28	9	36
Sistema Operacional	3	5	15	8	24
Facilidade de Uso	3	6	18	8	24
Mecado Potencial	2	10	20	6	12
Praticidade	3	9	27	6	18
Total		132		150	
Média		6,9		7,9	

Figura 4 - Matriz de Decisão

Além disso, com o desenvolvimento do projeto, notou-se que não fazia sentido o uso de GPS, uma vez que o objetivo do *software* é apenas o de informar a condição do trânsito nas ruas que serão utilizadas pelo usuário, sendo que este último deve acioná-lo antes de entrar no seu veículo.

Os sistemas anti-trânsito dos aparelhos integrados com GPS têm uma abordagem diferente, que é calcular a rota desejada levando em consideração apenas vias com condições de trânsito boas. Isso foge do escopo desse projeto, uma vez que o Traffic Monitor busca justamente encontrar as vias que possuem trânsito ruim de maneira mais fácil e deixar a tomada de decisão nas mãos do usuário. O sistema procura focar na facilidade de consulta às condições de trânsito da via escolhida pelo usuário e não em guiar o mesmo até seu destino por caminhos que ele não conhece, eventualmente desviando das ruas congestionadas.

2.3 Escolha da Linguagem e Ambiente de Programação

Optou-se por utilizar o ambiente do Visual Studio, pois já se conhecia suas ótimas funções por projetos anteriores. Além disso, este foi escolhido por facilitar a comunicação entre o servidor e o dispositivo móvel, o que é fundamental para o bom funcionamento do sistema.

O *software* Visual Studio também é interessante por possuir ferramentas que auxiliam na criação de interfaces gráficas. Além disso, para a realização de testes dos códigos, o Visual Studio possui um emulador de dispositivos móveis, o que facilita a realização de teste de funcionamento do sistema, pois não é necessário compilar o programa no dispositivo real para testá-lo.

A linguagem de programação selecionada foi o Visual Basic, por ser simples, objetiva e permitir programação orientada a objetos, sendo também adequada para chamar os métodos dos *Web Services* da MapLink, que serão comentados mais a frente.

Outro ponto importante, é que os PDAs geralmente são equipados com o Windows Mobile 6, o qual é baseado em Windows CE. Isso facilita a integração entre Visual Studio e dispositivo móvel, uma vez que ambos são da Microsoft e possuem um ótimo compartilhamento de funções.

2.4 Utilização do Web Service da MapLink

Com o intuito de aprimorar o sistema desenvolvido, buscou-se *Web Services* já disponíveis que pudessem agregar valor ao projeto. Uma preocupação do grupo era tornar o projeto o mais útil possível e, se tratando de monitoramento do trânsito, uma ferramenta extremamente útil seria o cálculo de rotas e obtenção de mapas. Dessa forma, o usuário poderia não só saber as condições do trânsito em uma determinada rua, como também, saber qual caminho deve fazer para chegar ao seu destino. Essa função é muito importante, pois o *software* se torna útil para todas as situações, inclusive aquelas na qual o usuário não domina completamente o caminho até o seu destino.

O serviço que foi utilizado nesse projeto foi o da MapLink. Essa empresa disponibiliza *Web Services* capazes de executar diversas funções como traçar rotas, gerar mapas, encontrar endereços, encontrar pontos de interesses, estimar o tempo necessário de uma rota, entre outras.

2.5 Obtenção das informações do trânsito

Para alimentar o Banco de Dados com as informações nos trânsitos da via da cidade, tinham-se dois cenários possíveis: encontrar um serviço, possivelmente da CET, com as informações reais do trânsito ou gerar dados fictícios.

Após algumas pesquisas feitas, percebeu-se que as informações reais não eram possíveis de serem conseguidas com rapidez e eficiência e, dado que o objetivo final do trabalho era viabilizar a lógica do sistema (e não torná-lo comercial), partiu-se para o uso de informações simuladas do trânsito, inseridas manualmente.

3 ESTRUTURA DO SISTEMA

3.1 Organização funcional

O programa de monitoramento do trânsito é composto por três elementos básicos:

- i) PDA, *Smartphone* ou Simulador com o Traffic Monitor compilado;
- ii) Servidor local: Computador com o *Web Service* publicado no IIS (Internet Information Services) e Banco de Dados MySQL com as informações do trânsito e dados dos usuários;
- iii) Servidor da MapLink, que contém seus *Web Services*.

A comunicação entre esses elementos pode ser vista no esquema da Figura 5.

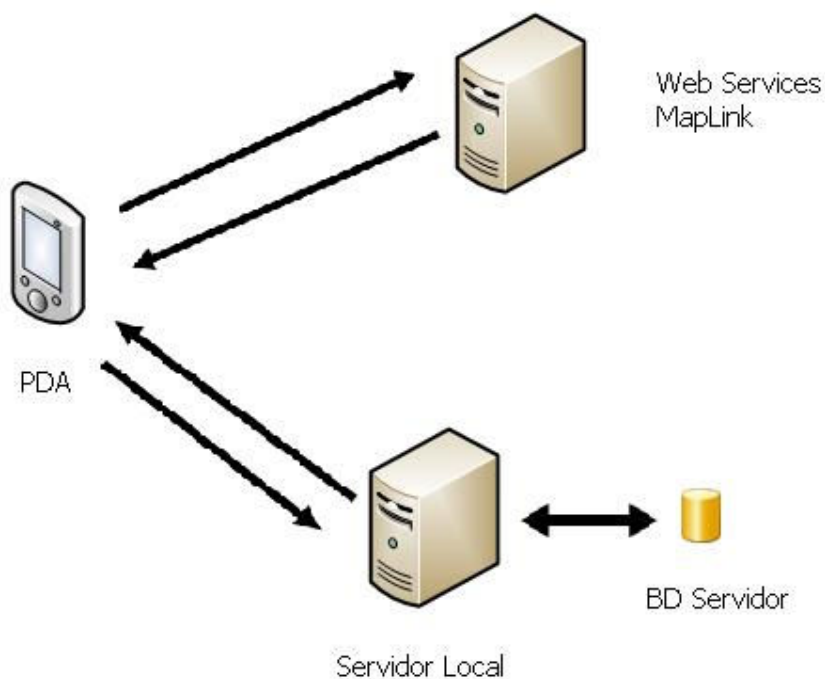


Figura 5 - Esquema de comunicação do sistema

A conexão entre dispositivo móvel e o servidor local será feita através da Internet por intermédio do IIS, presente na máquina servidor. Com essa estrutura, é possível fazer a maior parte do processamento de dados no servidor, enquanto o programa compilado no dispositivo apenas chama os métodos presentes no *Web Service*. Tal estrutura é importante, uma vez que PDAs e *Smartphones* não possuem o mesmo poder de processamento de um PC e isso torna o acesso às informações mais rápido para o usuário.

3.2 Detalhamento dos componentes

Uma vez definida as características do sistema, foi possível escolher componentes que tivessem as funcionalidades necessárias. Nessa seção, serão apresentadas as especificações dos componentes, bem como seus funcionamentos e interações com o sistema.

3.2.1 Smartphone

3.2.1.1 Especificação

Para poder realizar todas as funções do *software* projetado, o PDA ou *Smartphone* selecionado deveria possuir Windows Mobile, acesso à Internet, preferencialmente com pacote de dados (EDGE ou 3G, para facilitar o acesso em locais sem conexão Wi Fi) e capacidade de compilação de *softwares* no próprio aparelho.

Durante a fase de desenvolvimento, foi utilizado o simulador do Visual Studio Pocket PC 2003 SE Emulator, que possui as características citadas, com a diferença de que a conexão à Internet é feita por compartilhamento de dados com o computador *host*.

Para o usuário final, ficou escolhido o *Smartphone* HTC Touch, que possui todas as funções necessárias. Abaixo, uma descrição mais detalhada de suas características e no Apêndice A todos os dados do datasheet oficial.

Modelo: HTC Touch

Processador: TI's OMAP 850, 201MHz

Memória: ROM: 128MB, SDRAM: 64 MB DDR

Sistema Operacional: Windows Mobile 6 Professional

Conectividade: Bluetooth, Wi Fi e USB

Redes: GSM/GPRS/EDGE

3.2.1.2 Funcionamento

O dispositivo móvel, como é aquele que está em contato direto com o usuário final, é o responsável por fazer a comunicação entre esse, o servidor local e o servidor da MapLink.

Através de interfaces gráficas amigáveis, o usuário pode selecionar as funções desejadas e, pela Internet, o Traffic Monitor, automaticamente chama os métodos dos servidores necessários e devolve ao usuário as informações já processadas e tratadas. Dessa forma, o usuário final recebe as informações desejadas de forma prática e precisa sem precisar saber como o fluxo de informações é feito.

3.2.2 Servidor Local

3.2.2.1 Especificação

Para hospedar o *Web Service* principal, foi necessária a escolha de um computador que tivesse boa conexão com a Internet, alta capacidade de

processamento e alguns *softwares* que estão descritos nas especificações a seguir:

Processador: AMD Athlon 64 3200

Memória RAM: 1GB

Sistema Operacional: Windows XP Professional Service Pack 2

Conectividade: USB e Wi Fi

Redes: Internet Speedy 2MB

Plataforma para execução: Microsoft .NET Framework 2.0

Plataforma para simulação: Visual Studio 2005 e Pocket PC 2003 SE Emulator

Banco de Dados: MySQL Server 5.0

Servidor: IIS 5.1

Em função de o servidor local possuir acesso à internet com IP dinâmico e estar conectado a um roteador, foi necessário configurá-lo de modo a permitir que o dispositivo móvel consiga acessá-lo corretamente. Foi necessário abrir exceções no firewall do Windows XP e configurar o redirecionamento do IP do roteador para o IP da máquina servidor, além de se incluir uma linha de código no programa que permita a troca de IP.

3.2.2.2 Funcionamento

O servidor local contém uma página desenvolvida em ASP.NET, que funciona como um *Web Service*, o qual tem a capacidade de realizar consultas e inserir/excluir informações em um Banco de Dados MySQL

Nos casos de uso nos quais o *Web Service* local precisa ser acionado, este recebe do dispositivo móvel parâmetros de entrada que são tratados pelos seus métodos, os quais, por sua vez, devolvem ao dispositivo as respostas encontradas. Nesse processo, muitas vezes o Banco de Dados é consultado ou modificado.

Os principal método do *Web Service* local é o “PesqRua”, que possui como parâmetros de entrada o nome da rua escolhida, um cruzamento inicial e um cruzamento final. Esse método retorna um *DataSet* que mostra a condição do trânsito a cada cruzamento dessa via.

No caso, da determinação do trânsito em uma rota completa, o “PesqRua” é acionado diversas vezes para se obter as condições do trânsito em todo o trajeto.

Além desse, o servidor local possui outros importantes métodos:

- Login: faz a verificação de usuário e senha e retorna o número de identificação do usuário;
- SalvarRua: salva um trecho definido pelo usuário para posterior utilização;
- SalvarRota: salva uma rota definida pelo usuário para posterior utilização;
- Cons_Trechos: com o número de identificação do usuário, retorna o nome dos trechos salvos pelo mesmo;
- Cons_Rotas: com o número de identificação do usuário, retorna o nome das rotas salvas pelo mesmo;
- Trecho_Escolhido: com o número de identificação do usuário e nome do trecho, obtém no banco de dados as entradas do “PesqRua”, aciona este e devolve um *DataSet* com as condições do trânsito;
- Rota_Escolhida: com o número de identificação do usuário e nome da rota, obtém no banco de dados os endereços inicial e final da rota salva e os devolve em um *ArrayList*.
- Delete_Trecho: exclui o trecho desejado do banco de dados através do número de identificação do usuário e do nome do trecho;
- Delete_Rota: exclui a rota desejada do banco de dados através do número de identificação do usuário e do nome da rota.

3.2.3 Web Services MapLink

3.2.3.1 Especificação

Dentre os serviços disponibilizados pela empresa, optou-se pelo uso da versão 2.1, que é a mais usada e utilizaram-se apenas os *Web Services* necessários para a obtenção das respostas desejadas, que são os seguintes: Authentication, AddressFinder e Route.

3.2.3.2 Funcionamento

Nos casos de uso nos quais o usuário solicita cálculos de rota ou desenho de mapas, os *Web Services* da MapLink precisaram ser acionados.

Para o cálculo de rotas, o dispositivo envia à página da MapLink os endereços dos pontos de partida e chegada da rota, com dados de rua, número, cidade e estado.

A MapLink codificou o seu mapa através de coordenadas em um plano XY. Dessa forma, através do método “getXY” ela consegue, com os dados informados determinar quais são as coordenadas dos endereços fornecidos.

Essas coordenadas servem de parâmetros de entrada para a função “getRoute”, que devolve ao dispositivo móvel diversas *strings* que descrevem a rota a ser percorrida, entre elas as mais importantes são: *Command*, que mostra qual comando o usuário deve fazer (exemplos: vire à direita; continue em frente) e *Description*, que referencia o comando a uma rua (exemplo: Av. Paulista; Av. 23 de maio).

Assim como a função “getRoute”, o método “getRouteWithMap”, responsável por gerar o mapa da rota desejada, utiliza como parâmetros as coordenadas X e Y dos pontos de partida e chegada. Com esses dados, é possível delimitar um quadrado a ser mostrado ao usuário, bem como o traçado da rota desenhado no mapa, que é disponibilizado em um arquivo PNG em um *link* na Internet.

Na Figura 6, é possível ver um esquema da estrutura completa do sistema, que mostra a relação entre suas diversas classes e dá uma idéia de como é a comunicação entre os componentes. No Apêndice B é possível ver o detalhamento de cada classe do Traffic Monitor.

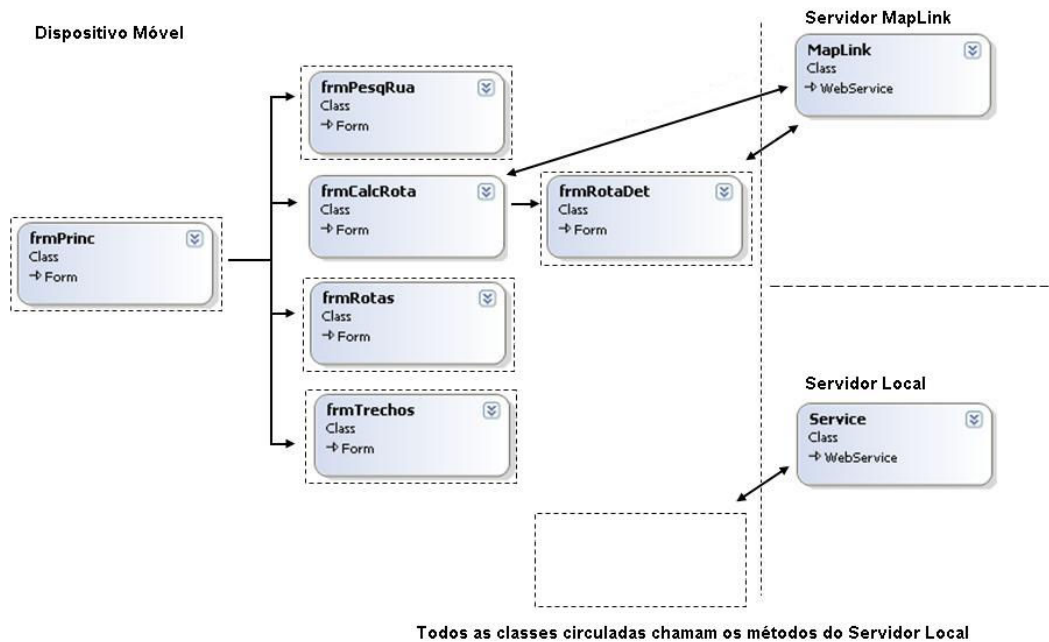


Figura 6 - Comunicação entre as classes do sistema

3.3 Estrutura do Banco de Dados

As informações sobre as condições do trânsito em alguns cruzamentos da cidade de São Paulo, dados dos usuários cadastrados, e trechos e rotas salvos pelos usuários são armazenados em um Banco de Dados. O mesmo possui uma estrutura que permite *logar* no sistema, consultar as condições do trânsito e procurar os trechos e rotas salvos da maneira mais fácil e rápida possível.

O Banco de Dados utilizado foi desenvolvido em MySQL. Ele possui informações simuladas sobre as condições do trânsito de algumas ruas da cidade de São Paulo. Este Banco de Dados, que é chamado de Traffic, possui quatro tabelas, são elas: Trânsito, Usuário, Rua e Rota. A seguir, cada uma delas é explicada com maiores detalhes.

3.3.1 Tabela Trânsito

Essa tabela contém informações fictícias sobre as condições do trânsito de diversos cruzamentos.

A tabela Trânsito possui seis campos, são eles:

- **id**: chave-primária de identificação;
- **Nome**: nome da rua onde será informada a condição do trânsito;
- **Cruzamento**: nome do cruzamento onde é informado o trânsito;
- **Estado1**: condição do trânsito em um sentido;
- **Estado2**: condição do trânsito no outro sentido;
- **Segmento**: mostra qual o segmento que está sendo informado.

Cada rua é dividida em diversos segmentos. A tabela tem informações sobre as condições do trânsito a cada cruzamento. Caso a rua seja mão única apenas um dos campos com a situação do trânsito é utilizado, ficando o outro vazio. Não há necessidade de que o sentido crescente dos segmentos seja o mesmo sentido do fluxo dos carros, dessa forma, é possível que tanto o campo “Estado1”, como o campo “Estado2” não seja preenchido.

Adotou-se que o campo “Estado1” mostra a situação da rua no sentido crescente dos segmentos, e o “Estado2” no sentido decrescente. Para que ambos os campos com informação do trânsito represente o mesmo segmento quando estiverem se referenciando à mesma rua e cruzamento, ou seja, na mesma linha da tabela, convencionou-se que o “Estado1” mostra a informação do trânsito na rua chegando ao determinado cruzamento. Já o “Estado2” apresenta a informação do trânsito nessa rua ao se afastar do mesmo cruzamento.

Dessa forma, consegue-se garantir a coerência entre os campos com as informações sobre o trânsito (“Estado1” e “Estado2”) e o campo “Segmento”. O mesmo não aconteceria caso fosse adotado o padrão de informar a condição do trânsito sempre ao se aproximar ou ao se afastar do determinado cruzamento.

O campo “Segmento” é fundamental para a tabela Trânsito, pois sem este campo não haveria informações sobre a sequência correta dos cruzamentos das diversas ruas. Isso seria um grande problema para o projeto, pois as informações sobre o trânsito em uma determinada rua seriam passadas ao usuário fora da

seqüência correta dos cruzamentos.

3.3.1.1 Exemplo de Informações na Tabela Trânsito

A Figura 7 mostra uma situação hipotética da situação do trânsito em uma rua e clarifica a estrutura dessa tabela do banco de dados.



Figura 7 - Situação do trânsito esquematizada em uma rua

A rua mostrada possui fluxo de veículos nos dois sentidos, portanto os campos “Estado1” e “Estado2” são utilizados. O campo “Estado1” informa a condição do trânsito no sentido crescente dos segmentos. Já o campo “Estado2” mostra a situação no sentido decrescente. Caso a rua fosse mão única um dos campos ficaria nulo.

A Tabela 1 mostra como é preenchida a tabela Trânsito para mostrar as informações da situação apresentada na Figura 7.

O trânsito no Traffic Monitor foi classificado de acordo com as seguintes categorias: Livre, Lento, Congestionado e Bloqueado.

Nome	Cruzamento	Estado1	Estado2	Segmento
Nome da Rua	Cruzamento 1	Livre	Congestionado	1
Nome da Rua	Cruzamento 2	Livre	Lento	2
Nome da Rua	Cruzamento 3	Bloqueado	Lento	3

Tabela 1 - Dados na tabela Trânsito

Vale ressaltar que com essa estrutura de banco de dados as condições do trânsito em ambos os sentidos para qualquer segmento ficam na mesma linha da tabela. Porém, o campo “Estado1” é a situação do trânsito para chegar ao cruzamento correspondente, enquanto o campo “Estado2” mostra a condição do trânsito ao se afastar do mesmo cruzamento, como mencionado anteriormente.

Para fins didáticos, ficou decidido no projeto, que a tabela Trânsito conteria informações simuladas do trânsito de uma região de Moema. A região escolhida está representada na Figura 8 e é delimitada pelas seguintes avenidas: Av. dos Bandeirantes, Av. Ibirapuera, Av. Indianópolis e Av. Moreira Guimarães.

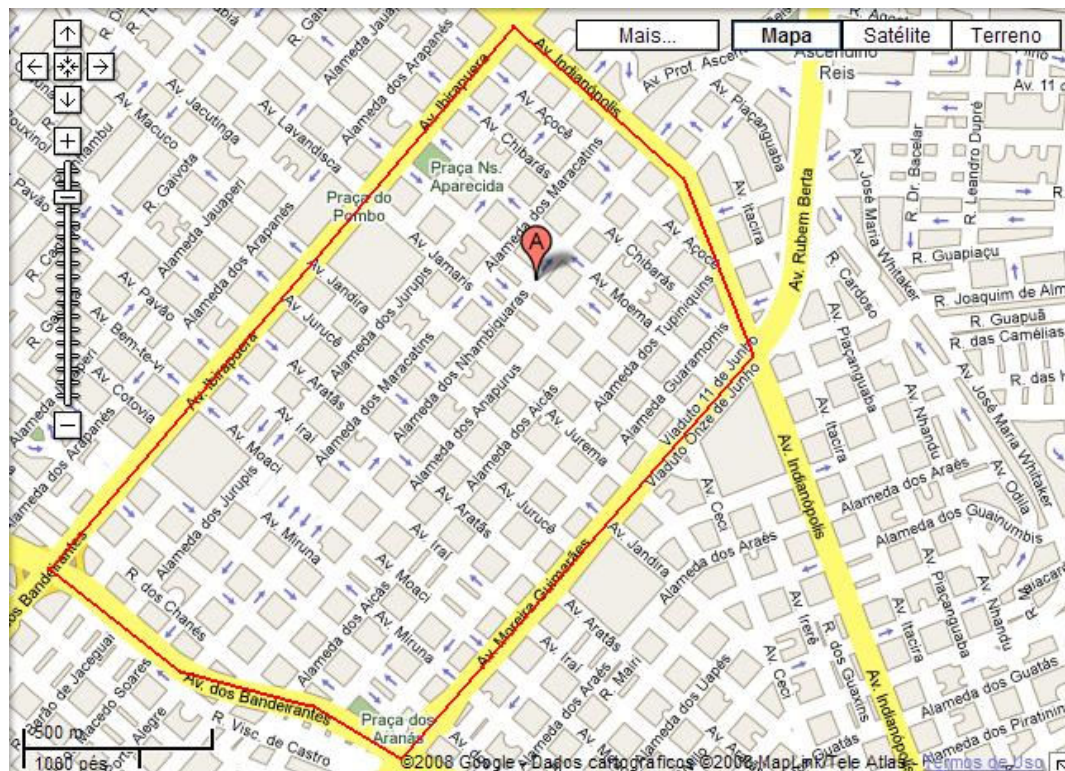


Figura 8 - Região mapeada na tabela Trânsito

3.3.1.2 Consulta de Informações sobre o Trânsito

Como exemplo, uma consulta desenvolvida para pesquisar a condição de uma determinada rua entre dois determinados cruzamentos é apresentada a seguir.

Para realizar essa consulta depende-se de três parâmetros, que são: a rua a ser pesquisada, o cruzamento inicial e o cruzamento final do trecho.

Com os três parâmetros, consegue-se obter, através de uma consulta, qual o número do segmento inicial e o segmento final do trecho que se precisa pesquisar. Para consultar o número do segmento inicial utiliza-se o seguinte código SQL:

```
“SELECT Segmento
FROM Transito
WHERE Nome=Nome_Rua And Cruzamento=Cruz_Inicial”
```

Com uma consulta similar à mostrada acima, apenas trocando-se o cruzamento inicial pelo cruzamento final, obtém-se o número do segmento final.

Caso o segmento inicial seja menor que o segmento final a pesquisa deverá ser feita no sentido crescente dos segmentos, e deve-se selecionar o campo “Estado1” para informar a condição do trânsito. Para tal consulta utiliza-se o seguinte código:

```
“SELECT Cruzamento, Estado1
FROM Transito
WHERE Nome=Nome_Rua And Segmento>Seg_Inicial And Segmento
<= Seg_Final”
```

Porém, se o segmento inicial for maior que o segmento final a pesquisa deverá ser feita no sentido decrescente dos segmentos, e, nesse caso, deve-se selecionar o campo “Estado2” para informar a condição do trânsito. Deve-se ordenar a consulta por ordem decrescente de segmentos, para que se tenha a informação do trânsito no sentido correto do fluxo de veículos. Para essa consulta utiliza-se o seguinte código:

```
“SELECT Cruzamento, Estado2
FROM Transito
WHERE Nome=Nome_Rua And Segmento>Seg_Final And Segmento
```

<= Seg_Inicial
ORDER BY Segmento DESC”

Através das consultas mostradas obtém-se a condição do trânsito em uma determinada rua a cada cruzamento. Para pesquisar a condição em uma rota, realiza-se a mesma consulta apresentada acima, porém, em todas as ruas que serão percorridas para se chegar ao destino desejado.

3.3.2 Tabela Usuário

Trata-se de uma tabela extremamente simples, com apenas três campos:

- **idUsuario:** chave-primária, para identificação do usuário;
- **Login:** nome do usuário;
- **Senha:** código de acesso do usuário.

Admitiu-se que os novos usuários, ao comprarem os serviços fornecidos pelo Traffic Monitor receberiam o seu *login* e senha que seriam manualmente inseridos no Banco de Dados.

A Tabela 2 mostra como é a tabela Usuário.

idUsuario	Login	Senha
1	Gustavo	1234
2	Vitor	4321

Tabela 2 - Dados na tabela Usuário

3.3.2.1 Consulta na Tabela Usuário

Para conseguir utilizar todas as funções do Traffic Monitor é necessário *logar* no sistema. É pedido ao usuário um *login* e uma senha logo que o *software* é inicializado.

Uma consulta é feita na Tabela Usuário, com os dados fornecidos pelo cliente

(login e senha), para encontrar a identificação do usuário (idUsuario) correspondente. O código utilizado para essa consulta é o seguinte:

```
“SELECT idUsuario
FROM Usuário
WHERE Login=Login and Senha=Senha”
```

Caso a busca não retorne nenhum valor o sistema informa que a senha e/ou usuário estão incorretos. O acesso só será permitido se o “idUsuario” encontrado for maior do que zero, pois esse campo será sempre um número positivo.

3.3.3 Tabela Rua

A tabela Rua possui as informações de todos os trechos salvos por todos os usuários. Para definir um trecho são necessários três parâmetros, o nome da rua, o cruzamento inicial e o cruzamento final. Cada usuário pode salvar quantas ruas desejar.

A tabela Rua possui os seguintes campos:

- **idUsuario:** identificação do usuário que salvou o trecho;
- **Trecho:** nome do trecho (será escolhido usuário);
- **Nome_Rua:** nome da rua;
- **Cruzamento_Inicial:** nome da rua do cruzamento inicial do trecho;
- **Cruzamento_Final:** nome da rua do cruzamento final do trecho;
- **idRuaSalva:** chave-primária de identificação do trecho salvo.

3.3.3.1 Salvar Trecho

Após o usuário escolher a rua (e os cruzamentos) a qual deseja consultar a situação do trânsito, ele tem a alternativa de salvar o trecho selecionado. Para isso, escolhe um nome para o trecho e o mesmo será armazenado na tabela Rua do Banco de Dados através do seguinte código:

```

“INSERT INTO Rua
VALUES
(user, Trecho, Nome_Rua, Cruz_In, Cru_Fin)”

```

O usuário pode ter quantos trechos desejar como favoritos. As informações preenchidas na tabela de ruas ficam armazenadas em variáveis globais do sistema.

A Tabela 3 mostra como fica preenchida a tabela Rua.

idUsuario	Trecho	Nome_Rua	Cruzamento_Inicial	Cruzamento_Final	idRuaSalva
1	Maracatins	Al. dos Maracatins	Av. Moema	Av. Miruna	1
2	Jandira	Av. Jandira	Al. dos Aicás	Al. dos Jurupis	2

Tabela 3 - Dados na tabela Rua

3.3.3.2 Consultar Trecho Salvo

Ao escolher a opção de ver um trecho salvo, o sistema faz uma busca na tabela Rua para ver quais são os trechos salvos pelo usuário. Para isso é utilizado o seguinte código:

```

“SELECT Trecho
FROM Rua
WHERE idUsuario=user
ORDER BY Trecho”

```

O nome de todos os trechos salvos pelo usuário é mostrado no visor do celular, e ele pode escolher qual deseja consultar.

Em seguida, com o nome do trecho escolhido, o sistema faz uma busca na mesma tabela para ver qual o nome da rua e os cruzamentos que o usuário deseja consultar. Essa consulta é feita através do seguinte código:

```

“SELECT Nome_Rua, Cruzamento_Inicial, Cruzameto_Final
FROM Rua
WHERE idUsuario=user AND Trecho=Nome_Trecho”

```

A consulta retorna tudo o que se precisa para realizar uma pesquisa em uma

rua (nome da rua e cruzamentos inicial e final). Com isso, o sistema chama o método de pesquisar rua e informa a condição do trânsito ao usuário.

3.3.3.3 Excluir Trecho

O usuário tem a possibilidade de excluir os seus trechos que não são mais utilizados. Para isso, ele pode consultar todos os trechos salvos e selecionar qual deve ser excluído.

Com o nome do trecho, o mesmo é excluído da tabela Rua através do seguinte código:

```
“DELETE FROM Rua
WHERE idUsuario=user AND Trecho=Nome_Trecho”
```

Após a exclusão do trecho é realizada novamente a consulta para encontrar os trechos salvos do usuário. Dessa forma, são mostrados todos os seus trechos já sem o trecho excluído.

3.3.4 Tabela Rota

A tabela Rota contém todas as rotas salvas por todos os usuários do Traffic Monitor. Não há limite de rotas para ser salva por cada um dos usuários.

A tabela Rotas contém sete campos, são eles:

- **idUsuario:** identificação do usuário que salvou a rota;
- **Nome_Rota:** nome da rota que será escolhido usuário;
- **Rua_Inicial:** nome da rua inicial da rota;
- **N_In:** número inicial da rota;
- **Rua_Final:** nome da rua final da rota;
- **N_Fin:** número final da rota;
- **idRotaSalva:** chave-primária de identificação da rota salva.

A Tabela 4 apresenta um exemplo de como é preenchida a tabela Rota.

idUsuario	Nome_Rota	Rua_Inicial	N_In	Rua_Final	N_Fin	idRotaSalva
1	Casa-médico	Av. Divino Salvador	398	Av. Açocê	340	1
2	Trabalho-dentista	Av. Miruna	420	Av. Aratãs	120	2

Tabela 4 - Dados na tabela Rota

3.3.4.1 Salvar Rota

Após o usuário fazer a sua rota ele tem a opção de salvar a mesma. Para isso o sistema solicita que ele escreva o nome desejado para a rota que será salva. Essa rota é armazenada na tabela Rota através do seguinte código:

```
“INSERT INTO Rota
VALUES
(user, Nome_Rota, Rua_In, N_In, Rua_Fin, N_fin)”
```

Tudo que será preenchido na tabela fica armazenado em variáveis globais do sistema.

3.3.4.2 Consultar Rota Salva

O usuário pode escolher a opção de consultar suas rotas salvas. O sistema faz uma busca na tabela Rotas com o número de identificação do cliente (idUsuario) para obter todas as rotas que ele possui salvas. Essa consulta é feita através do seguinte código:

```
“SELECT Nome_Rota
FROM Rotas
WHERE idUsuario=user
ORDER BY Nome_Rota”
```

O nome de todas as rotas que o usuário possui aparece na tela do dispositivo

móvel e ele pode selecionar a rota que desejar.

Em seguida, com o nome da rota desejada é realizada uma nova consulta na mesma tabela para obter o endereço inicial e final da rota selecionada. Essa consulta é feita através da seguinte consulta:

```
“SELECT Rua_Inicial, N_In, Rua_Final, N_Fin  
FROM Rotas  
WHERE idUsuario=user AND Nome_Rota=Rota”
```

Essa consulta encontra tudo o que se precisa para se definir uma rota (rua e número do endereço inicial e final). Com isso, é chamada a alternativa de calcular rota do sistema e todos os campos de informações dos endereços inicial e final já estarão preenchidos com as informações obtidas através da consulta.

3.3.4.3 Excluir Rota

O usuário pode excluir uma das suas rotas salvas. Para isso ele deve consultar todas as suas rotas, selecionar aquele que ele deseja excluir.

Com o nome da rota que deve ser excluída, o sistema apaga a mesma através do seguinte código:

```
“DELETE FROM Rota  
WHERE idUsuario=user AND Nome_Rota=Nome_Rota”
```

3.3.5 Diagrama de Entidade-Relacionamento

A Figura 9 mostra o diagrama de Entidade-Relacionamento do Banco de Dados do servidor do sistema.

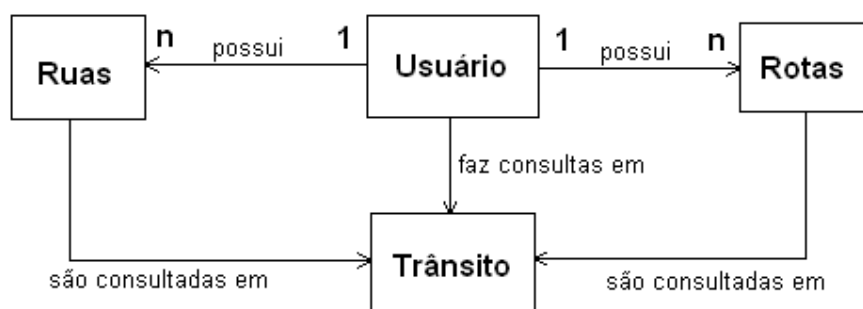


Figura 9 - Diagrama de Entidade-Relacionamento

4 CASOS DE USO

Neste item, serão apresentados detalhadamente os possíveis casos de uso do sistema.

i) Fazer Login

Descrição:

Usuário digita *login* e senha e caso os mesmos estejam corretos, as funções do programa são exibidas.

Ator:

Usuário

Pré-condições:

O dispositivo precisa estar ligado, conectado à Internet e com o Traffic Monitor rodando.

Fluxo Principal:

1. Usuário entra no sistema
2. Sistema exibe tela de *login*
3. Usuário digita os seus dados
4. Caso os dados estejam corretos, o sistema exibe os botões de acesso. Caso estejam incorretos, o sistema exibe uma mensagem de erro.

Ilustrações:

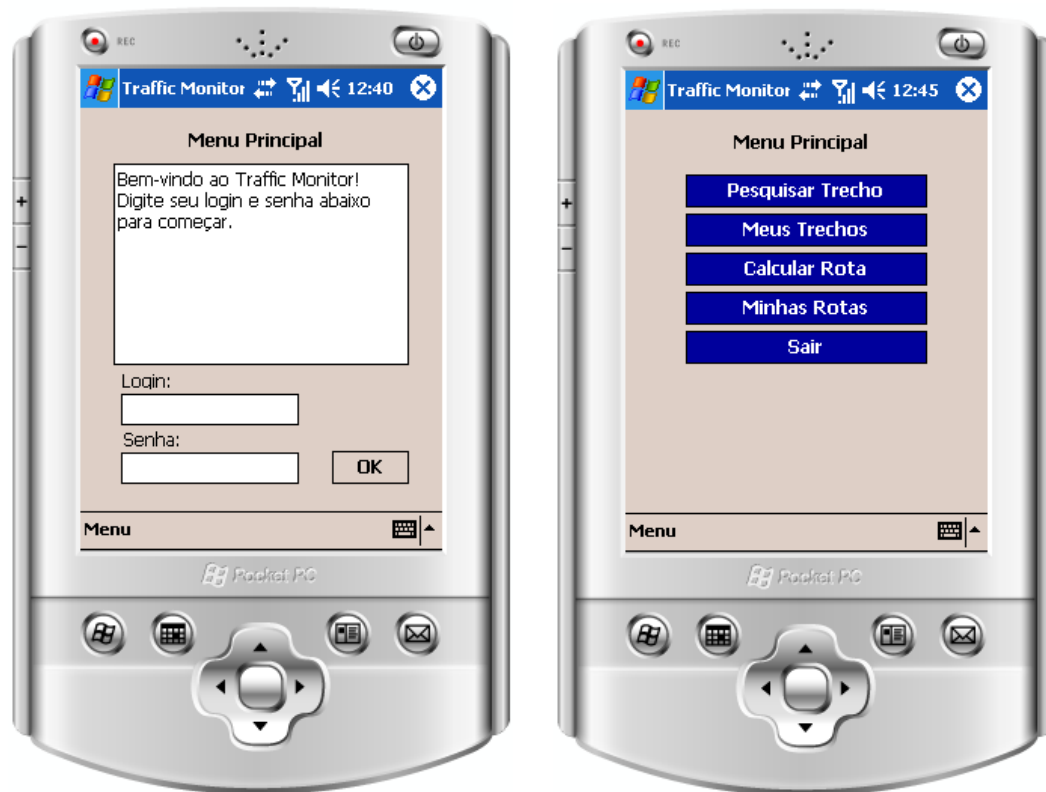


Figura 10 - Login no sistema

ii) Pesquisar Trecho

Descrição:

Usuário escreve nome da rua que deseja consultar e informa os cruzamentos inicial e final, delimitando um trecho a ser analisado. Sistema devolve condição do trânsito a cada cruzamento do trecho.

Ator:

Usuário.

Pré-condições:

O usuário precisa estar *logado*.

Fluxo Principal:

1. Usuário clica no botão “Pesquisar Trecho”.

2. Sistema exibe campos para *input* dos parâmetros.
3. Usuário preenche os campos.
4. Usuário clica no botão “Pesquisar”.
5. Sistema devolve tabela com condição do trânsito a cada cruzamento.

Fluxo Alternativo:

- 4a. Usuário escolhe salvar o trecho, clicando no botão “Salvar Trecho”.
- 5a. Sistema exibe janela para *input* do nome do trecho.
- 6a. Sistema exibe janela de confirmação.

Ilustrações:



Figura 11 - Pesquisar Trecho

iii) Consultar Trecho Salvo

Descrição:

Usuário escolhe entre seus trechos salvos e o sistema devolve a condição do trânsito a cada cruzamento do mesmo.

Ator:

Usuário.

Pré-condições:

O usuário precisa estar *logado* e ter trechos salvos.

Fluxo Principal:

1. Usuário clica no botão “Meus Trechos”.
2. Sistema exibe lista com todos os trechos salvos.
3. Usuário seleciona trecho desejado.
4. Usuário clica no botão “Consultar”.
5. Sistema exibe tabela com condição do trânsito a cada cruzamento do trecho selecionado.

Fluxo Alternativo:

- 4a. Usuário clica no botão “Excluir”.
- 5a. Sistema exclui o trecho selecionado.
- 6a. Sistema exibe trechos remanescentes.

Ilustrações:



Figura 12 - Consultar Trecho

ii) Pesquisar Rota

Descrição:

Usuário escreve nome e número do endereço inicial e final da rota que deseja consultar. Sistema acessa *Web Services* da MapLink e devolve ao usuário a rota calculada e opções para analisá-la detalhadamente.

Ator:

Usuário.

Pré-condições:

O usuário precisa estar *logado*.

Fluxo Principal:

1. Usuário clica no botão “Calcular Rota”.

2. Sistema exibe campos para *input* dos parâmetros.
3. Usuário preenche os campos.
4. Usuário clica no botão “Calcular Rota”.
5. Sistema exibe descrição passo-a-passo da rota.
6. Usuário clica no botão “Rota Detalhada”.
7. Sistema exibe *Menu* de opções.
8. Usuário clica no botão “Ver condições do trânsito.
9. Sistema exibe o início da rota.
10. Usuário clica no botão “Próxima” até o fim da rota.
11. Sistema vai exibindo a condição do trânsito em cada rua da rota.

Fluxo Alternativo “a”:

- 4a. Usuário escolhe salvar a rota, clicando no botão “Salvar Rota”.
- 5a. Sistema exibe janela para *input* do nome da rota.
- 6a. Sistema exibe janela de confirmação.

Fluxo Alternativo “b”:

- 8b. Usuário clica no botão “Ver Mapa”.
- 9b. Sistema exibe mapa com a rota traçada.

Fluxo Alternativo “c”:

- 12c. Usuário clica no botão “Menu”.
- 13c. Sistema exibe opções.
- 14c. Usuário clica no botão “opções de rota”
- 15c. Sistema volta ao Menu de opções.

Ilustrações:



Figura 13 - Calcular rota



Figura 14 - Opções de rota detalhada



Figura 15 - Voltar às opções de rota

iii) Consultar Rota Salva

Descrição:

Usuário escolhe entre suas rotas salvas e o sistema traça a rota desejada.

Ator:

Usuário.

Pré-condições:

O usuário precisa estar *logado* e ter rotas salvas.

Fluxo Principal:

1. Usuário clica no botão “Minhas Rotas”.
2. Sistema exibe lista com todos os trechos salvos.
3. Usuário seleciona rota desejada.

4. Usuário clica no botão “Consultar”.
5. Sistema mostra a tela de Calcular Rota com os campos preenchidos.
6. Usuário clica em “Calcular Rota”.

Fluxo Alternativo:

- 4a. Usuário clica no botão “Excluir”.
- 5a. Sistema exclui a rota selecionada.
- 6a. Sistema exhibe rotas remanescentes.

Ilustrações:

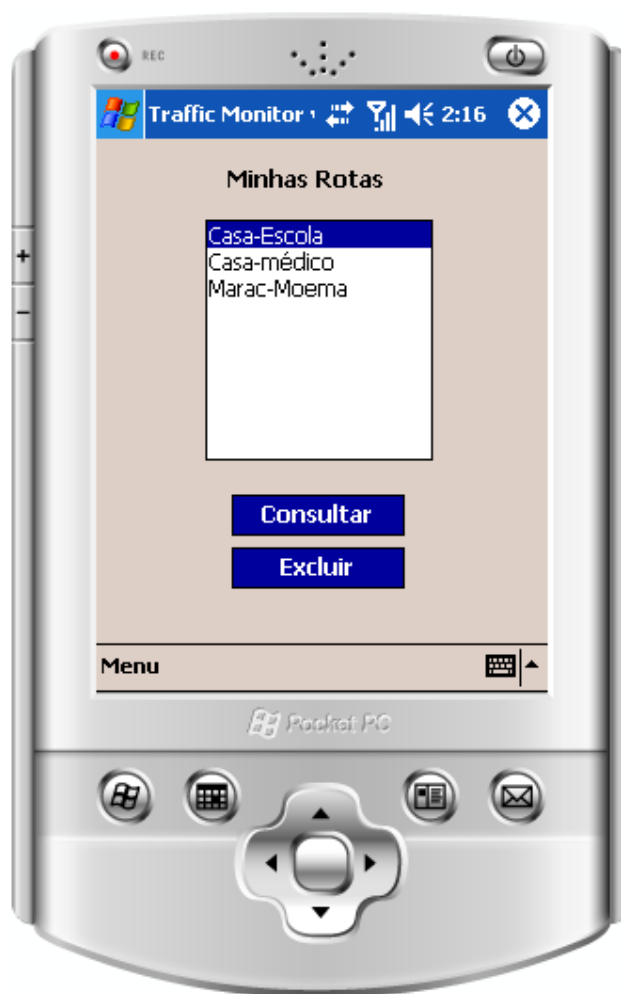


Figura 16 - Consultar rota

5 CONCLUSÕES

Neste projeto, tinha-se como principal objetivo o aprendizado de programação para dispositivos móveis e tal meta foi alcançada, pois foi possível criar um *software* inteiramente novo, com características previamente definidas pela equipe, e executá-lo em um *Smartphone* real.

Além disso, durante a realização desse projeto, pôde-se conhecer e aprender a trabalhar com duas ferramentas muito poderosas: o *Web Service* e o Microsoft Visual Studio.

O *Web Service* é uma ferramenta que vêm sendo utilizada cada vez mais e parece ser uma nova tendência para os programadores. Sua praticidade é o que o torna tão importante. Através dele, muitos programadores podem evitar horas de desenvolvimento repetitivo, pois, ao invés de reprogramar uma rotina, podem-se acessar métodos prontos de *Web Services* publicados na rede. Como exemplos, podem ser citadas rotinas de cálculo de rotas, previsão do tempo, buscas na Internet, entre outras. Dessa forma, muitos *webmasters* podem incluir em seus sites serviços como esses sem precisar desenvolvê-los novamente, mas apenas “chamando” métodos já disponíveis na rede.

Como é fácil imaginar, essa tecnologia abre espaço para muitos programadores, que podem desenvolver métodos úteis a outros *webmasters* ou desenvolvedores e cobrar pelo seu compartilhamento, como é o caso feito pela MapLink, que disponibiliza serviços de cálculos de rotas rodoviárias para serem utilizados por outros programas ou sites e cobra uma taxa por cada chamada de função.

Outra importante contribuição desse serviço é a facilidade de se centralizar bancos de dados de grande interesse como CEPs e CPFs. Seria inviável se todos os sistemas que precisam fazer consultas a essas informações tivessem que possuir seu próprio banco de dados. Com a tecnologia dos *Web Services*, consegue-se ter apenas um banco atualizado, com inúmeros programas acessando suas informações. Dessa maneira, consegue-se fazer com que apenas uma empresa ou equipe fique responsável pela atualização desses dados.

No caso de dispositivos móveis, fica evidente a importância desse tipo de

serviço, uma vez que tais aparelhos geralmente possuem baixas capacidades de processamento. Assim, pode-se deixar o trabalho “pesado” para ser executado por uma máquina mais robusta, que recebe as informações necessárias e devolve a resposta já processada ao dispositivo.

No caso do Traffic Monitor, a existência desse tipo de tecnologia permitiu sua programação e o tornou muito mais completo. Seria praticamente inviável manter um banco de dados completo sobre o trânsito dentro do próprio dispositivo e fazer consultas tão elaboradas como as que o projeto demandava. Além de que as rotinas para cálculo de rotas e geração de mapas, fornecidas para a equipe pela MapLink seriam, por elas só, temas completos para dissertações de conclusão de curso, por serem rotinas muito complexas que foram primeiramente desenvolvidas por equipes numerosas e altamente qualificadas.

Com relação ao desenvolvimento do projeto, o Microsoft Visual Studio foi ferramenta fundamental. Foi possível utilizar muitas de suas funcionalidades, tais como criação de interfaces gráficas, criação e publicação de *Web Services*, desenvolvimento e compilação de programas para dispositivos móveis e testes através do simulador de *Pocket PC*. A cada etapa do projeto, foi possível conhecer novas funções desse potente programa e perceber como ele facilita e auxilia diversos programadores do mundo inteiro.

Através do Visual Studio, foi possível programar em linguagem Visual Basic, que é uma linguagem de alto nível bastante objetiva, e compilar o *software* diretamente no *Smartphone* com Windows Mobile, sem a preocupação de alterar configurações no computador ou no aparelho.

Como outra fonte de motivação para o trabalho, estava a vontade da equipe de criar algo que pudesse ajudar os motoristas de São Paulo a se informar e, eventualmente, desviar do caótico trânsito da cidade.

Um ponto que chamou bastante a atenção foi o interesse apresentado pelas pessoas em um *software* como o Traffic Monitor, o que serviu para alertar a equipe à gravidade da situação do trânsito e, ao mesmo tempo, do espaço que um programa como esse poderia atingir no mercado.

Finalmente, é possível afirmar que se conseguiu perceber a grande importância do desenvolvimento de um projeto como esse. Isso ficou evidente pelo grande número de novas ferramentas aprendidas e pelo forte interesse da população. Provavelmente, isso se deve ao fato de que, cada vez mais, as pessoas

estão utilizando dispositivos móveis e estes condensam mais funções a cada lançamento. A tendência é que, no futuro, um pequeno aparelho possa reunir todas as funções móveis desejadas por um usuário, como máquina fotográfica, celular, agenda, navegador de Internet, TV, rádio, GPS, entre outras. Isso mostra que, aplicativos para dispositivos móveis se tornarão ainda mais difundidos e *softwares* cada vez mais personalizados serão necessários para atender a demanda do mercado.

Fica muito claro, então, que o aprendizado sobre programação para dispositivos móveis é de grande importância, assim como é o estudo sobre possíveis soluções para o grave problema do trânsito nas cidades. Portanto, a idéia, praticada nesse projeto, de se aliar os dois temas se revelou altamente valiosa.

6 REFERÊNCIAS

MSDN. Tutorial de Aplicação Móvel para Celular

Disponível em <<https://www.msdnbrasil.com.br/secure/tutoriais/celular/default.aspx>>.

Acesso em: 12 abril 2008.

MSDN. Ensina a usar e criar *Web Services* ASP.NET

Disponível em <<http://msdn.microsoft.com/en-us/library/8wbhsy70.aspx>>. Acesso em: 10 maio 2008.

NETSCAPE DEVEDGE. Informações sobre *Web Services*

Disponível em <http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html>. Acesso em: 10 maio 2008.

UOL ÚLTIMAS NOTÍCIAS. Números sobre o trânsito de São Paulo

Disponível em

<<http://noticias.uol.com.br/ultnot/especial/2008/transito/2008/04/24/ult5848u12.jhtm>>

Acesso em: 20 maio 2008.

MAPLINK WEB SERVICE. Serviços para cálculos de rota e obtenção de mapas.

Disponível em <<http://doc.maplink2.com.br>>. Acesso em: 27 maio 2008.

INTERNET INFORMATION SERVICES (IIS). Plataforma Web para hospedar

aplicações e *Web Services*. Disponível em <<http://www.iis.net>>. Acesso em 10 agosto 2008.

MACORATTI.NET. Acessando o MySQL 5.0 via código.

Disponível em <http://www.macoratti.net/07/10/vbn_mysql.htm>. Acesso em: 16 agosto 2008.

AKHUNE's WEBLOG. Configuração do Emulador de *Pocket PC*.

Disponível em <<http://blogs.msdn.com/akhune/archive/2005/11/16/493329.aspx>>.

Acesso em: 24 agosto 2008.

JÚNIOR, M. P. B. **Desenvolvendo Web Services**: Guia Rápido usando Visual Studio .NET com Banco de dados Sql Server. Rio de Janeiro. Ciência Moderna, 2005. 128 p.

WIGLEY, A. **Microsoft .NET Framework**. Washington, EUA. Microsoft Press, 2003.
888 p.

APÊNDICE A

Especificações do Datasheet do HTC Touch GSM

	Touch GSM
Processor	TI's OMAP™ 850, 201 MHz
Operating System	Windows Mobile® 6 Professional
Memory	ROM: 128 MB
	SDRAM: 64 MB DDR
Dimensions	99.9mm (L) X 58mm (W) X 13.9mm (T)
Weight	112g with battery
Display	2.8" LCD touch screen with backlight
	240 X 320 dots resolution with 65,536 colors
Network	GSM/GPRS/EDGE Tri-band: 900, 1800, 1900 (The device will operate on frequencies available from the cellular network.)
Device Control	HTC TouchFLO™
	5-Way navigation control
Connectivity	Bluetooth® 2.0
	Wi-Fi® IEEE 802.11 b/g
	HTC ExtUSB™ (11-pin mini-USB and audio jack in one)
Camera	2 megapixel CMOS color camera
Audio	Built-in microphone and speaker
	Windows Media Player supported formats: AAC, AAC+, eAAC+, MP3, WMA, WAV, QCELP, MPEG4, AMR-NB, AMR-WB
Battery	Rechargeable Li-Ion battery
	Capacity: 1100 mAh
	Standby time: Up to 200 hours
	Talk time: Up to 5 hours
Expansion Slot	microSD™ memory card (SD 2.0 compatible)
AC Adapter	Voltage range/frequency: 100 ~ 240V AC, 50/60Hz
	DC output: 5V and 1A

APÊNDICE B

Detalhamento das classes do Traffic Monitor

The image displays three screenshots of the Visual Studio Class Browser, showing the structure of three classes: **frmPrinc**, **frmPesqRua**, and **frmCalcRota**.

frmPrinc Class:

- Fields:** btnCalcRota, btnMeusTrechos, btnMinhasRotas, btnok, btnPesqRua, btnSair, components, ip, lblLogin, lblSenha, mainMenu1, MenuItem1, MenuItem2, MenuItem3, MenuItem4, MenuPrinc, txtIntro, txtlogin, txtsenha, user.
- Methods:** btnCalcRota_Click, btnMeusTrechos_..., btnMinhasRotas_..., btnok_Click, btnPesqRua_Click, btnSair_Click, Dispose, frmPrinc_Load, InitializeCompon..., MenuItem2_Click, MenuItem3_Click, MenuItem4_Click.

frmPesqRua Class:

- Fields:** btnPesq, btnSalvarRua, components, DataGrid1, DataGridTableSty..., DataGridTextBox..., FrmPesqRuaBindi..., Label1, lblCruz1, lblNomeRua, mainMenu1, MenuItem1, MenuItem2, MenuItem3, MenuItem4, PesqRua, trecho, txtCruzF, txtCruzI, txtNomeRua.
- Methods:** btnNovaPesq_Click, btnPesq_Click, btnSalvarRua_Click, btnVoltar_Click, Dispose, InitializeCompon..., MenuItem2_Click, MenuItem3_Click, MenuItem4_Click.

frmCalcRota Class:

- Fields:** btnCalcRota, btnRotaDet, btnSalvarRota, com, components, des, desc_rota, Label1, Label2, Label3, lblCalcRota, lblEndFin, lblEndIn, mainMenu1, MenuItem1, MenuItem2, MenuItem3, MenuItem4, MenuPrinc, ro, rota, rs, txtNFin, txtNIn, txtRota, txtRuaFin, txtRuaIn.
- Methods:** btnCalcRota_Click, btnNovaPesq_Cli..., btnRotaDet_Click, btnSalvarRota_Cli..., btnVoltar_Click_1, Dispose, InitializeCompon..., MenuItem1_Click, MenuItem2_Click, MenuItem3_Click, MenuItem4_Click.

frmRotaDet
 Class
 → Form

Fields

- a
- btnCondTrans
- btnDescRota
- btnProxRua
- btnRuaAnt
- btnVerMapa
- com_2
- components
- cont
- DataGrid2
- des_2
- i
- lblCom
- lblDes
- lblRotaDet
- mainMenu1
- MenuItem1
- MenuItem2
- MenuItem3
- MenuItem4
- pctMapa
- txtRota2

 Methods

- btnCondTrans_Click
- btnDescRota_Click
- btnProxRua_Click
- btnRuaAnt_Click
- btnVerMapa_Click
- btnVoltar_Click
- btnVoltarRota_Click
- Dispose
- InitializeCompon...
- MenuItem2_Click
- MenuItem3_Click
- MenuItem4_Click

frmRotas
 Class
 → Form

Fields

- btnConsultar
- btnExRota
- components
- lblRotas
- ListBox1
- mainMenu1
- MenuItem1
- MenuItem2
- MenuItem3

 Methods

- btnConsultar_Click
- btnConsultar_Click...
- btnExRota_Click
- Dispose
- Form9_Load
- InitializeCompon...
- MenuItem2_Click
- MenuItem3_Click

frmTrechos
 Class
 → Form

Fields

- btnConsultar
- btnExTrecho
- components
- lblTrechos
- ListBox1
- mainMenu1
- MenuItem1
- MenuItem2
- MenuItem3
- parametros

 Methods

- btnConsultar_Click
- btnExTrecho_Click
- Dispose
- frmTrechos_Load
- InitializeCompon...
- MenuItem2_Click
- MenuItem3_Click

