

Centro Universitário de Brasília - UNICEUB
Faculdade de Ciências Exatas e Tecnologia - FAET
Engenharia de computação



SISTEMA DE GERÊNCIA PARA TRANSPORTE COLETIVO UTILIZANDO TRANSMISSÃO CELULAR

Brasília - DF

2005

Centro Universitário de Brasília - UNICEUB
Faculdade de Ciências Exatas e Tecnologia - FAET
Engenharia de computação

SISTEMA DE GERÊNCIA PARA TRANSPORTE COLETIVO UTILIZANDO TRANSMISSÃO CELULAR

por

JUAREZ DE CARVALHO
2001597-9 – FAET - UNICEUB

Trabalho Final de Graduação

Prof. Msc. Francisco Javier de Obaldía Díaz

Orientador

Brasília/DF – junho de 2005.

Agradecimentos

A Deus, que sempre esteve adiante de mim, como fez com o seu povo, durante o dia, numa coluna de nuvem para me guiar pelos caminhos; durante a noite, numa coluna de fogo, para me alumiar. Assim, caminhei nesta jornada, de dia e de noite, e graças a Deus chego ao final.

A minha esposa Maria, por estar sempre ao meu lado, pelo seu imenso amor e paciência, sem a qual eu jamais teria conseguido.

Aos meus filhos Arthur, Bernardo e Fernanda, meus tesouros.

Aos meus pais, que descansam no descanso do Senhor. A eles tudo devo.

Aos meus irmãos e a toda a minha família. Essa vitória é nossa.

Ao professor Javier, que não me deixou desanimar mesmo nos momentos de maior dificuldade.

Resumo

Desenvolver um Sistema de gerência de transporte público coletivo, como uma proposta para modernizar o sistema atual, introduzindo uma abordagem diferente das que hoje operam, buscando, como objetivo principal, a melhoria dos níveis de satisfação do usuário final com o sistema. Propondo, para isso, uma maneira de levar ao usuário final informações on-line sobre o tráfego da linha de transporte desejada, como tempo estimado de chegada do veículo da rota ao ponto no qual se encontra esperando.

Para atingir os objetivos propostos, o sistema fará uso de ferramentas de tecnologias distintas, tais como GPS (Sistema de Posicionamento Global), celular GPRS, S.O Linux, entre outras, buscando sempre, como premissa básica a economia de custos de implementação e custos operacionais. Como documentação de implementação será apresentado todos os softwares desenvolvidos durante a implementação do protótipo de validação do projeto, bem como os resultados das simulações feitas em campo.

Palavras-Chaves: Transporte público, GPS, GPRS, Telefone Celular.

Abstract

To develop a Urban Transportation Management System, as a suggestion to optimize the operation of the actual system, introducing a different abordage. The main purpose of the project is to increase the satisfaction of urban transportation users. The project propose a way to make information on-line about the traffic of a specific urban transportation route available to the final system users, information such estimated time to arrive in a specific bus stop for example.

To reach this objective, the system will make use of different technologies from different sectors, such as GPS (Global Position System), GPRS mobile phone, Linux operational system and others. Such documentation will be presented all developed software and the results of field tests and simulations.

Key Word : Urban Transportation, GPS, GPRS, mobile phone.

Sumário

CAPITULO 1 - INTRODUÇÃO	11
CAPITULO 2 - PLATAFORMAS DE HARDWARE E SOFTWARE	13
2.1 - A PLATAFORMA GPS	13
2.1.1 - A CONSTITUIÇÃO DO SISTEMA GPS	14
2.1.2 - O SEGMENTO ESPACIAL.....	15
2.1.3 - O SEGMENTO DE CONTROLE	16
2.1.4 - O SEGMENTO DOS USUÁRIOS	17
2.2 - A CONEXÃO COM O MUNDO INTERNET.....	18
2.2.1 - A ARQUITETURA DA PLATAFORMA GPRS	19
2.2.2 - FUNCIONAMENTO DA REDE GPRS	21
2.2.3 - TERMINAIS GPRS	21
2.2.4 - CLASSES DE TERMINAIS MULTISLOT	22
2.3 - A LINGUAGEM DE PROGRAMAÇÃO PERL	23
2.4 - SISTEMAS OPERACIONAIS LINUX	24
2.4.1 - A DISTRIBUIÇÃO FEDORA CORE 3	25
2.5 - O PROTOCOLO WAP	26
2.5.1 - A LINGUAGEM DE PROGRAMAÇÃO WML.....	26
CAPITULO 3 – ARQUITETURA DO SISTEMA.....	28
3.1 - UNIDADE REMOTA.....	30
3.1.1 - DESENVOLVIMENTO DO MÓDULO GPRS DA UNIDADE REMOTA	30
3.1.2 - DESENVOLVIMENTO DO MÓDULO GPS DA UNIDADE REMOTA	32
3.1.3 - DESENVOLVIMENTO DO MÓDULO PROCESSAMENTO UNIDADE REMOTA	36
3.2 - UNIDADE DE CONSULTA.....	41
3.2.1 - DESENVOLVIMENTO DO MÓDULO CONEXÃO DE REDE DA UNIDADE DE CONSULTA	42
3.2.2 - DESENVOLVIMENTO DO MÓDULO DE PROCESSAMENTO DA UNIDADE DE CONSULTA	43
3.3 - SISTEMA DE GERÊNCIA DE REDE	49
3.3.1 - DESENVOLVIMENTO DO MÓDULO CONEXÃO DE REDE DO SISTEMA DE GERÊNCIA DE REDE.....	50
3.3.2 - DESENVOLVIMENTO DO MÓDULO DE PROCESSAMENTO DO SISTEMA DE GERÊNCIA DE REDE.....	50
CAPITULO 4 - VALIDAÇÃO DO SISTEMA	55
4.1 - DEFINIÇÃO DO PROBLEMA	55
4.2 - CÁLCULO E OBTENÇÃO DOS DADOS DE TRÂNSITO DA UNIDADE REMOTA.....	56
4.3 - DADOS SOBRE A ROTA	57
4.4 - PROBLEMAS ENFRENTADOS	58
4.5 - EXPERIMENTO EM CAMPO DA FUNCIONALIDADE DO SISTEMA	60
4.6 - CONSULTA VIA TERMINAL MÓVEL	63
CAPITULO 5 - CONCLUSÃO	65
BIBLIOGRAFIA	66
ANEXOS	67
ANEXO A - CÓDIGO FONTE DA UNIDADE DE CONSULTA	67
GPSCOMCONSULTA.PL	67

ANEXO B - CÓDIGO FONTE DA UNIDADE REMOTA	75
GPSCOM.PL.....	75
MONITORA.PL	84
ALARMES_DATABASE.PL.....	89
ALARMES_GPRS.PL	90
ALARMES_GPS.PL	91
ALARMES_SERIAL.PL	92
HELP.PL.....	93
SOBRE.PL	94
GPRS	95
GPRS-OFF	97
GPRS-ON.....	98
GPRS-ON.CONF.....	99
ANEXO C - CÓDIGO FONTE DO SISTEMA DE GERÊNCIA DE REDE.....	100
MOVEL.WML	100
MOVEL.CGI.....	101

Lista de Figuras

FIGURA 2.1 - SATÉLITE DO SEGMENTO ESPACIAL	15
FIGURA 2.2 - CONSTELAÇÃO DOS SATÉLITES GPS	15
FIGURA 2.3 - DISTRIBUIÇÃO DOS SATÉLITES GPS	16
FIGURA 2.4 - LOCALIZAÇÃO DOS SATÉLITES DO SEGMENTO DE CONTROLE.....	17
FIGURA 2.5 - EXEMPLO DE RECEPTOR GPS	18
FIGURA 2.6 - VISÃO GERAL GPRS.....	19
FIGURA 2.7 - TOPOLOGIA DETALHADA DE UMA REDE GSM/GPRS.....	20
FIGURA 2.8 - EXEMPLO DE MÓVEL GSM/GPRS	22
FIGURA 2.9 - CAMINHO DE UMA REQUISIÇÃO WAP	26
FIGURA 3.1 - DIAGRAMA ESQUEMÁTICO DO PROJETO	28
FIGURA 3.2 - DIAGRAMA MODULAR DA UNIDADE REMOTA.....	30
FIGURA 3.3 - FUNCIONAMENTO DO MÓDULO GPRS.....	31
FIGURA 3.4 - RECEPTOR GARMIN GPS 45XL.....	33
FIGURA 3.5 - CABO DE CONEXÃO DO GPS AO COMPUTADOR.....	33
FIGURA 3.6 - INTERFACE GRÁFICA DA UNIDADE REMOTA	38
FIGURA 3.7 - GERÊNCIA DE ALARMES NA UNIDADE REMOTA	39
FIGURA 3.8 - APRESENTAÇÃO DA OCORRÊNCIA DE UM ALARME NO RECEPTOR GPS	39
FIGURA 3.9 - MENU HELP DO SISTEMA DE GERÊNCIA DA UNIDADE REMOTA.....	41
FIGURA 3.10 - DIAGRAMA MODULAR DA UNIDADE DE CONSULTA	42
FIGURA 3.11 - FUNCIONAMENTO DO MÓDULO CONEXÃO DE REDE	43
FIGURA 3.12 - EXEMPLO DO MÉTODO DE CÁLCULO DA DISTÂNCIA	44
FIGURA 3.13 - DISTÂNCIA ENTRE DOIS PONTOS NA SUPERFÍCIE TERRESTRE	45
FIGURA 3.14 - INTERFACE GRÁFICA DA UNIDADE DE CONSULTA	46
FIGURA 3.15 - APRESENTAÇÃO DAS ROTAS DISPONÍVEIS AOS USUÁRIOS	47
FIGURA 3.16 - MENSAGEM DE ERRO APRESENTADA	47
FIGURA 3.17 - APRESENTAÇÃO DO RESULTADO DA CONSULTA	48
FIGURA 3.18 - DIAGRAMA MODULAR DO SISTEMA DE GERÊNCIA DE REDE	49
FIGURA 3.19 - DETALHAMENTO DAS APLICAÇÕES DO SISTEMA INTEGRADO AO SISTEMA DE GERÊNCIA	51
FIGURA 3.20 - FLUXO DO BANCO DE DADOS DO SISTEMA DE GERÊNCIA.....	52
FIGURA 3.21 - SIMULADOR DE CONEXÃO WAP UTILIZADO DURANTE O DESENVOLVIMENTO.....	53
FIGURA 3.22 - CAMINHO DE UMA REQUISIÇÃO DE CONSULTA VIA TERMINAL MÓVEL	54
FIGURA 4.1 - ROTA DE VALIDAÇÃO.....	55
FIGURA 4.2 - RESPOSTA AO USUÁRIO EM CASO DE INTERRUPÇÕES NO DESLOCAMENTO MUITO LONGAS	57
FIGURA 4.3 - DEMONSTRAÇÃO DA ESTRATÉGIA DE CÁLCULO	59
FIGURA 4.4 - STATUS DO SISTEMA DE GERÊNCIA DA UNIDADE REMOTA	61
FIGURA 4.5 - PONTO DA REALIZAÇÃO DA CONSULTA	62
FIGURA 4.6 - RESULTADO DE UMA CONSULTA	62
FIGURA 4.7 - INFORMAÇÃO DAS ROTAS DISPONÍVEIS AO USUÁRIO	63
FIGURA 4.8 - INFORMAÇÕES DAS PARADAS DISPONÍVEIS PARA A ROTA	64
FIGURA 4.9 - RESULTADO FINAL DA CONSULTA	64

Lista de Tabelas

TABELA 2.1 - CLASSES DE TERMINAIS DE ACORDO COM A CAPACIDADE DE SLOTS UTILIZADOS	22
TABELA 3.1 - CORRESPONDÊNCIA CONECTOR GPS GARMIN/DB9 PARA MODELO 45	34
TABELA 3.2 - FORMATO DA SENTENÇA NMEA \$GPRMC	36

Lista De Simbolos

ADSL – ASYMETRIC DIGITAL SUBSCRIBER LINE
BSC – BASE STATION CONTROLER
BTS – BASE TRANSMITION SYSTEM
GPRS – GENERAL PACKET RADIO SERVICE
GPS – SISTEMA DE POSICIONAMENTO GLOBAL - GLOBAL POSITIONING SYSTEM
GSM – GLOBAL SYSTEM MOBILE
HTML –HYPERTEXT MARKUP LANGUAGE
IP – INTERNET PROTOCOL
ISP – INTERNET SERVICE PROVIDER
KBPS – KILO BITS POR SEGUNDO
NMEA – NATIONAL MARINE ELETRONICS ASSOCIATION
PPP – POINT TO POINT PROTOCOL
PPS – PRECISE POSICIONING SERVICE
RF – RADIO FREQUENCE
RX – RECEPÇÃO
SPS – STANDARD POSICIONING SERVICE
TX – TRANSMISSÃO
WAP –WIRELESS APPLICATION PROTOCOL
WML –WIRELESS MARKUP LANGUAGE
WWW – WORLD WIDE WEB
2G – SEGUNDA GERAÇÃO
DoD – DEPARTAMENT OF DEFENSE

CAPÍTULO 1 – INTRODUÇÃO

Em um mundo moderno, com um número cada vez maior de automóveis particulares circulando pelas ruas, fazendo com que o trânsito em muitas cidades fique, a cada dia mais e mais caótico, torna-se imperioso a procura constante de soluções que busquem a melhoria da qualidade dos serviços prestados pelas empresas de transportes coletivos existentes, buscando sobretudo resgatar a confiança do usuário.

É do conhecimento de todos, principalmente daqueles que fazem uso constante do sistema de transporte público coletivo no Brasil, das deficiências que fazem parte de sua estrutura. Os problemas são inúmeros, e as tentativas de soluções muitas vezes deixam a desejar ou são onerosas demais, inviabilizando sua implementação.

Melhorar esses sistemas, fazer com que cheguem com a mesma qualidade a todos os pontos de uma região, é um desafio técnico e político de grande envergadura, e a busca de soluções que visem esse objetivo terá que se tornar em algum momento o alvo de políticas públicas prioritárias para o setor.

Buscando contribuir para alcançar esse objetivo, minha proposta com este projeto, é propor a implementação de um sistema inteligente de monitoramento remoto de linhas de transporte coletivo, coletando informações do sistema de transporte e transmitindo essas informações ao usuário de maneira rápida, eficiente e barata.

Propõe-se assim, um sistema que possa, a baixo custo, levar até o usuário final de forma dinâmica, informações sobre o posicionamento da rota desejada e o tempo que deverá ficar esperando em uma determinada parada por um veículo de uma determinada rota. Estas informações são de extrema importância para o usuário de transportes coletivos, uma vez que uma das maiores reclamações acerca do sistema é não se saber se o coletivo desejado está para chegar, vai demorar ou não vai passar. Tal indecisão leva muitos usuários a perderem o seu precioso tempo, muitas vezes horas, com a espera.

A minimização dos custos estará presente como um dos objetivos em cada fase da elaboração do projeto, e por esse motivo ele será todo baseado em plataformas de software livre. A simplicidade e eficiência das interfaces de interação com os usuários, também será uma busca constante.

Neste trabalho será apresentado, no Capítulo 2, um breve referencial teórico e técnico de todas as tecnologias utilizadas na busca de uma solução para

atingir os objetivos propostos acima, como minimização dos custos, simplicidade e eficiência, procurando sempre explicitar o motivo da adoção dessa ou daquela tecnologia, como premissa no desenvolvimento do Sistema de Gerência de Transporte Público Coletivo.

No Capítulo 3, será mostrado todo o desenvolvimento da idéia do projeto, a proposta em seus detalhes, todas as topologias e premissas escolhidas para serem adotadas na busca do desenvolvimento das partes e na integração em um todo simples, eficiente e de baixo custo de implementação e operação. Também será mostrada no Capítulo 3 a execução da construção de cada módulo do projeto, dentro das condições de elaboração adotadas anteriormente, todas as limitações enfrentadas, elaboração das interfaces gráficas, descrição de cada etapa seguida na implementação do modelo de protótipo adotado.

No Capítulo 4, serão apresentados os resultados da implementação do protótipo sugerido pelo projeto, os testes realizados em campo, os resultados coletados, as comparações feitas e as melhorias implementadas em função da análise dos resultados obtidos com as simulações.

Finalmente, as conclusões e sugestões para a evolução e implementações futuras.

CAPÍTULO 2 – PLATAFORMAS DE HARDWARE E SOFTWARE

Com o grande número de tecnologias em plataformas de hardware e software existentes hoje no mercado, serão apresentadas nesse capítulo as diferentes ferramentas adotadas para o desenvolvimento do projeto, tanto de hardware quanto de software, bem como a justificativa pela escolha de cada uma delas.

As principais ferramentas utilizadas no desenvolvimento do projeto aqui apresentado têm como referencial teórico algumas tecnologias que apresentamos nas próximas seções.

2.1 - A plataforma GPS

Ao procurar uma tecnologia de posicionamento global existente, que pudesse fornecer os dados de posicionamento espacial para o projeto a baixo custo e com alta precisão, foi encontrado o GPS – Global Positioning System.

A vantagem do uso do GPS está em ser o mesmo um sistema muito difundido e com um grande número de implementações técnicas em operação. Como exemplo pode-se citar sistemas de navegação marítima, navegação aérea e também sistemas de telecomunicações, que utilizam o GPS como ferramenta de coleta de informações de posicionamento global.

Também se tem como vantagem do GPS, uma grande disponibilidade de equipamentos comerciais a um custo não muito alto para os modelos mais simples.

Sempre foi de interesse científico do homem, ainda que primitivo, ter uma noção de onde se encontrava, dentro de um ambiente infinitamente superior. Expandir as fronteiras, registrar suas extensões sempre foi desejo dos imperadores da antiguidade.

E, com a necessidade, foram surgindo os primeiros instrumentos de orientação, que iam muito além da fornecida pelo sol, estrelas e pelos planetas.

A invenção da bússola pelos chineses por volta do ano 850 foi uma verdadeira revolução na história da orientação, principalmente na navegação. Instrumentos como o Astrolábio, inventado por volta de 150 A.C pelos gregos, o quadrante de Davis, o sextante, também ajudavam muito, porém, a determinação da posição exata de um navio em alto mar, precisar sua latitude e longitude ainda estava longe de uma solução definitiva [Galdino, 2000].

Mesmo com os melhores instrumentos de época, a solução de tal desafio ainda demorou muito para encontrar uma resposta que superasse as expectativas.

Esta resposta só surgiu junto com os anos 70 com a proposta do GPS (Global Positioning System) [Galdino, 2000]. O GPS, que é um sistema de radionavegação, que foi desenvolvido pelo Departamento de Defesa dos Estados Unidos da América. O propósito do sistema GPS era o de ser a principal ferramenta de navegação a ser utilizada pelas forças armadas americanas. Um exemplo de seu uso militar teve-se na Guerra do Golfo em 1991, onde já se tinham sistemas balísticos integrados com o sistema de posicionamento, e os ataques eram planejados a alvos com posição global conhecida, ou seja, latitude e longitude. O que se viu foi uma precisão no acerto nunca antes vista.

Logo após o início do projeto do sistema GPS, ocorreu um grande envolvimento de comunidades da sociedade civil, que seriam possíveis usuárias do sistema, tais como agricultura, navegação marítima, entre outras [Galdino, 2000].

Em sua proposta mais simples, o sistema GPS oferece ao usuário, que se encontra posicionado em qualquer ponto da superfície terrestre, ou a certa altitude, o mínimo de quatro satélites para rastreamento, permitindo assim um posicionamento global em tempo real.

Existem dois tipos de serviços, que são o SPS (Standard Positioning Service – Serviço de Posicionamento Padrão) e o conhecido como PPS (Precise Positioning Service – Serviço de Posicionamento Preciso) [Galdino, 2000].

O serviço padrão SPS está disponível a todos os usuários do globo, gratuitamente, e, até o dia 1 de maio de 2000, oferecia uma precisão entre 100 e 140 metros. Já o PPS, que possui melhores resultados, está restrito a uso militar e também a usuários autorizados, oferecendo uma precisão entre 1 e 20 metros.

Nos dias de hoje até o serviço SPS oferece níveis satisfatórios de precisão, pois, no dia 2 de maio de 2000 foi retirada a deterioração dos níveis de acurácia do SPS, melhorando em até 10 vezes a precisão [Galdino, 2000].

2.1.1 - A constituição do Sistema GPS

O sistema GPS é constituído por três segmentos distintos, que são o segmento espacial, o de controle e o segmento que é utilizado pelos usuários.

2.1.2 - O Segmento espacial

O segmento espacial, que é constituído por 24 satélites distribuídos em 6 planos orbitais a uma altitude aproximada de 20.200km, com um período orbital de aproximadamente 12 horas siderais, o que faz com que as posições dos satélites se repitam a cada dia, o que garante que no mínimo 4 satélites sejam visíveis em qualquer ponto da superfície terrestre [Galdino, 2000].

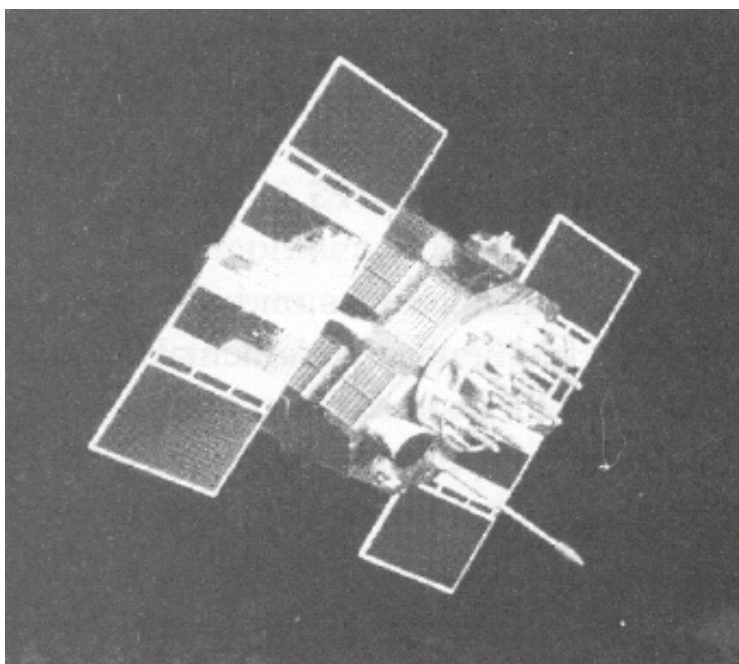


Figura 2.1 – Satélite do Segmento Espacial (Galdino, 2000)

A Figura 2.2 ilustra os satélites GPS em suas órbitas.

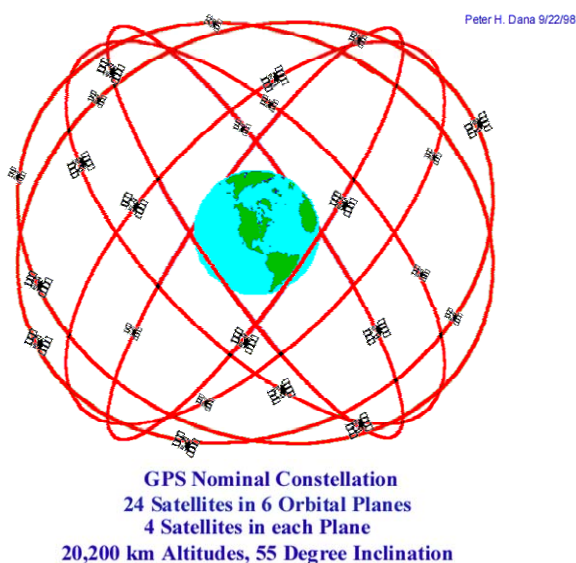


Figura 2.2 – Constelação dos satélites GPS.

(<http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>)

Na Figura 2.3, pode-se verificar a distribuição dos satélites GPS em cada um dos planos orbitais.

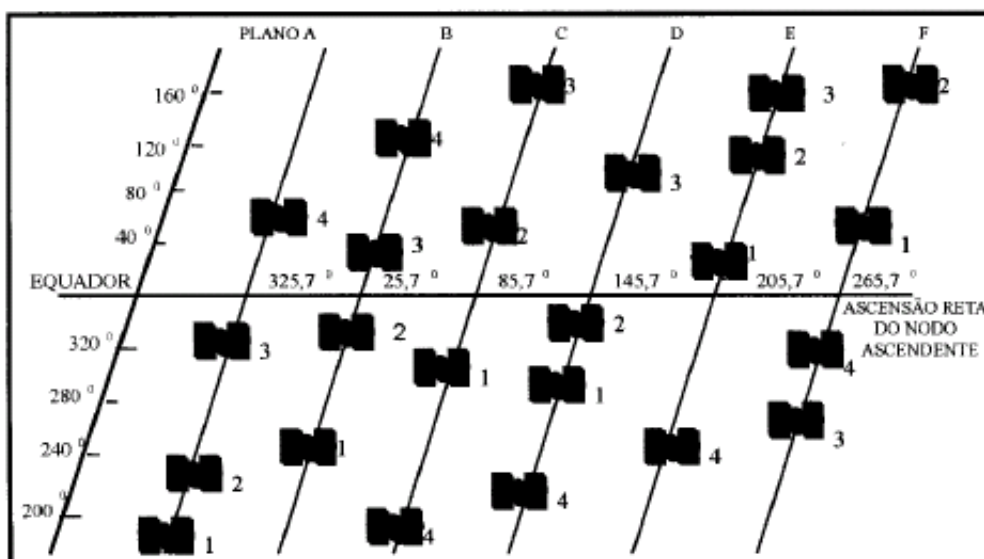


Figura 2.3 – Distribuição dos satélites GPS.

(<http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>)

Para estabelecer uma referência de tempo precisa cada satélite utiliza padrões de frequência altamente estáveis, utilizando para isso Césio e Rubídio, que possuem estabilidade de referência entre 10^{-12} e 10^{-13} por dia, o que oferece ao sistema GPS uma das melhores referências atuais de tempo, muito utilizada para sincronia de equipamentos de redes de telecomunicações [Galdino, 2000].

2.1.3 - O Segmento de Controle

É um segmento à parte do espacial, e possui as funções de monitorar e gerenciar o sistema espacial, determinar o seu sistema de tempo, proceder a correções no sistema de relógios dos satélites, dentre outras.

“O sistema de controle é composto por cinco estações monitoras (Hawaii, Kwajalein, Ascension Island, Diego Garcia, Colorado Springs), três delas com antenas para transmitir os dados para os satélites (Ascension Island, Diego Garcia, Kwajalein), e uma estação de controle central (MCS: Master Control Station) localizada em Colorado Springs, Colorado. Essas cinco estações de monitoramento pertencem à AAF (American Air Force); em conjunto com as sete do NIMA (National Imagery and Mapping Agency), compõem as estações monitoras GPS do DoD” [Galdino, 2000, 9].

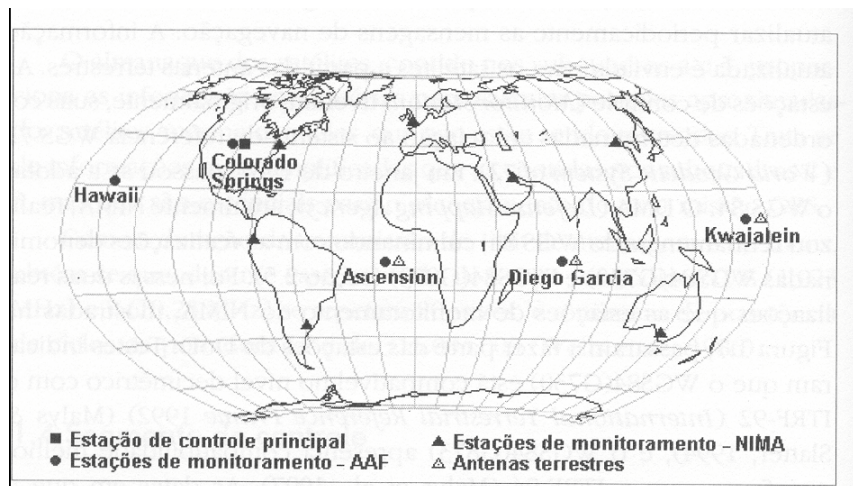


Figura 2.4 – Localização dos satélites do segmento de controle.
 (http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html)

2.1.4 - O Segmento dos Usuários

Este segmento é composto pelos receptores GPS, que são dos mais variados modelos, dependendo da finalidade a que se destinam. As aplicações dos receptores são as mais variadas, passando pela navegação, geodésica, posicionamento, uso militar dentre muitos outros [Galdino, 2000].

Não importando se o objeto monitorado se desloca eventualmente ou está em constante movimento.

Um receptor GPS, como o que será usado no presente projeto, é composto principalmente pelos seguintes módulos:

- **antena:** responsável pela detecção das ondas emitidas pelos satélites. É recomendável que as antenas sejam montadas com amplos ângulos de visada, para evitar a atenuação do sinal que já chega muito fraco ao receptor.
- **seção de RF:** fornece todo o tratamento de RF ao sinal recebido do satélite.
- **microprocessador de controle do receptor:** fornece o controle de todas as operações do receptor, tais como processar o sinal, cálculo de posição, velocidade e também as funções da interface com o usuário.
- **interface com o usuário:** é composta pelo display e pelos botões de comandos disponíveis para o usuário.
- **alimentação de energia:** os receptores mais modernos podem funcionar com pilhas comuns além de dispor de entrada para alimentação externa.

- **memória**: usado para armazenar os dados coletados e também os dados usados no setup inicial automático.



Figura 2.5 - Exemplos de receptores GPS.

2.2 – A conexão com o mundo Internet

No caso do sistema proposto, como o objeto a ser monitorado é um veículo normalmente em movimento, se fez necessário encontrar uma maneira de conectar o sistema de monitoramento, do qual faz parte o sistema GPS, com uma rede externa, preferencialmente usando a Internet e o protocolo IP, amplamente difundidos no mercado e de fácil configuração e conhecimento,

Com o advento da internet, é possível ampliar os pontos de monitoramento com o envio e recebimento de informações a um baixo custo de acesso para os usuários em um ambiente integrado com o sistema de monitoramento, ou seja, não é mais necessário, para operar um sistema como o proposto neste projeto, um alto investimento em desenvolvimento e equipamentos proprietários para se integrar uma rede com algumas dezenas de unidades monitoradas.

As redes conhecidas como fixas, que permitem acesso aos usuários por meio de cabos metálicos ponto a ponto ou fibras ópticas são uma alternativa de conexão. As redes wireless, em especial a rede celular que permite o acesso a redes externas e a comunicação sem fios com o uso de terminais celulares, também é uma outra alternativa.

A rede celular, utilizada nos sistemas móveis, já implantada pelas operadoras, mostrou-se como a alternativa mais viável, devido ao fato que com a rede fixa, o usuário é obrigado a estar no local onde está instalado o terminal, o que limitaria a mobilidade do coletivo monitorado e do usuário.

Os sistemas móveis de segunda geração (2G), como são conhecidos, são construídos com tecnologia 100% digital e trazem vantagens significativas com

relação a serviços oferecidos, dispondo o usuário de uma série de serviços agregados que pode fazer uso, dependendo de suas necessidades. Um exemplo de sistemas 2G é a rede celular construída com a tecnologia GSM.

O GSM é uma tecnologia 2G, porém, com o aumento da demanda para acesso móvel à Internet, levou ao desenvolvimento de tecnologias dentro da geração 2 de tecnologia móvel, fazendo assim surgir a geração 2,5G, que é o sistema 2G acrescido da tecnologia GPRS [GSM, 1998].

O sistema GPRS é um sistema baseado na tecnologia 2,5G de comunicações móveis capaz de realizar a comutação por pacotes, possibilitando o uso móvel de Internet a alta velocidade, como mostrado na Figura 2.6.

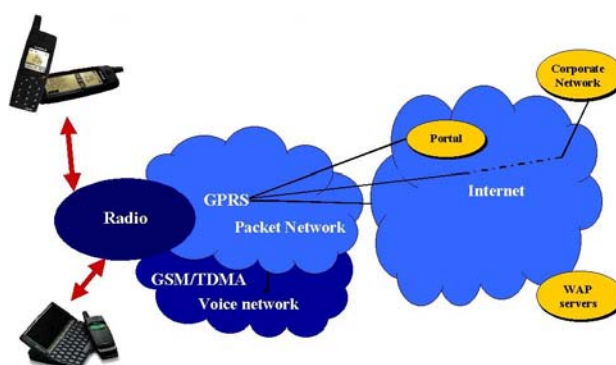


Figura 2.6 – Visão geral GPRS
(<http://www.wirelessdevnet.com/images/gpr 1>)

2.2.1 – A arquitetura da plataforma GPRS

O sistema de dados GPRS é um serviço adicional e se utiliza da rede celular baseada em tecnologia GSM. Ele é construído para suportar usuários finais que desejem acessar a internet, usando uma estação móvel como um dispositivo de conexão, com capacidade para tráfego de dados, fornecendo uma solução básica de comunicação com o protocolo IP.

Trata-se o GPRS de uma tecnologia de comutação por pacotes. Onde o circuito é estabelecido de fim a fim no momento em que é necessária a transmissão de informações. Adicionalmente, uma conexão diferente pode ser estabelecida com a transmissão de cada pacote [GSM, 1998].

No sistema GPRS, a transmissão de dados em pacotes é executada no modo fim a fim, o que inclui a interface aérea da rede celular GSM.

Por definição de norma, cada canal de RF na rede GSM, ocupa uma banda de 200KHz, dentro de uma estrutura de frame com 8 intervalos de tempo. Ao se estabelecer uma conexão de dados ou de voz, a utilização de um slot de tempo garante uma taxa de 9,6 Kbps [GSM, 1998].

Para se conseguir um aumento significativo nesta taxa de transmissão, a solução implementada foi a utilização simultânea de mais de um time-slot para a transmissão de dados.

Com a implementação da tecnologia GPRS, a transmissão de dados acontece através de vários time-slots. Por ser orientado a conexão, o GPRS consegue o estabelecimento de uma conexão permanente sem a necessidade de reserva constante de recursos, pois os mesmos só são alocados quando é necessária sua utilização. Este recurso permite que a tarifação seja efetuada somente pela utilização e não pelo tempo de conexão [GSM, 1998].

A configuração de uma rede GPRS pode ser feita de diferentes maneiras pelo operador, que pode reservar recursos dedicados ao tráfego GPRS ou dando prioridade ao tráfego de voz, sendo que nesse caso, o tráfego de dados irá ocupar somente os recursos livres no momento da utilização.

E, uma vez que os recursos só são alocados no momento da necessidade, a taxa conseguida pode variar a cada requisição de alocação de recursos.

A Figura 2.7, apresenta a arquitetura da rede GPRS dentro de uma rede celular GSM.

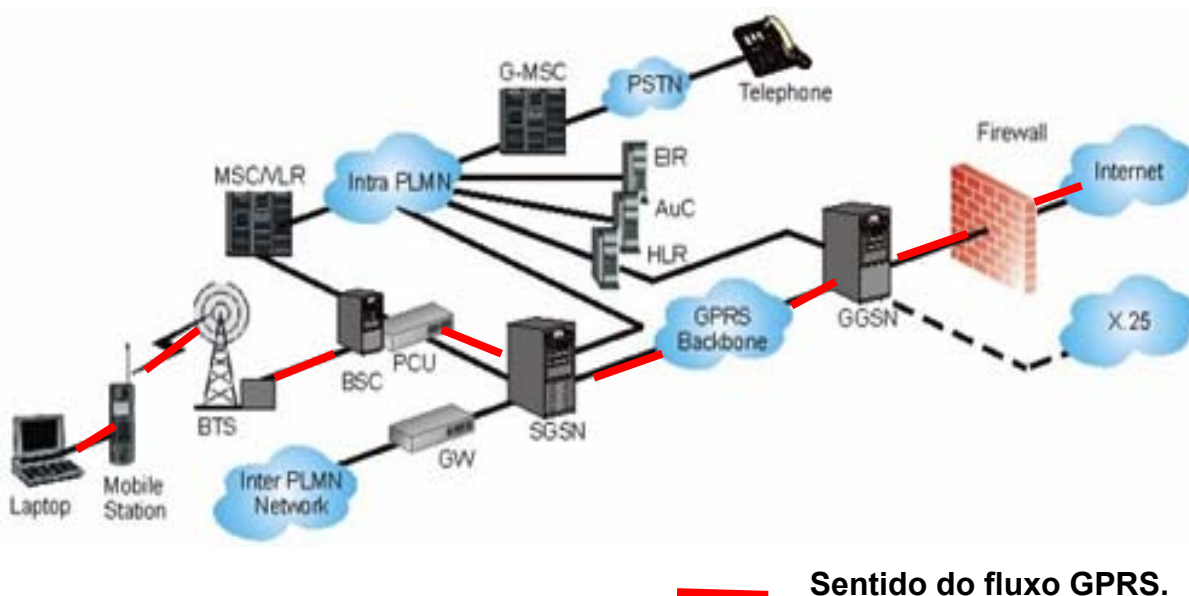


Figura 2.7 – Topologia detalhada de uma rede GSM/GPRS
(<http://www.radcom.com/radcom/test/images/gprs1.png>)

2.2.2 – Funcionamento da rede GPRS

O assinante ao fazer uma requisição de GPRS à rede GSM, executa o estabelecimento de uma conexão entre o seu móvel e a rede de dados, através de uma BTS, que é o módulo da rede que provê a comunicação entre o telefone móvel do assinante e o restante da rede. A BTS ao reconhecer o assinante como válido o encaminha para a BSC, que possui a função de gerenciamento de varias BTS dentro da rede.

A BSC ao reconhecer a chamada como sendo requisição de dados a encaminha para o próximo módulo da rede GPRS que é o responsável pelo gerenciamento e encaminhamento dos pacotes trafegados dentro da rede.

Neste ponto, os pacotes são enviados ao SGSN, cuja principal função dentro da rede GPRS é manter a conexão lógica ativa para os usuários móveis, quando em mobilidade, são deslocados para a área de cobertura de uma outra BTS, entrando no backbone GPRS. Neste ponto da rede, o pacote é entregue ao GGSN, módulo da rede que possui a função de prover a conexão com a Internet e também com outras redes de dados e ao sistema de bilhetagem.

2.2.3 – Terminais GPRS

Um terminal móvel para ter acesso a rede GPRS tem que possuir suporte a esse serviço. As normas GSM definem 3 classes de terminais, que são [GSM, 1998]:

- **Classe A** – São terminais que podem tratar dados e voz ao mesmo tempo.
- **Classe B** – Terminais que possuem a capacidade de tratamento de voz e dados, porém em tempos diferentes.
- **Classe C** – São terminais com capacidade para tratar somente dados, utilizando para isso cartões GPRS PCM/CIA para portáteis.



Figura 2.8 – Exemplo de Móvel GSM/GPRS

2.2.4 – Classes de terminais Multislot

De acordo com a norma GSM, os terminais são divididos em 29 classes, de acordo com a capacidade de utilização de slots na recepção e transmissão [GSM,1998].

Como exemplo se teria em um terminal Classe 8 (4 + 1), a capacidade de receber dados em 4 slots e de transmitir em 1 slot.

A tabela abaixo exemplifica as 8 classes de móveis mais comuns, e a sua respectiva capacidade de RX e TX [GSM, 1998]:

Classe de Multislot	Número máximo de slots		
	Rx	Tx	Soma
1	1	1	2
2	2	1	3
3	2	2	3
4	3	1	4
5	2	2	4
6	3	2	4
7	3	3	4
8	4	1	5

Tabela 2.1 – Classes de terminais de acordo com a capacidade de slots utilizados.

2.3 – A linguagem de programação Perl.

Ao buscar a linguagem principal para o desenvolvimento da parte lógica do projeto, se fez necessário encontrar uma linguagem de fácil programação, nativa no Linux, que possuísse recursos poderosos para tratamento de expressões regulares, que são recursos de programação avançados para pesquisa e tratamento de padrões de texto em arquivos, recurso muito usado pelos algoritmos de pesquisa nos sites de busca na Internet.

Também teria que ser uma linguagem com capacidade de conexão remota a banco de dados, que possuísse bibliotecas gráficas, fácil conexão a dispositivos seriais conectados ao computador, recursos matemáticos de programação, com capacidade para programação de conteúdo dinâmico para a WWW.

Após intensa pesquisa, baseada nos requisitos para o projeto do sistema proposto, encontrou-se o Java e o Perl.

A opção pela linguagem Perl deveu-se à natureza aberta do código fonte da linguagem.

Criada em 1986 quando um programador chamado Larry Wall, que trabalhava para a Agencia Nacional de Segurança americana, resolveu construir uma nova linguagem com alta capacidade para geração e formatação de relatórios [DEI, 2002].

Por ser uma linguagem com extrema flexibilidade, portabilidade e de grande eficiência, Perl ganhou muitos adeptos entre as comunidades de código aberto pelo mundo todo [DEI, 2002].

Trata-se de uma linguagem interpretada, aberta, que, por ter ganhado a simpatia de uma grande parte da comunidade internacional, possui hoje recursos que vão muito além dos iniciais projetados para tratamento de textos.

Ao chegar a sua versão 5, Perl sofreu uma grande reorganização, que eliminou muitas deficiências estruturais da linguagem, tornando-a poderosa, se tornando uma linguagem de programação de propósito geral, com um ambiente de desenvolvimento próprio [DEI, 2002].

Perl se tornou uma linguagem extensível por módulos, que são arquivos escritos na linguagem com extensão .pm, esses arquivos são desenvolvidos pela comunidade Perl em todo mundo, reunidos em centros de distribuição espalhados por todos os lugares. O principal deles é o Comprehensive Perl Archive Network (CPAN) – www.cpan.org, onde se encontram módulos, documentação, scripts e o

código-fonte propriamente dito, que podem ser baixados por meio de FTP anônimo [DEI, 2002].

“Perl é uma das linguagens de programação existentes mais altamente portáveis. Ela foi originalmente implementada em UNIX, mas desde então espalhou-se para muitas outras plataformas. Um programa Perl pode frequentemente ser portado de um sistema operacional para outro, sem nenhuma modificação, e ainda funcionar adequadamente” (Deitel, 2002, 51).

Por possuir todos os recursos necessários para o desenvolvimento do projeto, inclusive uma biblioteca gráfica, que será usada no desenvolvimento de todas as interfaces gráficas necessárias ao sistema, a linguagem Perl é a alternativa mais apropriada para ser utilizada com a principal ferramenta de programação para o desenvolvimento do projeto proposto.

2.4 – Sistema Operacional Linux

Para o Sistema Operacional usado em todas as unidades do projeto, a escolha natural foi o LINUX.

O Linux é um Sistema Operacional de código fonte aberto que possui suas origens no UNIX, deve-se a esse fato também ser conhecido como sistemas UNIX-LIKE. Por não ser fechado, não existe um Linux, mas uma grande quantidade de distribuições disponíveis na Internet, bancas de revista e em muitos outros lugares.

As distribuições do Linux diferenciam-se pelas suas limitações, objetivos, estabilidade, confiabilidade dentre outros fatores, e entre as mais populares encontram-se o Conectiva, Slackware, Mandrake, Fedora, Debian e muitas outras.

Nos dias de hoje, já se tem notícia que muitas empresas de grande porte estão migrando para o Linux, muitos bancos já estão usando o Linux em seus servidores.

Em muitas esferas do poder público no Brasil e em outros países, existem diversos projetos de migração de sistemas fechados e pagos como o Windows para o Linux, visando com isso a economia de gastos com o pagamento de licenças de uso e de suporte a software proprietários.

A escolha pelo Linux com Sistema Operacional deste projeto, deveu-se à tentativa de economia de custos em todas as fases e módulos do projeto, sem, contudo abrir mão da eficiência.

2.4.1 – A distribuição Fedora Core 3

Trata-se de um projeto de código fonte aberto, com o objetivo de desenvolver um Sistema Operacional completo, seguro, livre e sem custo de suporte.

O Fedora, que tem origem na distribuição RED HAT, é um sistema que tanto pode ser usado em desktop quanto em um servidor com a mesma eficiência. No ambiente operacional do projeto, era necessário um software que se comportasse bem tanto como cliente quanto como servidor, que fosse de fácil uso, com recursos avançados de programação nativos, banco de dados e com um excelente reconhecimento para periféricos novos conectados, como celulares GSM e o receptor GPS. Por esses motivos foi escolhida essa distribuição, além de atender aos requisitos de redução de custos de desenvolvimento do projeto por se tratar de software livre.

2.5 – O protocolo WAP

Para viabilizar a consulta às informações de tráfego das rotas monitoradas por intermédio de telefones celulares, a utilização do protocolo WAP apresentou-se como a alternativa mais viável, uma vez que o servidor http Apache, já nativo no Linux é capaz de responder a requisições WAP.

O WAP é uma tecnologia que torna possível o acesso à Internet a aparelhos celulares, PDA's entre outros, utilizando para isso a linguagem de programação WML.

Como é proposto pelo sistema em questão somente uma consulta simples, cuja totalidade do processamento ocorrerá na máquina do Sistema de Gerência de Rede e que receberá como retorno uma resposta de tamanho curto, no padrão de apresentação WML, cujo comprimento será perfeitamente compatível com a capacidade de exibição de qualquer móvel, não se tornando necessário, nesse caso, o desenvolvimento de uma aplicação exclusiva para processamento da consulta no próprio móvel.

Embora muitos fabricantes tentassem buscar o desenvolvimento de alternativas que provessessem o acesso de conteúdo Internet a celulares, essa iniciativas sempre esbarravam na incompatibilidade com os recursos já estabelecidos para a rede mundial.

EM 1997, empresas como Nokia, Motorola e Ericsson fundaram o chamado Wap Fórum, com o objetivo de estabelecer um conjunto de normas para o desenvolvimento de aplicações baseadas em Internet para celulares. Hoje o protocolo Wap encontra-se em sua versão 2.0.

A Figura 2.9 demonstra como funciona uma requisição Wap.

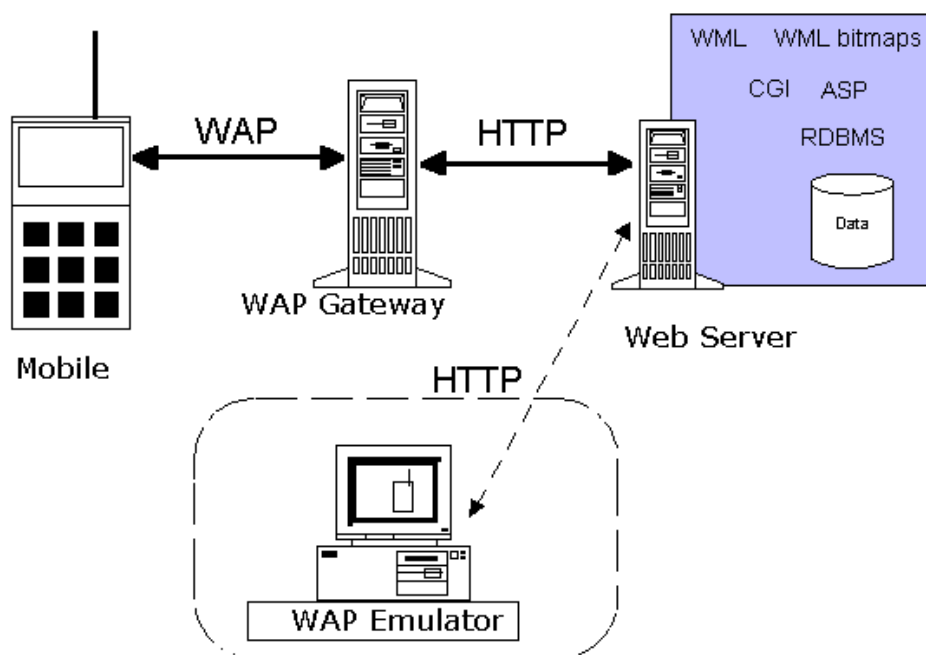


Figura 2.9 – Caminho de uma Requisição WAP
(<http://www.regione.emilia-romagna.it/wap/images/Image6.gif>)

2.5.1 – A linguagem de programação WML

O WML é uma linguagem de marcação que tem base no XML, criada especialmente com a finalidade de criar aplicações WEB que possam ser visualizadas em browser de telefones celulares, que possuem os recursos de navegação WAP instalados.

As páginas Internet construídas com o WML possuem muita semelhança com as escritas em HTML.

Trata-se de uma linguagem simples, de fácil entendimento e bastante eficiente, que mostrou atender a todos os requisitos e necessidades do projeto

nas consultas realizadas via terminais móveis, além de ser livre a sua utilização, e com ampla documentação, tanto na Internet como na literatura especializada.

Para o WML, todos os elementos e os tipos de dados válidos foram definidos pelo Wapforum, o que obriga a todos os programas WML a seguirem as normas estabelecidas pela especificação da linguagem.

Todas as tecnologias apresentadas nas seções anteriores deste capítulo, tanto de hardware quanto de software, serão utilizadas durante o desenvolvimento dos módulos e componentes deste projeto, conforme descrito no próximo capítulo.

CAPÍTULO 3 – ARQUITETURA DO SISTEMA

A idéia de utilizar o sistema celular integrado com o sistema GPS, e ao mesmo tempo, integrá-lo em um sistema de gerenciamento de posicionamento de veículos (coletivos) é até certo ponto pouco explorada, sendo que agora que começam a surgir alguns projetos comerciais com esse objetivo.

Nas próximas seções serão apresentadas as especificações dos módulos que compõem o sistema de gerenciamento proposto, a integração dos mesmos e as especificações do projeto.

Em particular, o modo como as interfaces serão integradas é novo, sendo, o principal objetivo do projeto, apresentando uma maneira não onerosa de se ter um sistema de posicionamento, e ao mesmo tempo, podendo ser utilizado para melhoria dos serviços prestados à população em geral.

Na Figura 3.1 é mostrado de maneira esquemática o funcionamento do sistema.

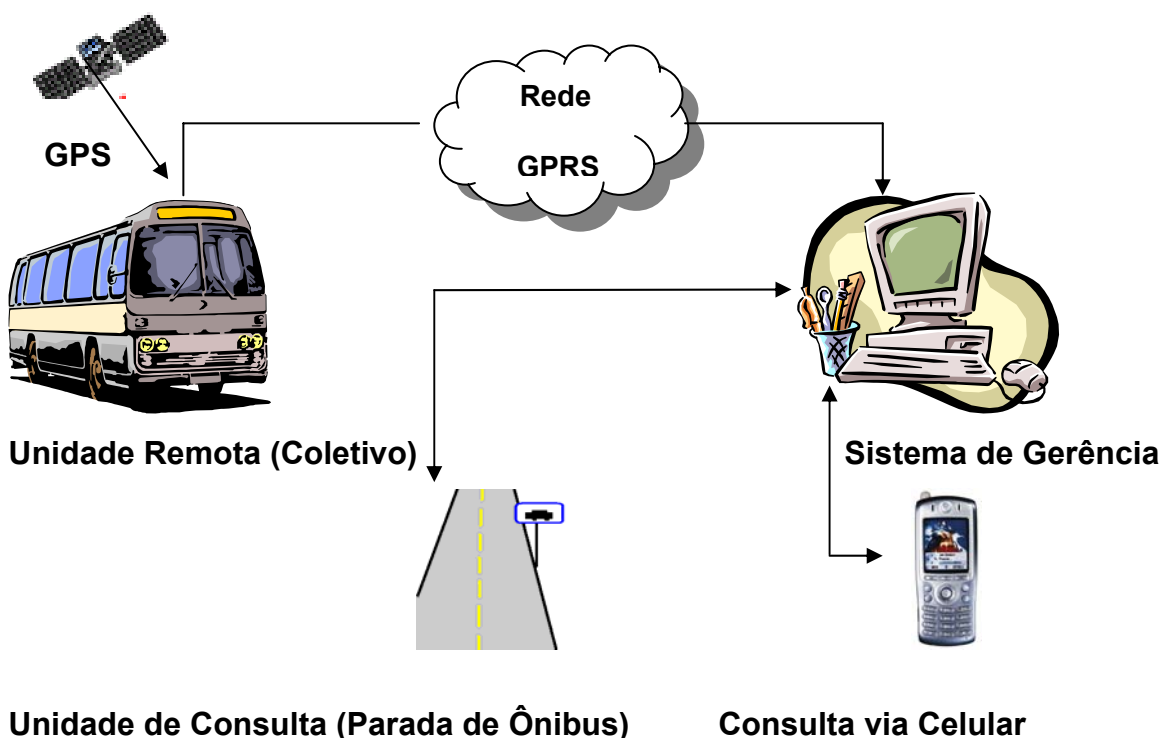


Figura 3.1 – Diagrama esquemático do projeto.

De maneira simplificada, o funcionamento do protótipo implementado pelo projeto é o seguinte: O veículo monitorado coleta informações de posicionamento através de um receptor GPS. Essas informações são armazenadas dinamicamente em um banco de dados servidor, através de uma conexão com a

Internet estabelecida por uma unidade móvel (celular) GPRS. Com os dados de posicionamento do veículo armazenados, um sistema de consulta montado em uma parada de ônibus, composto por um computador, software específico e uma conexão com a Internet. Esta conexão, que tanto pode ser através de um terminal móvel GPRS ou uma conexão física, como ADSL, por exemplo, coleta as informações de deslocamento do veículo no banco de dados do Sistema de Gerência de Rede e calcula o tempo aproximado de sua chegada à parada. Este sistema também pode ser consultado, ou seja, será provida pelo sistema, a consulta a partir de uma unidade móvel (celular), sem a necessidade do usuário se encontrar fisicamente na parada e obtendo o mesmo resultado.

A estratégia de desenvolvimento do protótipo cobrirá as seguintes unidades principais:

- **UNIDADE REMOTA (COLETIVO MONITORADO)** – Será chamada de UNIDADE REMOTA a parte do sistema que usará como base a linha de transporte coletivo regular, que terá seu tráfego monitorado pelo sistema. A UNIDADE REMOTA será formada por:
 - Receptor GPS;
 - Computador com o S.O Linux;
 - Telefone Celular GSM/GPRS.
- **SISTEMA DE GERÊNCIA DE REDE** – Módulo do Sistema que será responsável pelo armazenamento das informações de tráfego que chegam das Unidades Remotas monitoradas. Também terá a função de prover as informações de determinada linha para as Unidades de Consulta e para as consultas realizadas via celular. O SISTEMA DE GERENCIA DE REDE será formado por:
 - Servidor WEB Apache;
 - Servidor de Banco de dados Mysql;
- **UNIDADE DE CONSULTA (PARADA DE ÔNIBUS)** – Será chamada de UNIDADE DE CONSULTA o módulo do sistema que se encarregará de prover uma interface de consulta aos usuários de transporte coletivo, viabilizando a busca de informações sobre as linhas monitoradas. A UNIDADE DE CONSULTA será formada pelos seguintes componentes:
 - Computador com o S.O Linux;

- Conexão de rede com o Sistema de Gerencia.

3.1 – UNIDADE REMOTA

O desenvolvimento da UNIDADE REMOTA será feito em três módulos separados como mostrado na Figura 3.2.

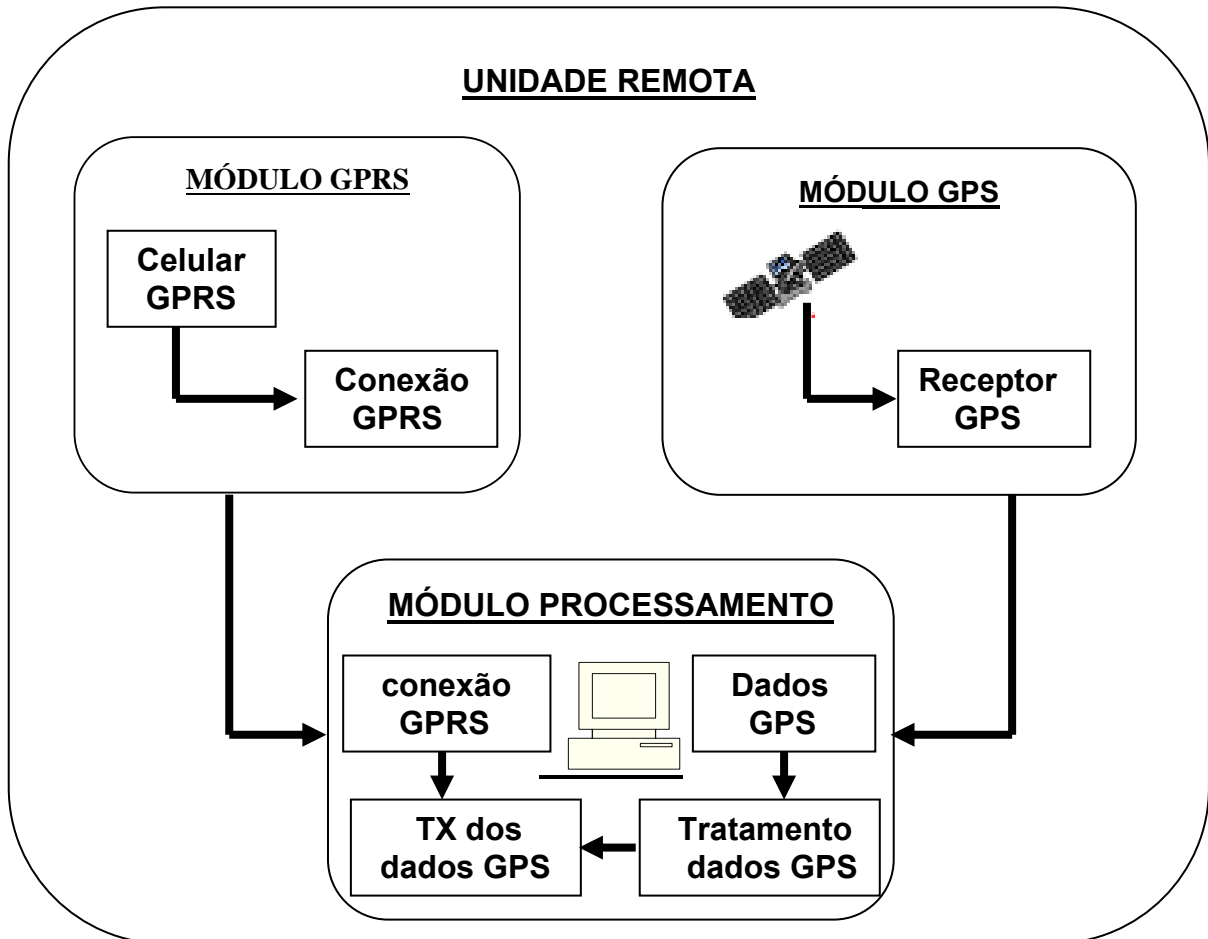


Figura 3.2 – Diagrama modular da UNIDADE REMOTA

3.1.1 – Desenvolvimento do MÓDULO GPRS da UNIDADE REMOTA

Este módulo possui a única função de estabelecer uma conexão entre o Módulo de Processamento da Unidade Remota e a Internet.

Uma vez que o veículo monitorado possui a característica de estar em constante movimento, também se faz necessário um método de conexão móvel com a Internet para possibilitar o estabelecimento da comunicação entre o Módulo de Processamento e o banco de dados do Sistema de Gerência de Rede com a sua conseqüente atualização pelos dados de trânsito do veículo monitorado.

O Módulo GPRS garantirá essa conexão móvel de maneira constante permitindo a comunicação necessária entre a Unidade Remota e o Sistema de Gerência.

Este módulo é composto por um telefone celular GSM/GPRS e scripts capazes de estabelecer uma conexão GPRS entre o sistema operacional Linux do módulo de processamento com a Internet.

Para estabelecer essa conexão será usado o daemon pppd que é nativo no Linux e é capaz de estabelecer conexões PPP com vários dispositivos, podendo servir tanto como servidor como cliente.

O PPP habilita o estabelecimento de uma conexão TCP/IP através de uma interface serial. Cliente e servidor podem se comunicar durante o estabelecimento da conexão e definir os parâmetros necessários de maneira correta. O servidor pode também configurar um endereço IP e um nome para o cliente.

Para que aconteça o estabelecimento de uma conexão, será necessário o uso de programas como wvdial ou chat, que são ferramentas poderosas para estabelecer conexões com Internet Service Providers (ISP).

Para viabilizar a conexão com o ISP GPRS, foram construídos scripts que se utilizam dos recursos do daemon pppd e dos programas CHAP e WVDIAL.

Uma vez obtido sucesso na tentativa de conexão entre o Módulo de Processamento e a Internet, passará a Unidade Remota a ter contato direto via TCP/IP com o Sistema de Gerencia de Rede, fazendo uso dos recursos de roteamento providos pelo provedor utilizado.

Na Figura 3.3 é mostrado o funcionamento esquemático do Módulo GPRS.

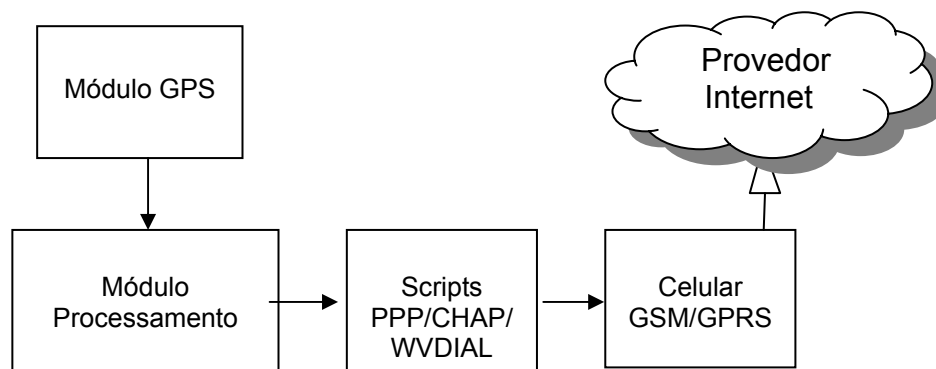


Figura 3.3 – Funcionamento do Módulo GPRS.

O script abaixo, responsável no sistema pelo estabelecimento da conexão de rede GPRS, mostra como é estabelecida a conexão.

Primeiramente é acionado o daemon pppd pelo script GPRS mostrado abaixo, onde são configurados alguns parâmetros tais como porta de conexão do modem GPRS, velocidade, entre outros. Este script também é o responsável por acionar ou o script CHAT ou o script WVDIAL, que irá negociar os parâmetros de conexão com o provedor.

```

#nodetach
debug
#####
# script para inicializar o modem GPRS. Alguns telefones #
# costumam trabalhar com o wvdial, outros com o chat. #
#####

#####
# script de conexão pelo chat. #
#####

connect /etc/ppp/peers/gprs-on

# script de conexão pelo wvdial. #
  
```

```
#####
#connect "/usr/bin/wvdial --chat --config /etc/ppp/peers/gprs-on.conf "
#####
# wvdial comentado, por estar sendo usado o chat. #
#####

#####
# script para desconexão.#
#####

disconnect /etc/ppp/peers/gprs-off

#####
# Porta serial e configuração de parâmetros. #
#####
/dev/ttyUSB0
57600
crtcts
local
noipdefault
ipcp-accept-local
defaultroute
usepeerdns

#####
# Username para uso no chap-secrets ou pap-secrets. #
#####
user "claro"
#####
# O comando persist irá tentar restabelecer a #
# conexão em caso de falha. #
#####
persist
maxfail 99
```

Com o estabelecimento da conexão, o Módulo de Processamento estará livre para proceder a atualização do Banco de Dados do servidor com as informações de deslocamento coletadas pelo Módulo GPS.

3.1.2 – Desenvolvimento do MÓDULO GPS da UNIDADE REMOTA

Esta unidade tem a função de estabelecer uma conexão entre o aparelho receptor GPS ao Módulo de Processamento da Unidade Remota.

O Módulo de Processamento é composto por um computador operando com sistema Operacional Linux, com capacidade de conexão de dispositivos externos através de interface serial e também por USB.

O receptor GPS será conectado ao computador do Módulo de Processamento pela sua interface serial.

O modelo de GPS utilizado no projeto é o Garmin GPS45xl, que possui as seguintes especificações técnicas especificadas em seu manual [Garmin, 1996]:

- FISICAS

- **Tamanho:** 15.6 x 5.1 x 1.23 cm
- **Peso:** Aproximadamente 269g com as baterias
- **Range de temperatura:** -15° a 70°C

- PERFORMANCE

- **Receptor:** Differential-ready MultyTrac8™

- **Tempo de Aquisição:** Aprox. 20 segundos (warm start)
Aprox. 2 minutos (cold start)
Aprox. 7.5 minutos (AutoLocate)
- **Update:** 1/segundo, continuamente
- **Acurácia de Posição:** 5 -10 metros com DGPS
15 metros com GPS

- **ALIMENTAÇÃO DE ENERGIA**

- **Input:** 4 baterias 1.5 volt AA ou 10 – 40 VDC por cabo de alimentação



Figura 3.4 – Receptor Garmin GPS 45xl

O cabo para estabelecer a conexão entre o GPS e o computador é o cabo próprio do fabricante Garmin como mostrado na figura abaixo [Garmin, 1996].



Figura 3.5 – Cabo de conexão do GPS ao computador

O conector utilizado para conexão ao computador é um DB-9 comum, e o GPS utiliza um conector específico de 4 pinos. A conexão entre os pinos do conector DB-9 e o utilizado pelo GPS segue o padrão abaixo [Garmin, 1996]:

Conector GPS GARMIN	Pino		Pino	Conector DB-9
Ground	2	->	5	Ground
Data output	4	->	2	Receptor
Data input	5	->	3	Transmissor

Tabela 3.1 – Correspondência do conector GPS GARMIN / DB-9 para o modelo 45.

Uma vez definida a maneira de se estabelecer a conexão física entre o GPS e o computador, foi desenvolvido um software capaz de fazer a abertura lógica da porta serial específica onde se encontra conectado o receptor GPS. A linguagem de programação escolhida para a escrita do programa foi a linguagem Perl, que possui recursos para controle de interfaces externas em computadores.

A comunicação com o GPS é estabelecida através da interface serial ttyS0, que é o nome padrão Linux para a correspondente serial COM1 no Windows. Uma vez obtido o sucesso na conexão, o programa fica aguardando o recebimento dos dados enviados pelo GPS.

Caso ocorra alguma falha no estabelecimento dessa conexão, uma mensagem de alarme é gravada em um arquivo específico, que poderá ser posteriormente apresentada na tela pelo menu gerenciador de alarmes, como será mostrado.

O trecho do script abaixo, desenvolvido com exclusividade para o projeto [DEITEL, 2002], mostra como é feita a comunicação com o receptor GPS.

```
#####
# Sub-rotina de abertura da porta serial #
# na velocidade de 4800 bps.          #
# Mesma velocidade do protocolo de    #
# comunicação do GPS.                 #
#####
my $date;
for ( my $i = 0; $i < 3; $i++ )
{
    $date = `date`;
    my $porta = shift;
    my $alarmfile;
    my $serialalert;
    open ($alarmfile, ">\home\jcarvalho\gpscom\alarmes_serial");
    open ($serialalert, ">\home\jcarvalho\gpscom\serialalert");

    open ($porta, "+<ttyS0")
    or
    print $alarmfile ( "$date SERIAL ALARME - Falha na Conexao com a Porta Serial.
                                                                \n"), &&
    print $serialalert ( "v" );

    system("stty 4800 raw < /dev/ttyS0");
}

#####
# Inicio da parte funcional do #
# programa GPSCOM.           #
#####

sub leitura
{
    my $par = shift;
```

```

my $data = 0;
my $buffer = 0;
my $gp = "";
my $timeout = 130;

#####
# Sub-rotina de configuracao #
# de parametros de leitura. #
#####
do
{
    sysread($porta,$buffer,256);
    $data .= $buffer
}
while ($data !~ /$par/i && $timeout--);
return $data;
}

#####
# Sub-rotina de definicao de qual porta #
# serial sera usada para leitura. #
#####
# Sub-rotina de gravacao dos dados coletados #
# em um arquivo texto "gpsdata"e no array especifico. #
#####

my $output = leitura($porta,"#");
my $outfile;
open ($outfile,">gpsdata");
print $outfile $output;
close $outfile;

```

Muitos dos receptores GPS possuem diversas maneiras para externar os dados coletados da constelação de satélites, algumas dessas maneiras são proprietárias dos fabricantes, requerendo software específico para leitura dos protocolos, porém existem outras maneiras que são abertas, dentre as maneiras abertas está o protocolo NMEA.

Trata-se de um protocolo muito bem definido que foi desenvolvido pelo National Marine Electronics Association (NMEA - USA).

O NMEA possui alguns padrões de formato, sendo o que está atualmente em uso é a versão. 0183, que padroniza a interface física pela norma EIA-422, permitindo a implementação da interface RS-232 de comunicação com um dispositivo externo, como um computador [NMEA].

A configuração de formato da interface não é programável, ou seja, funcionando sempre a 8 bits sem paridade e mais 1 de stop bit, a 4800 bps, o que obriga a serial ttyS0 do Módulo de Processamento ter a mesma configuração [Garmin, 1996].

O protocolo NMEA se comunica com dispositivos externos de controle através de sentenças padrão, são as chamadas SENTENÇAS NMEA, que são sempre formadas por caracteres ASCII, possuindo o seguinte formato [NMEA]:

- **Começam por um caractere - "\$" ;**
- **Seguindo mais duas letras - o TALKER ID;**
 - usado para identificar o equipamento transmissor:
 - GP para GPS, entre outros.
- **Seguindo mais três letras - a SENTENCE ID;**
 - usado para identificar a sentença NMEA específica.
- **Seguindo vários campos de dados separados por vírgula.**
 - latitude, longitude, velocidade, altitude, etc.

O projeto em questão irá utilizar a sentença NMEA \$GPRMC, que é uma sentença específica para dados de trânsito.

A tabela abaixo mostra a estrutura da sentença \$GPRMC, com um exemplo de coleta de posição.

GPRMC	Dados de Trânsito
\$GPRMC,104435,A,5325.034,N,03918.223,W,064.1,046.7,210504,019.8,E*HH	
104435	Horário de determinação do fixo. UTC
A	Alerta do receptor – A = normal V = alarme
5325.034,N	Latitude
03918.223,W	Longitude
064.1	Velocidade no solo em Nós
046.7	Curso verdadeiro
100305	Data de determinação do fixo
019.8	Variação magnética local
HH	Checksum

Tabela 3.2 – Formato sentença NMEA \$GPRMC.

3.1.3 – Desenvolvimento do Módulo de Processamento da UNIDADE REMOTA

Com a conexão entre o receptor GPS e o computador do módulo de processamento estabelecida, uma rotina do mesmo programa irá coletar todas as sentenças NMEA ativas, no receptor usado, e gravá-las em uma variável vetor, até que a sentença \$GPRMC esteja completa.

Em uma terceira fase, já com as sentenças armazenadas em memória e usando os poderosos recursos de EXPRESSÕES REGULARES do Perl, é feita uma pesquisa em todo o vetor em busca da expressão RMC, encontrada somente na sentença \$GPRMC.

Ao encontrar a expressão, e já com a sentença dividida em campos separados por vírgula, uma vez que todos os campos são separados por vírgula, o programa extrai da sentença somente as informações necessárias descartando as demais.

Entre as informações fornecidas pela sentença \$GPRMC as de interesse para a implementação do projeto e recuperadas pelo programa são somente as relativas a hora, alerta do receptor, latitude, longitude, data e velocidade de deslocamento.

De posse dessas informações e com as mesmas armazenadas em variáveis de memória, fica a cargo agora da rotina responsável por estabelecer uma conexão remota ao Banco de Dados do Sistema de Gerência de Rede e proceder a atualização da tabela responsável pela rota específica, conseguindo, ao final dessa operação sucesso no cumprimento da finalidade do software da Unidade Remota.

As operações de pesquisa e tratamento dos dados coletados descritos acima são realizadas pelo trecho do programa abaixo desenvolvido para o projeto.

```
#####
# Sub-rotina de pesquisa do arquivo de entrada e posterior #
# gravação dos dados procurados no arquivo de saída e nas #
# variáveis que serão inseridas no Banco de dados. #
#####
while(!eof($infile))
{
  $line=<$infile>;
  @word=split(/./,$line);
  if ($word[0] =~/RMC/)
  {
    $hora = $word[1];
    $alerta = $word[2];
    $lat = $word[3];
    $longit = $word[5];
    $speed = $word[7]*1.85;
    $data = $word[9];
  }
}
#####
# Sub-rotina para gravação dos dados #
# no arquivo de saída. #
#####
print $outfile ( "HORA - $hora\nALERTA - $alerta\nLATITUDE - $lat\n");
print $outfile ( "LONGITUDE - $longit\nVELOCIDADE - $speed\nDATA - $data\n");
close $infile; close $outfile;
```

A conexão com o Banco de Dados remoto é estabelecida utilizando recursos de programação DBI da linguagem Perl, como mostrado abaixo.

```
#####
# Conexão com o Banco de Dados.#
#####
my $dbalert;
my $db_alarm;
my $date = `date`;
open ($db_alarm, ">/home/jcarvalho/gpscom/alarmes_database");
open ($dbalert, ">/home/jcarvalho/gpscom/dbalert");
$dsn = "DBI:mysql:$dbname:$dbhost";
$dbh = DBI->connect("$dsn","$login", "",{PrintError =>0})
or
  print $db_alarm ( "$date DATABASE ALARME - Falha na Conexão com o Banco de
Dados.
_____ \n"), &&
  print $dbalert ( "\n" );
$dbh-> do ("UPDATE gcomtab SET
  hora = '$hora',
  alerta = '$alerta',
  latitude = '$lat',
  longitude = '$longit',
  speed = '$speed',
  data = '$data' ");
$dbh->disconnect;
$i--;
}
```

Para controlar todos os dispositivos e conexões externas da Unidade Remota, junto ao Módulo de Processamento foi desenvolvida uma interface gráfica capaz de controlar todos os processos envolvidos no sistema de maneira fácil e eficiente.

Para o desenvolvimento das interfaces gráficas dos programas utilizados, foi a biblioteca TK, incorporada ao Perl.

Na Figura 3.6 é apresentada a interface gráfica da Unidade Remota.

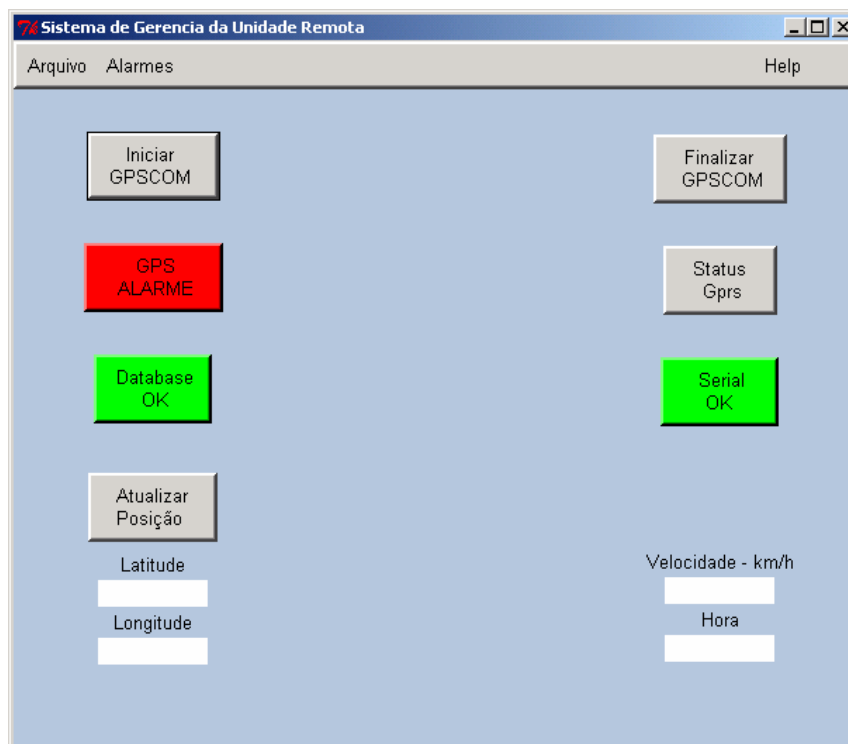


Figura 3.6 – Interface Gráfica da Unidade Remota.

Pode-se perceber, pela Figura 3.6, ícones relacionados ao GPS, ao GPRS, ao Banco de Dados e à Interface Serial, pressionando os bottons respectivos, é possível obter o status no momento de cada módulo ou conexão, pelo exemplo mostrado é fácil verificar que neste momento, ao iniciar o programa, obteve-se sucesso de conexão com o receptor GPS, tendo em vista o status da Serial estar OK, também obteve-se sucesso de conexão com o Banco de dados remoto, verificado pelo botton Database OK e ainda pode-se inferir que ocorreram problemas com o receptor GPS, como por exemplo perda de cobertura, o que ocorre em locais cobertos.

Cada conexão ou device controlado pelo Módulo de Processamento possui também um receptor de alarmes capaz de armazenar algumas ocorrências sem sucesso, tais como falha de conexão com o Banco de Dados, perda de cobertura de satélite GPS, perda da conexão GPRS, entre outras.

As ocorrências são geradas quando se é necessária a utilização do dispositivo e o mesmo não responde com sucesso.

No caso do receptor GPS, por exemplo, quando a rotina encarregada de obter e tratar as medidas de trânsito do veículo encontra perda de cobertura GPS, fato explicitado na sentença NMEA \$GPRMC, através do campo ALERT, que possui o valor "A" para uma medida realizada com sucesso e valor "V" para falha na realização da medida, uma mensagem de alarme é gravada em um arquivo texto chamado "alarmes_gps". Este arquivo será a fonte de consulta para se verificar o estado do receptor GPS. O indicativo de alarme nos dispositivos do sistema pode ser consultado a qualquer momento, possuindo inclusive a hora exata do evento, como mostrado na Figura 3.7.

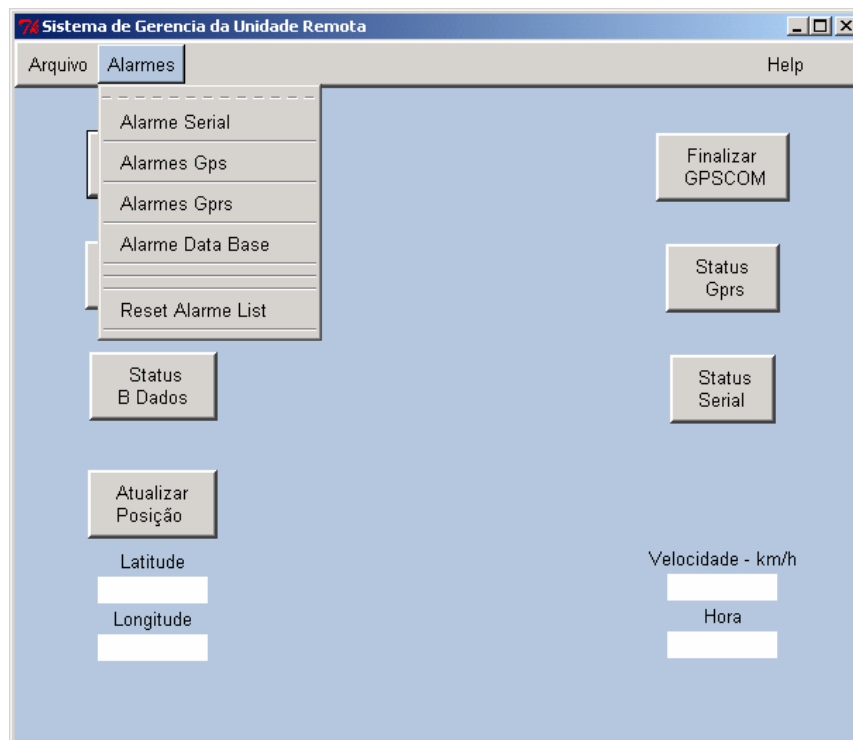


Figura 3.7 – Gerência de alarmes na Unidade Remota.

Ao se selecionar Alarmes Gps no menu Alarmes, o script alarmes_gps.pl é acionado, fazendo uma pesquisa no arquivo texto alarmes_gps. Apresentando para o operador, em caso da ocorrência de alarmes a seguinte tela.

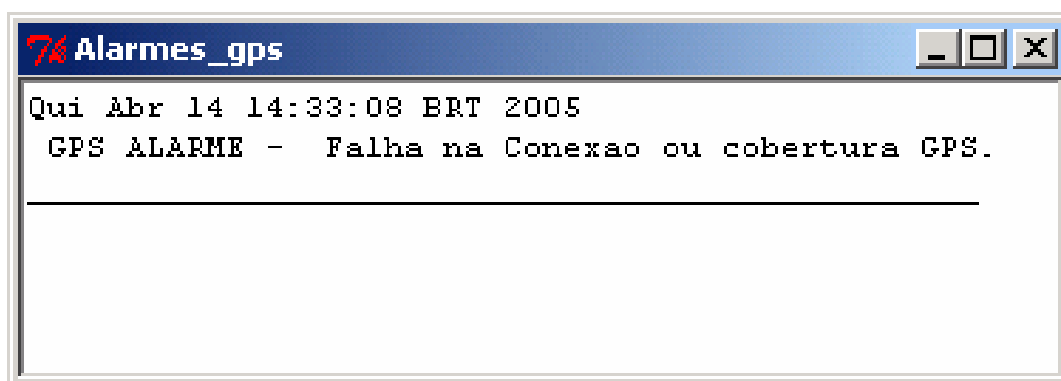


Figura 3.8 – Apresentação da ocorrência de um alarme no receptor GPS.

O script alarmes_gps.pl, responsável pela coleta e apresentação de alarmes no receptor GPS, desenvolvido para o projeto é o apresentado a seguir.

```
#####
# Aplicativo para apresentar na tela #
# informações sobre os alarmes no #
# GPSCOM #
#####
#!/usr/bin/perl
use strict;
use warnings;
use Tk;
use Tk::Text::Viewer;
```

```
#####
# Rotina de configuração da janela      #
# e chamada do arquivo alarmes_gps     #
#####

my $mw = new MainWindow();
our $t1 = $mw->Scrolled("Viewer", -wrap => 'none');

$t1 = $mw->Viewer()->pack();
$t1 ->Load("alarmes_gps");
      MainLoop();

#####
```

Os eventos de alarmes que são monitorados pelo Módulo de Processamento da Unidade Remota, possuem a seguinte estrutura de arquivos:

- Falhas no receptor GPS

- script responsável pela apresentação dos alarmes na tela:
 - > "alarmes_gps.pl"
- arquivo texto onde os alarmes são gravados:
 - > "alarmes_gps".

- Falhas na conexão com o Banco de Dados remoto

- script responsável pela apresentação dos alarmes na tela:
 - > "alarmes_database.pl"
- arquivo texto onde os alarmes são gravados:
 - > "alarmes_database".

- Falhas na conexão com a porta serial

- script responsável pela apresentação dos alarmes na tela:
 - > "alarmes_serial.pl"
- arquivo texto onde os alarmes são gravados:
 - > "alarmes_serial".

- Falhas na conexão GPRS

- script responsável pela apresentação dos alarmes na tela:
 - > "alarmes_gprs.pl"
- arquivo texto onde os alarmes são gravados:
 - > "alarmes_gprs".

A Gerência de Alarmes possui no sistema a função de alertar o operador para uma eventual falha ocorrida durante um trajeto, por exemplo. Uma vez que, dadas as características do sistema, de estar sempre em movimento por terrenos que são muitas vezes acidentados, fator causador de falhas em qualquer tipo de sistema embarcado e também por não ser possível em uma situação normal a sua observação constante, torna-se de suma importância uma gerência de falhas, ainda que sumária, com a função de alerta, como qualquer alarme.

Uma falha no receptor GPS, que pode ser causada por diversos fatores, tais como perda de cobertura, falha no cabo de interconexão com o computador, defeito no aparelho, desligamento acidental, deve ser diagnosticada com a maior brevidade possível, pois, dada as característica e funções da Unidade Remota, a perda do receptor GPS inutiliza toda a unidade.

E a não observância da falha, poderia impactar na performance do sistema, pois vários trajetos poderiam ser cumpridos sem que o sistema estivesse funcionando em perfeitas condições.

O que causaria também prejuízos à credibilidade do sistema, se implementado de forma operacional, junto a seus possíveis usuários. Indo na contramão de uma dos objetivos iniciais do projeto, que é propor um sistema que seja capaz, se operacional, de ajudar a resgatar a confiança dos usuários em sua qualidade.

Além da Gerência de Alarmes, a interface gráfica da Unidade Remota possui o menu Help, que externa algumas informações sobre o objetivo do projeto no item “Sobre o Gpscom” e uma ajuda resumida de como operar a Unidade Remota de maneira eficiente no item “Ajuda”.

No item “Ajuda” é apresentado um arquivo texto com um resumo sobre o funcionamento dos diferentes módulos da unidade, e também, algumas possíveis ações e procedimentos no caso de se constatar uma possível falha em alguma das unidades ou componentes do sistema.

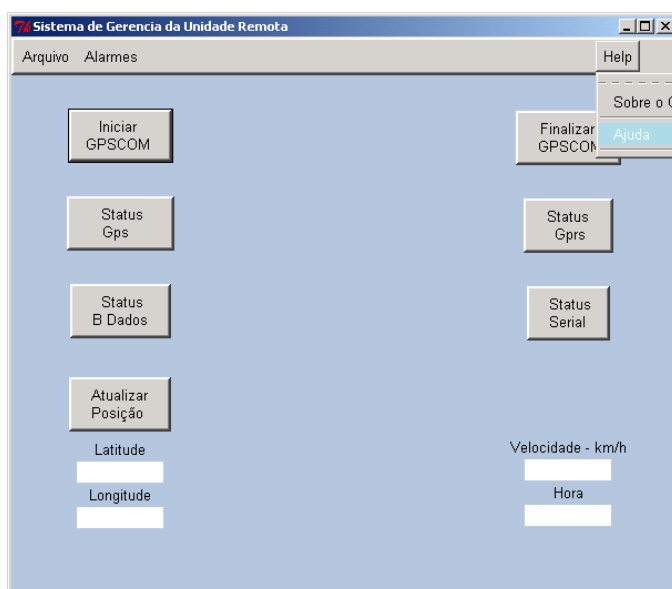


Figura 3.9 – Menu Help do Sistema de Gerência da Unidade Remota.

3.2 – UNIDADE DE CONSULTA

É chamada UNIDADE DE CONSULTA o módulo do sistema que se encarregará de prover uma interface amigável para que os usuários de determinada linha de transporte coletivo possam obter as informações sobre o tráfego daquela linha específica. Esta unidade tem a função de viabilizar tecnicamente a busca de informações sobre as linhas monitoradas e apresentá-las aos usuários com a maior rapidez possível.

A UNIDADE DE CONSULTA é formada pelos seguintes componentes:

- Computador com o S.O Linux;
- Conexão de rede via ADSL ou celular GSM/GPRS com o Sistema de Gerência de Rede.
- Aplicação responsável por prover a interface com o usuário além de executar todo o processamento necessário ao cumprimento das funções da unidade.

O desenvolvimento da UNIDADE DE CONSULTA será realizado como mostrado na Figura 3.10.

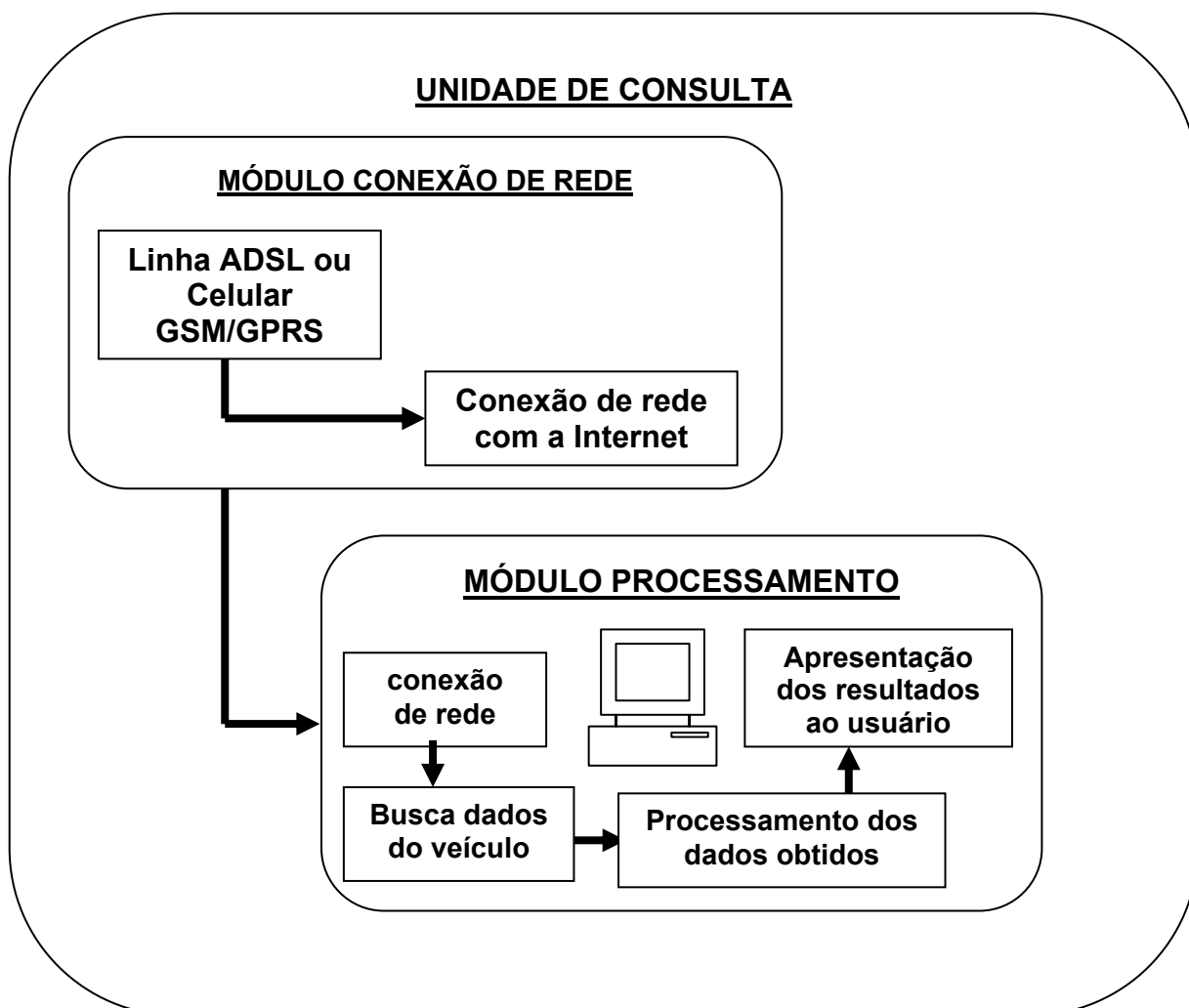


Figura 3.10 – Diagrama modular da UNIDADE DE CONSULTA

3.2.1 – Desenvolvimento do MÓDULO CONEXÃO DE REDE da UNIDADE DE CONSULTA

Estabelecer uma conexão de rede, que seja capaz de colocar a Unidade de Consulta em contato com o Sistema de Gerência de Rede, torna-se uma tarefa relativamente mais fácil que a necessária para a construção do módulo com a mesma função desenvolvida para a Unidade Remota.

Uma vez que a Unidade de Consulta, que se localiza em uma parada de ônibus, opera de maneira estática, ou seja, não possui deslocamento espacial, é possível o uso da rede fixa entre a Unidade de Consulta e a rede Internet. Tal conexão pode ser feita por um modem ADSL ou uma linha discada, por exemplo, além da possibilidade de se usar uma conexão móvel como a desenvolvida para a Unidade Remota, utilizando um móvel com tecnologia GSM/GPRS.

Para efeitos de demonstração do protótipo desenvolvido, será usada uma conexão móvel via celular, nos mesmos moldes da conexão desenvolvida para a Unidade Remota.

Assim como no Módulo GPRS da Unidade Remota, este módulo será composto por um telefone celular GSM/GPRS e scripts capazes de estabelecer uma conexão GPRS entre o Módulo de Processamento e a Internet, fazendo uso também aqui do daemon pppd, como o software que estabelecerá a conexão, e do Sistema Operacional Linux.

O diagrama abaixo mostra o funcionamento esquemático do Módulo Conexão de Rede.

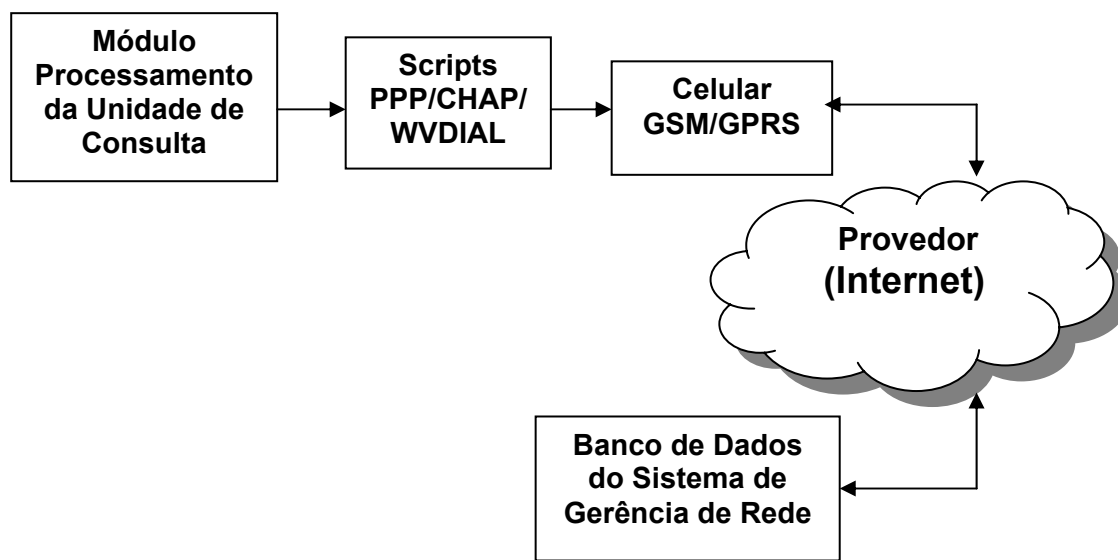


Figura 3.11 – Funcionamento do Módulo Conexão de Rede.

3.2.2 – Desenvolvimento do MÓDULO DE PROCESSAMENTO da UNIDADE DE CONSULTA

Este módulo possui a função de processamento, ou seja, controlar todos os processos relativos a essa Unidade, dentre os quais estão o estabelecimento e gerenciamento da conexão de rede com a Internet, coleta dos dados de trânsito relativos à rota solicitada pelo usuário, processamento dos dados obtidos e posterior apresentação ao usuário solicitante das informações requeridas.

Uma vez estabelecida a conexão com o Sistema de Gerência de Rede, os dados de deslocamento relativos à rota, são coletados através de uma conexão com o Banco de Dados do Sistema de Gerência utilizando para isso recursos de programação DBI da linguagem Perl. Os dados obtidos são armazenados em variáveis no computador local para posterior processamento.

Com as informações da Unidade Remota monitorada, como latitude, longitude e velocidade média armazenadas localmente, e sendo conhecidos os dados relativos à posição espacial da Unidade Remota, ou seja, sua latitude e longitude, o mesmo software inicia os cálculos para estimar o tempo aproximado de chegada daquele veículo àquela parada.

Como a maioria das linhas de transporte coletivo obedece a uma rota de ida e volta, algumas vezes utilizando a mesma estrada e sendo as distâncias

medidas pelas coordenadas coletadas pelo sistema GPS, tomam como base linhas retas, para obter um resultado da distância mais próximo do real, o cálculo será realizado utilizando uma técnica de ponto a ponto, como mostrado na Figura 3.12.

Para calcular a distância do veículo até a parada (Unidade Remota), primeiramente é calculado a distância entre o veículo e o Ponto B, do Ponto B ao Ponto C e do Ponto C à Unidade de Consulta. Ambos os pontos A e B são pontos escolhidos estrategicamente, com suas coordenadas armazenadas em variáveis dentro do software.

Este método de cálculo fornece uma distância bem mais próxima da realidade, se comparado com a medida da distância entre as coordenadas dos pontos X e Y diretamente.

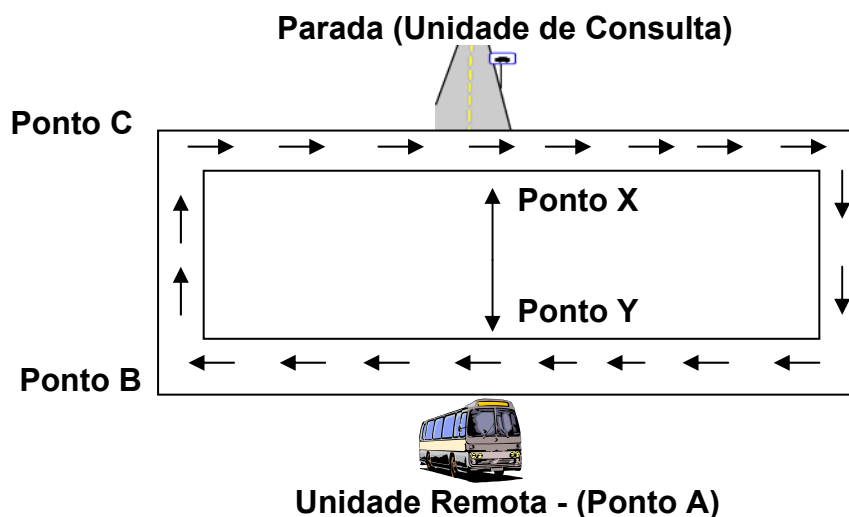


Figura 3.12 – Exemplo do método de Cálculo da distância.

O método de localização de um ponto na superfície terrestre pode ser feito com base nas suas coordenadas, previamente conhecidas, a latitude que para efeito dos cálculos dessa demonstração será representada por α e a longitude, que será representada por β .

A latitude do Ponto A da Figura 3.12, onde se encontra a Unidade Remota, por exemplo, é o arco de meridiano que se inicia no Equador e vai até o paralelo do ponto A, sendo que a latitude pode ser denominada norte ou sul (N ou S), dependendo do hemisfério ao qual pertencer o ponto. Quando se usa a latitude para fins matemáticos, é considerada a latitude norte com sinal positivo e a sul com sinal negativo.

A longitude do Ponto A, é o comprimento do arco do Equador que se inicia no meridiano de Greenwich e vai até o ponto A, sofrendo também diferenciação dependendo do hemisfério, sendo leste quando o Ponto A se localizar no Oriente ou oeste quando se localizar no Ocidente. A Figura 3.13 ilustra a distribuição e marcações de latitudes e longitudes no globo terrestre.

Pelo fato de ser a terra não exatamente uma esfera, o comprimento do arco relativo a um minuto varia dependendo do local da medida. Para contornar esse problema, convencionou-se a milha marítima internacional, correspondente a 1852 metros, que é uma média aritmética das milhas medidas em diversos pontos do globo [Colorado, 2005].

Como exemplo, guiando-se pela Figura 3.13, conhecidas as coordenadas das cidades São Paulo e Lisboa, determina-se a distância entre as mesmas

como sendo a medida do arco formado entre os pontos entre os seus pontos, sendo o Polo Norte o vértice, como mostrado na Figura 3.13.



Figura 3.13 – Distância entre dois pontos na superfície terrestre. ([http:// www.geografiaparatodos.com.br](http://www.geografiaparatodos.com.br))

Sendo assim, o cálculo da distância entre as cidades de São Paulo e Lisboa passa a ser apenas uma aplicação da Lei dos Cossenos, ou seja, dadas as coordenadas de São Paulo, representada como “A” na fórmula abaixo e as coordenadas de Lisboa, representadas como “B” e ainda representando latitude como “ α ” e longitude como “ β ”, temos as coordenadas de São Paulo $A(\alpha, \beta)$ e Lisboa $B(\alpha', \beta')$.

Ajustando as latitudes, para que se use como referência o Polo Norte, temos, aplicando a Lei dos Cossenos [Gelson, 1984]:

$$\cos c = \frac{\cos d - \cos a \cos b}{\sin a \sin b} \quad [1]$$

tem-se:

$$\cos \text{distancia} = \cos(\beta' - \beta) \sin(90^\circ - \alpha') \sin(90^\circ - \alpha) + \cos(90^\circ - \alpha') \cos(90^\circ - \alpha) \quad [2]$$

para aplicação matemática, deve-se transformar todas as coordenadas em números decimais, utilizando a fórmula:

$$\left(\frac{\text{graus} + \frac{\text{minutos}}{60} + \frac{\text{segundos}}{3600}}{\quad} \right) \quad [3]$$

Com a aplicação da equação [3], todas as coordenadas, latitude e longitude, dos dois pontos são transformados em números decimais. Os cálculos são feitos coordenada por coordenada [Colorado, 2005].

Obtem-se a distância multiplicando o arco (em radianos) pelo raio médio da terra, de valor igual a 6366,707 [Colorado, 2005], como abaixo:

$$d = 6366.707(\arccos(\cos(\beta' - \beta) \cos(90^\circ - \alpha') \cos(90^\circ - \alpha) + \sin(90^\circ - \alpha') \sin(90^\circ - \alpha))) \pi/180 \quad [4]$$

Com os cálculos mostrados na equação [4], é obtida a distância (d) em linha reta entre os pontos A e B, como explicado na seção 3.2.2.

Uma vez que não convém ao sistema basear-se apenas neste cálculo, na distância em linha reta entre dois pontos, considerando que uma rota sofre muitas variações em seu trajeto, obedecendo à construção da estrada, que varia muito. Todas as medidas necessárias para se obter a distância entre o veículo e uma parada de ônibus específica serão feitas calculando a distância entre pontos estratégicos, que devem ser especificados rota por rota.

A escolha desses pontos estratégicos, que devem buscar sempre o cumprimento de toda a rota percorrida pelo veículo, precisa ser definida de maneira tal a montar uma malha de pontos que consigam o melhor grau de precisão possível quando da realização das medições para o cálculo da distância entre a posição atual do veículo e a parada.

Com uma consulta solicitada por um usuário do sistema e com os cálculos necessários prontos, resta ao Módulo de Processamento da Unidade de Consulta apresentar os resultados ao usuário requisitante.

O objetivo no desenvolvimento dessa interface gráfica foi construir uma ferramenta de consulta capaz de informar ao usuário os resultados, da maneira mais amigável e intuitiva possível, de fácil entendimento para pessoas de qualquer nível de escolaridade.

Para a construção dos módulos gráficos da Unidade de Consulta, também foi utilizado o módulo TK de bibliotecas gráficas, juntamente com a linguagem de programação Perl.

Na Figura 3.14 é mostrada a visão que o usuário teria quando se dirigisse a um terminal de parada de ônibus com a intenção de realizar uma consulta a respeito de uma determinada rota.

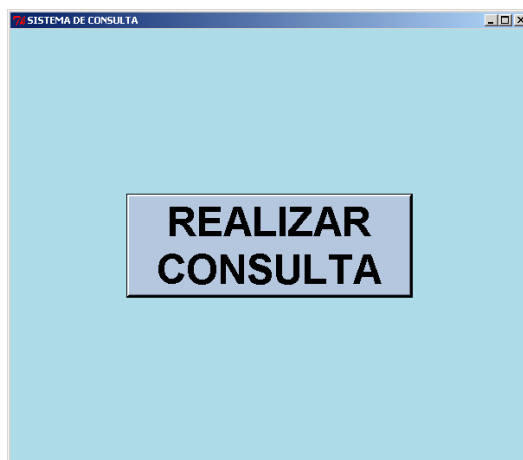


Figura 3.14 – Interface gráfica da Unidade de Consulta.

Ao clicar no botão REALIZAR CONSULTA, mostrado na Figura 3.14, uma outra tela gráfica é apresentada ao solicitante com a lista de todas as rotas disponíveis para aquela parada, ou seja, todas as linhas de transporte coletivo monitoradas que utilizam aquela parada específica como ponto de trânsito de passageiros.

Na Figura 3.15 é mostrada a tela gráfica com o menu de rotas disponíveis para consulta que é apresentada ao usuário solicitante.

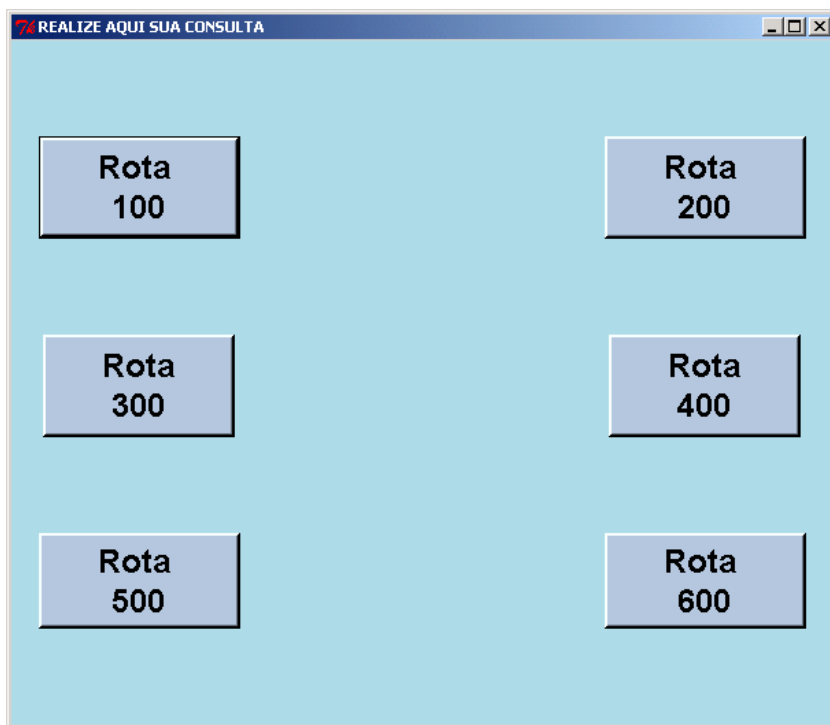


Figura 3.15 – Apresentação das rotas disponíveis ao usuário.

Em caso de falha de conexão com o Banco de Dados do Sistema de Gerência de Rede, uma mensagem de erro é apresentada ao usuário solicitando que tente a realização de uma nova consulta.

A Figura 3.16 mostra a mensagem de erro apresentada ao usuário.

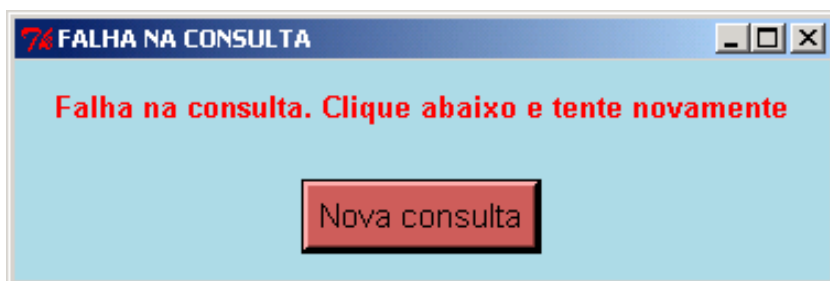


Figura 3.16 – Mensagem de erro apresentada.

Em não ocorrendo nenhum erro na conexão com o Banco de Dados, e os dados relativos ao trânsito da Rota solicitada sejam obtidos, o sistema realiza

todos os cálculos relativos à distância entre os dois pontos, que são executados localmente, como mostrados na seção 3.2.2.

Após a execução do processamento da consulta de interesse, os resultados são mostrados ao solicitante através de uma mensagem apresentada na tela, como mostrado na Figura 3.17, que apresenta o retorno de uma consulta de exemplo realizada com sucesso.

A Figura 3.17 apresenta os resultados de uma simulação de consulta realizada na rota nomeada como Rota 100, durante os testes de implementação, apresentado o resultado de tempo aproximado de chegada do veículo monitorado àquela parada.

Uma vez conferida a resposta à consulta solicitada, o usuário pode acionar o botão FECHAR CONSULTA, e o programa retorna para a tela apresentada na Figura 3.14, possibilitando que outro usuário realize uma nova consulta rapidamente, repetindo todo o processamento apresentado acima.

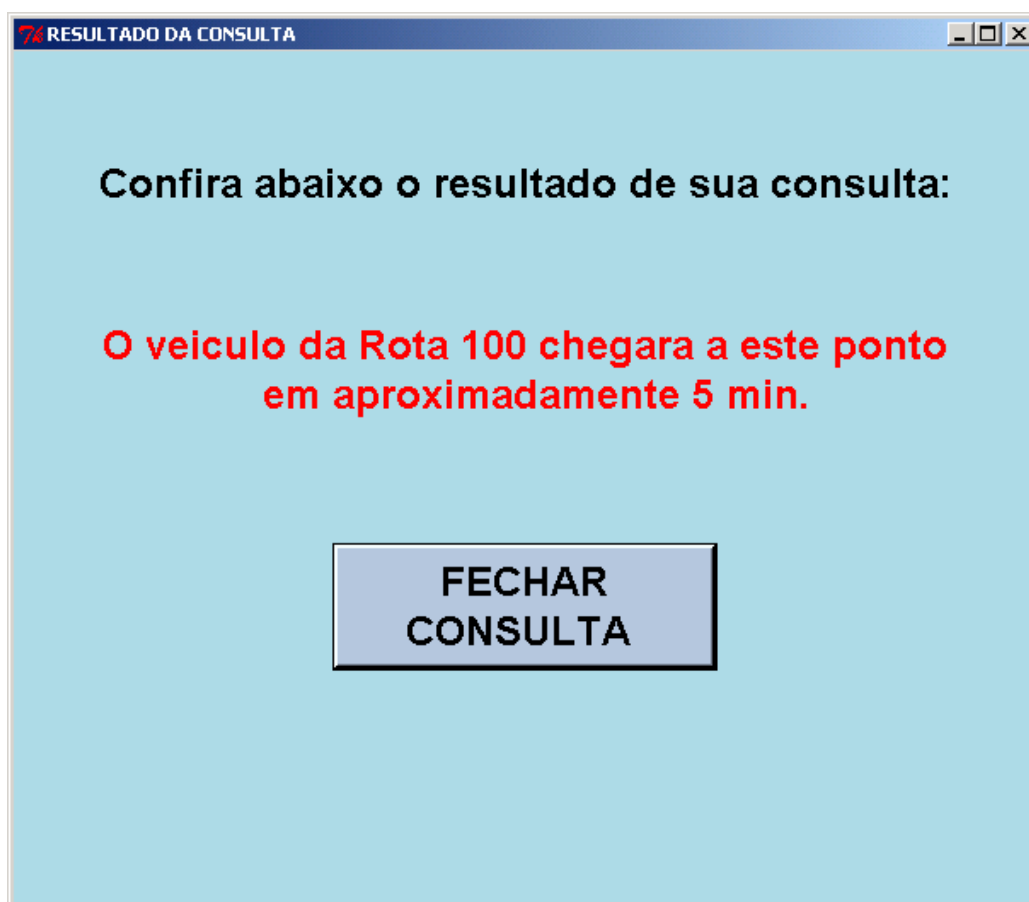


Figura 3.17 – Apresentação do resultado da consulta.

Com a apresentação do resultado mostrado na Figura 3.17 se encerra todo o ciclo de processamento de uma consulta e também de todas as funções desenvolvidas pela Unidade de Consulta.

3.3 – SISTEMA DE GERÊNCIA DE REDE

É o sistema responsável pelo armazenamento em Banco de Dados das informações de trânsito coletadas pelas Unidades Remotas monitoradas.

Uma vez possuindo essas informações armazenadas, também será o sistema que terá a função de prover essas informações quando solicitadas pelas Unidades de Consulta e também para as consultas realizadas via terminais móveis.

Será formado pelos seguintes componentes:

- Computador com o S.O Linux;
- Conexão de rede com a Internet utilizando um endereço IP válido.

O desenvolvimento do Sistema de Gerência de Rede será realizado como mostrado da Figura 3.18.

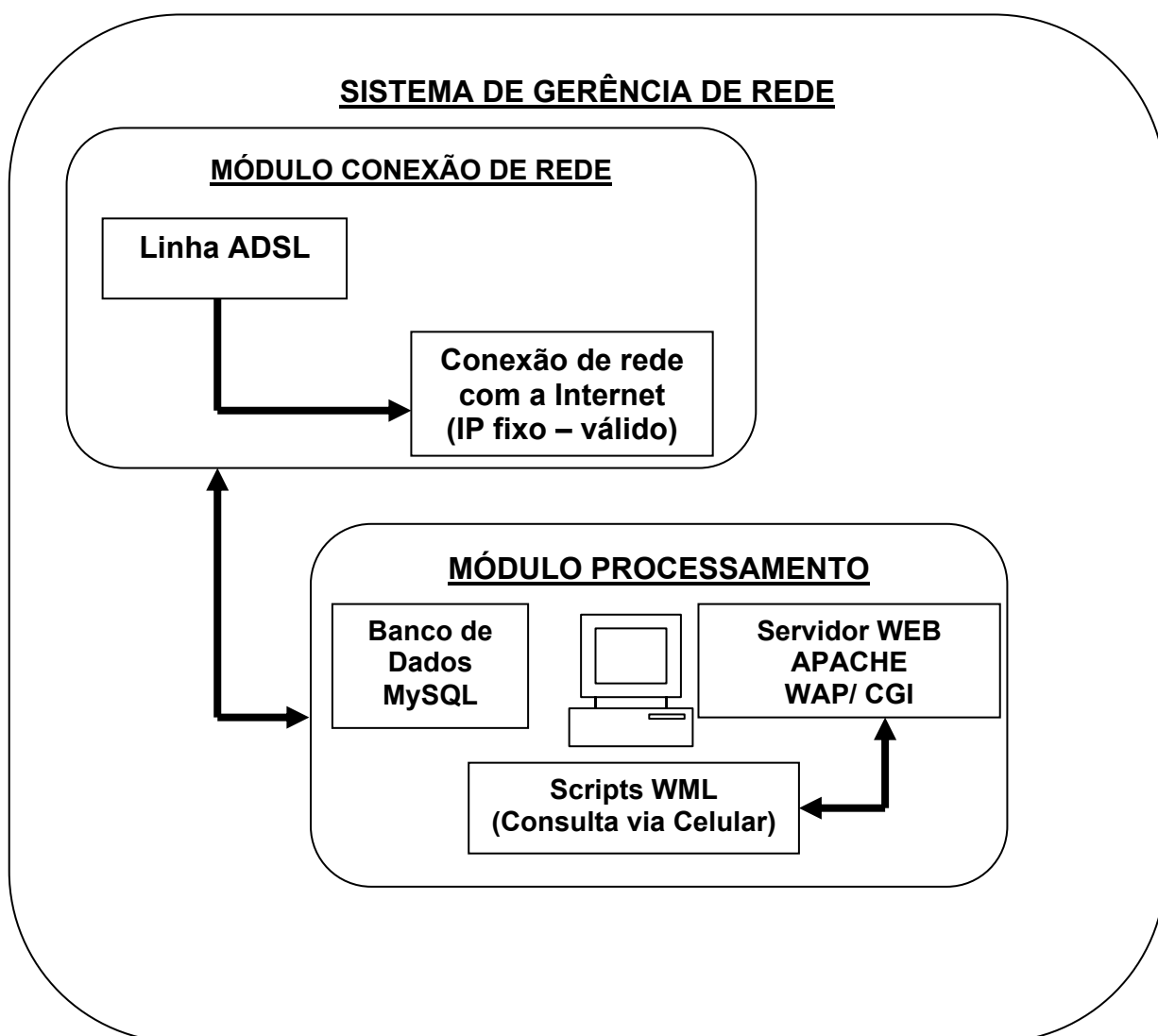


Figura 3.18 – Diagrama modular do SISTEMA DE GERÊNCIA DE REDE

3.3.1 – Desenvolvimento do MÓDULO CONEXÃO DE REDE do SISTEMA DE GERÊNCIA DE REDE

Este módulo não requer desenvolvimento, apenas que sejam disponibilizados recursos de conexão com a rede Internet para a máquina do Sistema de Gerência.

Para esse fim, e para colocar o Sistema de Gerência em contato com a rede Internet, foi escolhida a contratação de um link ADSL empresarial de 400Kbps. A escolha desse tipo de serviço se fez necessário devido ao fato de o mesmo disponibilizar ao contratante um endereço IP fixo de uso constante na Internet.

O Sistema de Gerência tem a necessidade de um IP fixo para seu uso devido a ser essa a forma que a Unidade Remota e a Unidade de Consulta irão acessar as aplicações do Sistema de Gerência.

A conexão da aplicação da Unidade Remota com o Banco de Dados da Gerência é feita utilizando os protocolos TCP/IP, que possibilitam à Unidade Remota o acesso à máquina servidora através de seu endereço IP, válido na Internet. Da mesma forma ocorre com a Unidade de Consulta, que acessa as aplicações do Sistema de Gerência.

A configuração de um IP fixo junto ao Sistema de Gerência também será de fundamental importância para o servidor de Internet WAP, que será o responsável de prover o conteúdo para as consultas realizadas via celular.

3.3.2 – Desenvolvimento do MÓDULO DE PROCESSAMENTO do SISTEMA DE GERÊNCIA DE REDE

Este módulo possui a função de armazenar as informações de trânsito coletadas pelas Unidades Remotas monitoradas e disponibiliza-las para as requisições realizadas pelas Unidades de Consulta e também para as consultas feitas através de terminais móveis, como mostrado na Figura 3.19.

As informações de trânsito coletadas pelas Unidades Remotas são transmitidas para o Sistema de Gerência. A maneira mais eficiente encontrada para o armazenamento dessas informações foi através de um Banco de Dados relacional, uma vez que se tem a garantia do armazenamento das informações mesmo em caso de pane do sistema, e também por ser a maneira mais eficiente de se disponibilizar esses dados aos requisitantes.

Sendo assim, não seria aconselhável armazenar os dados recebidos das Unidades Remotas em arquivos texto no sistema, por exemplo. Sendo o Banco de Dados e as suas tabelas específicas usadas como variáveis auxiliares de armazenamento e consulta pelas diferentes aplicações do sistema.

O Banco de Dados escolhido foi o MySQL, por ser um Banco de Dados de código aberto muito popular em todo o mundo, de fácil manejo e já conhecido por trabalhar bem com operações críticas em todo o mundo. Sendo que também tem a facilidade de já ser uma das opções de Banco de Dados disponíveis com o pacote Fedora Core 3, que é o Sistema Operacional escolhido para o projeto [MySQL, 2002].

A Figura 3.19 apresenta um detalhamento das aplicações nas unidades do sistema, mostrando como ocorre a interação entre os diferentes processos e aplicações do sistema junto ao Sistema de Gerência.

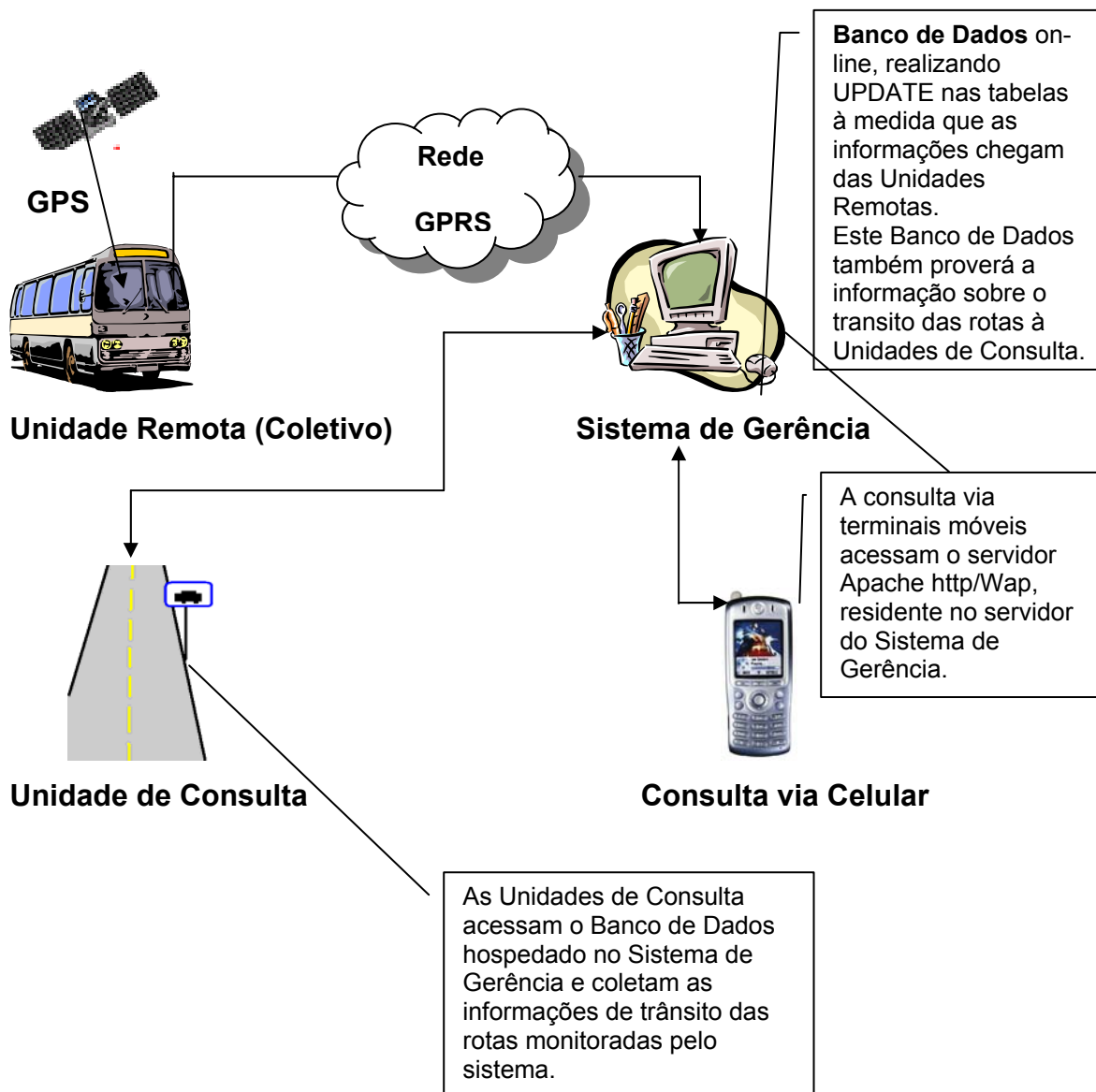


Figura 3.19 – Detalhamento das aplicações do sistema integrado ao Sistema de Gerência.

Por não ser o Banco de Dados o foco principal do projeto, funcionando apenas como uma ferramenta auxiliar durante a integração dos diferentes módulos do sistema, os trabalhos com essa ferramenta se restringirão apenas à criação de um Banco de Dados específico para o sistema e às tabelas necessárias para repositório das informações provenientes das Unidades Remotas.

Como para efeitos de demonstração da eficácia prática do sistema desenvolvido, será implantada a aplicação da Unidade Remota em uma rota específica, ainda a ser definida, e como a função do Banco de Dados no sistema funciona apenas como uma variável auxiliar, que sofrerá um UPDATE a cada

medida feita pela Unidade Remota monitorada, torna-se necessário apenas a criação de uma tabela relativa às informações da rota específica.

A Figura 3.20 demonstra o fluxo de informações entre a aplicação da Unidade Remota e de Consulta e o Banco de Dados do Sistema de Gerência.

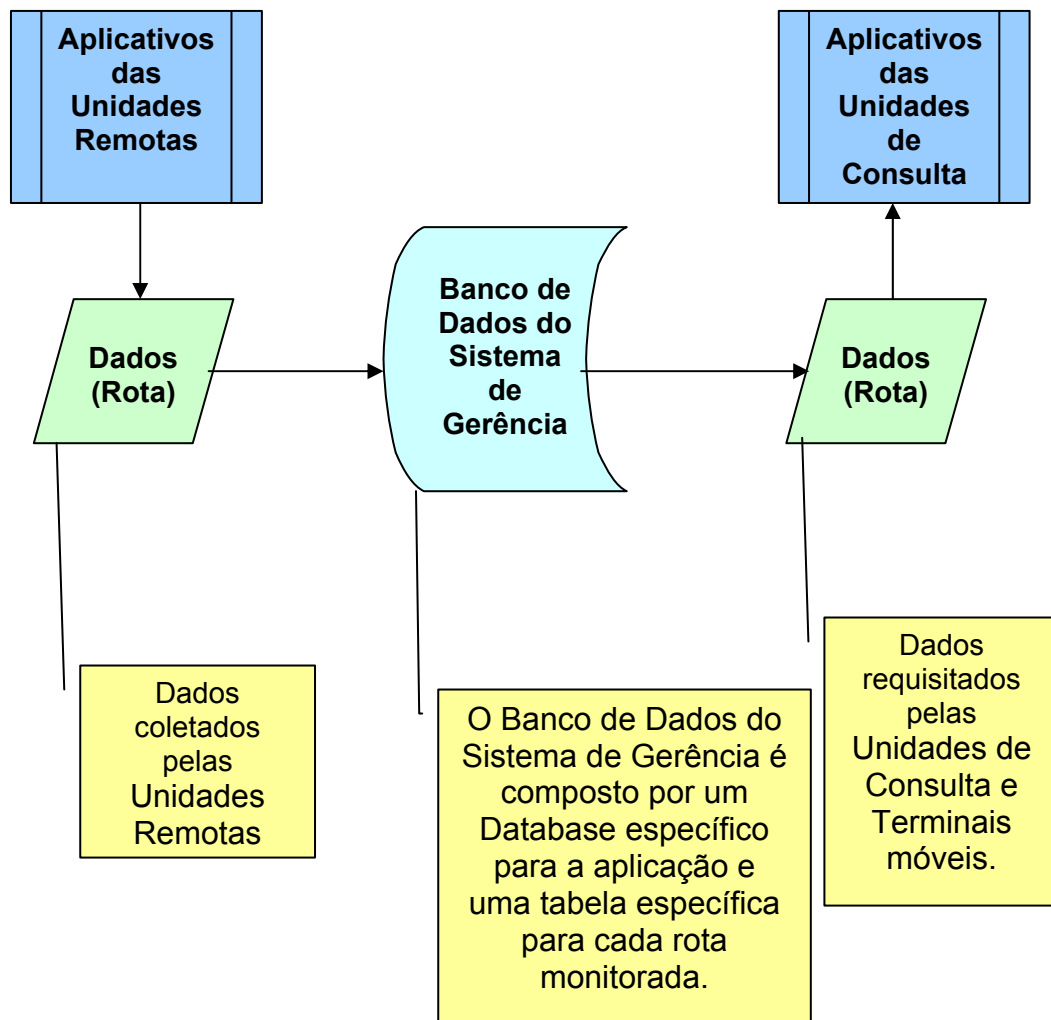


Figura 3.20 – Fluxo do Banco de Dados do Sistema de Gerência.

Com a implantação e perfeito funcionamento do Banco de Dados do Sistema de Gerência e com a interface de rede junto a Internet operando dentro da normalidade, a integração do sistema, ou seja, podemos ter Unidade Remota, Unidade de Consulta e Sistema de Gerência integrados, sem considerar as consultas via terminal móvel.

Para tornar operacional e disponível a consulta por celulares, torna-se necessário apenas o start do servidor http Apache, que foi o escolhido para ser o servidor Web para o projeto.

O Apache usa configuração default de instalação e já vem com a capacidade de atender às requisições feitas utilizando protocolo WAP, desde que os scripts WML estejam corretos e de acordo com os padrões da linguagem. Os

scripts WML desenvolvidos com exclusividade para o projeto serão os responsáveis pelo conteúdo disponibilizado nos terminais móveis.

Também aqui se busca durante o desenvolvimento, a construção de uma aplicação eficiente e simples, que pudesse ser facilmente acessada de um terminal móvel e principalmente facilmente entendida e utilizada. Esta busca torna-se necessária principalmente devido às limitações gráficas do protocolo WAP e das interfaces gráficas dos terminais móveis, fato que limitou muito a penetração, o uso e o desenvolvimento de aplicações WAP em diferentes segmentos de mercado.

Uma vez operando de forma on-line o servidor Web, o Sistema de Gerência estará apto para responder as requisições de consulta formuladas por terminais móveis.

O terminal móvel será o responsável pela seleção da rota e da Unidade de Consulta (Parada de ônibus) específica. A Figura 3.21 mostra a simulação realizada durante o desenvolvimento das aplicações necessárias.



Figura 3.21 – Simulador de conexão WAP utilizado durante o desenvolvimento.

A Figura 3.22, demonstra o fluxo da requisição WAP até o seu atendimento e resposta ao móvel solicitante.

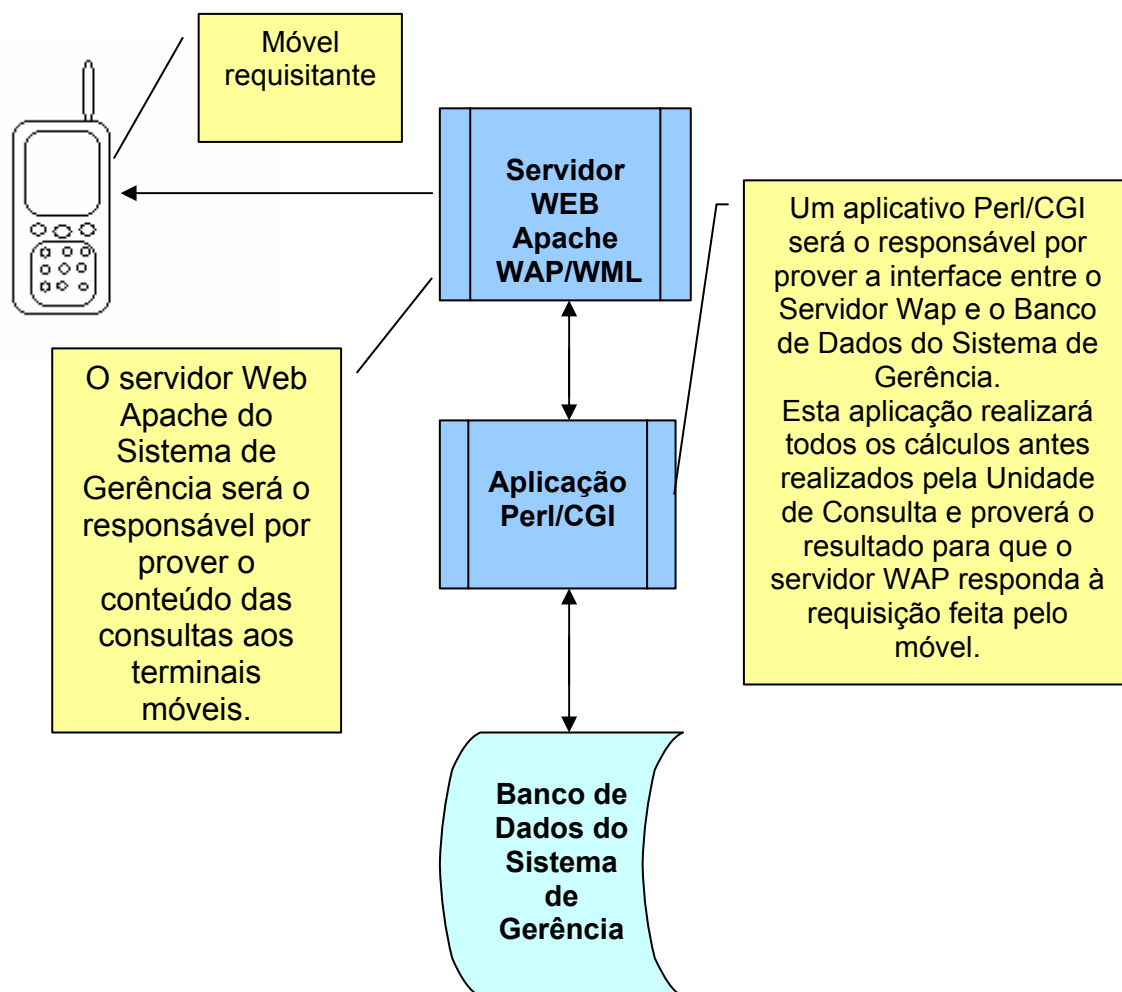


Figura 3.22 – Caminho de uma requisição de consulta via terminal móvel.

CAPÍTULO 4 – VALIDAÇÃO DO SISTEMA

Para validar a arquitetura proposta, foram definidos alguns ensaios, sendo que o principal é a realização de uma simulação com todas as unidades funcionando de maneira integrada, ou seja, a realização de uma consulta feita por uma Unidade de Consulta em uma rota de transporte coletivo hipotética, feita somente para a realização da simulação.

Além de uma consulta feita por uma Unidade Remota, cobrirá também a simulação, uma consulta feita por um terminal móvel durante a mesma simulação de validação.

A Figura 4.1 demonstra a rota escolhida para a realização da simulação, ela compreende o trajeto que circunda a Superquadra localizada na Asa Norte, Distrito Federal, onde se encontra as instalações do Campus Uniceub, progredindo em direção ao Colégio Militar (CMB) e retornando pelo retorno em frente ao mesmo colégio, voltando novamente em direção ao campus Uniceub. Completando assim o trajeto efetuado pela rota. A localização exata da Unidade de Consulta (Parada de Ônibus) será em frente à entrada do bloco de número 7 do campus Uniceub.

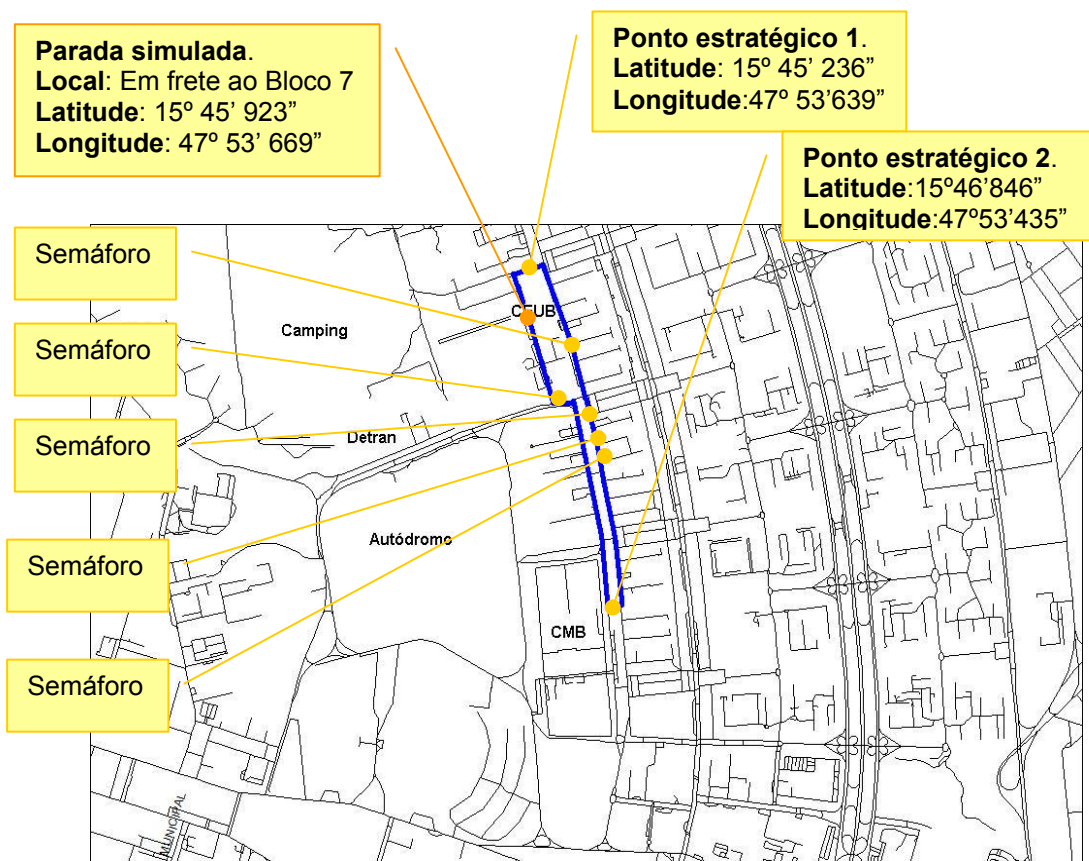


Figura 4.1 – Rota de validação.

4.1 – Definição do problema

O cálculo do tempo estimado de chegada do veículo monitorado à parada definida na rota deve levar em conta os seguintes passos:

- Obter os dados de trânsito da Unidade Remota monitorada.
- Calcular o tempo previsto para que a Unidade Remota chegue à Unidade de Consulta em questão.
- Apresentar os resultados ao solicitante.

4.2 – Cálculo e obtenção dos dados de trânsito da Unidade Remota

Como visto no Capítulo 3, a Unidade Remota é a responsável pela coleta dos dados de posicionamento do veículo, que são fornecidos pelo receptor GPS e o armazenamento desses dados no Sistema de Gerência de Rede.

Os dados coletados de importância para o sistema são os de localização, latitude e longitude e velocidade média de deslocamento do veículo.

Para contornar o problema da velocidade média, ou seja, para tentar anular interferências externas que alteram os dados coletados, tais como, paradas em sinais vermelhos, pequenos engarrafamentos, entre outros, a Unidade Remota não reportará somente a última medida de velocidade, evitando com isso que ocorra a atualização do banco de dados da rota com uma medida feita em um semáforo bloqueado, por exemplo, o que atualizaria o banco de dados com a informação de velocidade igual a zero e por consequência, inserindo no sistema uma informação não verdadeira.

Durante a execução do processamento da Unidade Remota, o algoritmo responsável pela atualização das informações de trânsito no Sistema de Gerência irá atualizar o Sistema de Gerência somente com a média das últimas cinco velocidades médias medidas pelo receptor GPS.

As simulações realizadas em campo, nessa mesma rota, mostraram que as cinco medidas são suficientes para anular eventuais interrupções na movimentação do veículo monitorado, como semáforo fechado, por exemplo.

Essas mesmas simulações mostraram que o tempo médio gasto para que se proceda a uma atualização de posicionamento, entre a medida realizada pelo GPS, tratamento dos dados coletados, estabelecimento de uma conexão GPRS, conexão com o Banco de Dados do Sistema de gerência e atualização das informações no Banco é de cerca de 7 segundos, ou seja, cerca de 8 atualizações de posição e velocidade média por minuto.

Para se obter na medida certo grau de precisão, não são necessárias 8 medidas por minuto, quando se leva em conta principalmente a velocidade média de deslocamento de um veículo coletivo comum, regulamentar em cerca de 60 km/h, oito medidas de posição por minuto significariam a atualização da posição do veículo de 30 em 30 metros.

Em uma linha regular cujo trajeto é de cerca de 30 km, situação comum no Distrito Federal nas linhas de ligação com as cidades satélites, medir-se o seu deslocamento de 30 em 30 metros seria um exagero de precisão, porém, o sistema irá realizar as atualizações de 7 em 7 segundos, ou seja, de cerca de 30 em 30 metros, e ainda irá, nas atualizações considerar as últimas cinco médias de velocidade, procurando assim aumentar o grau de precisão no cálculo do tempo estimado de chegada do veículo monitorado à parada.

Com esse método de cálculo, o sistema irá atualizar o campo velocidade média do banco de dados da rota com a média das últimas cinco velocidades reportadas pelo receptor GPS.

Tal metodologia, como já foi dito, irá evitar a atualização de medidas com o veículo fora de movimento, em paradas eventuais, tal como em sinais de trânsito.

O que não seria evitado seria a atualização de posição em momentos de congestionamentos mais demorados, em virtude de acidentes, blitz entre outros.

Para evitar que a situação acima apresente ao usuário uma informação errada, considerando que o tempo de interrupção no deslocamento do veículo possa ser superior às cinco medidas de velocidade média, o algoritmo irá considerar como velocidade média igual a zero um tempo indeterminado de chegada do veículo à parada, como apresentado na Figura 4.2, uma vez que ao encontrar uma velocidade média igual a zero a resposta igual a zero reportaria um tempo de chegada infinito à parada, ocasionando um erro de lógica no programa ao tentar fazer uma divisão por zero.

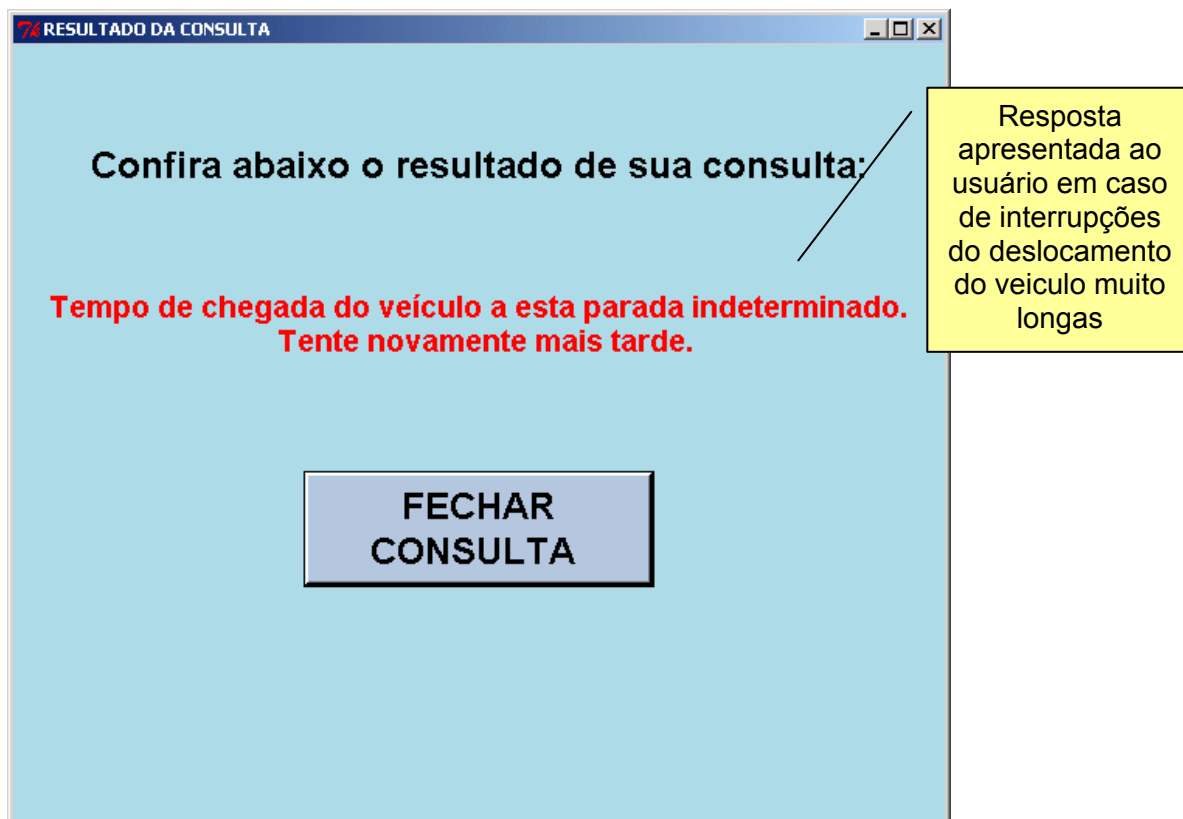


Figura 4.2 – Resposta ao usuário em caso de interrupções no deslocamento muito longas.

4.3 – Dados sobre a rota

A rota escolhida (Figura 4.1) possui um comportamento uniforme de tráfego ao longo do dia, como horas específicas em que ocorrem congestionamentos, ocasionados principalmente pelas escolas que se localizam na região. Seus pontos de retenção fixos são 5 semáforos apontados na Figura 4.1, que podem reter por alguns segundos o deslocamento do veículo monitorado.

Seu trajeto total possui uma extensão de 4 km, o que a uma velocidade média de 60 km/h, levaria 4 minutos para completar o total do percurso. Levando em conta os acidentes e interrupções na via, nos testes realizados levou-se em média 6,5 minutos para completar o trajeto.

O sentido que será seguido pelo veículo da rota será o mesmo oferecido pelo sentido da via. Esta via possui tráfego em uma só direção, indo do UNICEUB

na direção onde se localiza o CMB, realizando o retorno localizado em frente ao referido colégio e retornando na direção do final da Asa Norte, fazendo o percurso até o retorno que se localiza próximo ao Bloco 10 do campus UNICEUB, completando o trajeto quando chegar ao ponto de partida, onde encontra-se a Unidade de Consulta, localizada em frente à entrada do bloco 7 no campus UNICEUB.

4.4 – Problemas enfrentados

Além da necessidade de se anular interrupções eventuais no deslocamento do veículo, como mostrado na seção 4.2, com vistas a se obter uma medida mais precisa se fez necessário também atribuir o sentido de deslocamento do veículo na rota.

Tal cuidado é importante devido ao fato de que o receptor GPS, como mostrado na seção 3.2.2, ilustrado pela Figura 3.12, executa suas medições de posicionamento não levando em conta os acidentes artificiais no terreno, tais como construções, curvas, retornos, entre outros.

Uma medida realizada entre duas posições coletadas pelo receptor GPS mostrará sempre o resultado em linha reta, indo de uma coordenada à outra, sem considerar os acidentes listados acima, que serão percorridos pelo veículo durante o trajeto.

Como o veículo monitorado cumpre um trajeto pré-estabelecido, em solo, que obedece a uma rota que leva em conta as interferências artificiais e naturais em seu trajeto, uma média em linha reta não seria um cálculo muitas vezes confiável.

A estratégia adotada para se obter a distância exata entre a posição do veículo monitorado e a Unidade de Consulta em questão é efetuar o cálculo da distância não levando somente em conta a posição espacial ocupada no momento pela Unidade Remota e a Unidade de Consulta, mas sim efetuar uma medida ponto a ponto; de acordo com a posição da Unidade Remota.

De acordo com a posição espacial da Unidade Remota, por exemplo, logo após iniciar o seu trajeto, ela estará espacialmente mais próxima da Unidade de Consulta. No entanto, devido ao trajeto ainda a ser percorrido, este será o momento em que uma consulta realizada retornará para o usuário do sistema o tempo mais longo de retorno do veículo àquele ponto, uma vez que o veículo ainda deverá percorrer todo o trajeto e paradas da rota.

Como se pode perceber, se a medida fosse feita de maneira direta, sem levar em conta todo o trajeto ainda a ser percorrido pelo veículo, o resultado fornecido pelo sistema ao usuário seria um tempo muito inferior ao real, o que prejudicaria muito a confiabilidade no sistema, no caso de uma implementação operacional.

A Figura 4.3 ilustra a estratégia de cálculo exposta anteriormente.

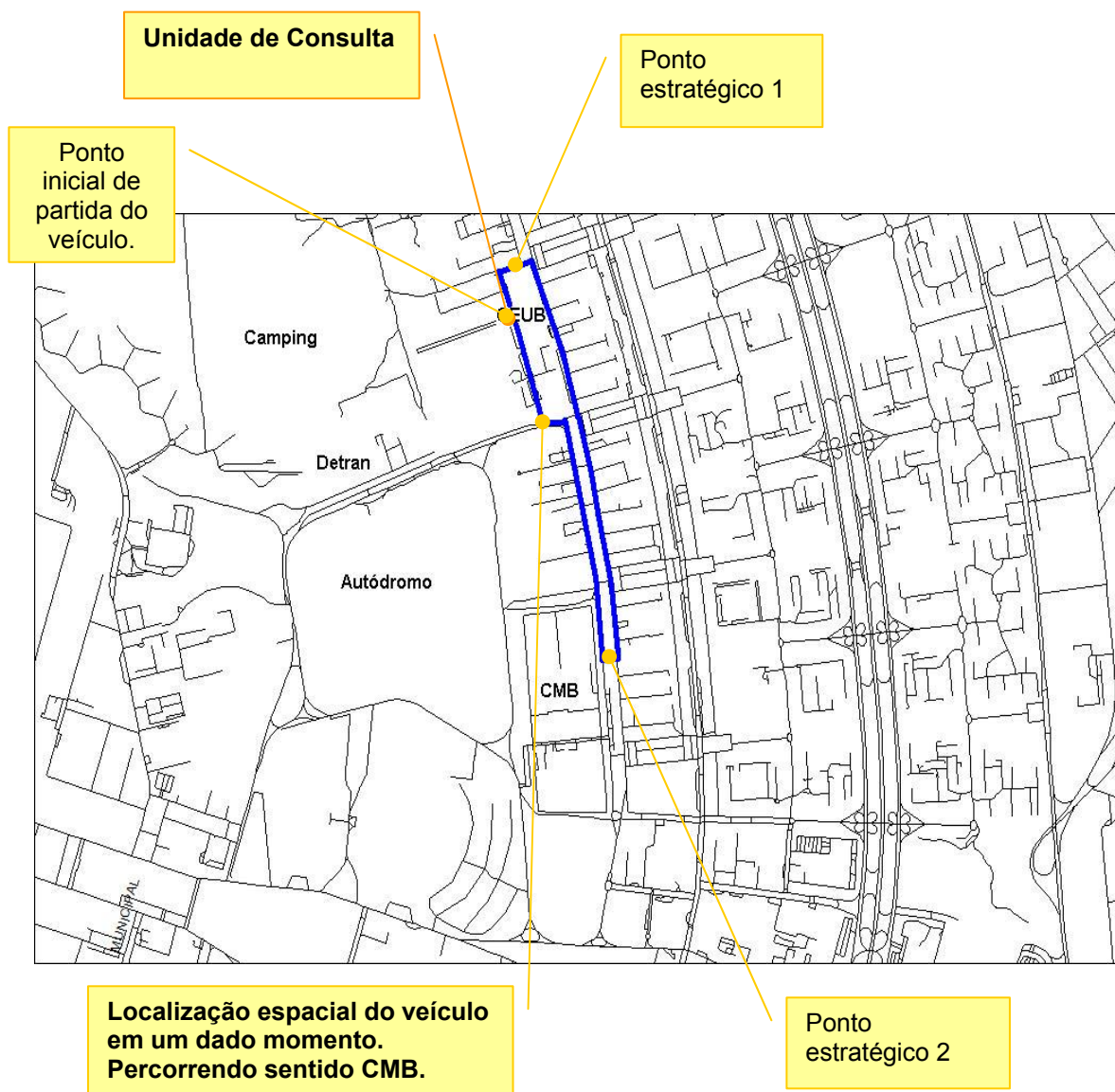


Figura 4.3 – Demonstração da estratégia de cálculo

Pode ser observado na Figura 4.3, que o ponto indicativo da localização espacial do veículo monitorado é um exemplo da estratégia de cálculo descrita acima e de sua eficiência.

Se o software levasse em consideração somente as coordenadas da Unidade de Consulta e Unidade Remota, o cálculo seria incorreto com relação à distância e com relação ao sentido percorrido pelo veículo na rota, fatos que podem ser visualizados na Figura 4.3.

Como a Unidade Remota localiza-se em uma posição espacial na rota anterior ao ocupado no espaço pelo Ponto estratégico 2, o cálculo deve ser feito da seguinte maneira:

- 1 - Cálculo da distância entre a localização espacial do veículo e o Ponto estratégico 2.
- 2 - Cálculo da distância entre o Ponto estratégico 2 e o Ponto estratégico 1.

- 3 - Cálculo da distância entre o Ponto estratégico 1 e a Unidade de Consulta.
- 4 – Soma das três distâncias medidas.

Se forem associadas as três medidas feitas com a Figura 4.3, pode-se perceber a proximidade entre o cálculo efetuado e o encontrado quando da realização em campo da medida de distância real entre os dois pontos, sendo percorrida a rota.

É claro que a definição de pontos estratégicos em uma rota depende de suas características, uma rota muito sinuosa necessita da definição de mais pontos estratégicos.

4.5 – Experimento em campo da funcionalidade do sistema

Utilizando a rota descrita nas seções anteriores deste capítulo, foi realizado teste em campo para verificar a funcionalidade do sistema e detectar possíveis falhas durante a execução prática de todas as operações.

Em todos os testes realizados foi utilizado um veículo particular fazendo as funções da Unidade Remota monitorada, com todos os módulos pertencentes ao sistema da Unidade Remota (Figura 3.2) embarcados no veículo.

Os testes ocorreram em horário no qual o volume de tráfego estava dentro da média já observada no setor em outras oportunidades, o que garantiu ao teste a credibilidade de ser realizado em um ambiente perfeitamente compatível com o real observado para o local. Nenhuma situação foi simulada, uma vez que dentre os objetivos do teste estava a detecção de possíveis erros, principalmente de timing entre os diferentes módulos do sistema.

Antes de se dar início ao teste aqui descrito, foi inicialmente verificado todo o Sistema de Gerência de Rede (Figura 3.18) e todos os seus processos, banco de dados, servidor WEB e conexão de rede com a Internet.

Devido ao fato do Sistema de Gerência não se localizar no mesmo espaço físico da realização dos testes, em virtude de limitações técnicas de conexão com a Internet, como descrito na seção 3.3.1, todas as verificações e conferências ao Sistema de Gerência foram feitos previamente.

O sistema pertencente à Unidade de Consulta (Figura 3.10), especificamente para o teste em questão, se utilizou de uma conexão GPRS feita por um terminal móvel, similar ao utilizado pela Unidade Remota, para se conectar ao Sistema de Gerência de Rede e realizar a consulta, uma vez que, como explicado na seção 3.2.1, a Unidade de Consulta tanto pode se utilizar de uma conexão fixa como de uma conexão móvel para exercer a sua funcionalidade dentro do sistema.

Uma vez verificado todas as unidades e recebido um status positivo de sua operacionalidade, partiu-se para a verificação da Unidade Remota e de seus módulos, composto pelos seguintes passos:

- verificação do receptor GPS e de sua conexão com o Módulo de Processamento da Unidade Remota,
- verificação do terminal móvel,
- verificação da conexão GPRS (botton indicativo de alarme mostrado na Figura 4.4),
- verificação da conexão com Banco de Dados do Sistema de Gerência.

A Figura 4.4 mostra o status do sistema de gerência da Unidade Remota antes do início dos testes.

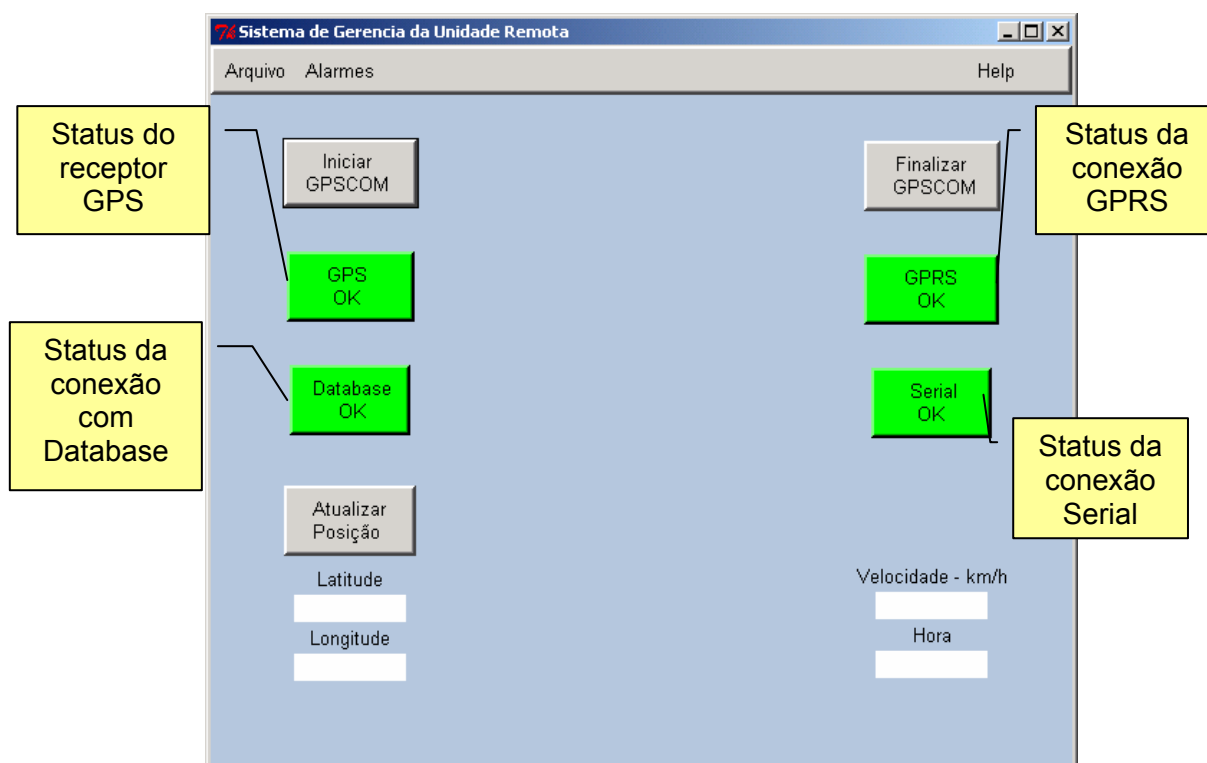


Figura 4.4 – Status do sistema de gerência da Unidade Remota.

Estando o sistema de Gerência como mostrado na Figura 4.4, os testes puderam ser iniciados. O veículo usado como Unidade Remota foi deslocado a uma velocidade de segurança, considerada como a média utilizada na via, igual a 50 km/h.

Durante a execução do trajeto previsto para a rota, não foi observado nenhuma atividade ou anormalidade que prejudicasse a sua execução e todos os resultados obtidos foram os previstos.

Com relação aos possíveis erros ocorridos durante a execução dos testes e a posterior correção dos mesmos, não foi verificada a ocorrência de nenhum erro fatal, ou seja, todos os módulos do sistema funcionaram perfeitamente e atenderam às expectativas. Cumpre salientar que muitos erros foram previamente corrigidos durante a execução de testes realizados na fase de desenvolvimento dos módulos, procedimento que minimizou os riscos de insucesso nos testes realizados com o sistema completamente integrado.

Durante o decorrer do percurso foram feitas diversas consultas ao Sistema de Gerência de Rede sobre a rota, não obtendo em nenhuma ocasião um erro como resultado. Os procedimentos adotados para anular a interferência de fatores externos interruptores de tráfego, descritos na seção 4.2, mostraram-se suficientes e cumpriram o objetivo.

Das diversas consultas realizadas no decorrer dos testes e desenvolvimento, serão mostrados os resultados de uma consulta específica, realizada exatamente quando o veículo monitorado se encontrava no ponto indicado na Figura 4.5.

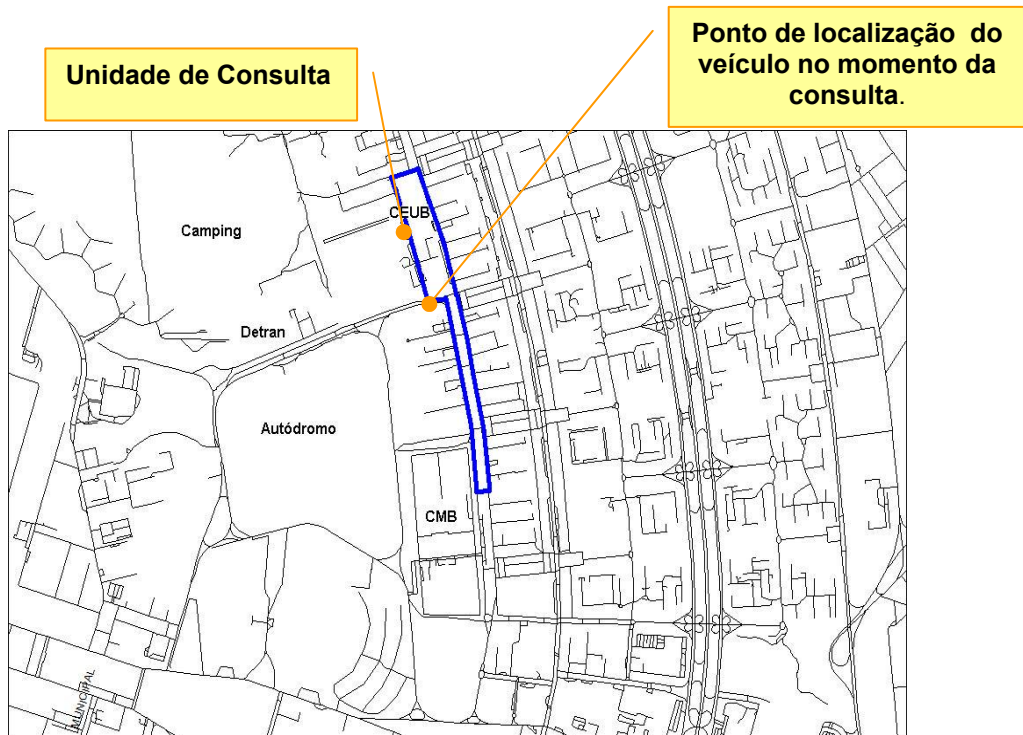


Figura 4.5 – Ponto da realização da consulta.

Como contraprova, e para verificar se os resultados obtidos com as consultas realizadas ao sistema desenvolvido correspondiam realmente ao tempo real gasto pelo veículo até a sua chegada à Unidade de Consulta, juntamente com o veículo da Unidade Remota, utilizou-se um cronômetro que foi acionado no momento da realização da consulta, sendo somente desligado no momento da chegada do veículo à Unidade de Consulta.

O resultado externado pela Unidade de Consulta é mostrado na Figura 4.6.

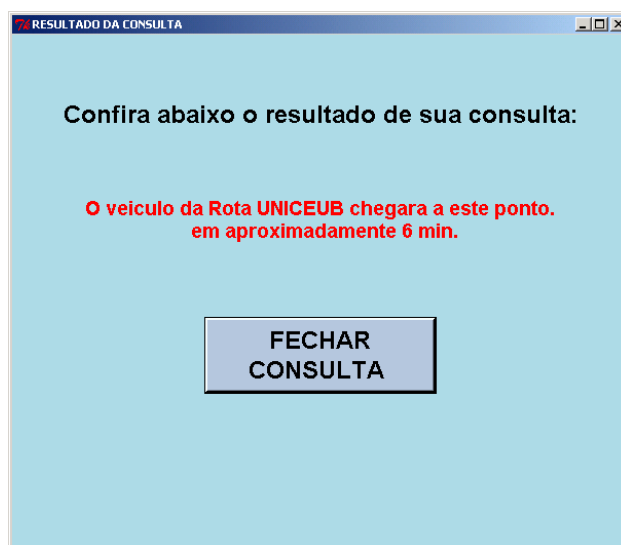


Figura 4.6 – Resultado de uma consulta.

Comparando os resultados obtidos na consulta (6 minutos) com o tempo medido pelo cronômetro, que foi exatos 6 minutos e 38 segundos, pode-se perceber que o tempo estimado pelo software é perfeitamente compatível com o medido pelo cronômetro. Conclui-se, então, que se obteve sucesso com o resultado do teste.

Paralelo ao teste realizado utilizando a Unidade de Consulta foi realizado uma consulta utilizando um terminal móvel.

4.6 – Consulta via Terminal Móvel

A fim de proporcionar ao protótipo do sistema uma maneira mais flexível de apresentar os resultados de uma consulta, também como uma alternativa ao método adotado das Unidades de Consulta, foi desenvolvida uma versão simplificada de uma ferramenta com capacidade para prover conteúdo às consultas feitas via celular.

Assim, em uma implementação operacional, o usuário poderá consultar a linha de seu interesse antes de se deslocar para a parada de ônibus, bastando para isso dispor de um aparelho celular.

Uma requisição de consulta por um terminal móvel é atendida pelo banco de dados do Sistema de Gerência de Rede, como explicado na Figura 3.22.

Ao solicitar uma consulta, uma tela com as rotas disponíveis é mostrada ao usuário, como é mostrado na Figura 4.7.



Figura 4.7 – Informação das rotas disponíveis ao usuário.

Como a dinâmica de uma consulta feita por um terminal móvel é diferente de uma consulta realizada por um usuário que se localiza em uma Unidade de Consulta, nessa consulta, por celular, é necessário que o usuário também informe

qual a parada desejada. Para que o usuário a selecione, uma lista com todas as paradas da rota é apresentada ao usuário, como mostrado na Figura 4.7.



Figura 4.8 – Informação das paradas disponíveis para a rota.

Com a rota e a parada selecionada, o sistema se encarrega de processar a informação, se utilizando da mesma dinâmica utilizada pela Unidade Remota e retorna ao usuário solicitante o resultado, como mostrado na Figura 4.8.



Figura 4.8 – Resultado final da consulta.

CONCLUSÃO

O estudo apresentado neste projeto é de uma importância estratégica, uma vez que o setor de transporte público tem passado por um processo de descrédito cada vez maior junto ao público alvo.

No Distrito Federal, por exemplo, o transporte público de passageiros por linhas regulares vem perdendo casa vez mais mercado para sistemas alternativos, as chamadas lotações. Um dos motivos para que isso ocorra é o modelo tecnológico ultrapassado no qual o transporte público brasileiro baseia seu funcionamento.

Um método moderno de se disponibilizar informações, capaz de solucionar uma das maiores reclamações dos usuários, que é o tempo de espera excessivo nas paradas, pode ajudar a dar esse salto tecnológico tão necessário para o sistema.

E é exatamente o que o projeto aqui apresentado foi capaz de mostrar, isto é, a viabilidade de se implementar solução tecnológica que permite apresentar ao usuário o posicionamento da rota e o tempo necessário de espera por um determinado veículo de transporte coletivo.

Uma vez que o projeto apresenta soluções para as informações serem disponibilizadas nas paradas de ônibus e ainda podendo ser feitas por celulares, seria uma maneira bastante eficiente de reconquistar a confiança do usuário para que o mesmo possa voltar a usar o sistema com mais regularidade.

Durante o desenvolvimento do projeto, principalmente na implementação do protótipo de validação, que funcionou como primeiro teste operacional do sistema, foram utilizadas informações de uma linha hipotética sugerida, onde se verificou a funcionalidade do modelo apresentado.

Este projeto também prova o sucesso que se obtém com a combinação de diferentes tecnologias de diferentes setores como telecomunicações móveis, GPS, Banco de Dados, Internet e software livre.

Como sugestão para trabalhos futuros, pode-se estudar a substituição do uso da tecnologia celular GSM/GPRS por uma outra de menor custo, atendendo ainda mais à premissa de diminuição dos gastos operacionais. Outro avanço seria a implementação de uma maneira de disponibilizar para a comunidade as informações através de uma interface WEB.

BIBLIOGRAFIA

DEITEL, H. M. et al., Perl Como Programar. Tradução Carlos Arthur Lang Lisboa. Porto Alegre: Bookman, 2002.

GSM System Survey - Ericsson Radio Systems – 1998.

MONICO, J.F.G., Posicionamento pelo NAVSTAR-GPS Descrição, fundamentos e aplicações. 1. ed. Presidente Prudente: Editora UNESP, 2000.

MANN, Steve., Programando Aplicativos WAP. Tradução Lávio Pareschi – São Paulo: Makron, 2001.

Nemeth, Evi. Et al., Manual de administração do sistema Unix. Tradução Edson Furmankiewicz. Porto Alegre: Bookman, 2002.

Iezzi, Gelson, et al., Matemática. São Paulo: Atual, 1984

ANEXOS

ANEXO A – Código fonte da UNIDADE DE CONSULTA

Gpscomconsulta.pl

```
#####  
# Programa principal do Modulo da #  
# Unidade de Consulta. #  
#####  
  
#!/usr/bin/perl  
  
#####  
# Inclusão das Bibliotecas necessárias #  
# de conexão ao Banco de Dados PG. #  
#####  
  
use strict;  
use DBI;  
use DBD::mysql;  
use Tk;  
use Math::Trig;  
  
#####  
# Configuração das interfaces #  
# gráficas com o usuário #  
#####  
  
our $buttonA;  
our $buttonB;  
our $buttonC;  
our $buttonD;  
our $buttonE;  
our $buttonF;  
our $gcom;  
my $body;  
my $startfrm;  
my $radiofrm;  
my $i;  
my $filexit;  
my $chamada = new MainWindow();  
my $buttonconsult = $chamada-> Button();  
  
$chamada-> minsize ( qw(600 500));  
$chamada-> maxsize ( qw(600 500));  
$chamada-> geometry ('+180+120');  
$chamada-> title ("SISTEMA DE CONSULTA");  
$chamada-> configure (-backgroun =>'light blue');  
$buttonconsult-> configure(-text=> ' REALIZAR
```

```

CONSULTA ',-font =>"helvetica 70 bold",-default=>'active',-command =>
\&consulta);
$buttonconsult-> configure (-background=>'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonconsult-> pack(-padx=>'50', -pady=>'190');

```

```

MainLoop( );

```

```

sub consulta

```

```

{
our $main = new MainWindow();
$main-> minsize ( qw(600 500));
$main-> maxsize ( qw(600 500));
$main-> geometry ('+180+120');
$main-> title ("REALIZE AQUI SUA CONSULTA");
$main-> configure (-backgroun =>'light blue');
our $body = $main->Frame (-background=>'light blue')->
pack(-side=>'top', -fill =>'x');
our $startfrm = $body->Frame(-background=>'light blue')->
pack(-side=>'left', -fill=>'y');
our $radiofrm = $body->Frame(-background=>'light blue')->
pack(-side=>'right');
our $buttonC = $startfrm-> Button();
our $buttonB = $startfrm-> Button();
our $buttonD = $radiofrm-> Button();
our $buttonE = $radiofrm-> Button();
our $buttonF = $startfrm-> Button();
our $buttonA = $radiofrm-> Button();
our $gcom = "";

$buttonD-> configure( -text => ' Rota
400 ',-font =>"helvetica 22 bold");
$buttonD-> configure( -background=> 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonC-> configure( -text => ' Rota
300 ',-font =>'helvetica 22 bold');
$buttonC-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonE-> configure( -text => ' Rota
600 ',-font =>'helvetica 22 bold');
$buttonE-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonF-> configure( -text => ' Rota
500 ',-font =>'helvetica 22 bold');
$buttonF-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonB-> configure( -text => ' Rota
UNICEUB ',-font =>'helvetica 22 bold',-default=>'active',-command =>
\&consultareal);
$buttonB-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonA-> configure( -text => ' Rota
200 ',-font =>'helvetica 22 bold');

```

```
$buttonA-> configure( -background => 'light steel blue',-relief=>'raised',-borderwidth=>'3');
```

```
$buttonA-> pack(-padx=>'20',-pady=>'70');  
$buttonD-> pack(-padx=>'20');  
$buttonE->pack(-padx=>'20', -pady=>'70');  
$buttonB-> pack(-padx=>'20', -pady=>'70');  
$buttonC-> pack(-padx=>'20');  
$buttonF-> pack(-padx=>'20', -pady=>'70');  
}
```

```
#####  
# sub-rotinas para configuração #  
# das funções desempenhadas pela #  
# Interface gráfica. #  
#####
```

```
#####  
# sub-rotina de ir realizar a #  
# consulta a Gerencia. #  
#####
```

```
sub consultareal  
{
```

```
#####  
# Parametros de conexão ao Banco de Dados. #  
#####
```

```
my $dbhost;  
my $dbname;  
my $login;  
my $senha;  
our $dsn;  
our $dbh;  
our $sth;  
our $erroflag = "A";  
$dbhost = "201.2.126.157";  
$dbname = "gcomdb";  
$login = "gpscom";
```

```
#####  
# Conexão com o Banco de Dados. #  
#####
```

```
$dsn = "DBI:mysql:$dbname:$dbhost";  
$dbh = DBI->connect("$dsn", "$login", "", {PrintError =>0})  
or  
erro();
```

```

sub erro
{
    our $erroflag = "V";
    our $erro = new MainWindow ();
    $erro ->title("FALHA NA CONSULTA");
    $erro ->minsize( qw(370 100));
    $erro ->maxsize( qw(370 100));
    $erro-> geometry ('+300+280');
    our $label = $erro -> Label();
    $label-> configure( -text => 'Falha na consulta. Clique abaixo e tente
novamente', -font=>'arial 13 bold',
        -foreground=>'red');
    our $buttony = $erro -> Button();
    $label-> configure( -background => 'light blue');
    $erro ->configure ( -background => 'light blue');
    $buttony-> configure( -text => 'Nova consulta', -font=>'arial 16 bold',-
default=>'active',
        -command => \&destroy1);
    $buttony-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3',
        -activebackground=>'indian red');
    $label-> pack(-pady=>'10');
    $buttony-> pack(-side =>'left', -padx=>'100');
}

if ( $erroflag =~ /A/)
{
    maini();
}

sub maini
{
    my $dbtest;
    open ($dbtest,">/home/jcarvalho/gpscom/dbtest");

    our $sth = $dbh->prepare ("SELECT * FROM gcomA");
    $sth -> execute();
    my @results = $sth->fetchrow_array();
    $sth->finish();
    our $dbh->disconnect;

#####
# Sub-rotina para realizar o calculo da distancia          #
# entre os pontos e o tempo estimado de chegada.          #
#####

    my $latstop = 1545.92;
    my $longstop = 4753.66;
    my $p2lat = "1546.84";
    my $p2long = "4753.43";
    my $p1lat = "1545.83";
    my $p1long = "4753.63";

```

```

my $stoptagrau = substr ( $latstop, 0, 2 );
my $stoptamin = substr ( $latstop, 2, 2 );
my $stoptasec = substr ( $latstop, 5, 2 );
my $stoplograu = substr ( $longstop, 0, 2 );
my $stoplomin = substr ( $longstop, 2, 2 );
my $stoplosec = substr ( $longstop, 5, 2 );
my $buslagrau = substr ( $results[2], 0, 2 );
my $buslamin = substr ( $results[2], 2, 2 );
my $buslasec = substr ( $results[2], 5, 2 );
my $buslograu = substr ( $results[3], 0, 2 );
my $buslomin = substr ( $results[3], 2, 2 );
my $buslosec = substr ( $results[3], 5, 2 );

```

```

my $stoplong = -1 * ($stoplograu + ($stoplomin /60) + ($stoplosec/3600));
my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));

```

```

my $longdifer = ($stoplong - $buslong); #A#

```

```

my $stoplat = 90 + ($stoptagrau + ($stoptamin /60) + ($stoptasec/3600));
#C#

```

```

my $buslat = 90 + ($buslagrau + ($buslamin /60) + ($buslasec/3600));#B#

```

```

my $cosA = cos($longdifer);
my $cosB = cos($stoplat);
my $cosC = cos($buslat);
my $senB = sin($stoplat);
my $senC = sin($buslat);
my $arco = ($cosB * $cosC) + ($senB * $senC * $cosA);
my $acos = 6366.707 * acos($arco) * 3.14159265359/180;

```

```

#####
# Calculo de distancia ponto a ponto (1). #
#####

```

```

if ($results[2] <= $p2lat)
{
my $p2lagrau = substr ( $p2lat, 0, 2 );
my $p2lamin = substr ( $p2lat, 2, 2 );
my $p2lasec = substr ( $p2lat, 5, 2 );
my $p2lograu = substr ( $p2long, 0, 2 );
my $p2lomin = substr ( $p2long, 2, 2 );
my $p2losec = substr ( $p2long, 5, 2 );
my $buslagrau = substr ( $results[2], 0, 2 );
my $buslamin = substr ( $results[2], 2, 2 );
my $buslasec = substr ( $results[2], 5, 2 );
my $buslograu = substr ( $results[3], 0, 2 );
my $buslomin = substr ( $results[3], 2, 2 );
my $buslosec = substr ( $results[3], 5, 2 );
my $p2long = -1 * ($p2lograu + ($p2lomin /60) + ($p2losec/3600));
my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));
my $longdif = ($p2long - $buslong); #A#

```

```

my $p2lat = 90 + ($p2lagrau + ($p2lamin /60) + ($p2lasec/3600)); #C#
my $busla = 90 + ($buslagrau + ($buslamin /60) + ($buslasec/3600));

my $cosA = cos($longdif);
my $cosB = cos($p2lat);
my $cosC = cos($busla);
my $senB = sin($p2lat);
my $senC = sin($busla);
our $arcos = ($cosB * $cosC) + ($senB * $senC * $cosA);
}
our $acoss = 6366.707 * acos(our $arcos) * 3.14159265359/180;
our $distancia = $acoss + 1.420;

#####
# Calculo de distancia ponto a ponto (1). #
#####

if ($results[2] > $p2lat)
{
my $p1lagrau = substr ( $p1lat, 0, 2 );
my $p1lamin = substr ( $p1lat, 2, 2 );
my $p1lasec = substr ( $p1lat, 5, 2 );
my $p1lograu = substr ( $p1long, 0, 2 );
my $p1lomin = substr ( $p1long, 2, 2 );
my $p1losec = substr ( $p1long, 5, 2 );
my $buslagrau = substr ( $results[2], 0, 2 );
my $buslamin = substr ( $results[2], 2, 2 );
my $buslasec = substr ( $results[2], 5, 2 );
my $buslograu = substr ( $results[3], 0, 2 );
my $buslomin = substr ( $results[3], 2, 2 );
my $buslosec = substr ( $results[3], 5, 2 );

my $p1long = -1 * ($p1lograu + ($p1lomin /60) + ($p2losec/3600));
my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));

my $longdif = ($p1long - $buslong); #A#

my $p1lat = 90 + ($p1lagrau + ($p1lamin /60) + ($p1lasec/3600)); #C#
my $busla = 90 + ($buslagrau + ($buslamin /60) +
($buslasec/3600));#B#

my $cosA = cos($longdif);
my $cosB = cos($p1lat);
my $cosC = cos($busla);
my $senB = sin($p1lat);
my $senC = sin($busla);
our $arcos = ($cosB * $cosC) + ($senB * $senC * $cosA);
}
our $acoss = 6366.707 * acos(our $arcos) * 3.14159265359/180;
our $distancia = $acoss + 280;

```



```
#####
# Calculo do tempo estimado de chegada ao ponto,      #
# levando em conta a distancia encontrada e a         #
# velocidade média                                    #
#####
```

```
my $velocidade = $results[4];
$velocidade = "30";
if ( $velocidade == 0 )
{
    $velocidade = 1;
}
my $tempo = int(($distancia / $velocidade ) * 60);
```

```
if ( $tempo < 1 )
{
    $tempo = 1;
}
```

```
#print "velocidade $velocidade\n";
#print "tempo $tempo\n";
```

```
#####
# Sub-rotina grafica para apresentar                  #
# o resultado na tela.                              #
#####
```

```
our $maini = new MainWindow();
$maini-> minsize ( qw(600 500));
$maini-> maxsize ( qw(600 500));
$maini-> geometry ('+180+120');
$maini-> title ("RESULTADO DA CONSULTA");
$maini-> configure (-backgroun =>'light blue');
our $body = $maini->Frame (-background=>'light blue')->
pack(-side=>'top', -fill =>'x');
our $startfrm = $body->Frame(-background=>'light blue')->
pack(-side=>'top', -fill=>'y');
our $radiofrm = $body->Frame(-background=>'light blue')->
pack(-side=>'top');
our $labelC = $startfrm-> Label();
our $entryA = $startfrm-> Label();
our $buttonB = $startfrm-> Button();

$labelC-> configure( -text => 'Confira abaixo o resultado de sua consulta.'
,-font =>'helvetica 30 bold');
$labelC-> configure( -background => 'light blue');
$entryA-> configure( -background => 'light blue', -width => '50', -
font=>'helvetica 25',
-foreground=>'red');
```

```

if ($velocidade == 1)
{
    $entryA-> configure( - text=>"Falha na consulta. Tente novamente
mais tarde ou consulte outra Rota.");
    $buttonB-> configure( -text => ' FECHAR
CONSULTA  ', -font =>'arial 28 bold',
    -default=>'active',-command => \&destroy);
    $buttonB-> configure( -background => 'light steel blue', -relief=>'raised',-
borderwidth=>'3');
}

if ( $velocidade > 1 )
{
    $entryA-> configure( - text=>"O veiculo da Rota UNICEUB chegara a este
ponto
em cerca de $tempo min.");
    $buttonB-> configure( -text => ' FECHAR
CONSULTA  ', -font =>'arial 28 bold',
    -default=>'active',-command => \&destroy);
    $buttonB-> configure( -background => 'light steel blue', -relief=>'raised',-
borderwidth=>'3');
}
    $labelC-> pack(-pady=>'60');
    $entryA-> pack( );
    $buttonB-> pack(-padx=>'50', -pady=>'70');
}
}

sub destroy
{
    our $main ->destroy();
    our $maini ->destroy();
}

sub destroy1
{
    our $erro ->destroy();
    our $main ->destroy();
}

```

ANEXO B – Código fonte da UNIDADE REMOTA

Gpscom.pl

```
#####  
# Programa principal do Modulo da #  
# Unidade Remota. #  
#####
```

```
#!/usr/bin/perl
```

```
#####  
# Inclusao das Bibliotecas necessárias e #  
# de conexao ao Banco de Dados PG. #  
#####
```

```
use strict;  
use Tk;
```

```
#####  
# Configuração das interfaces #  
# gráficas com o usuário #  
#####
```

```
our $buttonA;  
our $buttonB;  
our $buttonC;  
our $buttonD;  
our $buttonE;  
our $buttonF;  
our $buttonG;  
our $labelA;  
our $labelB;  
our $labelC;  
our $labelD;  
our $labelE;  
our $entryA;  
our $entryB;  
our $entryC;  
our $entryD;  
our $gcom;  
my $menu_bar;  
my $arquivo;  
my $body;  
my $alarmes;  
my $startfrm;  
my $radiofrm;  
my $help;  
my $i;
```

```
my $filexit;
```

```
my $VERSION="0.1";
```

```
my $main = new MainWindow();
$main-> minsize ( qw(600 500));
$main-> maxsize ( qw(600 500));
$main-> geometry ('+180+120');
$main-> title ("Sistema de Gerencia da Unidade Remota");
$main-> configure (-background =>'light blue');
$menu_bar = $main->Frame( -relief=>'raised', -borderwidth=>'3',
    -background=>'light sky blue')->
    pack(-side=>'top', -fill=>'x');
$body = $main->Frame (-background=>'light blue')->
    pack(-side=>'top', -fill =>'x');
$startfrm = $body->Frame(-background=>'light blue')->
    pack(-side=>'left', -fill=>'y');
$radiofrm = $body->Frame(-background=>'light blue')->
    pack(-side=>'right');
$buttonC = $startfrm-> Button();
$buttonB = $startfrm-> Button();
$buttonD = $radiofrm-> Button();
$buttonE = $radiofrm-> Button();
$buttonF = $startfrm-> Button();
$buttonG = $startfrm-> Button();
$buttonA = $radiofrm-> Button();
$labelA = $startfrm-> Label();
$labelB = $startfrm-> Label();
$labelC = $radiofrm-> Label();
$labelD = $radiofrm-> Label();
$labelE = $radiofrm-> Label();
$entryA = $startfrm-> Label();
$entryB = $startfrm-> Label();
$entryC = $radiofrm-> Label();
$entryD = $radiofrm-> Label();
$arquivo = $menu_bar -> Menubutton();
$help = $menu_bar -> Menubutton();
$alarmes = $menu_bar -> Menubutton();
    $arquivo = $menu_bar -> Menubutton();
    $help = $menu_bar -> Menubutton();
    $alarmes = $menu_bar -> Menubutton();
$gcom = "";
```

```
$labelA->configure( -text =>'Latitude', -background => 'light blue', -font => 'arial
12 bold', -activebackground=>'light blue');
$labelB->configure( -text =>'Longitude', -background => 'light blue', -font => 'arial
12 bold', -activebackground=>'light blue');
$labelC->configure( -text =>'Velocidade - km/h', -background => 'light blue', -font
=> 'arial 12 bold', -activebackground=>'light blue');
$labelD->configure( -text =>'Hora', -background => 'light blue', -font => 'arial 12
bold', -activebackground=>'light blue');
$labelE->configure( -background => 'light blue');
$entryA->configure( -background => 'white', -width => '12');
$entryB->configure( -background => 'white', -width => '12');
$entryC->configure( -background => 'white', -width => '12');
```

```

$entryD->configure( -background => 'white', -width => '12');

$arquivo -> configure( -text => 'Arquivo',-background => 'light sky blue',
    -font =>'arial 12 bold',-activebackground=>'light blue',
    -foreground=> 'black');
$help -> configure( -text => 'Help', -background => 'light sky blue',
    -font =>'arial 14 bold',-activebackground=>'light sky blue',
    -foreground=> 'black');
$alarmes -> configure( -text => 'Alarmes',-background => 'light sky blue',
    -font =>'arial 14 bold',-activebackground=>'light sky blue',
    -foreground=> 'black');
$buttonD-> configure( -text => ' STATUS
GPRS   ',-font =>'arial 12 bold',-relief=>'raised',-borderwidth=>'3');
$buttonD-> configure( -background=> 'light steel blue', -command=>
\&gprs_status);
$buttonC-> configure( -text => ' STATUS
GPS     ',-font =>'arial 12 bold',-relief=>'raised',-borderwidth=>'3');
$buttonC-> configure( -background => 'light steel blue', -command=>
\&gps_status);
$buttonE-> configure( -text => ' STATUS
SERIAL ',-font =>'arial 12 bold', -command =>\&serial_status);
$buttonE-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonF-> configure( -text => ' STATUS
B DADOS ',-font =>'arial 12 bold', -command =>\&db_status);
$buttonF-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonG-> configure( -text => ' ATUALIZAR
POSICAO ', -font=>'arial 12 bold',-relief=>'raised',-borderwidth=>'3');
$buttonG-> configure( -background => 'light sky blue', -command =>
\&informe);

$arquivo-> command( -label=> 'SAIR', -activebackground => 'light blue',
    -font =>'arial 12 bold',-command =>\&stop);
$arquivo-> separator();
$help-> command( -label=> 'Sobre o GPSCOM', -activebackground => 'light
blue',
    -font =>'arial 12 bold', -command =>\&sobre);
$help-> separator();
$help-> command( -label=> 'Ajuda', -activebackground => 'light blue',
    -font =>'arial 12 bold', -command =>\&help);
$help-> separator();
$alarmes-> command( -label=> 'Alarme SERIAL', -activebackground => 'light
blue',
    -font =>'arial 12 bold', -command =>\&almserial);
$alarmes->separator();
$alarmes-> command( -label=> 'Alarmes GPS', -activebackground => 'light blue',
    -font =>'arial 12 bold', -command =>\&almgps);
$alarmes->separator();
$alarmes-> command( -label=> 'Alarmes GPRS', -activebackground => 'light
blue',
    -font =>'arial 12 bold', -command =>\&almgprs);
$alarmes-> separator();

```

```

$alarmes-> command( -label=> 'Alarme DATA BASE', -activebackground =>
'light blue',
    -font =>'arial 12 bold', -command =>\&almdatabase);
$alarmes-> separator();
$alarmes-> separator();
$alarmes-> separator();
$alarmes-> command( -label=> 'Reset Alarme List', -activebackground => 'light
blue',
    -font =>'arial 12 bold', -command =>\&reset_alm);
$alarmes-> separator();

$buttonB-> configure(-text=> ' INICIAR
GPSCOM ',-font =>'arial 12 bold',-default=>'active',-command => \&gcomini);
$buttonB-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
$buttonA-> configure( -text=> ' FINALIZAR
GPSCOM ',-font =>'arial 12 bold',-command => \&finalgcom);
$buttonA-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');

$help->pack(-side =>'right', -padx=>'30');
$arquivo->pack(-side =>'left');
$alarmes->pack(-side =>'left');
$buttonA-> pack(-padx=>'50',-pady=>'30');
$buttonD-> pack(-padx=>'50');
$buttonE->pack(-padx=>'50', -pady=>'30');
$labelE->pack(-pady=>'18');
$buttonB-> pack(-padx=>'50', -pady=>'30');
$buttonC-> pack(-padx=>'50');
$buttonF-> pack(-padx=>'50', -pady=>'30');
$buttonG-> pack (-pady=>'5');
$labelA-> pack( ),$entryA-> pack( );
$labelB-> pack( ),$entryB-> pack( );
$labelC-> pack( ),$entryC-> pack( );
$labelD-> pack( ),$entryD-> pack( );

```

```

MainLoop();

```

```

#####
# sub-rotinas para configuração      #
# das funções desempenhadas pela #
# Interface gráfica.                #
#####

```

```

sub destroymain

```

```

{
    $main->destroy();
}

```

```

sub sobre

```

```

{
    system('perl /home/jcarvalho/gpscom/./sobre.pl &');
}

```

```

}

sub help
{
    system('perl /home/jcarvalho/gpscom/./help.pl &');
}

sub serial_status
{
    my $line;
    my @word;
    my $serialalm;
    open ($serialalm, "/home/jcarvalho/gpscom/serialalert");
    $line=<$serialalm>;
    @word=split(/\s+/, $line);
    if ($word[0] =~ /v/)
    {
        $buttonE-> configure( -text => ' SERIAL
ALARME ', -font => 'arial 12 bold');
        $buttonE-> configure( -background => 'red', -activebackground => 'red');
    }else
    {
        $buttonE-> configure( -text => ' SERIAL
OK ', -font => 'arial 12 bold');
        $buttonE-> configure( -background => 'green', -activebackground
=> 'green');
    }
}

sub db_status
{
    my $line;
    my @word;
    my $dbalm;
    open ($dbalm, "/home/jcarvalho/gpscom/dbalert");
    $line=<$dbalm>;
    @word=split(/\s+/, $line);
    if ($word[0] =~ /v/)
    {
        $buttonF-> configure( -text => ' ALARME
B DADOS ', -font => 'arial 12 bold');
        $buttonF-> configure( -background => 'red', -activebackground => 'red');
    }else
    {
        $buttonF-> configure( -text => ' DATABASE
OK ', -font => 'arial 12 bold');
        $buttonF-> configure( -background => 'green', -activebackground
=> 'green');
    }
}

sub gps_status
{

```

```

my $line;
my @word;
my $gpsalm;
open ($gpsalm, "/home/jcarvalho/gpscom/gpsalert");
$line=<$gpsalm>;
@word=split(/\s+/, $line);
if ($word[0] =~ /A/ )
{
    $buttonC-> configure( -text => '    GPS
OK    ', -font => 'arial 12 bold');
    $buttonC-> configure( -background => 'green', -activebackground
=>'green');
}
else
{
    $buttonC-> configure( -text => '    GPS
ALARME ', -font => 'arial 12 bold');
    $buttonC-> configure( -background => 'red', -activebackground => 'red');
}
}

sub gprs_status
{
    my $gprs_alarm;
    my $date = `date`;
    open ($gprs_alarm, ">/home/jcarvalho/gpscom/alarmes_gprs");
    my @ppp = `ifconfig ppp0`;

    if ($ppp[0] =~ /ppp0/)
    {
        $buttonD-> configure( -text => '    GPRS
OK    ', -font => 'arial 12 bold');
        $buttonD-> configure( -background => 'green', -activebackground
=>'green');
    }
    else
    {
        $buttonD-> configure( -text => '    GPRS
ALARME ', -font => 'arial 12 bold');
        $buttonD-> configure( -background => 'red', -activebackground => 'red');
        print $gprs_alarm ( "$date GPRS ALARME - Falha na Conexao GPRS.
_____ \n");

    }
}

sub almserial
{
    system('perl /home/jcarvalho/gpscom/./alarmes_serial.pl &');
}

sub almgps
{
    system('perl /home/jcarvalho/gpscom/./alarmes_gps.pl &');
}

```



```

sub almgprs
{
    system('perl /home/jcarvalho/gpscom/./alarmes_gprs.pl &');
}

sub almdatabase
{
    system('perl /home/jcarvalho/gpscom/./alarmes_database.pl &');
}

sub reset_alm
{
    system('rm -f /home/jcarvalho/gpscom/./alarmes_serial');
    system('rm -f /home/jcarvalho/gpscom/./alarmes_gps');
    system('rm -f /home/jcarvalho/gpscom/./alarmes_gprs');
    system('rm -f /home/jcarvalho/gpscom/./alarmes_database');
    system('touch /home/jcarvalho/gpscom/alarmes_serial');
    system('touch /home/jcarvalho/gpscom/alarmes_gps');
    system('touch /home/jcarvalho/gpscom/alarmes_gprs');
    system('touch /home/jcarvalho/gpscom/alarmes_database');
}

sub finalgcom
{
    my $label;
    my $button;
    our $stop = new MainWindow ();
    $stop ->title ("STOP GPSCOM");
    $stop ->minsize( qw(250 60));
    $stop ->maxsize( qw(250 60));
    $stop-> geometry ('+367+580');
    $label = $stop -> Label();
    $label-> configure( -text => 'Gpscom finalizado', -font=>'arial 12 bold');
    $button = $stop -> Button();
    $label-> configure( -background => 'light blue');
    $stop ->configure ( -background => 'light blue');
    $button-> configure( -text => 'OK', -font=>'arial 12 bold',-default=>'active',
        -command => \&stop );
    $button-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
    $label-> pack();
    $button-> pack(-side =>'left', -padx=>'100');
}

sub stop
{
    system ('killall pppd');
    system ('killall perl');
}

```

```
#####
# sub-rotina de confirmação do #
# início do GPSCOM #
#####
```

```
sub gcomini
{
  our $label;
  our $buttona;
  our $button;
  our $button1;
  our $window = new MainWindow ();
  $window ->title ("INICIAR GPSCOM");
  $window ->minsize( qw(250 60));
  $window ->maxsize( qw(250 60));
  $window-> geometry ('+367+580');
  $label = $window -> Label();
  $buttona = $window -> Button();
  $label-> configure( -text => 'Iniciar Gpscom?', -font=>'arial 12 bold');
  $button = $window -> Button();
  $button1 = $window -> Button();
  $label-> configure( -background => 'light blue');
  $window ->configure ( -background => 'light blue');
  $button-> configure( -text => 'SIM', -font=>'arial 12 bold',-command => \&gcom);
  $button-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
  $button1-> configure( -text => 'FECHAR', -font=>'arial 12 bold',-command =>
\&destroy1 );
  $button1-> configure( -background => 'light steel blue',-relief=>'raised',-
borderwidth=>'3');
  $label-> pack();
  $button-> pack(-side =>'left', -padx=>25);
  $button1-> pack(-side =>'right', -padx=>25);
}

```

```
#####
# sub-rotina de informações sobre a #
# posição do veículo. #
#####
```

```
sub informe
{
  my $data;
  my $lines;
  my $hora;
  my $horacerta;
  my $ponto = ":";

  open ($data, "/home/jcarvalho/gpscom/newgpsdata");
  while(!eof($data))
  {
    $lines=<$data>;
    my @word=split(/\s+/, $lines);

```

```

if ($word[0] =~/HORA/)
{
    $hora = $word[2];
    $entryD->configure( -text =>"$hora");
};
if ($word[0] =~/LATITUDE/)
{
    my $lat = $word[2];
    my $latgrau = substr ( $lat, 0, 2 );
    my $latmin = substr ( $lat, 2, 2 );
    my $latsec = substr ( $lat, 5, 2 );
    my $test = cos ( 1 );
    $entryA->configure( -text =>"$latgrau $latmin' $latsec""");
}
if ($word[0] =~/LONGITUDE/)
{
    my $long = $word[2];
    my $longgrau = substr ( $long, 1, 2 );
    my $longmin = substr ( $long, 3, 2 );
    my $longsec = substr ( $long, 6, 2 );
    $entryB->configure( -text =>"$longgrau $longmin' $longsec""");
}
if ($word[0] =~/VELOCIDADE/)
{
    my $speed = $word[2];
    $entryC->configure( -text =>"$speed");
}
}

sub gcom
{
    system ('perl /home/jcarvalho/gpscom/./monitora.pl &');
    our $label-> configure( -text => 'Click em FECHAR', -font=>'arial 12 bold', -
foreground =>'red');
    our $button-> configure( -text => 'INICIADO', -font=>'arial 12 bold',-background
=> 'green',
        -activebackground =>'green');
}

sub destroy1
{
    our $window ->destroy();
}
}

```

```
#####  
# Programa principal do Modulo da #  
# Unidade Remota. #  
#####  
  
#!/usr/bin/perl  
  
#####  
# Inclusao das Bibliotecas necessárias e #  
# de conexao ao Banco de Dados PG. #  
#####  
  
use strict;  
use DBI;  
use DBD::mysql;  
  
#####  
# Sub-rotina de abertura da porta serial #  
# na velocidade de 4800 bps. #  
# Mesma velocidade do protocolo de #  
# comunicacao do GPS. #  
#####  
  
my $date;  
for ( my $i = 0; $i < 3; $i++ )  
{  
  
    $date = `date`;  
    my $porta = shift;  
    my $alarmfile;  
    my $serialalert;  
        open ($alarmfile,">/home/jcarvalho/gpscom/alarmes_serial");  
        open ($serialalert,">/home/jcarvalho/gpscom/serialalert");  
  
        open ($porta, "+</dev/ttyS0")  
            or  
                print $alarmfile ( "$date SERIAL ALARME - Falha na  
Conexao com a Porta Serial.  
_____  
print $serialalert ( "v" );  
                \n"), &&  
  
        system("stty 4800 raw < /dev/ttyS0");  
  
#####  
# Inicio da parte funcional do #  
# programa GPSCOM. #  
#####  
  
sub leitura  
{
```

```

my $par = shift;
my $data = 0;
my $buffer = 0;
my $gp = "";
my $timeout = 130;

#####
# Sub-rotina de configuração #
# de parâmetros de leitura. #
#####

do
{
    sysread($porta,$buffer,256);
    $data .= $buffer
}
while ($data !~ /$par/i && $timeout--);

return $data;
}

#####
# Sub-rotina de definicao de qual porta #
# serial será usada para leitura. #
#####
#####
# Sub-rotina de gravacao dos dados coletados #
# em um arquivo texto "GPSDATA.TXT", para #
# posterior tratamento. #
#####

my $output = leitura($porta,"#");
my $outfile;
open ($outfile,">gpsdata");
print $outfile $output;
close $outfile;

#####
# Fim do programa de leitura e gravacao #
# dos dados enviados pelo GPS. #
#####

#####
# Início do programa de passar dos dados #
# coletados pelo GPSCOM, e tratamento #
# da sentença $GSRMC. #
#####

my $line;
my $hora;
my $alerta;
my @word;

```

```
my $lat;
my $longit;
my $speed;
my $data;
my $infile;
my $latgrau;
my $latmin;
my $latsec;
my $longgrau;
my $longmin;
my $longsec;
my @speed;
```

```
#####
# Abre os arquivos de entrada e saída. #
#####
```

```
open ($infile, "/home/jcarvalho/gpscom/gpsdata");
open ($outfile,">/home/jcarvalho/gpscom/newgpsdata");
```

```
#####
# Sub-rotina de pesquisa do arquivo de #
# entrada e posterior gravação dos #
# dados procurados no arquivo de saída. #
#####
```

```
while(!eof($infile))
{
    $line=<$infile>;
    @word=split(/,/, $line);
    if ($word[0] =~/RMC/)
    {
        $hora = $word[1] - 30000;
        $alerta = $word[2];
        $lat = $word[3];
        $longit = $word[5];
        $speed = $word[7]*1.85;
        $data = $word[9];
    }
}
```

```
#####
# Sub-rotina para gravação dos dados #
# no arquivo de saída. #
#####
```

```
print $outfile ( "HORA - $hora\nALERTA - $alerta\nLATITUDE - $lat\n");
print $outfile ( "LONGITUDE - $longit\nVELOCIDADE - $speed\nDATA - $data\n");
close $infile;
close $outfile;
```

```
#####
# Sub-rotina de gravação da variável #
# alerta no arquivo GPSALERT.      #
#####

my $outalert;
my $alarmfile_gps;
my $date = `date`;
open ($alarmfile_gps, ">/home/jcarvalho/gpscom/alarmes_gps");
open ($outalert, ">/home/jcarvalho/gpscom/gpsalert");
print $outalert ($alerta);
    if ($alerta =~ /V/)
    {
        print $alarmfile_gps ( "$date GPS ALARME - Falha na Conexao ou
cobertura GPS.
_____ \n");
    }
}
```

```
#####
# Parâmetros de conexão ao Banco de Dados. #
#####
```

```
my $dbhost;
my $dbname;
my $login;
my $senha;
my $dsn;
my $dbh;
my $sth;
```

```
$dbhost = "201.2.126.157";
$dbname = "gcomdb";
$login = "gpscom";
```

```
#####
# Conexão com o Banco de Dados.#
#####
```

```
my $dbalert;
my $db_alarm;
my $date = `date`;
open ($db_alarm, ">/home/jcarvalho/gpscom/alarmes_database");
open ($dbalert, ">/home/jcarvalho/gpscom/dbalert");
$dsn = "DBI:mysql:$dbname:$dbhost";
$dbh = DBI->connect("$dsn", "$login", "")
    or
        print $db_alarm ( "$date DATABASE ALARME - Falha na Conexao com
o Banco de Dados.
_____ \n"), &&
print $dbalert ( "v" );
```

```
$dbh-> do ("UPDATE gcomA SET
```

```
hora = '$hora',  
alerta = '$alerta',  
latitude = '$lat',  
longitude = '$longit',  
speed = '$speed',  
data = '$data' ")
```

or

```
print $db_alarm ( "$date DATABASE ALARME - Falha na Conexao com o  
Banco de Dados.
```

```
_____ \n"), &&  
print $dbalert ( "v" );
```

```
$dbh->disconnect;
```

```
$i--;
```

```
}
```


alarmes_database.pl

```
#####  
# Aplicativo para apresentar na tela #  
# informações sobre os alarmes no #  
# GPSCOM #  
#####  
  
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;  
  
#####  
# Rotina de configuracao da janela #  
# e chamada do arquivo alarmes_database. #  
#####  
  
my $mw = new MainWindow();  
my $t1 = $mw->Scrolled('Viewer', -wrap => 'none');  
$t1 = $mw->Viewer()->pack();  
$t1->Load("/home/jcarvalho/gpscom/alarmes_database");  
  
MainLoop();  
  
#####
```

alarmes_gprs.pl

```
#####  
# Aplicativo para apresentar na tela #  
# informações sobre os alarmes no #  
# GPSCOM #  
#####  
  
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;  
  
#####  
# Rotina de configuração da janela #  
# e chamada do arquivo alarmes_gprs #  
#####  
  
my $mw = new MainWindow();  
my $t1 = $mw->Scrolled('Viewer', -wrap => 'none');  
$t1 = $mw->Viewer()->pack();  
$t1->Load("/home/jcarvalho/gpscom/alarmes_gprs");  
  
MainLoop();  
  
#####
```

alarmes_gps

```
#####  
# Aplicativo para apresentar na tela #  
# informações sobre os alarmes no #  
# GPSCOM #  
#####  
  
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;  
  
#####  
# Rotina de configuração da janela #  
# e chamada do arquivo alarmes_gps #  
#####  
  
my $mw = new MainWindow();  
our $t1 = $mw->Scrolled('Viewer', -wrap => 'none');  
  
$t1 = $mw->Viewer()->pack();  
$t1 ->Load("/home/jcarvalho/gpscom/alarmes_gps");  
  
MainLoop();  
  
#####
```

alarmes_serial

```
#####  
# Aplicativo para apresentar na tela #  
# informações sobre os alarmes no #  
# GPSCOM #  
#####  
  
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;  
  
#####  
# Rotina de configuração da janela #  
# e chamada do arquivo almarmes_serial. #  
#####  
  
my $mw = new MainWindow();  
my $t1 = $mw->Scrolled('Viewer', -wrap => 'none');  
$t1 = $mw->Viewer()->pack();  
$t1->Load("/home/jcarvalho/gpscom/alarmes_serial");  
  
MainLoop();  
  
#####
```

help.pl

```
#####  
#      Aplicativo para apresentar na tela      #  
#      informações de ajuda sobre o programa  #  
#              GPSCOM                          #  
#####
```

```
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;
```

```
#####  
#      Rotina de conFiguração da janela      #  
#      e chamada do arquivo help.          #  
#####
```

```
my $mw = new MainWindow();  
my $t1 = $mw->Scrolled('Viewer', -wrap=> 'char');  
    $t1 = $mw->Viewer()->pack();  
    $t1->Load("/home/jcarvalho/gpscom/help");
```

```
    MainLoop();
```

```
#####
```

sobre.pl

```
#####  
#      Aplicativo para apresentar na tela      #  
#      informações sobre o programa            #  
#      GPSCOM                                  #  
#####  
  
#!/usr/bin/perl  
use strict;  
use warnings;  
use Tk;  
use Tk::Text::Viewer;  
  
#####  
#      Rotina de configuração da janela        #  
#      e chamada do arquivo sobre.            #  
#####  
  
my $mw = new MainWindow();  
my $t1 = $mw->Scrolled('Viewer', -wrap => 'none');  
    $t1 = $mw->Viewer()->pack();  
# $t1 = $mw->configure ( -background => 'sky blue');  
    $t1->Load("/home/jcarvalho/gpscom/sobre");  
  
    MainLoop();  
  
#####
```

gprs.sh

```
#####  
#      Arquivo de Conexão pelo pppd ao modem GPRS      #  
#####  
  
#nodetach  
debug  
  
#####  
#      script para inicializar o modem GPRS. Alguns telefones      #  
#      costumam trabalhar com o wvdial, outros com o chat.      #  
#####  
  
#####  
#      script de conexão pelo chat.      #  
#####  
  
connect /etc/ppp/peers/gprs-on  
  
#####  
#      script de conexão pelo wvdial.      #  
#####  
  
#connect "/usr/bin/wvdial --chat --config /etc/ppp/peers/gprs-on.conf "  
  
#####  
#      wvdial comentado, por estar sendo usado o chat.      #  
#####  
  
#####  
#      chamada script para desconexão.      #  
#####  
  
disconnect /etc/ppp/peers/gprs-off  
  
#####  
#      Porta serial e conFiguração de parâmetros.      #  
#####  
  
/dev/ttyUSB0  
57600  
crtscts  
local  
noipdefault  
ipcp-accept-local  
defaultroute  
usepeerdns
```

```
#####  
# Username para uso no chap-secrets ou pap-secrets. #  
#####
```

```
user "claro"
```

```
#####  
# O comando persist irá tentar restabelecer a #  
# conexão em caso de falha. #  
#####
```

```
#persist  
maxfail 99
```

```
#####
```


gprs-off.sh

```
#####  
#   Arquivo para desconectar o modem GPRS.   #  
#####  
  
#!/bin/sh  
  
exec /usr/sbin/chat -V -s -S      \  
ABORT          "BUSY"             \  
ABORT          "ERROR"            \  
ABORT          "NO DIALTONE"      \  
SAY            "\nSending break to the modem\n"  \  
""            "\K"                 \  
""            "\K"                 \  
""            "\K"                 \  
""            "+++ATH"           \  
""            "+++ATH"           \  
""            "+++ATH"           \  
SAY            "\nPDP context detached\n"
```

gprs-on.sh

```
#####  
#   Arquivo chat dos comandos AT de inicialização do modem.   #  
#####
```

```
#!/bin/sh
```

```
exec chat \
TIMEOUT      5 \
ECHO         ON \
ABORT       "\nBUSY\r" \
ABORT       "\nERROR\r" \
ABORT       "\nNO ANSWER\r" \
ABORT       "\nNO CARRIER\r" \
ABORT       "\nNO DIALTONE\r" \
ABORT       "\nRINGING\r\n\r\nRINGING\r" \
"           \rAT \
TIMEOUT      10 \
SAY         "Press CTRL-C to close the connection at any stage!" \
SAY         "\ndefining PDP context...\n" \
OK          ATH \
OK          ATE1 \
OK          'AT+CGDCONT=1,"IP","claro.com.br"' \
OK          ATDT*99***1# \
TIMEOUT      15 \
SAY         "\nwaiting for connect...\n" \
CONNECT     "" \
SAY         "\nConnected." \
SAY         "\nIf the following ppp negotiations fail,\n" \
SAY         "try restarting the phone.\n"
login:      "claro"
password:   "claro"
```

gprs-on.conf

```
#####  
#   Arquivo conf wvdial para o modem.   #  
#####
```

```
Init1 = ATH  
Init2 = ATE1  
Init3 = AT+CGDCONT=1,"IP","claro.com.br"  
Dial Command = ATDT  
Phone = *99***1#  
username = claro  
Password = claro
```

ANEXO C – Código fonte do SISTEMA DE GERÊNCIA DE REDE

movel.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id ="rota" title="Selecione a Rota">
<p>
<a href="#parada">Rota UNICEUB</a><br />
<a href="">Rota 200</a><br />
<a href="">Rota 300</a><br />
<a href="">Rota 400</a><br />
<a href="">Rota 500</a><br />
<a href="">Rota 600</a>
</p>
</card>
<card id ="parada" title="Selecione a parada">
<p>
<a href="cgi-bin/movel.cgi">Parada Bloco 7</a><br />
<a href="">Parada Bloco 1</a><br />
<a href="">Parada CMB</a><br />
<a href="">Parada Biblioteca</a>
</p>
</card>
</wml>
```

move1.cgi

```
#####  
# Programa principal da #  
# consulta via celular. #  
#####  
  
#!/usr/bin/perl  
  
use DBI;  
use DBD::mysql;  
use Math::Trig;  
use CGI::WML;  
$query = new CGI::WML;  
  
my $dbhost;  
my $dbname;  
my $login;  
my $senha;  
my $dsn;  
my $dbh;  
my $sth;  
my $erroflag = "A";  
$dbh = DBI->connect("DBI:mysql:gcomdb","root","");  
$sth = $dbh->prepare("SELECT * FROM gcomA");  
$sth -> execute();  
my @results = $sth->fetchrow_array();  
$sth->finish();  
$dbh->disconnect;  
my $latstop = 1545.92;  
my $longstop = 4753.66;  
my $p2lat = "1546.84";  
my $p2long = "4753.43";  
my $p1lat = "1545.83";  
my $p1long = "4753.63";  
my $stoplagrau = substr ( $latstop, 0, 2 );  
my $stoplamin = substr ( $latstop, 2, 2 );  
my $stoplasec = substr ( $latstop, 5, 2 );  
my $stoplograu = substr ( $longstop, 0, 2 );  
my $stoplomin = substr ( $longstop, 2, 2 );  
my $stoplosec = substr ( $longstop, 5, 2 );  
my $buslagrau = substr ( $results[2], 0, 2 );  
my $buslamin = substr ( $results[2], 2, 2 );  
my $buslasec = substr ( $results[2], 5, 2 );  
my $buslograu = substr ( $results[3], 0, 2 );  
my $buslomin = substr ( $results[3], 2, 2 );  
my $buslosec = substr ( $results[3], 5, 2 );  
my $stoplong = -1 * ($stoplograu + ($stoplomin /60) + ($stoplosec/3600));  
my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));  
my $longdifer = ($stoplong - $buslong); #A#  
my $stoplat = 90 + ($stoplagrau + ($stoplamin /60) + ($stoplasec/3600));  
my $buslat = 90 + ($buslagrau + ($buslamin /60) + ($buslasec/3600));#B#  
my $cosA = cos($longdifer);
```

```

my $cosB = cos($stoplat);
my $cosC = cos($buslat);
my $senB = sin($stoplat);
my $senC = sin($buslat);
my $arco = ($cosB * $cosC) + ($senB * $senC * $cosA);
my $acos = 6366.707 * acos($arco) * 3.14159265359/180;
  if ($results[2] <= $p2lat)
  {
    my $p2lagrau = substr ( $p2lat, 0, 2 );
    my $p2lamin = substr ( $p2lat, 2, 2 );
    my $p2lasec = substr ( $p2lat, 5, 2 );
    my $p2lograu = substr ( $p2long, 0, 2 );
    my $p2lomin = substr ( $p2long, 2, 2 );
    my $p2losec = substr ( $p2long, 5, 2 );
    my $buslagrau = substr ( $results[2], 0, 2 );
    my $buslamin = substr ( $results[2], 2, 2 );
    my $buslasec = substr ( $results[2], 5, 2 );
    my $buslograu = substr ( $results[3], 0, 2 );
    my $buslomin = substr ( $results[3], 2, 2 );
    my $buslosec = substr ( $results[3], 5, 2 );
    my $p2long = -1 * ($p2lograu + ($p2lomin /60) + ($p2losec/3600));
    my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));
    my $longdif = ($p2long - $buslong); #A#
    my $p2lat = 90 + ($p2lagrau + ($p2lamin /60) + ($p2lasec/3600)); #C#
    my $busla = 90 + ($buslagrau + ($buslamin /60) + ($buslasec/3600));#B
    my $cosA = cos($longdif);
    my $cosB = cos($p2lat);
    my $cosC = cos($busla);
    my $senB = sin($p2lat);
    my $senC = sin($busla);
    our $arcos = ($cosB * $cosC) + ($senB * $senC * $cosA);
  }
  our $acoss = 6366.707 * acos(our $arcos) * 3.14159265359/180;
  our $distancia = $acoss + 1.420;
if ($results[2] > $p2lat)
{
  my $p1lagrau = substr ( $p1lat, 0, 2 );
  my $p1lamin = substr ( $p1lat, 2, 2 );
  my $p1lasec = substr ( $p1lat, 5, 2 );
  my $p1lograu = substr ( $p1long, 0, 2 );
  my $p1lomin = substr ( $p1long, 2, 2 );
  my $p1losec = substr ( $p1long, 5, 2 );
  my $buslagrau = substr ( $results[2], 0, 2 );
  my $buslamin = substr ( $results[2], 2, 2 );
  my $buslasec = substr ( $results[2], 5, 2 );
  my $buslograu = substr ( $results[3], 0, 2 );
  my $buslomin = substr ( $results[3], 2, 2 );
  my $buslosec = substr ( $results[3], 5, 2 );
  my $p1long = -1 * ($p1lograu + ($p1lomin /60) + ($p1losec/3600));
  my $buslong = -1 * ($buslograu + ($buslomin /60) + ($buslosec/3600));
  my $longdif = ($p2long - $buslong); #A#
  my $p1lat = 90 + ($p1lagrau + ($p1lamin /60) + ($p1lasec/3600)); #C#

```

```

        my $busla = 90 + ($buslagrau + ($buslamin /60) +
($buslasec/3600));#B#
        my $cosA = cos($longdif);
        my $cosB = cos($p2lat);
        my $cosC = cos($busla);
        my $senB = sin($p2lat);
        my $senC = sin($busla);
        our $arcos = ($cosB * $cosC) + ($senB * $senC * $cosA);
    }
    our $acoss = 6366.707 * acos(our $arcos) * 3.14159265359/180;
    our $distancia = $acoss + 280;
my $velocidade = $results[4];
if ( $velocidade == 0 )
{
    $velocidade = 1;
}
my $tempo = int(($distancia / $velocidade ) * 60);
if ( $tempo < 1 )
{
    $tempo = 1;
}
if ($velocidade == 1)
{
    My $scoringa = "Tempo de chegada indeterminado. Tente novamente
mais tarde.";
    print
        $query->header(),
        $query->start_wml(),
        $query->template(-content=>$query->prev()),
        $query->card(-id=>"resultado",
        -title=>"Resultado da consulta",
        -content=>($scoringa),
        $query->end_wml());
    print
    $query->wml_to_wmlc(-wml=> $wml_buffer,
        -errorcontext=>2);
    ($page_title,$content) = $query->html_to_wml($buffer);
} else
{
    my $scoringa = $query->p("Tempo aproximado de chegada: $tempo.");
    print
        $query->header(),
        $query->start_wml(),
        $query->template(-content=>$query->prev()),
        $query->card(-id=>"resultado",
        -title=>"Resultado da consulta",
        -content=>($scoringa),
        $query->end_wml());
    print
    $query->wml_to_wmlc(-wml=> $wml_buffer,
        -errorcontext=>2);
    ($page_title,$content) = $query->html_to_wml($buffer);
}

```

