


**UM SISTEMA PARA TELEMÁTICA DE TRANSPORTE UTILIZANDO
DISPOSITIVOS MÓVEIS**

DIEGO MÁRCIO OLIVEIRA BEZERRA

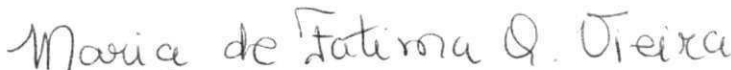
Dissertação Aprovada em 07.11.2008



ANGELO PERKUSICH, D.Sc., UFCG
Orientador



HYGGO OLIVEIRA DE ALMEIDA, D.Sc., UFCG
Componente da Banca



MARIA DE FÁTIMA QUEIROZ VIEIRA, Ph.D., UFCG
Componente da Banca

CAMPINA GRANDE - PB
NOVEMBRO - 2008

Um Sistema para Telemática de Transporte Utilizando Dispositivos Móveis

Diego Márcio Oliveira Bezerra

Dissertação de Mestrado submetida à Coordenadoria do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Campina Grande - Campus de Campina Grande como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências no Domínio da Engenharia Elétrica.

Área de Concentração: Engenharia da Computação

Angelo Perkusich, Dr.
Orientador

Campina Grande, Paraíba, Brasil
©Diego Márcio Oliveira Bezerra, novembro 2008

Resumo

A quantidade de problemas relacionados ao trânsito cresce à medida que aumenta a quantidade de automóveis trafegando nas ruas das cidades. O atual estágio de desenvolvimento das tecnologias de dispositivos eletrônicos permite que sistemas sejam criados para reduzir esses problemas. Muitos estudos são realizados na tentativa de encontrar as melhores maneiras de reduzir problemas como engarrafamento nas vias e emissão de poluentes. Uma das áreas mais estudadas é a telemática de transporte, que é a união entre sistemas de tecnologia da informação modernos e telecomunicações, aplicadas à indústria dos transportes. Nesse trabalho, é proposta uma infra-estrutura que visa reduzir os problemas de custo de implantação de sistemas de telemática de transporte. A arquitetura dessa infra-estrutura foi concebida de forma que o sistema pudesse ser extensível. Para validação da infra-estrutura, um software foi desenvolvido para telemática de transporte.

Abstract

The quantity of problems related to transit grows as the amount of cars transiting the city's streets increase. The current stage of development of electronic devices technology allows the creation of systems designed to reduce these problems. Many studies are performed in an attempt to find the best ways to reduce problems like traffic jam on roads and emission of pollutants. One of the most studied areas is the transport telematics, which is the joint of modern systems of information technology and telecommunications, applied to transport industries . In this dissertation, it is proposed an infrastructure to reduce the cost problems of introduction of transport telematics at automobiles. This infrastructure architecture was conceived in ways to permit the system to be extensible. To validate the infrastructure, a software was developed for transport telematics.

Agradecimentos

Agradeço primeiramente aos meus pais, José e Rita, e à minha avó, Severina, por me ajudarem moralmente e financeiramente durante a minha vida, e à minha irmã, Fernanda, por me incentivar no âmbito musical, sempre mentido quando diz que eu sou afinado.

Agradeço aos meus amigos: Marcos Morais, por ter dado a idéia inicial desse trabalho; Danilo Santos, por ter me ajudado no início da concepção desse trabalho com toda sua paciência; Yuri Gomes, por ter me ajudado durante a concepção desse trabalho com toda sua paciência; e Camila Souto, por ter me ajudado no fim da concepção desse trabalho com todo seu desespero, achando que não daria certo. Também agradeço à minha namorada, Raíssa, por ter me ajudado nas correções desse trabalho, ou seja, me ajudando a não assassinar o português (ele só entrou em coma profundo).

Agradeço a todos os meus amigos do laboratório embedded e aos meus orientadores, Angelo Perkusich e Hyggo Almeida. Agradeço também à copele e à Angela, por terem ajudado nas orientações.

Agradeço às minhas amigas e orientandas do laboratório embedded, Cecília, Fernanda, Taciana e Joelma. Aos meus amigos Genildo, Yuri e Thiago Onofre, os quais dividiram apartamento comigo e me ensinaram muitas coisas sobre convivências.

Por fim, agradeço aos meus amigos mais próximos Luiz Paulo, Yuri, Breno, Estela Maris, Taísa Felix, Ademar Virgulino, João André, Adelyana, Nathalie, Yasmin, Priscila (Tesouro), Anaíra, Alex, Marília, Bel, Zé, Mariana, Larissa e Lorena por terem participado dos meus momentos de descontração.

índice

1	Introdução	1
1.1	Problemática	3
1.2	Objetivo	4
1.3	Relevância	5
1.4	Estrutura da Dissertação	6
2	Fundamentação Teórica	7
2.1	Telemática	7
2.2	Computação Pervasiva	9
2.3	Computação em Ambientes Automotivos	11
2.4	Computação em Dispositivos Portáteis	14
2.5	Trabalhos Relacionados	17
3	Infra-estrutura para Sistemas de Telemática de Transporte	19
3.1	Visão geral	19
3.2	Módulos que formam a infra-estrutura para construção de sistemas de telemática	20
3.3	Restrições e requisitos	21
3.3.1	Módulo para obtenção de dados do automóvel	21
3.3.2	Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos	22
3.4	Componentes que compõem os módulos da infra-estrutura	23
3.4.1	Módulo para obtenção de dados do automóvel	24
3.4.2	Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos	25
3.4.3	Módulo do servidor remoto	27
4	Estudo de caso	28
4.1	Módulo para obtenção de dados do automóvel	28

4.2	Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos	31
4.3	Módulo do servidor remoto	33
4.4	Interface gráfica com o usuário	34
4.5	Resultados dos testes realizados	38
5	Conclusões e trabalhos futuros	42
	Referências Bibliográficas	44

Lista de Figuras

2.1	Cenário de telemática de transporte	7
2.2	Conector DLC	13
2.3	Exemplo de código de falha	14
3.1	Arquitetura da infra-estrutura proposta.	23
3.2	Módulo para a obtenção de dados do automóvel.	24
3.3	Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos.	25
3.4	Módulo do servidor remoto.	27
4.1	Diagrama UML do software criado para o <i>smartphone</i>	32
4.2	Diagrama UML do software criado para o servidor.	34
4.3	Menu inicial da interface gráfica com o usuário.	34
4.4	Mensagem de erro exibida quando não existe perfil.	35
4.5	Formulário para criação de perfil.	35
4.6	Estado de conexão.	35
4.7	Mensagem de espera para procura de servidor.	36
4.8	Menu para geração de rota.	37
4.9	Rota gerada.	37
4.10	Mapa contendo a localização dos postos de combustível.	38
4.11	Ponto de partida para os testes.	39
4.12	Resultado do primeiro teste.	39
4.13	Resultado do segundo teste.	40
4.14	Resultado do terceiro teste.	40

Capítulo 1

Introdução

Atualmente, é notável o aumento da quantidade de problemas relacionados ao trânsito de veículos em cidades grandes ou sem planejamento estrutural das ruas e avenidas. À medida que aumenta a frota de carros nas ruas das cidades, também aumentam os problemas como, por exemplo, engarrafamentos, falta de estacionamento, desrespeito às leis de trânsito, colisões, buracos nas vias, entre outros. Esses problemas podem ser sanados com o uso das tecnologias vigentes. Existem diversos dispositivos eletrônicos que podem solucionar os problemas supracitados, seja em conjunto com outros dispositivos ou não. Além disso, muitos estudos estão sendo feitos na área que envolve sistemas de tecnologia da informação, comunicação e automóveis, os quais caracterizam a chamada telemática de transporte [1].

Com o uso da telemática de transporte é possível auxiliar o motorista em suas diversas tarefas, incluindo a tarefa mais importante que é dirigir o automóvel. É comum que o motorista execute outras tarefas durante o seu trajeto como, por exemplo, fazer ajustes no som ou parar para solicitar informações de como chegar ao destino desejado. O auxílio a essas tarefas também é provido pela telemática de transportes, ou seja, é possível acessar serviços de multimídia, serviços de VoIP [2] ou serviços de localização utilizando as tecnologias disponíveis atualmente.

Para exemplificar, pode-se citar a situação em que o motorista está trafegando por uma cidade desconhecida e não sabe como chegar ao destino desejado. Uma opção para solucionar esse problema é parar e perguntar a algum habitante da cidade a melhor rota para se chegar ao destino. Porém, esta ação pode ser desaconselhável dado que o habitante pode não conhecer bem a cidade e sugerir o caminho errado ou, dado que o motorista não conhece a cidade, a localidade onde ele parou para conseguir informações pode ser de índice de criminalidade considerado alto pelas entidades de segurança. Outra opção é o uso da telemática de transportes para isso. É possível obter dados de localização atual do automóvel e dados de localização do destino utilizando os sistemas de comunicação

presentes nos diversos dispositivos eletrônicos, que podem ser utilizados na telemática de transporte. Além disso, com a utilização de alguns desses dispositivos, é possível exibir em uma tela o mapa da cidade indicando a localização atual do automóvel, a localização do destino e as rotas disponíveis.

A localização atual do automóvel pode ser obtida a partir de um sistema de localização GPS¹ (do inglês *Global Positioning System*). Porém, para se utilizar outros serviços na área da telemática de transporte, é necessário obter outros dados. Os automóveis atuais possuem sensores e computadores embarcados interligados por uma rede de comunicação, o que possibilita a obtenção de dados. Com os dados obtidos, é possível acessar outros serviços como, por exemplo, o serviço que, após obter o dado referente à velocidade do automóvel, avisa ao motorista quando ele está acima do limite de velocidade de uma determinada via. Para que a obtenção de dados do automóvel seja possível, é necessária a inclusão de um dispositivo eletrônico que possua processador, capacidade de armazenamento e sistemas de comunicação, os quais são necessários para, respectivamente, processar, armazenar e transmitir os dados obtidos. Para isso, existem no mercado computadores portáteis de baixo custo com os requisitos listados acima. Assim, todos os requisitos necessários para a criação de sistemas de telemática de transporte já podem ser contemplados.

Além de prover serviços relacionados ao ambiente interno ao automóvel, chamado de ambiente automotivo neste trabalho, também é possível que os sistemas de telemática de transporte provejam serviços relacionados ao ambiente externo. É possível que haja comunicação entre veículos próximos, permitindo a criação de diversos serviços de multimídia, segurança, entre outros. Como exemplo, pode-se citar a comunicação entre veículos para evitar colisões. Isso pode ser feito com a obtenção e troca dos dados de velocidade, de localização e de acionamento dos freios entre dois ou mais automóveis. Assim, os freios de um automóvel podem ser acionados logo que o automóvel à sua frente tiver os freios acionados bruscamente.

Diversos automóveis possuem computadores de bordo, ou seja, equipamentos integrados que mostram alguns dados do estado atual como, por exemplo, consumo de combustível. Porém, esse equipamento não é encontrado na maioria dos automóveis de baixo custo, ditos automóveis populares. Um dos motivos para que isso ocorra é o alto custo de implantação de um equipamento eletrônico que tem como propósito mostrar algumas informações que, na maioria dos casos, não influenciam criticamente na dirigibilidade. Assim, estas informações não têm grande contribuição no auxílio ao motorista na realização de tarefas. Além disso, esses sistemas não são expansíveis, ou seja, não é possível adicionar novas características e novos serviços.

¹<http://www.gps.gov/> acessado em 31/10/2008

Os sistemas de telemática são diferentes. Além de poder prover serviços que auxiliem os motoristas em suas tarefas, esses sistemas são expansíveis. O uso da telemática de transporte é ideal para solucionar problemas de trânsito ou problemas no ambiente automotivo. Para isto, basta que o automóvel proveja dados referentes ao seu estado atual como, por exemplo, velocidade, nível de combustível e temperatura do motor; que exista algum sistema que forneça dados de localização do automóvel como, por exemplo, o sistema global de posicionamento GPS; e que haja dispositivos eletrônicos disponíveis que possuam sistemas de comunicação e que sejam capazes de processar dados, exibir dados e receber comandos de usuários. Os componentes citados acima como requisitos podem ser facilmente interligados de diversas maneiras, permitindo a criação de sistemas de telemática de transporte diversificados.

1.1 Problemática

Os sistemas de telemática de transporte podem ser a solução para vários problemas do trânsito e é certo que eles auxiliam os motoristas em suas tarefas. Porém, os fabricantes têm dificuldades de incluir esses sistemas em seus automóveis de baixo custo. Isso ocorre por vários motivos. Um dos motivos principais é o custo de implantação de sistemas eletrônicos com capacidade de comunicação e que possuam tela para exibir dados. A inclusão desses dispositivos em carros populares aumentaria o custo de fabricação e, conseqüentemente, o preço do automóvel, diminuindo a procura por parte dos consumidores. Outro motivo é a falta de cooperação entre os fabricantes [3], que procuram fazer seu próprio sistema, dificultando a criação de um padrão para a telemática de transporte, o que permitiria a interoperabilidade e reduziria consideravelmente os custos de desenvolvimento e de inclusão na linha de produção. É notável que itens de série em automóveis sejam mais caros que em lojas de aparelhos eletrônicos como, por exemplo, sistemas de som, que chegam a custar duas vezes mais caro. Outro motivo é que o tempo de vida útil dos automóveis é bem maior que o tempo de vida útil dos dispositivos eletrônicos. Este fato não favorece os fabricantes de automóveis a introduzirem sistemas de telemática de transporte na sua linha de produção de produtos de baixo custo. Não é de forma alguma vantajoso incluir algo em uma linha de produção que aumentará o custo de produção e o valor final do automóvel, pois os consumidores não seriam motivados a comprar um automóvel mais caro devido à inclusão de um dispositivo que ficaria obsoleto em pouco tempo.

Existem outros motivos que dificultam a implantação e, principalmente, a popularização dos sistemas de telemática de transporte. Há diversos estudos nessa área que tentam sanar esses problemas. Porém, apesar dessa área estar em constante expansão e

existir interesse por parte dos fabricantes e dos consumidores, os problemas citados acima fazem com que o processo de popularização desses sistemas seja demorado.

1.2 Objetivo

Neste trabalho, tem-se como objetivo disponibilizar uma infra-estrutura para a implantação de sistemas de telemática de transporte de baixo custo e expansível. O sistema de telemática de transporte criado deve permitir o acesso a informações providas por computadores remotos, tendo como base os dados obtidos do automóvel e do sistema de localização global GPS. Para isso, essa infra-estrutura deve conter componentes capazes de se comunicar com o automóvel para obter dados do estado atual deste, comunicar-se com o dispositivo que provê os serviços remotos que serão acessados, ser capaz de obter dados de localização por GPS e ser capaz de armazenar, processar e exibir os dados obtidos de forma adequada para um usuário do sistema.

A infra-estrutura criada deve ser extensível, ou seja, à medida que as tecnologias utilizadas evoluem, a infra-estrutura também deve evoluir. Isso significa que, por exemplo, se uma nova tecnologia de comunicação com maior área de abrangência ou maior taxa de transmissão que a utilizada na infra-estrutura surgir no mercado, deve ser permitido que essa tecnologia substitua a anterior. Essa substituição também deve ser permitida para os outros componentes da infra-estrutura, se for necessário, para uma possível atualização.

O ponto central da infra-estrutura não deve ser o automóvel, pois, devido aos motivos citados anteriormente neste capítulo, os custos de implantação e atualização de dispositivos eletrônicos em automóveis são altos. Obviamente, o ponto central não deve ser o servidor remoto, pois o ambiente de trabalho é o automóvel. Assim, a arquitetura deve ter como ponto central um dispositivo eletrônico que agregue as funcionalidades de obter dados, tanto do automóvel quanto do sistema de localização GPS, armazenar, processar e exibir dados para o usuário do sistema e acessar informações providas por computadores remotos.

Para validação da arquitetura, será criado um software com os seguintes requisitos:

- **Obtenção de dados do automóvel:** O software deve ser capaz de se comunicar com o automóvel utilizando alguma tecnologia de comunicação e seguindo algum protocolo disponível no dispositivo utilizado como ponto central da infra-estrutura, para obter dados.
- **Acesso a serviços remotos:** O software deve ser capaz de obter informações providas por computadores remotos utilizando alguma tecnologia de comunicação sem fio e de grande abrangência geográfica, devido à característica de mobilidade do

automóvel, como explicado na seção 2.4 do Capítulo 2. A tecnologia de comunicação utilizada deve estar disponível no dispositivo utilizado como ponto central da infraestrutura.

- **Interface gráfica com o usuário:** O software deve exibir as informações obtidas e processadas de forma adequada para o usuário, ou seja, se o usuário for o motorista do carro, não deve haver distração com a utilização do software.

Serão criados dois serviços para validação da infra-estrutura. O primeiro deles é um serviço que indica o posto de combustível mais próximo ou que está em promoção, de acordo com a preferência do usuário, e indique a melhor rota para chegar nesse posto. Isto é feito automaticamente quando o nível de combustível está abaixo de um patamar definido. Dados referentes à autonomia serão considerados caso a preferência do usuário seja procurar por um posto em que o combustível esteja mais barato, evitando que o posto escolhido seja muito distante e o combustível restante no tanque não seja suficiente para se alcançar o destino. Além disso, o serviço de rota também será criado. Neste serviço o usuário indica o destino e obtém um caminho a ser seguido para alcançá-lo.

1.3 Relevância

À medida que aumenta a quantidade de automóveis nas ruas também aumenta a quantidade de problemas relacionados ao trânsito. Assim, é de grande importância o estudo de técnicas que venham a solucionar esses problemas. O estudo na área da telemática de transporte é cada vez mais abrangente, trazendo soluções que auxiliam os motoristas na realização de suas tarefas. Porém, o uso da telemática de transporte possui algumas restrições, as quais são relacionadas principalmente ao custo de implantação nas linhas de produção dos fabricantes de automóveis, como explicado anteriormente. Dessa forma, o desenvolvimento e popularização dos sistemas de telemática de transporte ficam comprometidos.

Este trabalho é bastante relevante, pois busca criar uma infra-estrutura de baixo custo e expansível, tornando possível a utilização de sistemas de telemática, mesmo para automóveis de baixo custo, ditos automóveis populares. Além de permitir a popularização, a arquitetura utilizada para a infra-estrutura permite que o sistema seja expandido e que serviços nas mais diversas áreas como, por exemplo, a área comercial e a área governamental, sejam criados.

1.4 Estrutura da Dissertação

O restante desta dissertação está organizado da seguinte forma:

- No Capítulo 2 são apresentadas fundamentações teóricas sobre as tecnologias ou áreas de estudo utilizadas neste trabalho. Alguns temas são abordados com mais detalhes como, por exemplo, telemática, computação pervasiva, computação em ambientes automotivo, computação em dispositivos portáteis, entre outros;
- No Capítulo 3 a arquitetura utilizada na infra-estrutura é detalhada. Para isto, foi feita uma separação em três módulos. O primeiro módulo trata da parte de aquisição de dados do automóvel. O segundo módulo trata da parte que obtém, processa, armazena e exibe os dados do automóvel e do sistema de localização GPS e também acessa serviços providos remotamente. Já o terceiro módulo trata da parte do servidor remoto;
- No Capítulo 4 é apresentado o estudo de caso que valida a infra-estrutura criada;
- No Capítulo 5 são apresentadas as conclusões do trabalho, descrevendo ainda os trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste trabalho, utiliza-se como base a telemática dos transportes e todas as tecnologias que dão suporte ao acesso a serviços remotos baseados em dados obtidos a partir de um veículo. Neste capítulo apresenta-se uma fundamentação teórica com ênfase nas tecnologias utilizadas como, por exemplo, a computação pervasiva, a computação em dispositivos portáteis e a computação em ambientes automotivos. Alguns trabalhos relacionados ao proposto nesta dissertação também são discutidos.

2.1 Telemática

A união entre sistemas de tecnologia da informação modernos e telecomunicações, aplicadas à indústria dos transportes, caracteriza a telemática de transporte. No contexto desse trabalho a telemática de transporte também será tratada apenas como telemática e será usada para definir um sistema que engloba GPS, computadores embarcados e sistemas de comunicação em ambientes automotivos.

Um possível cenário para a telemática de transporte é ilustrado na Figura 2.1.

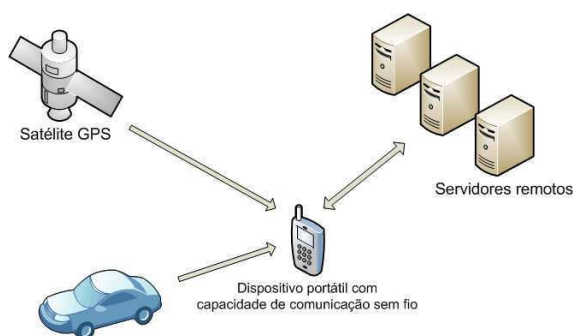


Figura 2.1: Cenário de telemática de transporte

Neste cenário o motorista ou passageiro do automóvel acessa serviços providos por servidores remotos utilizando uma tecnologia de comunicação sem fio. A escolha do serviço

é feita de acordo com dados obtidos do automóvel, dados providos pelo usuário e dados obtidos do sistema GPS. Um exemplo de serviço para esse cenário é a indicação de localização de postos de combustíveis. Esse serviço é acessado com base na informação de nível baixo de combustível no tanque obtida diretamente do automóvel. Neste campo há trabalhos que tratam de comunicação entre automóveis para evitar colisão [4] [5] [6]. Outros trabalhos tratam da comunicação entre automóveis e rodovias para evitar congestionamentos [7]. Há também os trabalhos que tratam de comunicação entre veículo e servidores remotos, o que possibilita a obtenção de serviços de caráter comercial, industrial ou governamental disponibilizados por esses servidores [8].

O campo da telemática ainda é imaturo, porém o seu uso é crescente e as tecnologias utilizadas para a sua concepção estão em constante evolução [9] [10]. Há várias razões para isso. Uma tecnologia que é comumente utilizada em sistemas de telemática é o GPS, que tem se tornado mais barato, sendo possível encontrar dispositivos móveis celulares com GPS embarcado. Os sistemas de telemática utilizam tecnologias de comunicação sem fio para troca de dados. Atualmente existem várias formas de comunicação sem fio como, por exemplo, *Bluetooth*¹, *WiMAX*², *WI-FI*³ e *GPRS*⁴. Das tecnologias citadas, o *WiMAX* e o *GPRS* se destacam por permitir acesso em uma área extensa, permitindo o uso da telemática em automóveis quando estes se encontram em rodovias. Os automóveis atuais utilizam diversos computadores e sensores, além disso, provêm uma forma de acesso aos dados que são obtidos pelos sensores e processados pelos computadores. Sendo assim, a criação de sistemas de telemática tem se tornado mais acessível e barata.

Como discutido por A. Karimi et al. [11], existem cinco objetivos principais para a telemática: navegação e acessibilidade, segurança, informação e entretenimento, manutenção do veículo e produtividade. Alcançar um desses objetivos pode significar o alcance de outros, pois a manutenção do veículo pode significar maior segurança. Porém o contrário pode acontecer, já que o uso de entretenimento em um veículo pode distrair o motorista, ocasionando acidentes. Assim, a telemática deve ser utilizada de forma correta, melhorando a navegação e o acesso a informações úteis ao motorista ou aos passageiros.

Existem dificuldades e desafios para a criação de sistemas de telemática, sendo alguns tecnológicos. A inclusão de um sistema eletrônico de telemática em um automóvel deve ser feita de forma que este sistema possa passar por um processo de atualização. Uma possível solução é a utilização de dispositivos portáteis, que podem ser atualizados tanto em hardware quanto em software facilmente. Porém, existem outras dificuldades que não são solucionadas com o simples uso de dispositivos portáteis. Uma das mais importantes

¹<http://www.bluetooth.com/bluetooth/> acessado em 31/10/2008

²<http://www.wimax.com/> acessado em 31/10/2008

³<http://www.wi-fi.org/> acessado em 31/10/2008

⁴<http://www.palowireless.com/gprs/> acessado em 31/10/2008

é a segurança dos dados enviados e recebidos por um sistema de telemática [12]. É crucial que dados referentes a transações financeiras sejam acessados apenas por usuários autorizados [13]. Para a criação de sistemas de telemática de transporte o usuário alvo deve ser levado em consideração. A interface gráfica com o usuário deve ser planejada de forma que se adapte bem às necessidades do motorista ou do passageiro. Sendo o motorista o usuário, a interação deve ser feita de modo a não atrapalhar a dirigibilidade, mantendo a segurança no ato de dirigir. Para garantir que o usuário interaja o mínimo necessário com o sistema um novo paradigma de computação, chamado de computação pervasiva, pode ser utilizado.

2.2 Computação Pervasiva

Suponha um cenário em que o motorista de um carro está dirigindo por uma cidade na qual ele acabou de chegar de viagem. Este motorista possui um dispositivo portátil capaz de obter informações de localização (latitude e longitude), exibir dados em sua tela e conectar-se à Internet sem fio. A hora do almoço se aproxima e, por não conhecer a cidade, o motorista não sabe aonde ir. Por sorte, o dispositivo portátil tem a informação de que a hora do almoço se aproxima, que o motorista tem preferência por comida japonesa e onde se localizam os restaurantes da cidade. O dispositivo portátil percebe que o automóvel está em movimento e emite um sinal sonoro para que o motorista pare e possa visualizar na tela a lista contendo o nome dos restaurantes da cidade. O motorista pára e escolhe um dos restaurantes da lista. O dispositivo portátil apresenta em sua tela o cardápio com os preços dos pratos do restaurante escolhido. Sabendo que os preços estão bons o motorista solicita a melhor rota para se chegar ao restaurante. Como a viagem havia sido longa o nível de combustível no tanque estava baixo, assim seria impossível chegar ao restaurante escolhido. Sendo assim, o motorista é avisado sobre a falta de combustível pelo dispositivo portátil que, logo em seguida, apresenta a melhor rota para se chegar ao restaurante escolhido passando por um posto de combustível. Este é apenas um exemplo de um cenário de computação pervasiva [14], também conhecida como computação ubíqua.

Em 1991, Mark Weiser, vislumbrou a computação para o século 21 de forma que esta se tornaria pervasiva, ou seja, estaria disponível em qualquer lugar a qualquer momento, existindo de uma forma quase imperceptível para os usuários [15]. Na computação pervasiva os computadores estão imersos no ambiente, como em eletrodomésticos, automóveis, aeronaves, residências, hospitais, entre outros ambientes. Como afirmado por Mark Weiser, a computação pervasiva pode ser alcançada com a utilização de três tecnologias: dispositivos baratos e com baixo consumo de energia, uma estrutura para a comunicação entre esses dispositivos e, finalmente, aplicações pervasivas. Quando esta afirmação foi

dita, a tecnologia disponível não oferecia suporte à computação pervasiva.

Atualmente existem dispositivos portáteis com tecnologia suficiente para que este tipo de computação seja concebido. Porém, existem problemas associados à criação de sistemas de software para ambientes pervasivos como, por exemplo, a necessidade de acesso completo a informações de usuários e do ambiente em que o sistema está situado que nem sempre estão disponíveis ou, quando estão disponíveis, precisam ser interpretadas e utilizadas de forma correta. Além disso, as mudanças no ambiente devem ser consideradas pelo sistema. No cenário apresentado no início deste capítulo, o ambiente pelo qual o motorista trafega é uma cidade, que é um ambiente em constante mudança. Em cidades grandes os congestionamentos das vias são freqüentes e este fato deveria ser considerado já que o automóvel está com o nível de combustível baixo. Neste caso, o dispositivo portátil informaria uma rota por ruas e avenidas não congestionadas.

Na computação pervasiva os dispositivos se comunicam através de uma rede de comunicação sem fio, chamada de rede pervasiva. Assim, aspectos fundamentais como mobilidade devem ser considerados. O uso da mobilidade em sistemas de computação enfrenta problemas que podem ser agrupados da seguinte forma [16]:

- **Limitação de recursos dos dispositivos portáteis:** os dispositivos portáteis possuem menos capacidade de processamento e armazenamento se comparados com computadores pessoais. Assim, os aplicativos de computação pervasiva devem ser criados levando em consideração as restrições dos dispositivos nos quais eles serão executados;
- **Restrições de energia:** dispositivos portáteis dependem de baterias como fonte de energia, que precisam ser recarregadas freqüentemente;
- **Variabilidade das conectividades sem fio:** o desempenho e a confiabilidade das redes de comunicação sem fio variam de uma para a outra;
- **Segurança das comunicações sem fio:** nas redes de comunicação sem fio os dados são mais facilmente obtidos do que nas redes de comunicação com fio. Assim, a segurança dos dados é um fator muito importante nas redes de comunicação sem fio.

Outro aspecto fundamental da computação pervasiva é a invisibilidade [14]. Os sistemas pervasivos devem requerer o mínimo de intervenção do usuário, de forma que se aproximem da invisibilidade. No cenário citado inicialmente, o sistema é ciente de que a hora do almoço se aproxima e que o motorista prefere comida japonesa e então toma a decisão de exibir a lista de restaurantes japoneses da cidade. Este processo foi feito de forma transparente, ou seja, sem intervenção direta do usuário. Porém, supondo que neste

dia o motorista gostaria de comer carne bovina, o sistema erraria em sua previsão. Assim, é necessário que o usuário informe previamente mais detalhes sobre as suas preferências para que os erros de previsão sejam reduzidos, evitando uma intervenção direta sobre o sistema.

Há uma crescente migração da computação tradicional baseada em computadores pessoais para a computação pervasiva [17]. Sendo assim, existem vários projetos nessa área. Entre eles se destacam o projeto Aura [18] da Carnegie Mellon University, o projeto Endeavour⁵ da University of California e o projeto Oxygen⁶ do Massachusetts Institute of Technology. Esses trabalhos comprovam o constante crescimento do estudo e da utilização da computação pervasiva como solução de problemas nas diversas áreas como, por exemplo, problemas na área da medicina, do entretenimento ou da indústria dos transportes.

Para que o cenário apresentado possa ser realizado, alguns dados do automóvel devem ser obtidos. Atualmente a computação em ambientes automotivos permite que alguns dados relevantes possam ser obtidos facilmente.

2.3 Computação em Ambientes Automotivos

No início da década de 1970, os únicos componentes eletrônicos presentes em um automóvel eram diodos e reguladores de tensão. A partir deste período, a quantidade de sistemas eletrônicos, utilizados para substituir sistemas mecânicos e hidráulicos, cresceu exponencialmente devido ao aumento do desempenho e da confiabilidade dos componentes utilizados [19]. O objetivo principal destes sistemas é incrementar o conforto e a segurança dos passageiros e do motorista. Como exemplo, tem-se o sistema de controle que auxilia a frenagem chamado de ABS, do inglês *Antilock Braking System*, o qual evita que as rodas travem durante o processo.

Atualmente os automóveis vêm equipados com sistemas eletrônicos que substituem sistemas mecânicos e hidráulicos. Estes sistemas são chamados de ECU, do inglês *Electronic Control Unit*. Os ECUs são compostos por um micro-controlador e um conjunto de sensores e atuadores, sendo distribuídos por grande parte do veículo. Algumas funcionalidades dependem de mais de um ECU, sendo necessário que haja troca de informação. Inicialmente as ligações entre os ECUs eram feitas ponto a ponto e a quantidade de canais de comunicação crescia de forma quadrática à medida que a quantidade de ECUs crescia. Quanto maior a quantidade de cabos e conectores maior o custo de fabricação e maior o peso do automóvel. Esse problema induziu a indústria automotiva a utilizar redes de comunicação multiplexadas sobre um meio compartilhado.

⁵endeavour.cs.berkeley.edu/ acessado em 31/10/2008

⁶www.oxygen.lcs.mit.edu/ acessado em 31/10/2008

Como todas as funcionalidades em um automóvel não possuem o mesmo desempenho e a mesma necessidade de segurança e, conseqüentemente, custo, é esperado que as redes de comunicação utilizadas por cada funcionalidade sejam diferentes entre si. Redes de comunicação para funcionalidades de multimídia não necessitam de desempenho e segurança maior que as redes de comunicação utilizadas pelas funcionalidades de controle do motor ou transmissão, por exemplo. Sendo assim, os fabricantes de automóveis utilizam redes de comunicação variadas internas a seus produtos.

A Sociedade para Engenheiros Automotivos (SAE, do inglês *Society for Automotive Engineers*) definiu uma classificação para protocolos de comunicação baseada na velocidade de transmissão dos dados e funções que são distribuídas sobre a rede, separando-as em classes A, B, C e D ⁷. Redes classe A possuem taxa de transmissão de dados menor que 10kb/s e são usadas para transmitir dados de controle simples com tecnologia de baixo custo como, por exemplo, as travas das portas ou iluminação. Redes classe B operam entre 10 e 125kb/s e são utilizadas para troca de dados entre ECUs. Redes classe C operam entre 125kb/s e 1Mb/s e são usadas para controle em tempo real com alta tolerância a falhas como o controle de freios eletrônicos.

Dentre as redes de comunicação utilizadas em automóveis a rede CAN (do inglês *Controller Area Network*) é a mais utilizada ⁸. Essa rede foi desenvolvida no início da década de 1980 pela Bosch e, com o passar dos anos, muitos padrões CAN foram desenvolvidos e utilizados em diferentes aplicações como, por exemplo, os padrões *low-speed CAN* e *high-speed CAN* [20] [21]. A rede de comunicação LIN⁹ (do inglês *Local Interconnect Network*) teve sua criação iniciada em 1998, foi padronizada em 2000 (LIN 1.1) e 2003 (LIN 2.0) e foi introduzida na produção de automóveis de série em 2001. Esta rede é bastante utilizada, coexistindo bem com a rede CAN, pois tem um baixo custo e uma velocidade de comunicação acima de 20KB/s. Como outros exemplos de redes de comunicação utilizadas em automóveis tem-se a Byteflight [22], com velocidade de comunicação acima de 10MB/s e com propósito de interligar sistemas críticos de segurança, e o MOST (do inglês *Media Oriented Systems Transport*) [23] com o propósito de prover comunicação para aplicações multimídia.

Apesar de existirem vários padrões de redes de comunicação internas a automóveis, os dados que trafegam por elas não estão disponíveis para o meio externo, sendo necessária a adição de sistemas que externem esses dados. Entre os anos de 1981 e 1987, a SAE, a EPA (do inglês *Environmental Protection Agency*) e a CARB (do inglês *California Air Resources Board*) trabalharam para a padronização e regulação do controle da emissão de poluentes de automóveis. A partir daí foi criado em 1988 um sistema para automóveis

⁷<http://support.intel.com/design/auto/autolxbk.htm> acessado em 31/10/2008

⁸<http://www.can.bosch.com/> acessado em 31/10/2008

⁹<http://www.lin-subbus.org/> acessado em 31/10/2008

com capacidade de monitorar a eficiência do motor e seus subsistemas e com o propósito de identificar e reportar maus funcionamentos que aumentassem a emissão de poluentes. Esse sistema foi chamado de OBD (do inglês *On-Board Diagnostic*) e, com a adição de novas funcionalidades, foi criado o sistema OBDII em seguida [24].

O OBDII possui um conector de diagnóstico chamado de DLC (do inglês *Diagnostic Link Connector*) que tem como função principal permitir a ligação de ferramentas de diagnóstico (*diagnostic scan tools*) com as unidades de controle do sistema OBDII. O DLC possui 16 pinos, como apresentado na Figura 2.2, cada um com uma função diferente. Alguns pinos têm sua função definida pelo fabricante do automóvel, porém eles não são utilizados pelas unidades de controle. Outros pinos estão ligados à bateria do automóvel, permitindo que as ferramentas de diagnóstico possam ser ligadas sem a necessidade de uma fonte de energia externa. Os pinos restantes são reservados para os padrões de comunicação (J1850, CAN e ISO 9141-2) que os fabricantes escolhem para o sistema de diagnóstico, sendo que cada padrão é associado a um pino específico [25]. Esse sistema, além de ser utilizado para fazer diagnóstico, também é utilizado para obtenção de valores dos sensores distribuídos por todo automóvel, sendo útil para se obter o estado atual deste. Para diagnóstico, o sistema OBDII retorna um código que, dependendo do fabricante do automóvel, pode ser comparado a uma tabela de código de falha, tornando possível verificar especificamente o problema ocorrido. Para dados que representam valores obtidos por sensores, o sistema OBDII retorna códigos que depois de tratados possibilitam acesso a diferentes informações como velocidade, rotações por minuto do motor, nível de combustível, posição do acelerador, entre outros dados comuns a automóveis.

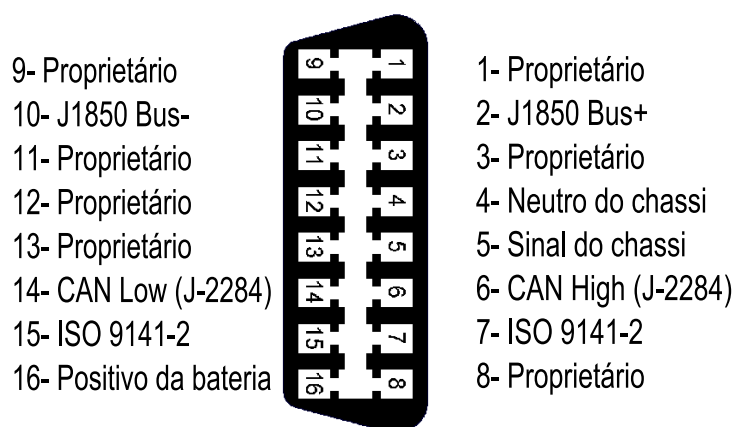


Figura 2.2: Conector DLC

O código de diagnóstico é formado por uma letra e quatro dígitos, como designado pela SAE J2012. A letra identifica em qual parte do automóvel ocorreu a falha, podendo ser P, B, C ou U, sendo P para motor e transmissão, B para o corpo, C para o chassi e U

para a rede de comunicação. O primeiro dígito determina se o código de falha foi definido pela SAE (valor 0) ou se foi definido pelo fabricante (valor 1). Essa diferença existe porque cada fabricante tem seu próprio sistema. Assim, a SAE separou os códigos de falha em duas categorias: os códigos padrão, criados por ela, e os códigos dos fabricantes. O segundo dígito indica o sistema afetado, variando de 1 a 8. O dígito 3, por exemplo, indica que existe falha no sistema de ignição e fagulha. Já o terceiro e o quarto dígitos são analisados juntos e representam a falha especificamente. Analisando o código P0301 da Figura 2.3, tem-se que P indica uma falha no motor ou transmissão, “0” indica que o erro é genérico, “3” indica que é no sistema de ignição e fagulha e “01” indica, neste caso, o cilindro que não tem fagulha.

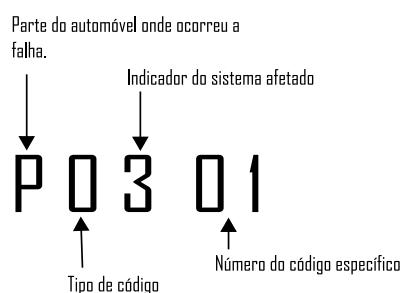


Figura 2.3: Exemplo de código de falha

Para que os dados providos pelos sistemas de computação em ambientes automotivos possam ser observados pelo motorista, um dispositivo que disponibilize alguma interface com o mesmo deve ser utilizado. Alguns automóveis saem de fábrica equipados com estes dispositivos, o que aumenta o seu valor final, fazendo com que este não seja acessível à maior parte dos compradores. Uma possível solução é a utilização de dispositivos portáteis para este fim.

2.4 Computação em Dispositivos Portáteis

Dispositivos portáteis são aqueles que possuem características que permitem que eles sejam transportados por humanos com facilidade sendo, geralmente, leves e pequenos. Dispositivos como: câmera fotográfica, telefone móvel celular e *laptops* são exemplos mais comuns deste tipo de dispositivos. Os avanços tecnológicos permitem o surgimento de dispositivos mais leves, menores e com mais funcionalidades agregadas. As câmeras fotográficas digitais, por exemplo, permitem o armazenamento de milhares de imagens com resolução adequada para impressão, possuem uma tela de cristal líquido colorida para exibição de imagens e algumas possuem capacidade de comunicação sem fio para transmissão da imagem. Os telefones móveis celulares evoluíram para *smartphones*, que são dispositivos móveis celulares com alta capacidade de processamento e armazenamento.

Os *smartphones* se comunicam por meio de tecnologias de comunicação sem fio, sendo então categorizados como dispositivos móveis [26]. A comunicação pode ser feita com a utilização de diversas tecnologias como, por exemplo, *Bluetooth*, Wi-Fi e GPRS, sendo ideais para fazer a ligação entre diferentes tecnologias de rede como, por exemplo, a rede de comunicação de um automóvel e a rede mundial de computadores. Porém, a mobilidade traz consigo alguns desafios, sendo um dos principais relacionado à comunicação. Os dispositivos móveis podem estar na área de cobertura de uma rede sem fio e em um determinado momento sair dessa área assim que são transportados para outro local, ocasionando a interrupção da transmissão e recepção de dados. Se este outro local estiver coberto pelo sinal de outra rede, o dispositivo móvel poderá transmitir e receber dados por essa rede. O novo local pode estar coberto pelo sinal de uma mesma tecnologia de rede do local anterior ou de outra tecnologia. O processo automático de mudança de rede é chamado de *handoff*, sendo este chamado de *handoff* horizontal quando se mantém a tecnologia de rede e de *handoff* vertical quando a tecnologia de rede é alterada [27].

Atualmente, é comum encontrar *smartphones* de baixo custo que oferecem uma interface para programação de aplicações bem definida, permitindo que o desenvolvedor acesse a maioria dos recursos disponíveis no dispositivo portátil. Recursos como transferência de arquivos usando a tecnologia *Bluetooth*, transferência de dados usando a tecnologia GPRS, que permite o acesso a informações providas remotamente, e construção de interface gráfica com o usuário podem ser acessados pelo desenvolvedor, permitindo que aplicações complexas sejam criadas. Outra vantagem dos *smartphones* é que eles são dispositivos de uso pessoal e podem conter informações de perfil de usuário.

A construção de interfaces gráficas com usuários para dispositivos portáteis é mais complexa devido a características inerentes à mobilidade e portabilidade. Esses dispositivos podem ser transportados por diversos ambientes com características diferentes. Por exemplo, é possível utilizá-lo em uma sala de teatro com pouca luminosidade ou em uma praia ensolarada, o que pode dificultar a visualização. Existe também a possibilidade de uso em situações em que o usuário não pode desviar a atenção em uma determinada tarefa como, por exemplo, dirigir um automóvel. Nesta situação a interface com o usuário não deve ser feita em modo visual, mas pode ser feita em modo sonoro, o que não seria recomendado no caso de uso em uma sala de teatro. Os dispositivos portáteis geralmente possuem uma tela pequena, principalmente os *smartphones*, dificultando a visualização de alguns componentes como, por exemplo, botões e barras de rolagem. Além disso, a interação dos usuários com os *smartphones* geralmente se dá por meio de um teclado com poucas teclas com relação a um teclado de computador, dificultando ainda mais a construção de interfaces. O seu desenvolvimento enfrenta problemas como:

- **A diferença de hardware entre dispositivos:** alguns dispositivos possuem pro-

cessadores poderosos e telas com resolução alta, que são características necessárias para exibir alguns tipos de interfaces mais amigáveis e dinâmicas, que não seriam possíveis de serem exibidas em outros dispositivos com capacidade de processamento e tela menores;

- **As diversas formas existentes de interação com o usuário:** a forma de interação com o usuário varia bastante, podendo ser feita de várias formas como, por exemplo, por meio de teclado, *mouse*, tela sensível ao toque ou por meio sonoro;
- **Os diversos propósitos de uso:** o propósito de uso de um *smartphone* é diferente de uma câmera fotográfica digital, apesar de ambos serem dispositivos portáteis;
- **A possível mudança de contexto:** com a mobilidade dos dispositivos portáteis o contexto pode variar durante o uso como, por exemplo, a diferença de luminosidade de ambiente que varia entre ambientes e pode dificultar a visualização.

Estes são alguns dos problemas que dificultam a implementação de interfaces gráficas em dispositivos portáteis [28]. Uma das soluções para resolver esses problemas é a construção de uma interface única para cada dispositivo. Isso é possível para o caso em que tem-se um dispositivo específico para uso. Caso contrário, problemas como inconsistência devem ser considerados, pois uma interface deve ser apresentada de forma similar em dispositivos diferentes.

Outras dificuldades são encontradas no desenvolvimento de aplicações para dispositivos portáteis. A maioria delas é relacionada à capacidade reduzida de armazenamento e de processamento desses dispositivos com relação aos computadores pessoais, os chamados *desktops*. Para suprir a necessidade de criar aplicações para dispositivos móveis, que exigem uma capacidade de armazenamento e de processamento de informações não encontradas em dispositivos portáteis, pode-se criar aplicações para servidores remotos e prover para os dispositivos portáteis o acesso às informações dispostas por esse servidor.

Os serviços providos remotamente e acessados, geralmente, via *web* são chamados de *Web services*¹⁰. Tecnicamente, *Web services* são sistemas de software que têm como objetivo prover suporte à interação entre máquinas em uma rede de comunicação. Eles possuem uma interface em um formato que é passível de processamento por uma máquina. Tipicamente essa interface é descrita em WSDL¹¹ que é um documento XML¹².

A arquitetura proposta pela W3C¹³ (do inglês *World Wide Web Consortium*) envolve muitas camadas e tecnologias inter-relacionadas, existindo várias formas de criar e

¹⁰<http://www.w3.org/TR/ws-arch/>

¹¹<http://www.w3.org/TR/2007/REC-wsdl20-primer-20070626/> acessado em 31/10/2008

¹²<http://www.w3.org/XML/> acessado em 31/10/2008

¹³www.w3.org acessado em 31/10/2008

usar *Web services*. A troca de mensagens entre provedores de serviços e clientes é feita com a utilização do protocolo SOAP¹⁴ (do inglês *Simple Object Access Protocol*), que é um protocolo simples e extensível para troca de informações estruturadas em ambientes distribuídos. Assim é possível criar diversos serviços de forma simples, distribuída e extensível, sendo acessíveis a partir de um dispositivo móvel, que esteja na área de cobertura de uma rede de comunicação, como, por exemplo, um serviço que indique uma livraria que contém um livro que o usuário está procurando nas proximidades de sua localização.

2.5 Trabalhos Relacionados

Millar et al apresentaram em [8] um trabalho que examina a contribuição que os *Web services* podem trazer no desenvolvimento de sistemas de telemática de transporte, tanto na área de comércio quanto nas áreas governamentais e de emergência. Para validar o trabalho, uma infra-estrutura composta por um PDA, um servidor remoto e uma aplicação foi criada. Nessa infra-estrutura, a aplicação que está sendo executada no PDA acessa dados do servidor remoto via GPRS.

Bilchev et al descreveram em [29] o projeto Traffimatics. Este projeto visa desenvolver o conceito de sistemas inteligentes cooperativos em rodovias, suportando aplicações que vão de acesso a informações em tempo real de tráfego a aplicações de entretenimento. Para isso, as tecnologias que possibilitam a criação desse projeto são descritas e as barreiras e oportunidades de mercado são discutidas. Este trabalho também trata da computação pervasiva como suporte para aplicações de segurança, controle de tráfego e aplicações para os passageiros.

Kim et al apresentaram em [30] um sistema aberto de telemática baseado em um *gateway* e um arcabouço de software, de forma que não fosse dependente de uma rede de comunicação móvel específica e que fosse interoperável com outros sistemas de telemática.

Alguns outros trabalhos merecem ser citados. Bisdikian et al descreveram em [3] um sistema de telemática de padrão aberto baseado no ambiente de serviços sem fio ts-PWLAN e em um *middleware* de gerenciamento de recursos. Já Reilly et al mostraram em [31] como desenvolver sistemas de telemática de transporte utilizando prototipagem rápida. Zhao, por sua vez, abordou sobre a segurança trazida pela telemática de transportes em [32]. Reilly et al descreveram em [33] o projeto EmergeITS, que tem como objetivo utilizar a telemática de transporte para acesso a serviços de emergência em caso de incêndio. Bauer et al mostraram em [34] uma estratégia de como integrar diferentes sistemas de telemática. Kim et al em [35] propuseram um software chamado de TeleASP que cria aplicações virtuais que podem ser acessadas e executadas sob demanda por di-

¹⁴<http://www.w3.org/TR/soap12-part1/> acessado em 31/10/2008

versos dispositivos em sistemas de telemática. Li et al descreveram em [36] um sistema uniforme chamado de *Intelligent Vehicle Telematics Service System*.

A partir de uma análise detalhado dos trabalhos citados acima, é possível observar que eles possuem características interessantes para serem utilizadas neste trabalho. Porém, diferenças importantes são encontradas entre os trabalhos citados e o trabalho aqui realizado.

O trabalho definido em [8] utiliza *Web services* como fonte de informação, assim como este trabalho. Porém, ele não considera a obtenção de dados do automóvel como condição para acesso a serviços remotos, pois apenas a obtenção de dados de localização GPS são considerados.

O trabalho definido em [29] não define como a obtenção de dados do automóvel é feita. Apenas cita-se um conjunto de redes de comunicação utilizadas em automóveis. Além disso, não se utiliza o conceito de *Web services* para provisão de serviços e seu foco principal é criar uma comunicação entre o automóvel e a estrada, para provisão de serviços relacionados a tráfego.

O trabalho definido em [30], além de não citar a obtenção de dados do automóvel, também não cita o uso de *Web services* para provisão de serviços. Ele apenas define um arcabouço de software para ser utilizado no nível de aplicação.

Capítulo 3

Infra-estrutura para Sistemas de Telemática de Transporte

Neste capítulo, apresenta-se a infra-estrutura proposta para sistemas de telemática de transporte proposta neste trabalho. Primeiramente, apresenta-se uma visão geral da infra-estrutura. Depois, apresenta-se os módulos que compõem a infra-estrutura. Em seguida, apresenta-se uma análise das restrições e requisitos impostos. Por fim, apresenta-se os componentes que compõem cada módulo apresentado, os quais seguem as restrições e requisitos citados.

Como descrito no Capítulo 1, existem muitas dificuldades para a construção de sistemas de telemática de transporte. Dessa forma, neste trabalho foi proposta uma infra-estrutura que viesse a reduzir essas dificuldades.

3.1 Visão geral

O atual estágio de desenvolvimento das tecnologias já permite a construção de sistemas de telemática. É possível encontrar sistemas de comunicação de baixo custo e, além disso, os automóveis já possuem sensores e computadores embarcados bem desenvolvidos e em grande quantidade. Porém, apesar dessas condições serem necessárias, elas não são suficientes. É necessária também a inclusão de um dispositivo que processe os dados recebidos pelos sensores e computadores embarcados no automóvel, exiba esses dados para o motorista de forma adequada e, utilizando os dados processados e dados do perfil do motorista, acesse serviços providos por servidores remotos. O principal problema para inclusão desses dispositivos é que o tempo de vida útil de um automóvel é bem maior que o tempo de vida útil de um dispositivo eletrônico, pois esse tipo de tecnologia se torna obsoleta mais rapidamente. Como exemplo, tem-se o sistema de comunicação móvel e sem fio GPRS. Esse sistema se adequa aos requisitos de mobilidade com grande alcance

geográfico. Porém, outras tecnologias de comunicação de baixo custo e longo alcance, como o WiMAX, podem se tornar mais vantajosas, dado que possuem uma maior taxa de transmissão que o GPRS. Seria necessária, então, a substituição ou atualização do hardware do dispositivo, que é uma tarefa difícil e de alto custo para os fabricantes de automóvel. A infra-estrutura apresentada a seguir é uma solução para os problemas relacionados à construção de sistemas de telemática de transporte.

3.2 Módulos que formam a infra-estrutura para construção de sistemas de telemática

A solução aqui apresentada é a criação de uma infra-estrutura que permita o uso de um dispositivo eletrônico para processamento, aquisição, tratamento, exibição de dados e também para acesso a serviços remotos que possa ser substituído ou atualizado facilmente e sem um custo operacional alto para o fabricante ou motorista. Para um melhor entendimento, a infra-estrutura foi separada em três módulos:

- **Módulo para obtenção de dados do automóvel:** este módulo é constituído por um sistema de diagnóstico de automóveis, como o OBDII, ou um sistema eletrônico para obtenção de dados de automóveis. O acesso aos dados é facilitado pelo conector DLC, o qual permite a ligação de dispositivos eletrônicos chamados de scantools¹. Esses dispositivos podem funcionar como *gateway*, convertendo os dados do protocolo utilizado no sistema OBDII para outros protocolos como, por exemplo, o RS232.
- **Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos:** este módulo é constituído por um dispositivo portátil e móvel com capacidade suficiente para se comunicar com o módulo para obtenção de dados do automóvel. Além disso, esse dispositivo deve ser capaz de obter dados de localização do sistema GPS, ou seja, deve possuir o hardware capaz de se comunicar com satélites que provêem dados de localização. Para armazenar dados de perfil de usuário, esse dispositivo deve possuir mídia de armazenamento e também deve possuir alguma tecnologia de comunicação sem fio, para que seja possível o acesso a serviços providos por servidores remotos, evitando que a característica de mobilidade seja suprimida. Para exibir dados de forma adequada para o motorista, o dispositivo utilizado deve possuir uma tela. Um dispositivo que possui essas características é o *smartphone*, pois nele estão presentes diversas tecnologias de comunicação, como, por exemplo, o *Bluetooth*, que possui o protocolo RFCOMM, o qual

¹<http://www.scantool.net/> acessado em 31/10/2008

provê uma emulação de portas seriais, e é capaz de se comunicar com dispositivos de localização GPS. Esse dispositivo também possui a tecnologia de comunicação GPRS, que permite o acesso, em grandes áreas, a serviços providos remotamente. Além disso, tem uma tela grande e colorida, a qual é suficiente para exibir dados de forma adequada, e são capazes de armazenar dados de perfil de usuário, pois possuem mídia de armazenamento.

- **Módulo do servidor remoto:** este módulo é constituído por um computador de propósito geral ligado à Internet, com ou sem fio, ou por algum dispositivo que possua capacidade de processamento, armazenamento e comunicação como, por exemplo, o TINI². A sua função é prover serviços via *Web services*, gerenciando as conexões com os usuários de forma eficaz e guardando as informações necessárias, tanto de usuários como de serviços cadastrados.

Com a infra-estrutura formada por esses três módulos é possível substituir ou atualizar o módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos de forma fácil, pois o sistema está centrado neste módulo e os outros dois módulos são apenas provedoras de informações. Assim, não existe um acoplamento forte entre eles, basta que os protocolos de comunicação sejam seguidos.

3.3 Restrições e requisitos

Para os módulos acima apresentados, uma série de requisitos deve ser seguida devido às restrições apresentadas por seus componentes. Cada módulo possui seus requisitos e estes serão apresentados separadamente.

3.3.1 Módulo para obtenção de dados do automóvel

Este módulo necessita que o automóvel que proverá os dados possua um sistema que facilite a aquisição de dados do automóvel como, por exemplo, o sistema OBDII. Caso o sistema utilizado seja o OBDII, é necessário verificar a diferença de potencial entre os pinos dezesseis e quatro do conector DLC, pois nem todos os automóveis saem de fábrica com esse sistema funcionando. Esta diferença de potencial deve ser em torno de doze volts, que é a tensão da bateria do automóvel. Mesmo que a diferença de potencial esteja dentro dos padrões, não é garantido que o sistema OBDII esteja em funcionamento, mas esta condição é necessária, pois a tensão provida pelo pino dezesseis do conector DLC serve como fonte de alimentação para o scantool. Assim, é necessário realizar mais testes, com a utilização de um *scantool*.

²<http://www.maxim-ic.com/products/microcontrollers/tini/> acessado em 31/10/2008

A restrição imposta ao *scantool* é que sua interface de comunicação externa siga um protocolo de comunicação que possibilite a troca de dados com alguma tecnologia de comunicação sem fio. Essa restrição é imposta para que o *smartphone*, que possui características de mobilidade e comunicação sem fio, possa requisitar dados providos pelo automóvel. Uma tecnologia de comunicação sem fio bastante utilizada é o *Bluetooth*. Esta tecnologia está presente na maioria dos *smartphones* e existem no mercado vários dispositivos *Bluetooth* de baixo custo, o que possibilita o seu uso no *scantool*. Outras tecnologias podem ser utilizadas como, por exemplo, infravermelho. Porém, esta tecnologia não é indicada para uso na infra-estrutura proposta, pois é necessário que o infravermelho do *smartphone* esteja próximo ao infravermelho do *scantool* e que eles estejam apontados diretamente entre si.

3.3.2 Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos

Este módulo é o que possui mais restrições. Além de ser o centro da arquitetura, ele gerencia os dados provindos do automóvel, do usuário, do sistema GPS e dos serviços providos remotamente. Como as fontes de informação são diversas, é necessário que o acesso e requisição dos dados sejam feitos de forma assíncrona para que informações importantes não sejam perdidas e a aplicação não seja bloqueada, ou seja, é necessário que o ato de obter dados do automóvel não bloqueie o acesso aos serviços providos remotamente ou a exibição de dados na tela, por exemplo.

A exibição de dados deve ser feita de forma cautelosa, pois o ambiente automotivo requer cuidados quando se trata de algo que pode tirar a concentração do motorista. Considerando a situação em que o usuário do sistema é o motorista, os dados só devem ser exibidos na tela do dispositivo utilizado para exibição quando o automóvel estiver parado. Porém, desta forma, dados importantes podem ser perdidos como, por exemplo, o nível de combustível no tanque. O usuário não será informado de que o nível de combustível no tanque está baixo se o automóvel estiver em movimento. Uma possível solução é emitir um aviso sonoro sempre que uma informação importante estiver disponível para a visualização na tela do dispositivo utilizado para exibição, forçando o usuário a parar o automóvel para visualizar a informação. Porém, esta solução não deve ser utilizada sempre, pois o usuário da aplicação pode ser um passageiro. Neste caso, o ideal é que a exibição de dados seja feita sempre. Assim, é necessário que o usuário informe à aplicação se ele é o motorista ou passageiro antes da utilização do sistema.

Para que seja permitido o acesso aos serviços providos remotamente, é necessário apenas que o cliente siga o protocolo para troca de mensagens utilizado no servidor remoto.

Assim, é possível que a linguagem de programação utilizada no servidor remoto seja diferente da linguagem de programação utilizada no cliente. Como neste caso o cliente é um dispositivo portátil com restrições de capacidade de memória e de processamento, a liberdade de escolha da linguagem de programação é um fator muito importante para a construção de sistemas de telemática utilizando a infra-estrutura proposta.

Além das restrições de capacidade de armazenamento e de processamento, os dispositivos portáteis possuem restrições com relação ao fornecimento de energia. A maioria dos dispositivos portáteis utiliza uma bateria como fonte de energia, que possui um limite de carga. O uso de tecnologias de comunicação sem fio consome bastante energia, fazendo com que o tempo de fornecimento de energia da bateria seja reduzido. Assim, é necessário que a bateria do dispositivo portátil permaneça em carregamento em alguns momentos do seu uso. Essa restrição é facilmente contornada com o uso de carregadores próprios para serem usados em automóveis.

3.4 Componentes que compõem os módulos da infra-estrutura

Cada módulo da infra-estrutura é subdividido em componentes, os quais são ilustrados na Figura 3.1 e detalhados a seguir.

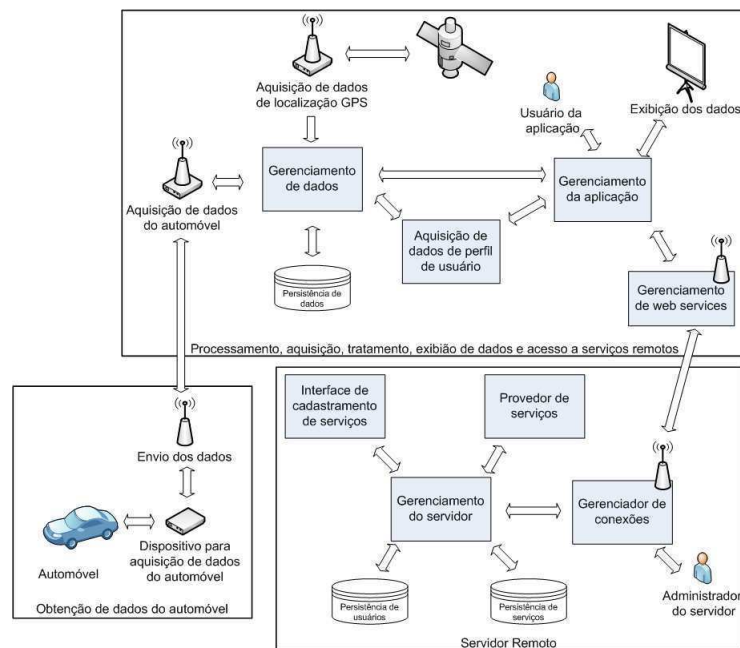


Figura 3.1: Arquitetura da infra-estrutura proposta.

3.4.1 Módulo para obtenção de dados do automóvel

Este módulo possui três componentes, como é ilustrado na Figura 3.2.



Figura 3.2: Módulo para a obtenção de dados do automóvel.

O componente “provedor de dados” é o automóvel. Ele possui sensores e computadores embarcados que provêm dados do seu estado atual. No caso de se utilizar o sistema OBDII, deve-se observar que o protocolo de comunicação utilizado varia de acordo com a escolha do fabricante, podendo ser J1850, CAN, ISO 9141-2, entre outros. Assim, existe a necessidade de inclusão de um componente que descubra qual protocolo de comunicação o sistema OBDII utiliza e esteja preparado para obter dados do automóvel utilizando este protocolo.

O componente “aquisição e tratamento de dados” pode ser um *scantool* ou outro dispositivo eletrônico para aquisição de dados do automóvel. Os *scantools* possuem um conector DLC fêmeo e conseguem descobrir qual protocolo de comunicação é utilizado pelo sistema OBDII de um determinado automóvel. Os *scantools* também possuem uma interface de comunicação externa para receber comandos e enviar dados de resposta. A depender do *scantool*, esta interface pode ser uma porta serial RS232, USB ou *Bluetooth*. Assim, os comandos de requisição de dados são enviados ao *scantool* que, após converter o comando para o protocolo utilizado pelo sistema OBDII, requisita os dados do estado atual do automóvel. Em seguida, o *scantool* converte a resposta enviada pelo automóvel para o protocolo utilizado pela interface de comunicação externa e envia a resposta por esta interface.

O componente “comunicação e envio de dados” é um módulo eletrônico de comunicação que é conectado ao dispositivo para aquisição de dados do automóvel. Este componente tem como função converter os dados para o protocolo de comunicação utilizado pelo componente para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos.

3.4.2 Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos

Este módulo possui duas camadas: o hardware e o software. O software possui diversos componentes internos, como é ilustrado na Figura 3.3, os quais foram separados em três partes para um melhor entendimento: gerenciamento de dados, exibição e acesso a serviços remotos.

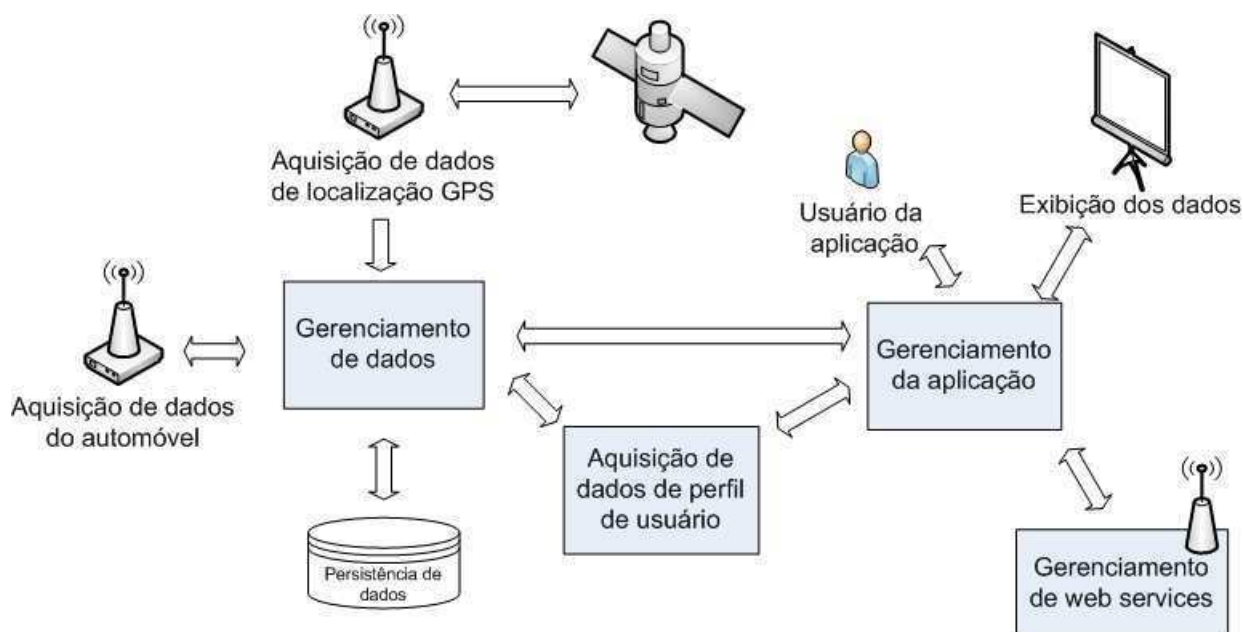


Figura 3.3: Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos.

Gerenciamento de dados

A parte de gerenciamento de dados é composta por: componente “gerenciamento de dados”, componente “aquisição de dados do automóvel”, componente “aquisição de dados de localização GPS”, componente “aquisição de dados de perfil de usuário” e componente “persistência de dados”. O componente “gerenciamento de dados” é o principal componente desta parte, pois ele recebe dados do automóvel, dados de GPS e dados do usuário. Além disso, esse componente guarda os dados de perfil de usuário obtidos. Para obter dados do automóvel, este componente faz uma requisição ao componente “aquisição de dados do automóvel” que, por sua vez, comunica-se com o componente “comunicação e envio de dados” do módulo para obtenção de dados do automóvel, utilizando *Bluetooth* ou *Wi-fi*. O componente “aquisição de dados de localização GPS” obtém os dados de GPS de forma autônoma e envia os dados obtidos ao componente “gerenciamento de dados” sempre que existir atualização nos dados de localização. O componente “aquisição de

dados de perfil de usuário” recebe os dados de perfil de usuário a partir do componente “gerenciador da aplicação” que está contido na parte de exibição.

Exibição

A parte de exibição é composta por: componente “interface gráfica com o usuário” e componente “gerenciador da aplicação”. O componente “interface gráfica com o usuário” é responsável por exibir as informações obtidas do componente “gerenciamento de dados”. Essas informações são tratadas pelo componente “gerenciador da aplicação” para que se adequem ao formato e tamanho da tela em que as informações serão exibidas. Este componente também possui a funcionalidade de receber eventos do usuário e alterar a interface gráfica da aplicação. Para obter dados de perfil de usuário, o componente “gerenciador da aplicação” requisita que o componente “interface gráfica com o usuário” exiba um formulário, o qual deve ser preenchido pelo usuário. Após o preenchimento e confirmação, os dados são enviados para o componente “aquisição de dados de perfil de usuário” da parte de gerenciamento de dados. Os dados são enviados em seguida para o componente “gerenciamento de dados” que requisita a gravação dos dados ao componente “persistência de dados”. Algumas ações do usuário são enviadas pelo componente “gerenciador da aplicação” para o componente “aquisição de dados de perfil de usuário”. Isso é feito para que o software possa tomar decisões baseadas em ações anteriores, evitando que o usuário tenha que agir sobre o sistema sempre que existir mais de uma opção de escolha.

Acesso a serviços remotos

A parte de acesso a serviços remotos é a mais simples. Ela é composta apenas pelo componente “gerenciador de serviços remotos”. Este componente segue uma API para acesso a serviços remotos definida de forma que seja possível acessar serviços providos pelo módulo do servidor remoto. O componente “gerenciador da aplicação” requisita o acesso aos serviços providos remotamente e, após receber a resposta, trata os dados recebidos e envia para o componente “interface gráfica com o usuário” os dados que forem necessários. Este componente também é o responsável por enviar ao servidor remoto os dados de perfil de usuário. Ele recebe os dados do componente “gerenciador da aplicação” e os envia para que o servidor trate informações relacionadas às preferências do usuário e, assim, possa selecionar os serviços que serão providos ao usuário de forma correta.

3.4.3 Módulo do servidor remoto

Este módulo provê os serviços que serão acessados remotamente pelo componente “gerenciador de serviços remotos” do módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos. Ela é composta pelos componentes: gerenciador de conexões, gerenciador do servidor, persistência de serviços, persistência de usuários, provedor de serviços e interface para cadastramento de serviços, como é ilustrado na Figura 3.4.

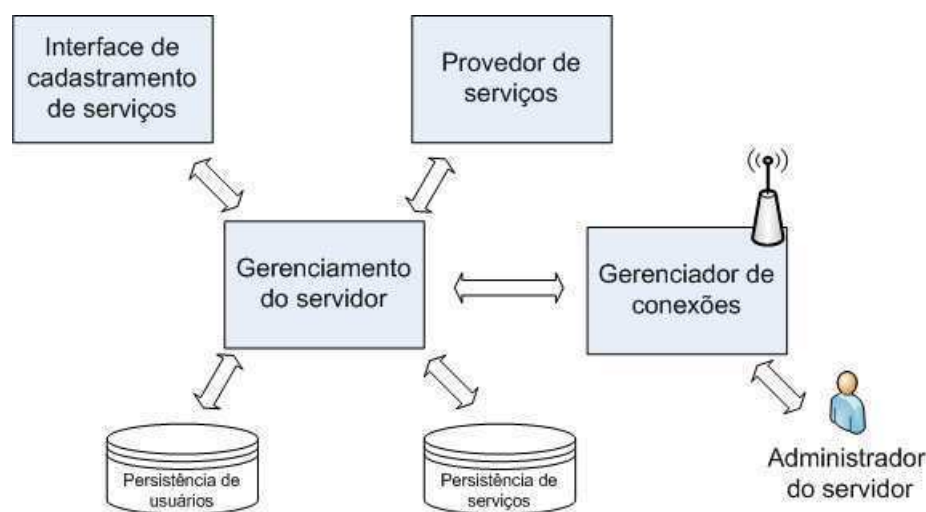


Figura 3.4: Módulo do servidor remoto.

Como o servidor será acessado por mais de um cliente ele necessitará gerenciar as conexões com esses clientes. O componente que realiza essa tarefa é o “gerenciador de conexões”. Quando um cliente se conecta, uma *thread* é criada para que a conexão com este cliente se mantenha e o “gerenciador de conexões” possa esperar por outra conexão. Após realizada a conexão, o componente “gerenciador do servidor” obtém os dados de perfil de usuário e os envia para o componente “persistência de usuários” para que sejam guardados. Este processo é repetido sempre que um cliente se conecta e, para cada cliente, um identificador é criado para que os usuários sejam diferenciados e seus dados possam ser buscados mais facilmente no banco de dados. O componente “interface de cadastramento de serviços” provê uma interface para que um cliente cadastre serviços. Neste caso o cliente é um administrador do sistema. Através dessa interface é possível cadastrar serviços localmente ou remotamente, pois esta interface possui métodos providos como serviços remotos. Os serviços cadastrados são então gravados no componente “persistência de serviços”. Depois de gravados, os dados são obtidos e tratados pelo componente “gerenciador de serviços” para então serem enviados para o componente “provedor de serviços” e serem providos como *Web services*.

Capítulo 4

Estudo de caso

Para validação da infra-estrutura proposta, um software, chamado de Tilapia, foi desenvolvido para *smartphone*, que é o ponto central da arquitetura. Além disso, um servidor remoto foi implementado para prover serviços remotamente e um *scantool* foi utilizado para permitir a obtenção de dados de um automóvel.

A seguir, os três módulos que compõem a infra-estrutura são apresentados em detalhes. Por fim, os resultados dos testes realizados para validação do software são apresentados.

4.1 Módulo para obtenção de dados do automóvel

Para que um automóvel possa prover dados referentes ao seu estado atual é necessário que ele possua sensores, computadores embarcados e uma rede de comunicação. A maioria dos automóveis fabricados a partir de 1996 possui esses requisitos e o sistema OBDII, que facilita a obtenção dos dados, como já explicado no Capítulo 2. Porém, os sistemas OBDII utilizam diversos protocolos de comunicação, escolhidos pelo fabricante, para troca interna de dados. Dentre estes protocolos encontram-se J1850, CAN e ISO 9141-2. Sendo assim é necessário conhecer antecipadamente qual protocolo de comunicação é utilizado em determinado automóvel para obter os dados. Cada fabricante utiliza um protocolo de comunicação para o sistema OBDII à sua escolha, o que torna difícil saber antecipadamente qual protocolo utilizar para a criação de dispositivos de aquisição de dados.

Existem no mercado diversos dispositivos eletrônicos capazes de descobrir qual protocolo de comunicação é utilizado no sistema OBDII do automóvel. Após o protocolo utilizado ser descoberto, a comunicação é feita e os dados providos pelo automóvel podem ser obtidos e convertidos para que se adequem a outro protocolo que seja amplamente utilizado como, por exemplo, o protocolo RS232. Esses dispositivos são chamados de *scantools*.

O *scantool* utilizado foi o ELM327¹, que interpreta e converte dados de OBDII para RS232. Este dispositivo permite que haja comunicação serial, seguindo o padrão RS232, com o automóvel. Para que a comunicação seja realizada, a taxa de transmissão deve ser 9600Kbps ou 38400Kbps, o tamanho da palavra de dados deve ser 8 bits, sem bit de paridade e com 1 (um) bit de parada. Os dados enviados para o conector de comunicação serial do *scantool* podem ser entendidos como comandos para uso interno ou entendidos como comandos que devam ser formatados e passados para o automóvel, ou seja, o *scantool* recebe um determinado comando em sua porta de comunicação serial e decide se deve utilizá-lo como comando interno ou como comando a ser enviado ao automóvel. O ELM327 pode decidir isso rapidamente, pois os comandos para uso interno são comandos AT², que são comandos normalmente utilizados em *modems*, enquanto que os comandos para o sistema OBDII do automóvel devem ser formados por códigos ASCII com dígitos hexadecimais (de 0 a 9 e de A a F). Independentemente do comando enviado ao ELM327 ser para uso interno ou para o automóvel, um carácter de retorno de carro (do inglês *carriage return*, representado pelo código hexadecimal “0D”) deve ser enviado após cada comando. Se este carácter não for enviado, o ELM327 irá esperar por 20s e, após esse tempo, enviará o carácter “?”, informando que o comando não foi entendido. O carácter “?” também é enviado se a mensagem for incompleta, se o comando AT estiver incorreto ou se o dígito hexadecimal for inválido. O ELM327 não diferencia maiúsculas de minúsculas, ignora carácter de espaço e repete o último comando quando recebe apenas um carácter de retorno de carro, o que facilita o envio de comandos. Como exemplo de comando AT, tem-se o comando “at rv”, que retorna a voltagem da bateria. Pode-se observar que este comando é executado com os caracteres em letras minúsculas, mas também poderia ser feito com letras maiúsculas. Pode-se também observar que existe um carácter de espaço no comando, porém este é opcional.

Se o comando enviado ao ELM327 não iniciar com os caracteres AT, ele é assumido como comando OBDII, o qual deve ser enviado ao automóvel. Os comandos OBDII são geralmente formados apenas por um ou dois bytes, porém alguns podem conter três ou mais bytes. A quantidade máxima de bytes suportados pelo ELM327 é quatorze, ou seja, qualquer comando que exceda essa quantidade será ignorado e o ELM327 responderá com o carácter “?”, que também é a resposta para comandos com quantidade ímpar de dígitos hexadecimais. O padrão do sistema OBDII especifica que cada grupo de bytes enviados ao veículo deve seguir o seguinte formato: o primeiro byte, conhecido como “modo”, sempre descreve o tipo de dado que é requisitado, enquanto os bytes restantes, conhecidos como PID, especificam a informação atual requisitada. Existem nove modos

¹<http://www.elmelectronics.com/> acessado em 31/10/2008

²<http://nemesis.lonestar.org/reference/telecom/modems/at/history.html> acessado em 31/10/2008

de diagnóstico no padrão do sistema OBDII, sendo que não é necessário que todos os modos sejam suportados pelo automóvel. Os nove modos são:

- 01-Show Current data;
- 02-Show freeze frame data;
- 03-Show diagnostic trouble codes;
- 04-Clear trouble codes and stored values;
- 05-Test results, oxygen sensors;
- 06-Test results, non-continuously monitored;
- 07-Show “pending” trouble codes;
- 08-Special control mode;
- 09-Request Vehicle information.

Para cada modo, o PID “00” é reservado para mostrar quais PIDs são suportados para aquele modo. Como exemplo, pode-se verificar o PID 05 para o modo 01, que requisita a atual temperatura do sistema de arrefecimento do motor. Este comando deve ser enviado da seguinte forma: 01 05, seguido do carácter de retorno de carro. A resposta para este comando será da forma: 41 05 7B, em que 4 indica que se trata de uma resposta, 1 indica o modo, 05 indica o PID requisitado e 7B é o dado desejado. Convertendo este valor de hexadecimal para decimal temos 123, que representa a temperatura atual em graus Celsius somada mais 40 (valor utilizado para permitir temperaturas abaixo de zero graus Celsius). Assim a temperatura atual é 83 graus Celsius. As informações de como converter os dados obtidos em hexadecimal para o valor real podem ser encontradas no site da SAE³, no site da ISO⁴ e em várias outras fontes na Internet.

Para que o módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos, a qual utiliza um *smartphone*, pudesse requisitar e obter informações do automóvel, um dispositivo para comunicação *Bluetooth* foi acoplado ao conector serial do *scantool*, pois, além de ser uma tecnologia de baixo custo, a maioria dos *smartphones* existentes no mercado a possui.

³www.sae.org acessado em 31/10/2008

⁴www.iso.org acessado em 31/10/2008

4.2 Módulo para processamento, aquisição, tratamento, exibição de dados e acesso a serviços remotos

Esta módulo é composto por componentes de hardware e software. Neste trabalho, foi utilizado um *smartphone* Nokia N95, que utiliza a plataforma s60, como hardware, pois ele possui: capacidade de processamento e armazenamento suficientes para suportar o software criado; hardware para obtenção de dados GPS; tecnologia de comunicação *Bluetooth* para troca de dados com o *scantool*; tecnologias de comunicação GPRS e wi-fi para acesso a serviços remotos; e uma tela grande e colorida, para que os dados possam ser exibidos de forma adequada para o usuário.

A plataforma s60 é uma plataforma dedicada para *smartphones*. Esta plataforma possui uma interface intuitiva e incorpora diversas tecnologias de comunicação. A maioria dos dispositivos que utilizam a plataforma s60 possui as seguintes características:

- Tela colorida com no mínimo 176 X 208-pixel de resolução;
- Aplicações de gerenciamento de informações pessoais (como contatos e calendário);
- Características avançadas de telefonia;
- Mensagem SMS e MMS;
- GSM e GPRS;
- Câmera e aplicações multimídia;
- Sistema operacional Symbian OS⁵;
- *Bluetooth*.

O sistema operacional Symbian possui gerenciamento de memória e característica multitarefa, que permite uma operação segura e eficiente sob condições de recursos reduzidos. Esse sistema operacional é aberto, ou seja, permite o desenvolvimento de software de forma livre em diversas linguagens de programação, incluindo a linguagem de programação Python, a qual foi utilizada. Esta linguagem de programação é bastante versátil, possui uma sintaxe limpa e permite que a utilização do paradigma de programação estruturada e também orientada a objetos.

Como o software foi criado utilizando o paradigma de programação orientada a objetos, um diagrama de classes UML (do inglês *Unified Modeling Language*) foi utilizado, sem a descrição dos métodos, para uma melhor visualização do modelo de classes do software. Este diagrama é ilustrado na Figura 4.1.

⁵[http:// www.symbian.com](http://www.symbian.com)

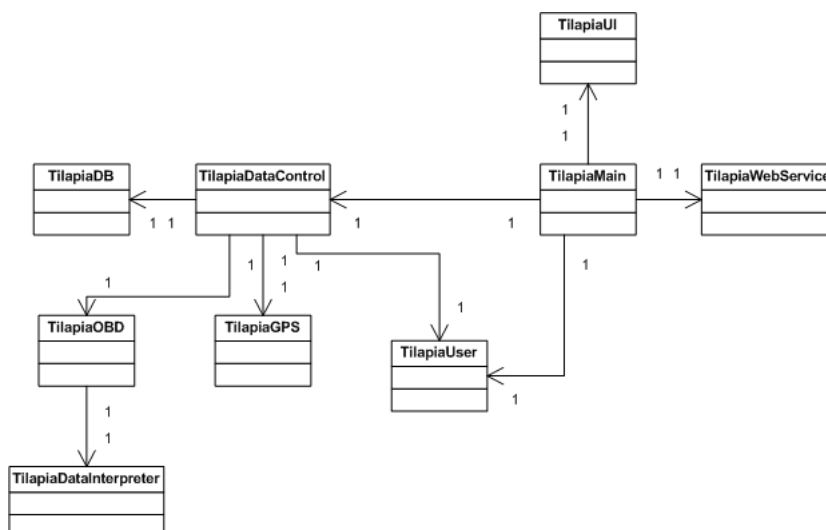


Figura 4.1: Diagrama UML do software criado para o *smartphone*.

A classe *TilapiaDataControl* tem a função de obter e tratar dados do automóvel, dados de localização GPS, dados de perfil de usuário e persistir esses dados em um banco de dados. Para isso, ela utiliza as classes: *TilapiaOBD*, *TilapiaGPS*, *TilapiaUser* e *TilapiaDB*, respectivamente.

A classe *TilapiaOBD* tem como função requisitar dados do automóvel e tratá-los para, então, poderem ser utilizados pela classe *TilapiaDataControl*. Para requisitar os dados do automóvel, a classe *TilapiaOBD* utiliza a API de *Bluetooth* de Python para *s60*. Com a utilização do protocolo RFCOMM⁶, é possível emular uma comunicação serial utilizando *Bluetooth*. Como o dispositivo *Bluetooth*, que está ligado ao *scantool*, provê um serviço de comunicação serial, basta enviar os PIDs do protocolo OBDII necessários. Neste caso, foram utilizados dois PIDs: o PID referente à velocidade do automóvel (hexadecimal 010D) e o PID referente ao nível de combustível no tanque (hexadecimal 012F). A obtenção da velocidade do automóvel é necessária para que não seja permitida a exibição de dados com o automóvel em movimento, caso o usuário seja o motorista. Já o nível de combustível no tanque é necessário para se determinar quando o sistema deve avisar ao usuário o momento certo de procurar um posto de combustível. Após a requisição, o *scantool* envia os dados do automóvel, porém esses dados recebidos são bytes. Para convertê-los para alguma grandeza física real como, por exemplo, quilômetros por hora, a classe *TilapiaOBD* utiliza a classe *OBDResponseInterpreter*. A classe *TilapiaGPS* tem como função requisitar dados de localização GPS. Para isso, ela utiliza a API de localização de Python para *s60*. A classe *TilapiaUser* é responsável por receber os dados de perfil de usuário da classe *TilapiaMain* e tratá-los, ou seja, filtrar as informações para que as mais relevantes sejam utilizadas pela classe *TilapiaDataControl*. A classe *TilapiaDB* tem

⁶<http://www.palowireless.com/infotooth/tutorial/rfcomm.asp> acessado em 31/10/2008

como função persistir todos os dados obtidos e tratados pela classe `TilapiaDataControl`. Ela utiliza a API de banco de dados de Python para `s60`.

A classe `TilapiaMain` é a classe principal da aplicação, ou seja, ela é o ponto inicial da aplicação e é responsável por controlar as ações do usuário e por exibir a interface gráfica com o usuário. Essa classe utiliza as classes: `TilapiaUser`, `TilapiaUI` e `TilapiaWebServices`. Após receber comandos do usuário, a classe `TilapiaMain` utiliza a classe `TilapiaUI` para requisitar a exibição de um formulário. Depois de preenchidos, os dados do formulário são então enviados para a classe `TilapiaDataControl` por intermédio da classe `TilapiaUser`. A classe principal também utiliza a classe `TilapiaDataControl` para acessar os dados do usuário, os dados do automóvel e os dados de localização GPS. Os dados que estão armazenados no banco de dados também podem ser acessados. Assim, essa classe possui a informação referente à velocidade e ao nível de combustível no tanque, podendo então informar ao usuário que ele procure um posto de combustível. Essa informação é dada em modo sonoro se o usuário for o motorista e se o carro estiver em movimento. Caso contrário, a informação é dada em modo visual. A classe `TilapiaWebServices` é responsável por permitir que a classe `TilapiaMain` acesse serviços providos remotamente. Ela utiliza a biblioteca `SOAPpy`⁷, a qual utiliza o protocolo soap e é feita para utilização na linguagem de programação Python.

4.3 Módulo do servidor remoto

O servidor remoto é composto por um computador ligado à Internet a cabo e à Internet sem fio. Assim, é possível que ele proveja serviços para dispositivos próximos utilizando a Internet sem fio ou para dispositivos distantes utilizando qualquer tecnologia de comunicação para acesso à Internet. Este módulo também possui um software, o qual está representado em diagrama UML na Figura 4.2 e é descrito a seguir.

A classe `ServerManager` é a classe principal, pois ela possui todas as outras classes e é o ponto inicial do software do servidor remoto, além de gerenciar todo o processo. Essa classe possui as classes: `ConnectionManager`, `UserDB`, `ServicesDB`, `ServicesProvider` e `ServicesInterface`. A classe `ConnectionManager` é responsável por gerenciar as conexões com os usuários. Ela espera por uma conexão e, quando o cliente se conecta, ela cria um novo objeto da classe `User`, a qual contém informações referentes à conexão com o usuário, de forma que a gerência da conexão com esse cliente não bloqueie a espera por conexão de outro cliente. Esta funcionalidade é provida através do uso de *threads*. A classe `User` possui um identificador para cada cliente, o qual é enviado para a classe `ServerManager` para que as informações do usuário possam ser persistidas, com o uso da classe `UserDB`,

⁷<http://pywebsvcs.sourceforge.net/> acessado em 31/10/2008

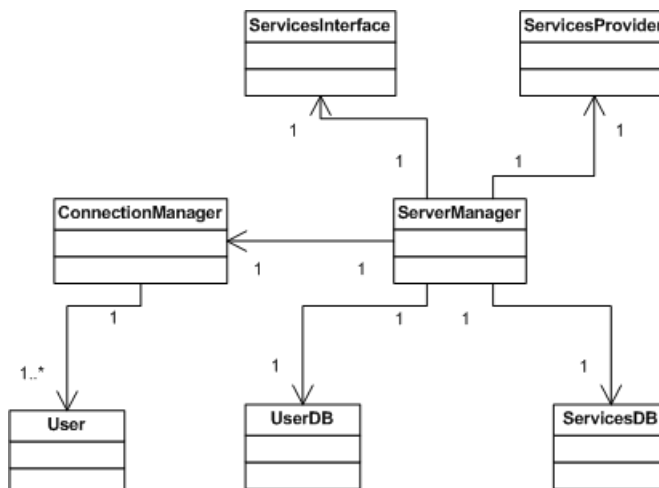


Figura 4.2: Diagrama UML do software criado para o servidor.

e para que possam ser requisitadas de forma mais fácil. A classe *ServicesInterface* é responsável por prover serviços via *Web services* para que um administrador do servidor remoto possa cadastrar novos serviços. Os novos serviços cadastrados são enviados à classe *ServicesDB*, para que possam ser persistidos e, posteriormente, são providos pela classe *ServicesProvider*.

4.4 Interface gráfica com o usuário

A interface gráfica com o usuário possui quatro estados. O primeiro estado é referente ao início da aplicação. Nele é exibida uma tela com instruções iniciais de uso e um menu. O menu possui três opções: conectar, criar perfil e sair, como é ilustrado na Figura 4.3.



Figura 4.3: Menu inicial da interface gráfica com o usuário.

Para que o usuário possa se conectar, é necessário que ele possua um perfil de usuário armazenado na aplicação, caso contrário, a aplicação exibe uma mensagem de erro, como é ilustrado na Figura 4.4.

Para criar um perfil de usuário, basta que a opção “criar perfil” do menu seja selecio-

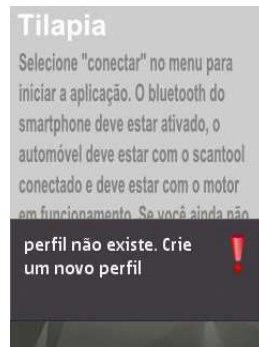


Figura 4.4: Mensagem de erro exibida quando não existe perfil.

nada. Assim, um formulário será exibido para que o usuário o preencha. O formulário é ilustrado na Figura 4.5.



Figura 4.5: Formulário para criação de perfil.

Com o formulário preenchido e o perfil de usuário criado, o usuário pode então se conectar. Para isso, basta que ele selecione a opção “conectar” no menu. Após a seleção, a interface muda de estado, ou seja, ela passa para o segundo estado, que é o estado de conexão, o qual é ilustrado na Figura 4.6. Nesse estado, a aplicação realiza a busca por dispositivos Bluetooth e exibe a lista dos dispositivos que podem ser selecionados. O usuário então deve selecionar o dispositivo que representa o *scantool*.

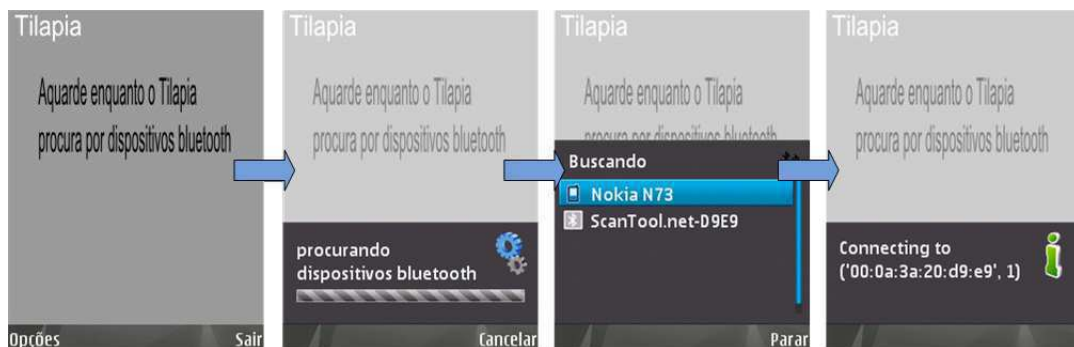


Figura 4.6: Estado de conexão.

Após a seleção, uma mensagem de espera é exibida. Nesse momento a aplicação está procurando por um servidor remoto, ao qual ela se conectará automaticamente. Se o servidor remoto não for encontrado ou se algum erro de conexão ocorrer, uma mensagem de erro será exibida. Caso contrário, uma mensagem de confirmação será exibida. A procura por um servidor remoto, a mensagem de erro e a mensagem de confirmação são, ilustradas na Figura 4.7.



Figura 4.7: Mensagem de espera para procura de servidor.

Após realizada a conexão com o *scantool* e com o servidor remoto, a aplicação passa ao estado de exibição de localização. Nesse estado, exibe-se uma tela contendo: um mapa, o qual é obtido utilizando o Google Maps⁸; um indicador para a localização atual do usuário, o qual é obtido a partir do sistema de localização GPS; e indicadores de localização de postos de combustível, os quais têm suas informações fornecidas pelo servidor remoto. O menu desse estado possui três opções: desconectar, gerar rota e sair. Ao selecionar a opção “desconectar”, o usuário faz com que a aplicação volte ao estado inicial. Já a opção “gerar rota” exibe uma caixa de entrada de texto para que o usuário possa indicar uma localização, como é ilustrado na Figura 4.8. A opção sair, faz com que a aplicação seja desconectada do *scantool* e do servidor remoto e volte ao estado inicial.

A partir do momento que o usuário indica uma localização ou o nível de combustível fica abaixo de um patamar determinado pelo usuário, a aplicação muda para o estado de

⁸<http://code.google.com/apis/maps/documentation/>



Figura 4.8: Menu para geração de rota.

exibição de rota. Nesse estado, um texto contendo a rota a ser seguida é exibido, como ilustrado na Figura 4.9.

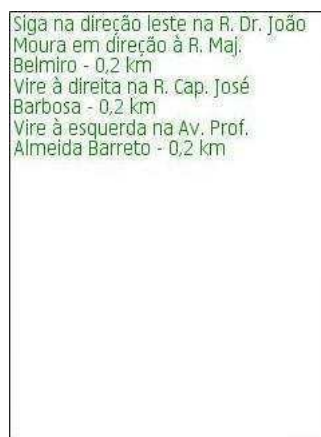


Figura 4.9: Rota gerada.

No caso em que o combustível está acabando, a aplicação escolhe a rota para o posto de combustível mais próximo ou para o posto em que o combustível é mais barato, a depender da preferência do usuário. As informações referentes à localização do posto, ao preço do combustível e ao nome do posto são obtidas do servidor remoto. No caso em que a opção é pelo posto em que o combustível é mais barato, a aplicação calcula se o combustível presente no tanque é suficiente para que o destino seja alcançado. Nesse estado, as opções de menu são as mesmas do estado anterior. Porém, ao selecionar a opção de menu “sair”, a aplicação volta para o estado de exibição de localização.

O último estado se refere ao estado de segurança. Nesse estado, apenas um texto indicando que o motorista pare o automóvel é exibido. Esse estado é alcançado apenas quando o usuário da aplicação é o motorista e quando o automóvel está em movimento.

4.5 Resultados dos testes realizados

Os testes foram realizados nas ruas da cidade de Campina Grande. Para isso, foram cadastrados três postos de combustível com seus respectivos nomes, localização e preço, em um servidor remoto. O posto Milênio, localizado na Avenida Prefeito Severino Cabral, foi cadastrado com o preço da gasolina em R\$2,61. O posto Marília, localizado na Avenida Professor Almeida Barreto, foi cadastrado com o preço da gasolina em R\$2,58. O posto Anel do Brejo, localizado na BR-230 próximo ao Aeroporto João Suassuna, foi cadastrado com o preço da gasolina em R\$2,36.

Para realização dos testes, foi utilizado um automóvel Fiat Palio 1.0, ano 2006. Esse automóvel possui o tanque de combustível com volume de 56l, percorre 9km por litro e é movido à gasolina.

Para verificar o funcionamento do servidor remoto, o software foi utilizado para requisitar um mapa contendo os postos de combustível cadastrados em suas respectivas localizações. Essa requisição foi feita por dois *smartphones* simultaneamente para testar o gerenciamento de conexões múltiplas do servidor remoto. O mapa obtido é ilustrado na Figura 4.10.



Figura 4.10: Mapa contendo a localização dos postos de combustível.

Para facilitar os testes, o posto Milênio foi removido do servidor. Assim, apenas dois postos foram considerados, sendo o posto Anel do Brejo o mais distante e mais barato e o posto Marília o mais próximo e mais caro. O ponto de partida para todos os testes foi a Rua Aprígio Veloso, em frente à Universidade Federal de Campina Grande, como ilustrado na Figura 4.11.

No total foram realizados três testes. O primeiro teste foi feito com o tanque de combustível do automóvel contendo 5,2% de seu volume total de gasolina. As preferências do usuário foram: usuário sendo o motorista, posto de combustível escolhido como sendo



Figura 4.11: Ponto de partida para os testes.

o mais próximo e o patamar para aviso de pouco combustível com valor de 5%. No ponto da Rua Rodrigues Alves próximo à Rua Alta Leite, o nível de combustível no tanque alcançou o patamar de 5%, fazendo com que o software emitisse um aviso sonoro para que o motorista parasse o automóvel para poder visualizar as informações exibidas. Como o posto de combustível mais próximo cadastrado foi o posto Marília, uma rota foi gerada para esse posto. Essa rota é ilustrada na Figura 4.12.



Figura 4.12: Resultado do primeiro teste.

O segundo teste foi feito novamente com o tanque de combustível do automóvel contendo 5,2% de seu volume total de gasolina. As preferências do usuário foram: usuário sendo o motorista, posto de combustível escolhido como sendo o mais barato e o patamar para aviso de pouco combustível com valor de 5%. Mais uma vez, no ponto da Rua Rodrigues Alves próximo à Rua Alta Leite, o nível de combustível no tanque alcançou o patamar de 5%, fazendo com que o software emitisse um aviso sonoro para que o motorista parasse o automóvel para poder visualizar as informações exibidas. Como o posto de combustível mais barato cadastrado foi o posto Anel do Brejo, uma rota foi gerada

para esse posto. Essa rota é ilustrada na Figura 4.13.



Figura 4.13: Resultado do segundo teste.

O terceiro e último teste foi feito com o tanque de combustível do automóvel contendo 1,2% de seu volume total de gasolina. As preferências do usuário foram: usuário sendo o motorista, posto de combustível escolhido como sendo o mais barato e o patamar para aviso de pouco combustível com valor de 1%. Ao chegar no ponto da Rua Rodrigues Alves próximo à Rua Alta Leite, o nível de combustível no tanque alcançou o patamar de 1%, fazendo com que o software emitisse um aviso sonoro para que o motorista parasse o automóvel para poder visualizar as informações exibidas. Como o posto de combustível mais barato cadastrado foi o posto Anel do Brejo, uma rota deveria ser gerada para esse posto. Porém, foi calculado que a quantidade de combustível no tanque do automóvel não era suficiente para que o posto mais barato fosse alcançado. Assim, uma rota para o posto Marília foi gerada. Essa rota é ilustrada na Figura 4.14.



Figura 4.14: Resultado do terceiro teste.

Os testes foram realizados apenas para o caso em que o usuário é o motorista, pois para o caso em que o usuário é o passageiro não há diferença significativa que justifique a repetição dos testes, já que a única diferença é que as informações serão exibidas em todos os momentos do uso. Os valores utilizados nos testes foram aproximados, pois as medidas utilizadas para o patamar de aviso de pouco combustível e de rendimento do automóvel não puderam ser determinadas com exatidão. Porém, mesmo com valores aproximados, os resultados foram bastante satisfatórios e atenderam as expectativas.

Capítulo 5

Conclusões e trabalhos futuros

O aumento da quantidade de automóveis nas ruas é inevitável. Medidas como construção de viadutos, criação de novas vias e melhorias no transporte coletivo, não são suficientes para reduzir o caos no trânsito das grandes cidades. Muitos estudos são feitos para tentar reduzir os efeitos do aumento da quantidade de veículos nas cidades, tanto para o trânsito como para o meio ambiente. A telemática de transporte é uma das áreas mais estudadas para reduzir esses efeitos.

Nesse trabalho foi feita uma análise de vários outros trabalhos sobre telemática de transporte. A partir dessa análise, foi criada uma infra-estrutura que viesse a reduzir os problemas ocorridos para a implantação e popularização dos sistemas de telemática de transporte. Para isso, aspectos como computação em ambientes automotivos, computação em dispositivos móveis, computação pervasiva, entre outros, foram estudados e integrados à infra-estrutura criada.

Para validação da infra-estrutura criada, foi desenvolvido um software, chamado de Tilapia, e foram feitas simulações com a utilização de um automóvel, de um *smartphone* e de um servidor remoto. Assim, identificou-se que é possível criar sistemas de telemática de transporte de baixo custo, expansíveis e de potencial popularização.

Como trabalho futuro, espera-se que o software seja aprimorado para que as rotas geradas sejam exibidas diretamente no mapa. Além disso, espera-se que novos serviços sejam adicionados como, por exemplo, serviços comerciais e governamentais. Também é esperado que os fabricantes de automóveis acrescentem em seus produtos sistemas de comunicação que permitam a obtenção de forma mais fácil. Como exemplo, pode-se acrescentar um sistema que obtém dados do estado do automóvel e possua uma interface de comunicação *Bluetooth* para prover esses dados. Assim, só haveria a necessidade de um dispositivo móvel com capacidade de comunicação por *Bluetooth* para requisitar dados do automóvel.

Outros trabalhos podem ser realizados com relação à provisão de serviços. Serviços

comunitários podem ser criados, possibilitando que um usuário faça indicações para outros usuários. Como exemplo, pode-se ter um serviço através do qual um usuário informe a localização de ruas com buracos. Assim, outros usuários poderiam utilizar essa informação para trafegar por rotas que não passassem por essas ruas.

Referências Bibliográficas

- [1] Chul-Su Kim, Jong-Ik Kim, Woo-Yong Han, and Oh-Cheon Kwon. Development of open telematics service based on gateway and framework. *Advanced Communication Technology, 2006. ICACT 2006. The 8th International Conference*, 2:1349–1352, Feb. 2006.
- [2] Diego Bezerra, Joelma Silva, Danilo Freire, Hyggo Almeida, and Angelo Perkusich. Desenvolvimento de um software livre de videophone baseado em voip. *Anais VII Workshop de Software Livre*, v 1:229–236, 2006.
- [3] C. Bidiskian, V. Rasin, I. Boamah, P. Castro, A. Misra, J. Rubas, N. Villotrex, D.Yeh, H. Huang, and C. Simonds. Intelligent pervasive middleware for context-based and localized telematics services. *WMC'02*, pages 15–24, September 2002.
- [4] <http://www.car-to-car.org/> acessado em 31/10/2008. Car 2 car communication consortium, 2008.
- [5] <http://www.cartalk2000.net/> acessado em 31/10/2008. Cartalk, 2008.
- [6] <http://www.informatik.uni-mannheim.de/pi4/projects/FleetNet/> acessado em 31/10/2008. Fleetnet, 2008.
- [7] <http://www-c.inria.fr/cybercars2/overview/> acessado em 31/10/2008. Cybercars 2, 2008.
- [8] R. Collingridge D. Ward W. Millar. Consumer vehicle telematics - an emerging market where web services offer benefits. *BT Technology Journal*, 22(1):99–106, January 2004.
- [9] Daqing Zhang and Xiao Hang Wang. Osgi based service infrastructure for context aware connected homes. In *In 1st International Conference on Smart Homes and Health Telematics*, pages 81–88. IOS Press, 2003.

- [10] Ying Song, Qing quan Li, Bi jun Li, Jing hai Xu, and Nian bo Zheng. Real-time dynamic display on spatial data for car navigation service. *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 94–97, June 2006.
- [11] A. Karimi, J. Olsson, and J. Rydell. A software architecture approach to remote vehicle diagnostics. *Master Thesis in Informatics of Goteborg University. School of Business, Economics and Law*, page 90, October 2004.
- [12] Stephan Eichler. Anonymous and authenticated data provisioning for floating car data systems. *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on*, pages 1–5, Oct. 2006.
- [13] M. McMorrow. Telematics - exploiting its potential. *IEE Manufacturing Engineer*, 83(1):46–48, Feb.-March 2004.
- [14] D. Saha and A. Mukherjee. Pervasive computing: A paradigm for the 21st century. *28th IEEE Conference on Local Computer Networks*, pages 25–30, 2003.
- [15] M. Weiser. The computer for 21st century. *Scientific American*, 265(3):94–104, September 1991.
- [16] M. Satyanarayanan. Pervasive computing: vision and challenges. *Personal Communications, IEEE*, 8(4):10–17, Aug 2001.
- [17] E. Loureiro, G. Ferreira, H. Almeida, and A. Perrkusich. Pervasive computing: What is it anyway? *Ubiquitous and Pervasive Knowledge and Learning Management: Semantics, Social Networking and New Media to their full potential*, pages 1–34, 2007.
- [18] David Garlan, Daniel P. Siewiorek, and Peter Steenkiste. Project aura: Toward distraction-free pervasive computing. *IEEE Pervasive Computing*, 1:22–31, 2002.
- [19] N. Navet, Y. Song, F. Simonot-Lion, and C. Wilwert. Trends in automotive communication systems. *Proceedings of the IEEE*, 93(6):1204–1223, June 2005.
- [20] Liang Chu, Jiayun Gu, Minghui Liu, Jun Li, Yimin Gao, and M. Ehsani. Study on CAN Communication of EBS and Braking Performance Test for Commercial Vehicle. *Vehicle Power and Propulsion Conference, 2007. VPPC 2007. IEEE*, pages 849–852, Sept. 2007.
- [21] Thorsten Gerke. Developing a robust can physical layer implementation for in-vehicle networks using the robust design method. *Automotive Electronics, 2007 3rd Institution of Engineering and Technology Conference on*, pages 1–6, June 2007.

- [22] G. Cena and A. Valenzano. Performance analysis of byteflight networks. *Factory Communication Systems, 2004. Proceedings. 2004 IEEE International Workshop on*, pages 157–166, Sept. 2004.
- [23] <http://www.mostcooperation.com> acessado em 31/10/2008. Media oriented systems transport, 2008.
- [24] R. Cox. *Introduction to OBDII*, volume 1. THOMSON DELMAR LEARNING, 2006.
- [25] P. David. *OBDII Diagnostic*. Kotzig Publishing, Inc., 2002.
- [26] George H. Forman and John Zahorjan. The challenges of mobile computing. *Computer*, 27(4):38–47, 1994.
- [27] Jon W. Mark and Weihua Zhuang. *Wireless Communications and Networking*. Prentice Hall, 2003.
- [28] J. Eisenstein, J. Vanderdoncki, and A. Puerta. Adapting to mobile contexts with user-interface modeling, 2000.
- [29] G. Bilchev, D. Marston, N. Hristov, E. Peytchev, and N. Wall. Traffimatics - intelligent co-operative vehicle highway systems. *BT Technology Journal*, 22(3):73–83, July 2004.
- [30] C. Kim, J. Kim, W. Ham, and O. Kwon. Development of open telematics service based on gateway and framework. *ICACT2006*, pages 1349–1352, February 2006.
- [31] D. Reilly and A. Taleb Bendiab. A rapid prototyping approach for the design of extensible in-vehicle telematics systems. *J. Integr. Des. Process Sci.*, 6(2):91–106, 2002.
- [32] Yilin Zhao. Telematics: safe and fun driving. *Intelligent Systems, IEEE*, 17(1):10–14, Jan/Feb 2002.
- [33] D. Reilly and A. Taleb-bendiab. A service-based architecture for in-vehicle telematics systems. In *Distributed Computing Systems Workshops, 2002.Proceedings.22nd International Conference on 2002*, pages 741–742.
- [34] Thomas Bauer, Jens Herrmann, Peter Liggesmeyer, and Christopher Robinson-Mallett. A flexible integration strategy for in-car telematics systems. *SIGSOFT Softw. Eng. Notes*, 30(4):1–7, 2005.

- [35] Jungsook Kim, Jungdane Choi, and ByungTae Jang. The next generation telematics service platform in ubiquitous environments. *Advanced Communication Technology, The 9th International Conference on*, 2:1263–1266, Feb. 2007.
- [36] Qing quan Li, Nian bo Zheng, Zhi xiang Fang, Jing hai Xu, Wen wu Gao, and Ying Song. A proposed intelligent vehicle telematics service system (ivts). *ITS Telecommunications Proceedings, 2006 6th International Conference on*, pages 866–869, June 2006.