

HECTOR ENRIQUE DE LA HOZ LEÓN

**DESENVOLVIMENTO DE PLATAFORMA DE
PUBLICIDADE BASEADA NA LOCALIZAÇÃO:
ESTUDO DE CASO PARA O SISTEMA DE
TRANSPORTE URBANO DE FLORIANÓPOLIS**

FLORIANÓPOLIS, 2012

**INSITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E
TECNOLOGIA DE SANTA CATARINA
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO DE PÓS-GRADUAÇÃO (LATO SENSU) EM
DESENVOLVIMENTO DE PRODUTOS ELETRÔNICOS**

HECTOR ENRIQUE DE LA HOZ LEÓN

**DESENVOLVIMENTO DE PLATAFORMA DE
PUBLICIDADE BASEADA NA LOCALIZAÇÃO:
ESTUDO DE CASO PARA O SISTEMA DE
TRANSPORTE URBANO DE FLORIANÓPOLIS**

Monografia submetida ao Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina como parte dos requisitos para obtenção do título de Especialista em Desenvolvimento de Produtos Eletrônicos.

Professor Orientador: Fernando S. Pacheco, Dr.

FLORIANÓPOLIS, 2012

CDD 526.1
L184d

La Hoz, Hector Enrique de
Desenvolvimento de plataforma de publicidade baseada na localização:
estudo de caso para o sistema de transporte urbano de Florianópolis
[monografia] / Hector Enrique de La Hoz; orientação de Fernando S.
Pacheco. – Florianópolis, 2013.

1 v. : il.

Monografia de especialização (Desenvolvimento de Produtos Eletrônicos) –
Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.
Curso de Especialização em Desenvolvimento de Produtos Eletrônicos.

Inclui referências.

1. GPS. 2. GIS. 3. Local Based Advertising (LBA). I. Pacheco, Fernando S.
II. Título.

Sistema de Bibliotecas Integradas do IFSC
Biblioteca Dr. Hercílio Luz – Campus Florianópolis
Catalogado por: Edinei Antonio Moreno CRB 14/1065
Rose Mari Lobo Goulart CRB 14/277

**DESENVOLVIMENTO DE PLATAFORMA DE
PUBLICIDADE BASEADA NA LOCALIZAÇÃO:
ESTUDO DE CASO PARA O SISTEMA DE
TRANSPORTE URBANO DE FLORIANÓPOLIS**

HECTOR ENRIQUE DE LA HOZ LEON

Este trabalho foi julgado adequado para obtenção do Título de Especialista e aprovado na sua forma final pela banca examinadora do Curso de Pós-Graduação (Lato Sensu) em Desenvolvimento de Produtos Eletrônicos do Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina.

Florianópolis, 21 de dezembro de 2012.

Banca Examinadora:

Prof. Fernando S. Pacheco, Dr.

Prof. Joel Lacerda, Dr.

Prof. Luís Carlos Martinhago Schlichting, Dr.

Dedico este trabalho aos meus familiares.

Aos meus professores.

AGRADECIMENTOS

Primeiro a Deus pela vida e saúde.

Aos meus pais, esposa e filho pelos sacrifícios, pela companhia e pelo apoio.

Ao professor Fernando por me dar a honra de ser seu orientado e por acreditar fortemente neste projeto.

Ao meu chefe Dhelyo Rodriguez Pereira e sua empresa CSP Controle e Automação pela chance de trabalhar e estudar nesta maravilhosa cidade.

Aos meus amigos de toda a vida que de alguma forma contribuíram para minha formação como cidadão.

Aos amigos do IF-SC.

RESUMO

Pelo seu crescimento e pelo modelo de transporte urbano empregado, Florianópolis tem na mobilidade um dos seus maiores desafios no futuro imediato. Contar com sistemas de transporte urbanos mais inteligentes, econômicos e eficazes é uma das soluções de maior ressonância na cidade. Com isso, um número maior de pessoas se sentirá atraído em utilizar o sistema, criando oportunidades para a divulgação de informações que dinamizem a economia e gerem um maior consumo de produtos por parte do cidadão. Este projeto visa o desenvolvimento de uma plataforma para sistemas de informação que procurem fornecer dados comerciais em lugares e momentos determinados para os usuários do sistema de transporte urbano da cidade de Florianópolis. Fundamentada no hardware aberto BeagleBoard, a plataforma desenvolvida neste projeto utiliza o sistema de posicionamento global (GPS) para determinar a localização do veículo monitorado e, com base nesses dados, exibir informações da região por onde o veículo está transitando. Além disso, dados relacionados com a viagem (como a rota, a velocidade, etc.) e com os locais comerciais da região também são exibidos pela mesma plataforma. Por meio de simulações foram feitos alguns testes que consistiam em alimentar a plataforma com coordenadas geográficas pertencente ao trajeto de uma linha de ônibus da cidade (TICEN-TITRI). Assim verificou-se a ativação de tarefas de cálculos de distância para determinar se o veículo se encontra ou não em uma região de interesse comercial. Ao final do processo de cálculo, o sistema decide se deve exibir informação relacionada ao lugar onde atualmente se encontra o veículo. Com esse teste foi possível visualizar, além do funcionamento da plataforma, a pertinência deste tipo de tecnologia no sistema de transporte da cidade de Florianópolis. O projeto foi desenvolvido visando baixo custo e com um orçamento para materiais de R\$ 230 foi construído um protótipo funcional da plataforma.

Palavras-chave: GPS. GIS. 3G. Local Based Advertising (LBA).

ABSTRACT

By its growth and the urban transport model employed, Florianópolis has in the mobility one of their biggest challenges in the immediate future. Having a more intelligent, more effective and more economical urban transport systems is one of the solutions with most resonance in the city. A better system would attract a larger number of users, creating opportunities to disclose information that streamline the economy and generate increased consumption of products and services by the citizens. This project aims to develop a information systems platform that seek to provide valuable data in certain places and times for users of the transportation system of Florianópolis. Based on the open hardware BeagleBoard, the platform developed in this project uses the global positioning system (GPS) to determine the location of monitored vehicles and, based on these data, display information for the area where this vehicle is traveling. In addition, data related to the travel (as the route, speed, etc.) and to commercial places in the region are also displayed by the same platform. Through simulations, some tests were carried out. In these tests, geographical coordinates belonging to the path of a city bus line (TICEN-TITRI), were fed to the platform forcing the activation of tasks such as distance calculations to determine if the vehicle is or is not in an area of commercial interest. At the end of the distance calculation process, the system decides whether to display information related to the place where the car is currently located. In addition to the operation of the platform, it was possible to visualize the relevance of this type of technology in the transportation system of the city of Florianópolis. The project was developed thinking in low cost and with R\$ 230 budget for materials a fully functional prototype was made.

Key-words: GPS. GIS. 3G. Local Based Advertising (LBA).

LISTA DE FIGURAS

Figura 1. Exemplos de LBS. a. City24x7, b. YELP, c. Find My Friends, d. NEER, e. Google Latitude.	24
Figura 2. Diagrama geral da aplicação desenvolvida neste trabalho.....	25
Figura 3. Diagrama de blocos da arquitetura dos LBS/LBAD. ..	33
Figura 4. Beagle Board e interfaces para conexão de periféricos utilizados no projeto.	37
Figura 5. Módulo GPS C2626.	40
Figura 6 Módulo GPS BU-353.....	41
Figura 7 Módulo de comunicação celular G24.....	43
Figura 8 Modem de comunicação celular E1750 com interface USB.	43
Figura 9 Diagrama de conexões entre a BeagleBoard e demais módulos de hardware no sistema de teste.	46
Figura 10. Diagrama de blocos do software desenvolvido.	48
Figura 11. Mapa de Florianópolis salientando bairros e terminais integrados de transporte.	50
Figura 12. Janelas do X11 e PPP-Config.	54
Figura 13 Diagrama de fluxo do software gerenciador.	56
Figura 14. Execução do arquivo HTML/JS no <i>chromium</i> apresentando a rota e a posição de um ônibus TICEN-TITRI. ..	59

LISTA DE TABELAS

Quadro 1 - Especificações do <i>hardware</i> BeagleBoard.....	38
Quadro 2 - Características do módulo GPS C2626.....	40
Quadro 3 - Características do módulo GPS BU-353.....	42
Quadro 4 - Características do módulo GSM G24.....	44
Quadro 5 - Características do módulo E1750.....	45
Quadro 6 - Consumo de corrente do sistema em diferentes cenários.....	63

ABREVIATURAS

API – Application Programming Interface

AVI – Audio Video Interface

BB – Beagle Board

DNS – Domain Name System

FTP – File Transfer Protocol

GIS – Geographic Information System

GSM – Global System for Mobile

GPS – Global Positioning System

HDMI – High Definition Multimedia Interface

HTML – Hyper Text Markup Language

IFSC – Instituto Federal de Santa Catarina

ISP – Internet Service Provider

JS – Java Script

JSON – Java Script Object Notation

LBAD – Location Based advertising Device

LBS – Location Based Services

OS – Operating System

PCI – Placa de Circuito Impresso

PPP – *Point to Point Protocol*

SCP – *Secure Copy*

SD – *Secure Digital Card*

SSH – *Secure Shell*

TCP – *Transmission Control Protocol*

UMTS – *Universal Mobile Telecommunications System*

USB – *Universal Serial Bus*

SUMÁRIO

1. INTRODUÇÃO	31
1.1 Objetivos do Trabalho	25
1.2 Estrutura da Monografia.....	26
2. FUNDAMENTOS TEÓRICOS.....	27
2.1 Modo de Uso e Modelo de Usuário	27
2.2 Modo de Localização e modelo de contexto	29
2.3 Classificação dos LBS.....	31
2.4 Componentes dos LBS	32
3. DESENVOLVIMENTO	35
3.1 Hardware	35
3.1.1 BeagleBoard.....	37
3.1.2 GPS	39
3.1.3 Modem 3G.....	41
3.1.4 Monitor.....	44
3.1.5 Hub USB.....	45
3.1.6 Resultado da etapa de desenvolvimento de hardware	46
3.2 Software	47
3.2.1 Considerações iniciais para a aplicação do LBS no sistema de transporte da cidade de Florianópolis.....	47
3.2.2 Instalação e configuração do sistema.....	50
3.2.1 Conceitos e definições do software gerenciador	54
3.2.2 Pré-Configuração do sistema e arquivo XML	57
3.2.3 Determinação da localização, análise de contexto e exibição de informação	57

3.2.3 Resultados da etapa de desenvolvimento de software.....	62
3.3 Resultados dos testes e simulações	62
4. CONCLUSÕES	65
REFERÊNCIAS.....	67
APÊNDICE A - Instalação Ubuntu Linux 11.10 na Beagle Board rev. C3.....	71
APÊNDICE B - Comunicação entre a Beagle Board e uma rede de computadores com acesso à Internet.....	74
APÊNDICE C - Configuração de comunicações GSM/3G com o PPP-Config no Linux.....	77
APÊNDICE D - Arquivo XML que condensa as informações da rota de exemplo TICEN-TITRI	79

1. INTRODUÇÃO

Tradicionalmente, as informações dos locais turísticos e comerciais das cidades têm sido divulgadas e pesquisadas em guias telefônicos e jornais. Com a difusão da internet e a criação de produtos de consumo geral que embarcam tecnologias de sensoriamento e telecomunicações (como *smartphones* e *tablets*), as tarefas de localização e identificação geográficas tem-se tornado atividades comuns na vida das pessoas. Aplicações web interativas e fáceis de utilizar (como o caso do Googlemaps) têm aparecido como substitutos dos métodos de busca tradicionais, contando atualmente com um grande número de usuários. Como consequência disso, alguns setores comerciais e industriais têm desenvolvido e explorado os chamados Serviços Baseados na Localização (LBS, do inglês *Location Based Services* ou LBA, do inglês *Location Based Advertising*), os quais geram informações que permitem aos seus usuários casar necessidades comerciais com a sua localização espacial. As aplicações atuais abrangem diversos setores econômicos como os de transporte, saúde, educação, comércio, etc. Dado o grande interesse de vários setores comerciais pelos LBS e graças aos avanços tecnológicos em matéria de interfaces e telecomunicações, uma grande quantidade de produtos deste gênero tem surgido nos últimos cinco anos, mudando a forma com que os cidadãos interagem com o seu entorno e com as cidades. É o caso do sistema chamado **City24x7** que está sendo instalado na cidade de Nova Iorque (BAND NEWS, 2012). Essa proposta consiste em trocar cabines de telefones públicos por sistemas com telas sensíveis ao toque para que os cidadãos possam consultar informações como as rotas e horários dos sistemas de transportes públicos, a programação dos teatros e cinemas, etc. Embora não possa ser acessado em todo momento (dado que não é um sistema móvel), seu sucesso vem levando a prefeitura de Nova Iorque a prever a instalação de

aproximadamente duzentos destes sistemas em toda a cidade até o final de 2013. A Figura 1(a) apresenta uma imagem da plataforma City24x7. Uma proposta diferente é oferecida pelo **YELP** (YELP, 2012). Esse produto é um motor de busca (*search engine*) que permite obter informações específicas sobre um dado local. Em uma consulta, o usuário fornece, além do objetivo da busca (cabeleireiros, hospitais, lojas, por exemplo), alguns dados relacionados ao local de interesse (CEP, endereço, etc.). Como resultado final, o sistema retorna uma lista dos locais que se encaixam na descrição fornecida e também uma qualificação (de zero a cinco estrelas) de cada um dos resultados encontrados. Tal qualificação é fornecida por outros usuários do YELP, criando assim uma verdadeira comunidade *online* que, até janeiro de 2012, contava com 71 milhões de acessos mensais (Van Grove, 2010). Outros produtos como **LAYAR**, **Find My Friends** da Apple, **Latitude** da Google e **NEER** da Qualcomm tem o seu objetivo não só em apresentar informações comerciais, mas também em informar sobre a localização de pessoas (em especial, amigos e familiares). Segundo Colin Johnson, editor especialista da revista eletrônica EETimes (JOHNSON, 2012) “[...] este tipo de serviço é considerado como uma das dez tecnologias que dominará o mercado nos próximos dez anos [...]”. Nas Figuras 1(b), 1(c), 1(d) e 1(e) são apresentadas as telas de interface de cada um dos produtos descritos anteriormente.

Ao contrário dos sistemas de informação tradicionais (também chamados de sistemas *broadcast* como TV, rádio e jornal), em que uma das principais preocupações é transmitir a maior quantidade de informação num intervalo de tempo (ou espaço, no caso dos jornais), os sistemas LBA procuram transmitir informação ao público-alvo no momento e no lugar em que gere maior eficiência socioeconômica. Isto marca uma grande diferença entre os dois tipos de sistema, apesar de muitas das técnicas dos sistemas *broadcast* serem reutilizadas pelos sistemas de marketing alternativos. Desta forma, uma das tarefas

mais importantes dos sistemas LBA é preencher a lacuna existente entre a arte de criar informação para os consumidores e a tarefa de levar informação em momentos e lugares estratégicos do ponto de vista comercial (PRASAD, 2007).

Pelo seu crescimento e pelo modelo de transporte urbano empregado, Florianópolis tem na mobilidade um dos seus maiores desafios no futuro imediato (FUNDAÇÃO F, 2010). O governo e a iniciativa privada estão conscientes da necessidade de sistemas de transporte urbanos coletivos mais inteligentes, econômicos e eficazes com o objetivo de atrair mais passageiros e reduzir o número de automóveis nas vias públicas. Cria-se, então, a necessidade do desenvolvimento de ferramentas que viabilizem o cenário descrito anteriormente e que ajudem na materialização dos sistemas de transporte urbanos coletivos inteligentes, os quais se mostram como a grande tendência não só em nível nacional, mas também internacional (OIHANA; LINAZA, 2010).

Nesse sentido, a proposta deste trabalho é o desenvolvimento de uma plataforma para LBA, cujo diagrama geral é mostrado na Figura 2. Trata-se de uma plataforma que utiliza o sistema de posicionamento global (GPS) para determinar a localização de um ônibus do sistema de transporte urbano e, com base nesse dado, exibir, dentro desse veículo, informações da região por onde o mesmo está transitando. Além disso, dados relacionados com a viagem (como a rota, a velocidade, etc.) e vídeos dos locais comerciais da região de interesse são exibidos através da plataforma com o objetivo de fornecer informações mais contextualizadas aos passageiros. As informações da rota e dos vídeos que serão exibidos são carregadas no sistema, sob demanda dos clientes do sistema (agentes de publicidade e agentes comerciais em geral), através de uma conexão à Internet.



a.



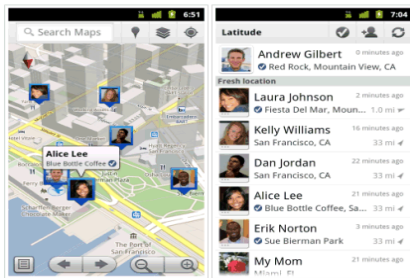
b.



c.



d.



e.

Figura 1. Exemplos de LBS. a. City24x7, b. YELP, c. Find My Friends, d. NEER, e. Google Latitude.

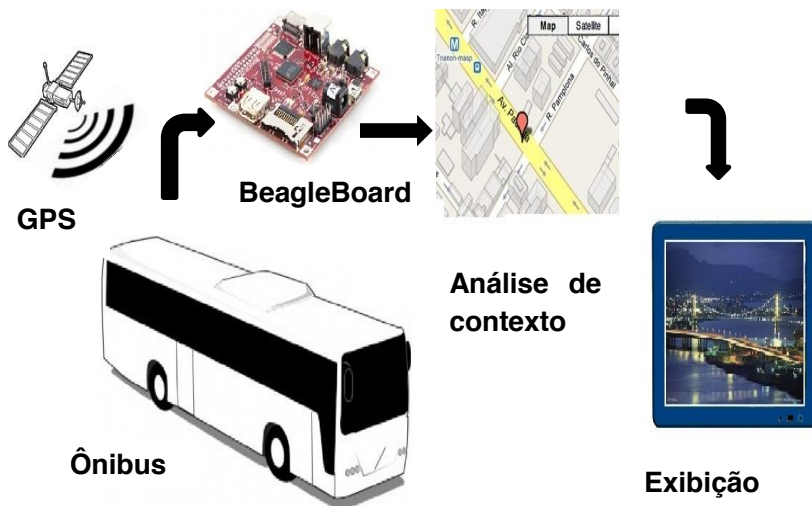


Figura 2. Diagrama geral da aplicação desenvolvida neste trabalho.

1.1 Objetivos do Trabalho

Este trabalho propõe projetar e construir um protótipo de plataforma de publicidade baseada na posição (do inglês *Location Based Advertising, LBA*) para a divulgação dinâmica de conteúdo audiovisual em sistemas de transporte coletivo urbanos.

Como objetivos específicos, este trabalho contempla os seguintes pontos:

- Projetar e desenvolver um plataforma para LBS com características de baixo custo e baixo consumo de energia, utilizando a plataforma de *hardware BeagleBoard* como base de desenvolvimento do equipamento.

- Desenvolver *softwares* específicos que permitam a interação entre a plataforma de *hardware BeagleBoard* e os módulos a serem utilizados para a localização e atualização do conteúdo no equipamento.
- Avaliar e testar o funcionamento da plataforma desenvolvida utilizando dados reais do sistema de transporte urbano da cidade de Florianópolis.

1.2 Estrutura da Monografia

Esta monografia está organizada como segue. O capítulo 2 apresenta conceitos teóricos fundamentais dos LBS/LBA. O Capítulo 3 apresenta os detalhes técnicos dos módulos de *hardware* utilizados, assim como também descreve detalhadamente os módulos de *software* que foram desenvolvidos e/ou adaptados para conseguir os objetivos deste trabalho. Ao final desse mesmo capítulo 3 serão apresentados os resultados dos testes preliminares feitos com a plataforma. Por fim, comentários e conclusões finais deste trabalho são apresentados no Capítulo 4.

2. FUNDAMENTOS TEÓRICOS

As abordagens utilizadas pelos LBS na geração de informação estão intimamente relacionadas ao modo em que o usuário pretende aproveitar o sistema. Modelos de contexto, de localização e de usuário também fazem parte das definições importantes no projeto de LBS. Neste capítulo são descritos cada um desses conceitos.

2.1 Modo de Uso e Modelo de Usuário

Perguntas como onde está o usuário em um dado momento, qual é o local de interesse mais próximo, onde está e como o usuário pode chegar até dito local de interesse ajudam na seleção do **modo de uso** no projeto de LBS (GROSSMANN, 2009).

Existem dois modelos predominantes na escolha da forma de uso dos LBS: modo de uso *push* e modo de uso *pull*, os quais resolvem de maneira diferente as questões expostas anteriormente. No modelo *push*, as informações geradas pelos sistemas LBS são transmitidas e apresentadas ao usuário dos dispositivos de informação sem participação ativa dele, isto é, as informações são transmitidas e apresentadas sem serem requeridas explicitamente pelo usuário. Já no modelo *pull*, as informações geradas pelos sistemas LBS são requisitadas pelos usuários. Os sistemas *pull* se caracterizam por incorporar filtros de busca que ajudam os usuários a definir melhor as suas necessidades comerciais, obtendo assim informações mais precisas e personalizadas (WIKIPEDIA, 2012; UNNI, 2010).

Tendo em conta que o sistema desenvolvido neste projeto é um sistema móvel que não tem interação direta com os passageiros do sistema de transporte na hora de exibir informações, o seu modo de uso em campo se enquadraria no modelo *push*. Por outro lado, agentes de publicidade e cidadãos (pessoas físicas e

jurídicas) em geral, que desejem utilizar o produto para apresentar informações dos seus locais comerciais ou negócios (daqui por diante chamados de clientes do sistema), terão interação com o sistema no modo de uso *pull*. Assim, o cliente, antes de transferir as informações do local comercial de interesse, pode requisitar informações da rota onde dita informação será apresentada.

Como na seleção do modo de uso, existem algumas perguntas que ajudam na escolha do **modelo de usuário** a ser utilizado pelo LBS. Respostas a questões como: quem são os usuários, quais suas necessidades e quando eles precisam dos serviços ajudam a definir para quem será projetado o LBS. Existem três modelos amplamente utilizados para o tipo de usuário dos sistemas LBS. O primeiro é o **usuário ambicioso** que é aquele que precisa de informação detalhada sobre todo o seu contexto geográfico (GROSSMANN, 2009). Exemplos deste tipo de usuários são os gerentes de frotas de carros fortes, onde o interesse não é só conhecer com clareza a posição atual dos veículos, mas também conhecer o que acontece ao seu redor. Outro tipo de usuário é o **usuário seletivo**, que precisa de informação detalhada sobre itens específicos (GROSSMANN, 2009). Exemplo deste tipo de usuários são os passageiros do sistema de transporte urbano, para os quais é interessante conhecer a maior quantidade de informação sobre o percurso do ônibus que irão usar (origem, destino, tempo estimado da viagem, estado do trânsito na região, etc.). Por último, há o **usuário atarefado**, que precisa de reduzida informação sobre todo o seu contexto geográfico (GROSSMANN, 2009). Usuários das redes sociais, para os quais é interessante conhecer (em um formato enxuto) não só informações geográficas sobre eles mesmos, mas sobre terceiros da mesma rede, são exemplos deste último tipo de usuários (GROSSMANN, 2009).

Considerando que a informação apresentada pela plataforma desenvolvida com este projeto está dirigida a um grupo específico de passageiros e que os locais que pretendem ser pautados tem que levar em conta regras de localização definidas pelas rotas do sistema de transporte, pode-se dizer que o modelo de usuário empregado no projeto é o modelo de usuário seletivo. Com esse modelo se ganha certa flexibilidade em relação à quantidade de informação apresentada pelo sistema, fazendo que esta seja dependente de fatores como a rota dos ônibus, os horários das rotas, a quantidade de locais de interesse definidos, entre outras condições.

2.2 Modo de Localização e modelo de contexto

A localização de um objeto ou de uma pessoa são tarefas fundamentais na operação de sistemas LBS. A escolha da forma em que estes dados são adquiridos impõe algumas necessidades e requisitos no software e no hardware a serem utilizados. Uma das condições que ajudam na escolha da forma em que a informação de localização é adquirida é o ambiente em que o sistema LBS vai ser utilizado, pois dependendo desse, será ou não conveniente eleger uma dentre diferentes técnicas e tecnologias. Caso se trate de um ambiente aberto (*outdoor*), os métodos baseados em satélite são os mais recomendados (GPS, *Wide Area Augmentation System*, GALILEO, etc). Já no caso de ambientes fechados (*indoor*), as tecnologias são mais variadas, indo desde sinalização por infravermelho, passando por rádio-frequência (RFID) e ultrassom e chegando até às visuais (crachás). Mais, recentemente, as tecnologias baseadas em redes também estão ganhando espaço neste tipo de aplicações. A infraestrutura das redes GSM/UMTS, WiFi, ZigBee, por exemplo, estão sendo aproveitadas para executar algoritmos de triangulação e RSSI (*Received Signal Strength Indication*) para

determinar a posição de um objeto em um determinado espaço (WIKIPEDIA, 2012).

Tendo definido o tipo de tecnologia que será utilizada na aquisição dos dados de localização, é necessário determinar a forma em que essa informação é representada. Existem diversas formas de representar a localização de um objeto num determinado espaço. A representação mais utilizada nos LBS é a **representação geométrica**, onde a posição do objeto de interesse é dada por sua latitude e longitude. Outra forma muito comum de definir a localização de um item de interesse é utilizando a **representação simbólica** onde alguns lugares bem conhecidos e representativos da cidade servem como pontos de referência (campus da universidade, shopping, etc). Por outro lado, a **representação absoluta** é a utilizada legalmente para descrever a posição de um local. Nessa forma de localização, são utilizadas as diretrizes definidas pelas autoridades governamentais para a organização territorial das cidades (endereço e CEP). Em contraste com as formas anteriores de localização, a **representação relativa** emite a posição do local de interesse como a distância entre tal local e um ponto geográfico fixo ou móvel, utilizando termos como por exemplo 100 metros ao norte, três quarteirões à esquerda, etc. (GRPSSMANN, 2009).

Como consequência do conhecimento da informação de localização, os sistemas LBS utilizam **análise de contexto** para gerar informação relevante e de acordo com as necessidades do usuário. O contexto é definido como qualquer informação que possa ser utilizada para caracterizar ou definir a situação de uma pessoa ou objeto (YIN, 2003). O contexto não só ajuda na identificação geográfica do usuário do LBS, mas também na identificação das pessoas e objetos que estão próximas, como por exemplo as condições climáticas do local. O contexto é influenciado pelo tipo de usuário, o ambiente social e a tarefa

para a qual o usuário precisa da informação fornecida. Fatores como a infraestrutura e a condição climática auxiliam, também, a definir a dinâmica das informações geradas pelos LBS (GROSSMANN, 2009).

Dado que o sistema desenvolvido neste projeto será utilizado em ambiente *outdoor*, a tecnologia escolhida para a tarefa de localização é o GPS. Um dos pontos a considerar no uso desse tipo de tecnologia é que a precisão com que a localização é definida está dada pelas características metrológicas do sensor (repetitividade, incerteza da medição, erro máximo, etc.). A representação geométrica da posição dos objetos de interesse, utilizada também neste projeto, traz o benefício de que o sistema possa ser utilizado em diversas cidades.

2.3 Classificação dos LBS

Geralmente, os LBS podem ser classificados em três grandes categorias: LBS-telemática, LBS-estáticos e LBS-móveis (YIN, 2003).

Os LBS classificados na categoria de **telemática** unem tecnologias de comunicação sem fio, sistemas de monitoração veicular e sistemas de localização de objetos. Dentre os exemplos mais representativos desta categoria estão: a localização automática de veículos (AVL), o rastreamento de fretes e de frotas, a navegação e assistência técnica online. Já nos **LBS-estáticos**, o serviço é fornecido considerando a posição especificada pelo usuário (que pode ser diferente de sua posição real). Assim, tais sistemas não exigem tecnologias de posicionamento. Exemplo de LBS-estáticos são os sistemas que geram anúncios publicitários em algumas páginas da internet a partir da informação referente ao país ou região informada pelo navegador (*browser*). Outros exemplos ainda são o YELP, apresentado no primeiro capítulo, e o serviço oferecido pelo

Google Maps para definir rotas de viagens só considerando os pontos de origem e destino. Por último, os **LBS-móveis** integram a categoria de maior crescimento. Aqui se enquadram os LBS que geram, transmitem e apresentam informações para dispositivos móveis tipo *smartphones*, *tablets* e PDAs. A diminuição dos preços dos dispositivos móveis assim como o uso massivo das redes sociais tem gerado uma grande demanda por esse tipo de serviços, forçando a criação de técnicas e algoritmos muito mais sofisticados para tratar o volume de informação desejado pelos usuários (PRASAD, 2007; WIKIPEDIA 2012; YIN, 2003).

2.4 Componentes dos LBS

A maioria dos sistemas baseados na localização recebe como entrada dados do GPS, retornando como saída informações comerciais da região onde o dispositivo de informação se encontra. Seguindo essa característica, a arquitetura do LBS desenvolvido neste projeto pode ser dividida em quatro subsistemas: (i) sistema de posicionamento, (ii) sistema de localização e de análise de contexto, (iii) sistema de telecomunicação e (iv) sistema de exibição (YIN, 2003). Os três primeiros são responsáveis pelo processamento de dados e preparação da informação a ser exibida. Na Figura 3, apresenta-se em diagrama de blocos a arquitetura do sistema de LBS desenvolvido.

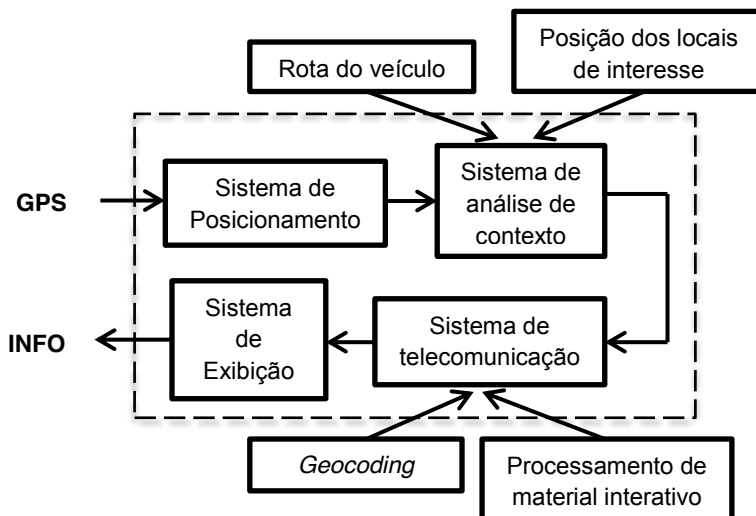


Figura 3. Diagrama de blocos da arquitetura dos LBS/LBAD.

A tarefa principal do **subsistema de posicionamento** é responder à pergunta: “Onde está o sistema neste momento?”. Contar com uma boa precisão é uma das características mais desejadas para este subsistema, em especial nas aplicações para LBS móveis onde o cálculo de rotas e a verificação de trajetos podem ser considerados como requisitos.

A tarefa principal do subsistema de **localização e de análise de contexto** é responder à pergunta: “O móvel está em uma região comercial de interesse?” ou “Que estabelecimentos comerciais e turísticos de interesse estão próximos da posição atual do móvel?”. Este subsistema pode ser considerado o coração do LBS, pois se encarrega de relacionar a posição atual do móvel com informações do entorno geográfico. Tais informações podem ser adquiridas de várias maneiras: neste trabalho, considera-se o emprego de um banco de dados estático e interno ao sistema de

arquivos da plataforma utilizada. Neste sentido, o subsistema de localização e análise de contexto realiza uma comparação direta dos dados do GPS com os dados do banco de dados local. Uma vez que o móvel se encontra em uma região de interesse, informações relacionadas aos locais comerciais da região são extraídas do banco de dados local e disponibilizadas para o **sistema de exibição**.

Já o subsistema de **telecomunicação** é o encarregado de fornecer o meio para que o LBS/LBA tenha acesso a informações que não se encontram no seu sistema de arquivos como, por exemplo, informações de endereçamento para descrever, em representação absoluta, a localização dos estabelecimentos de interesse. O sistema de telecomunicação permite ao LBS ser atualizado sob demanda (*on-line*), possibilitando aos clientes transferir informações de seus locais comerciais a qualquer momento do dia. Neste trabalho considerou-se utilizar um modem 3G para efetuar as tarefas de telecomunicações.

Finalmente, o sistema de **exibição** é o subsistema encarregado de fazer chegar as informações ao público-alvo. Sem ele, não seria possível atingir o objetivo máximo dos LBS: sensibilizar o consumidor para gerar maior participação deste na compra e venda de produtos. O subsistema de exibição pode ser local (fazer parte do mesmo LBS) ou externo (exibir as informações geradas num dispositivo remoto tipo celular ou *tablet*). Neste trabalho utiliza-se um sistema de exibição local, que consiste em um monitor de vídeo com interface HDMI ligado na plataforma de hardware BeagleBoard.

3. DESENVOLVIMENTO

No desenvolvimento de produtos eletrônicos, muitas vezes o *software* e o *hardware* não podem ser desenvolvidos separadamente. Neste projeto, o *software* e o *hardware* da plataforma de LBS foram desenvolvidos de maneira simultânea: a cada módulo de *hardware* selecionado, um módulo de *software* (mesmo que somente para teste do *hardware*) foi desenvolvido ou adaptado. Embora o desenvolvimento do *hardware* e do *software* tenha sido feito em conjunto, as explicações que seguem neste capítulo serão feitas por separado.

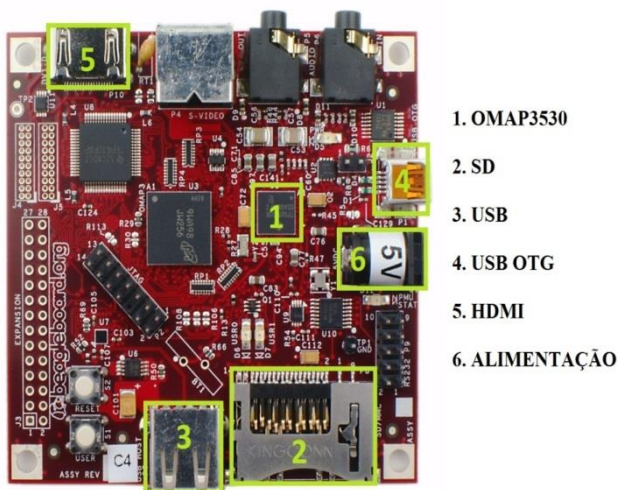
3.1 Hardware

No desenvolvimento de produtos eletrônicos e sistemas embarcados, a escolha do hardware adequado é uma etapa muito importante do projeto. Sem uma correta definição do *hardware*, o projeto de *software* ou *firmware* encontrará limitações que possivelmente tornem inviável o produto. Algumas propriedades dos LBS impõem certas características que são desejadas no *hardware* da plataforma desenvolvida, por exemplo: dada a quantidade de tarefas que tem que ser realizadas pelo LBS, é conveniente que o *hardware* utilizado tenha a capacidade de processar dados rapidamente e ser multitarefa. Com a restrição imposta pelo uso de um sistema de exibição local (Seção 2.4), obriga-se que o *hardware* tenha suporte para conexões de vídeo (TV e monitores). Além disso, o *hardware* desenvolvido deve proporcionar suporte para comunicação com outros módulos de *hardware* externos (GPS, modems e até outros computadores) via interfaces padrão como USB, RS232, ethernet, etc. Adicionalmente, uma das características desejadas é que a capacidade de armazenamento da plataforma possa ser alterada facilmente,

permitindo aumentá-la ou diminuí-la sem afetar o funcionamento geral do equipamento.

Nesta seção serão apresentados os módulos de *hardware* escolhidos no desenvolvimento deste projeto. É importante mencionar que este projeto além de servir como protótipo também é uma prova de conceito e, por isso, cada módulo foi escolhido separadamente com o objetivo de testar diferentes alternativas sem requerer alterações drásticas na etapa de desenvolvimento de *software*.

Outro destaque é o fato da escolha do *hardware* estar determinada pela disponibilidade imediata dos módulos de *hardware* para o desenvolvimento do projeto. Com isso a quantidade de módulos a serem analisados para determinar a sua pertinência técnica no projeto se reduz consideravelmente, tendo casos onde o conjunto de análise apresenta um elemento só.



1. OMAP3530
2. SD
3. USB
4. USB OTG
5. HDMI
6. ALIMENTAÇÃO

Figura 4. Beagle Board e interfaces para conexão de periféricos utilizados no projeto.

3.1.1 BeagleBoard

BeagleBoard (BB) é uma plataforma de *hardware* projetada com o objetivo de prover um computador de dimensões físicas reduzidas (uma placa única e sem cooler), de baixo consumo de energia (máximo 2 W com alimentação de 5 V), baixo custo, alto poder de processamento e aberta (*open hardware*) (PARETAS, 2009). Por possuir tais características, a BB foi escolhida como plataforma de desenvolvimento deste projeto.

A BB utilizada neste projeto é a versão C3, equipada com um microcontrolador de referência OMAP3530, fabricado pela Texas Instruments, baseado na arquitetura ARM Cortex A8 e operando a uma frequência de 600 MHz. Uma das vantagens de optar pela Beagle Board neste projeto é a disponibilidade de um conjunto

de ferramentas de *software* de código aberto (*open source*), reduzindo o custo do desenvolvimento em comparação com outras plataformas com compiladores e ferramentas de programação proprietárias. Contar com um processador como o OMAP3530 permite executar um sistema operacional que gerencie de forma eficiente todos os recursos desta plataforma. Na Figura 4, é apresentada uma foto da plataforma, destacando as interfaces utilizadas neste projeto e no Quadro 1 são apresentadas algumas características importantes da BB (BEAGLEBOARD, 2009). No Apêndice A, a sequência de inicialização executada pela plataforma para carregar o sistema operacional é apresentada. Essa sequência permite verificar também o correto funcionamento da plataforma.

Quadro 1 - Especificações do *hardware* BeagleBoard.

Especificação	Descrição
Processador	OMAP3530
Memória	2Gb NAND, 2Gb SDRAM
Interfaces	SD
	DVI-D, SVIDEO
	USB HOST, USB 2.0 OTG
Alimentação	Externa: 5 V -2 A
	USB: 5 V – 350 mA
Tamanho	78,74 x 76,2 mm
Preço	US\$ 150,00

3.1.2 GPS

Para adquirir a posição atual do móvel foram testados dois módulos de hardware: o receptor de GPS C2626 da Trimble e o receptor de GPS BU-353 da Global Sat. Conforme foi explicado anteriormente, a escolha destes dois módulos para avaliação é motivada mais pela sua disponibilidade do que por fundamentos técnicos. Mesmo assim, ainda é importante mencionar as características técnicas de cada um para tomar uma decisão sobre a utilização de um deles no projeto.

O módulo C2626, embora tenha boa precisão, boa sensibilidade e seja adequado para esta aplicação por suas especificações técnicas para medição, apresenta a dificuldade de exigir um conector fabricado pela empresa Samtech, muito caro (para a especificação de baixo custo deste projeto) e muito específico (não é padrão). Isso obrigaria ao projeto de uma placa de circuito impresso (PCI) só para a sua utilização. Além da dificuldade de conexão, o módulo C2626 não tem um *case* adequado para suportar as condições ambientais e mecânicas dos locais onde será instalado (exterior ou interior dos ônibus), exigindo assim, a construção de um gabinete especial que garanta a conservação do módulo nas condições de intempérie dos ambientes *outdoors*. Embora isso não seja uma preocupação neste projeto (pois o objetivo principal é obter um protótipo e uma prova de conceito de sistemas LBS), o ponto da estrutura mecânica do componente e a sua interface com o resto do sistema foram determinantes na hora de decidir por não utilizar este módulo, mesmo que testes preliminares tenham sido efetuados.

Na figura 5 é apresentada uma foto do módulo C2626 e, no quadro 2, suas características elétricas e dimensionais (TRIMBLE, 2010).



Figura 5. Módulo GPS C2626.

Quadro 2 - Características do módulo GPS C2626

CARACTERÍSTICA	DESCRIÇÃO
Precisão	Posição: <5 m
	Altitude: < 3 m
Sensibilidade	Aquisição: -146 dBm
	Rastreamento: -160 dBm
Tempo de aquisição	Primeira amostra: <38 s
	Demais amostras: 2 s
Dimensões mecânicas	26 x 26 x 6 mm
Preço	US\$ 65,00

O módulo BU-353 foi selecionado como sistema de posicionamento deste projeto. Esse módulo tem interface USB, que permite comunicação direta com a BB e estrutura mecânica a prova de água e com fixação por meio de ímã. Isso faz dele um *hardware* suficientemente robusto que viabiliza a sua instalação

no exterior de veículos do sistema de transporte público. Além disso, ao contrário do C2626, o BU-353 não precisa nem de fonte nem antena externas para adquirir os sinais dos satélites. Na Figura 6 é apresentada uma imagem do módulo e, no Quadro 3, são apresentadas as suas características elétricas e dimensionais (GLOBALSAT, 2012).



Figura 6 Módulo GPS BU-353.

3.1.3 Modem 3G

Como na escolha do sistema de posicionamento, para o sistema de telecomunicações foram testados dois módulos de *hardware* diferentes (os únicos disponíveis para o desenvolvimento do projeto). O primeiro se trata do modem celular de referência G24 fabricado pela Motorola e mantido no mercado atualmente pela Telit. Este modem tem interface de comunicação serial, para transmitir e receber comandos AT de configuração e controle do mesmo. Com este módulo também se tem o problema da exigência de conector especial, fabricado pela empresa Molex, que obriga à fabricação de uma PCI para sua utilização no projeto.

Quadro 3 - Características do módulo GPS BU-353.

CARACTERÍSTICA	DESCRIÇÃO
Precisão	Posição: <5 m
	Altitude: < 5 m
Sensibilidade	Aquisição: -159 dBm
	Rastreamento: -159 dBm
Tempo de aquisição	Primeira amostra: <45 s
	Demais amostras: 8 s
Dimensões físicas	53 (diâmetro) x 19,2 mm
Preço	US\$ 24,00

Pela exigência do projeto e fabricação de uma PCI, pela necessidade de uma fonte de energia exclusiva para sua alimentação e pelo fato de utilizar uma antena externa para sintonização de sinais GSM, o G24 é uma escolha pouco atraente para o desenvolvimento deste projeto.

Dadas essas condições, foi necessário utilizar uma alternativa muito mais simples: um modem celular padrão, fornece as características elétricas e mecânicas que são necessárias para o protótipo desenvolvido neste trabalho. O modem celular utilizado é o modelo E1750, fabricado pela empresa Huawei. Os Quadros 4 e 5 apresentam as especificações dos modems G24 e E1750, respectivamente, enquanto as Figuras 7 e 8 apresentam fotos de cada um deles, respectivamente (TELIT, 2001; HUAWEI C, 2010).



Figura 7 Módulo de comunicação celular G24.



Figura 8 Modem de comunicação celular E1750 com interface USB.

Quadro 4 - Características do módulo GSM G24.

Característica	Descrição
Bandas de operação	Quad-Band (850/900/1800/1900 MHz)
Tensão de operação	3,3 – 4,2 V
Potência	2W@850/900 MHz 1W@1800/1900 MHz
Sensibilidade	-106 dBm
Dimensões físicas	24,4 x 45,2 x 5,8 mm
Preço	US\$ 100,00

3.1.4 Monitor

Não existem definições formais das características ideais do sistema de exibição para este projeto. O objetivo é procurar um meio em que os passageiros dos ônibus do sistema de transporte possam apreciar o conteúdo apresentado pela plataforma sem esforço visual e com clareza em qualquer um dos assentos do ônibus. Aproveitando a interface *High Definition Multimedia Interface* (HDMI) da BB, neste projeto foi utilizado um monitor de vinte e quatro polegadas com entrada HDMI, para simular o sistema de exibição do LBS.

Quadro 5 - Características do módulo E1750.

Característica	Descrição
Bandas de operação	GSM,GPRS,EDGE,2G (850/900/1800/1900 MHz) HSDPA,HSUPA,UMTS,3G (2100 MHz)
Tensão de operação	5,0 V
Potência	2,5 W (máximo)
Sensibilidade	-106 dBm
Dimensões físicas	71,4 x 25,5 x 12,5 mm
Preço	US\$ 30,00

3.1.5 Hub USB

A plataforma de *hardware* Beagle Board utilizada no projeto tem só uma interface USB disponível com capacidade máxima de 500 mA de corrente, insuficiente para alimentar todos os dispositivos requeridos neste projeto. Assim, se faz necessário um hub USB para gerenciar as conexões com o *hardware* externo descrito neste documento. O hub USB utilizado neste projeto possui sete (7) portas independentes e fonte externa capaz de fornecer até 2 A, suficiente para energizar todos os dispositivos utilizados.

3.1.6 Resultado da etapa de desenvolvimento de hardware

No hardware desenvolvido neste projeto destaca-se a arquitetura modular empregada que permite alterar os módulos de *hardware* individualmente sem precisar alterar o resto da plataforma. Com a escolha dos módulos de *hardware* apresentados nas últimas seções, procurou-se atender às especificações de baixo consumo de energia e baixo custo, estabelecidas como objetivos específicos deste trabalho. Além disso, também se procurou dar confiabilidade ao sistema partindo das características e do desempenho teórico de cada um dos componentes selecionados. Na Figura 9 é apresentado um gráfico das interconexões dos módulos que fazem parte do projeto. Note-se o papel central da BB já que é o módulo encarregado do processamento de dados, da gestão de tarefas e de atuar como integrador de tecnologias.

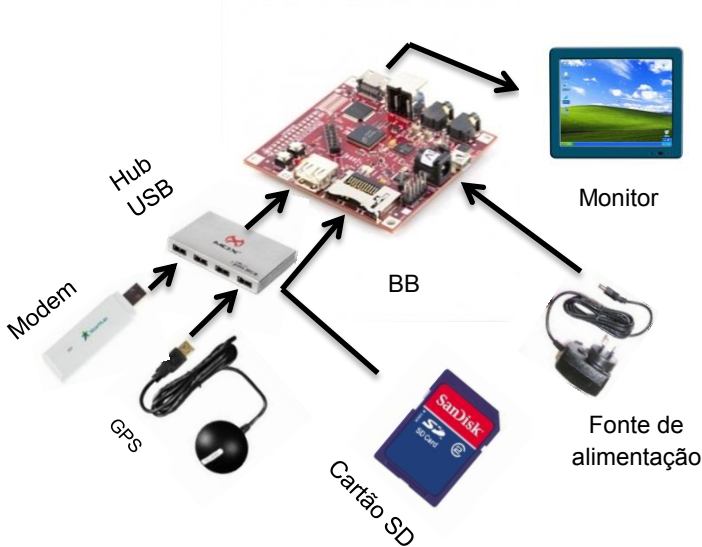


Figura 9 Diagrama de conexões entre a BeagleBoard e demais módulos de hardware no sistema de teste.

3.2 Software

Dependendo do nível de desenvolvimento, a criação de software pode ser vista como o processo de dar vida ao *hardware*. Sem o software, o hardware só seria um conjunto de peças sem um comportamento e sem objetivos.

Seguindo a mesma arquitetura modular utilizada no desenvolvimento do *hardware*, o desenvolvimento do *software* consistiu na programação de códigos fontes independentes cujo propósito é o controle de cada um dos módulos de *hardware* do LBS. Os códigos gerados foram integrados utilizando o modelo de programação *multithreading*¹ que permite o compartilhamento dos recursos informáticos da BB e a execução de cada uma das tarefas do LBS de forma autônoma.

Ao final se obteve um tipo de biblioteca de *software* que permite configurar e interagir com os módulos de *hardware* do sistema, atendendo conseqüentemente um dos objetivos específicos do projeto. Cabe salientar que em alguns casos não foi desenvolvido código novo, mas simplesmente foram feitas personalizações e configurações de códigos já disponíveis (*open source*). Na Figura 10 se apresenta o diagrama de blocos do software desenvolvido.

3.2.1 Considerações iniciais para a aplicação do LBS no sistema de transporte da cidade de Florianópolis

Dado que este projeto busca aplicar os conceitos dos LBS no desenvolvimento de uma plataforma para o sistema de transporte urbano da cidade de Florianópolis, considerações a

¹ *Thread*: é uma forma de um processo computacional dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente. O suporte à thread é fornecido pelo próprio sistema operacional (SO).

respeito do entorno geográfico da cidade foram levadas em conta e assim o software estar o mais perto possível da realidade.

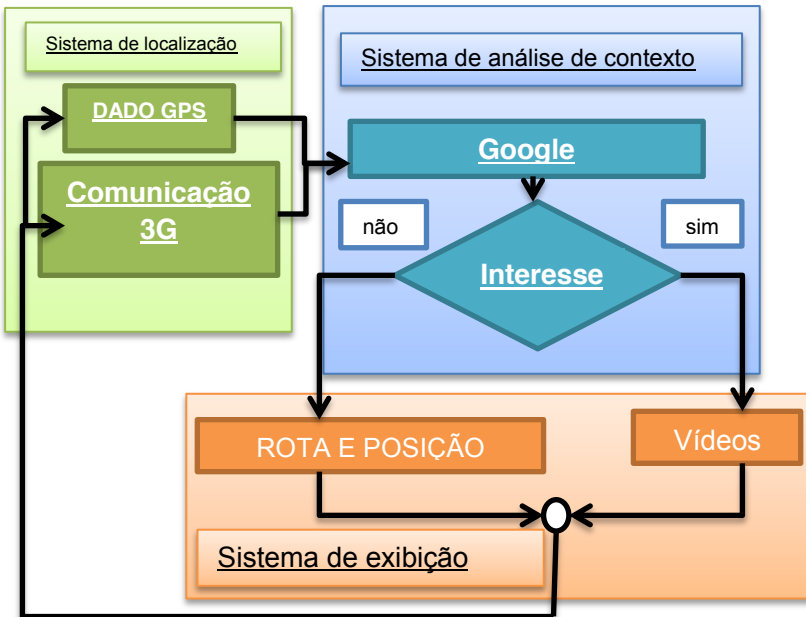


Figura 10. Diagrama de blocos do software desenvolvido.

Assim, a cidade foi dividida em quatro regiões de interesse comercial; cada região é identificada por uma série de pontos de referência, os quais também são utilizados como pontos de parada obrigatórios do veículo monitorado (FUNDAÇÃO F 2010). Como dado adicional se tem que a escolha tanto das regiões quanto dos pontos de referência foi baseada na localização dos terminais integrados de transporte da cidade de Florianópolis e dos lugares que por tradição e cultura são amplamente frequentados pelos seus habitantes; isso faz com que alguns bairros estejam fora da lista a seguir. Outro destaque é que estes pontos de referência só definem as regiões em que a cidade foi fracionada (máximo de oito pontos de referência por região) sem

afetar a quantidade de estabelecimentos comerciais que podem utilizar o sistema LBS para divulgar os seus serviços e produtos.

- **Região Centro:** Formada pelos bairros Trindade, Agronômica, Centro, Pantanal, Carvoeira e Saco dos Limões. Os pontos de referência escolhidos foram a Universidade Federal de Santa Catarina (UFSC), Terminal Integrado de Transporte da Trindade (TITRI), Casa D'Agronômica, Instituto Federal de Santa Catarina (IFSC), Terminal Integrado de Transporte do Saco dos Limões (TISAC), Avenida Beira Mar Norte, Ponte Hercílio Luz e Terminal Integrado de Transporte do Centro (TICEN).
- **Região Norte:** Formada pelos bairros Canasvieiras, Jurerê, Ingleses, Santo Antônio, Monte Verde e João Paulo. Os pontos de referência escolhidos foram o Terminal Integrado de Transporte de Canasvieiras (TICAN), Hotel Il Campanario, Terminal Integrado de Transporte de Santo Antônio (TISAN), Floripa Shopping e Parque Tecnológico ALFA.
- **Região Sul:** Formada pelos bairros Rio Tavares, Ribeirão da Ilha, Campeche e Pântano do Sul. Os pontos de referência escolhidos foram o Terminal Integrado de Transporte do Rio Tavares (TIRIO), Aeroporto Internacional Hercílio Luz, Estádio de Futebol da Ressacada, Lagoa do Peri e Avenida Pequeno Príncipe.
- **Região Leste:** Formada pelos bairros Santa Mônica, Itacorubi, Lagoa, Praia Mole e Barra da Lagoa. Os pontos de referência escolhidos foram o Shopping Iguatemi, Universidade Estadual de Santa Catarina (UDESC), Centrais Elétricas de Santa Catarina (CELESC), Terminal Integrado de Transporte da Lagoa (TILAG), Avenida das Rendeiras, Praia Mole e Barra da Lagoa.

Na Figura 11 é apresentado um mapa de Florianópolis onde são destacadas cada uma das regiões mencionadas assim como os terminais integrados de transporte da cidade (FUNDAÇÃO F, 2010).



Figura 11. Mapa de Florianópolis salientando bairros e terminais integrados de transporte.

3.2.2 Instalação e configuração do sistema

A Beagle Board (BB), módulo central do projeto, precisa de um *software* que consiga gerenciar de forma eficiente todas as suas interfaces. Como já comentado, é possível utilizar um

sistema operacional para efetuar esta tarefa. O sistema operacional escolhido para gerenciar a BB é o Linux, distribuição Ubuntu 11.10 (Oneiric-Ocelot). Para a sua instalação é preciso contar com um computador (chamado de *host*) com sistema operacional Linux e também com um cartão SD (de pelo menos 2GB) que serve como meio de armazenamento de dados na Beagle Board. Neste projeto, foi utilizada uma versão do Linux configurada para o microcontrolador OMAP3530, disponível na internet (Wiki Embedded Linux 2012). Uma vez baixado o pacote de instalação do Linux é possível utilizar alguns *scripts* que preparam o cartão SD para posterior gravação da imagem do OS. No Apêndice A são explicados os passos para instalar Ubuntu-Oneiric na BB.

Uma configuração importante consiste em conectar a Beagle Board numa rede de computadores. Isso é muito conveniente na etapa de desenvolvimento, já que permite a utilização de *cross-compilers*. Assim, o projetista pode desenvolver no *host* (que conta com muito mais poder de processamento do que a BB) e só deixar a BB como plataforma de teste e de execução. A versão da BB utilizada neste projeto (versão C3) não tem interface ethernet como acontece com revisões mais recentes do hardware (versão XM). Por isso, para esta tarefa, é utilizada a interface USB-OTG. Além disso, conectar a BB numa rede de computadores possibilita o acesso à internet, obtendo todas as funcionalidades de um computador moderno. No Apêndice B é apresentado um descritivo passo a passo para comunicar a BB com o *host* e estabelecer uma conexão entre a BB e internet utilizando como ponte (*bridge*) o *host* (CARLSON, 2012). Com a BB conectada à internet, é possível baixar e instalar alguns programas necessários e determinantes no desenvolvimento deste projeto. A seguir, são mencionados os programas instalados na BB:

- Atualização dos **repositórios de software do Ubuntu**.
- **X11**: Interface gráfica com browser *chromium*.
- **VLC**: Reprodutor de vídeo.
- **GPSD**: *Daemon* para comunicação com módulo GPS.
- **GPSD-clients**: Aplicativos exemplos para testar o GPSD.
- **PPPD**: *Daemon* para estabelecer o protocolo PPP entre o módulo celular 3G e a operadora de serviço celular (ISP).
- **PPPD-config**: Programa para configuração do PPPD.

Cada um desses programas tem um papel importante no funcionamento do projeto. O X11 é uma interface gráfica do Linux que se caracteriza por utilizar poucos recursos informáticos (LXDE, 2012). Junto com ele, é instalado automaticamente o browser *Chromium*, que é uma versão enxuta do browser Google Chrome. Com a interface gráfica e com o browser instalado na BB, é possível acessar o conteúdo dos mapas e exibir no monitor os vídeos e as informações de rotas dos ônibus onde o sistema for instalado. O conjunto VLC, browser e interface gráfica formam a parte do software que gerencia o sistema de exibição. O **VLC** é o programa que permite reproduzir os arquivos de vídeos dos locais de interesse definidos no sistema. Outras opções de reprodutores de vídeo foram avaliadas (por exemplo, o MPlayer) mas o VLC oferece grande quantidade de *codecs* e a possibilidade de reproduzir *streams* de vídeos utilizando o protocolo RTP. Essa propriedade pode ser aproveitada futuramente para reproduzir vídeos que não estejam armazenados no cartão SD da BB, mas em um computador remoto com acesso à internet (VIDEOLAN, 2012). Por sua parte, o **GPSD** é um programa que facilita a leitura de informações vindas de módulos GPS, abstraindo ao desenvolvedor os detalhes do hardware e possibilitando a troca de módulos de *hardware* sem precisar de grandes modificações no sistema em

geral (RAYMOND, 2012). Embora o GPSD seja um *daemon*², existem duas maneiras de interagir com ele: a primeira é por meio de uma interface de programação de aplicativos (API) e a segunda, utilizando uma comunicação *socket* TCP. Neste projeto foi decidido não utilizar a opção da API para evitar o processo de portá-la para a arquitetura do microcontrolador ARM; ao invés disso, foi aproveitado o fato do GPSD disponibilizar um servidor *socket* (*socket server* no endereço **127.0.0.1** e porta **2947**) que possibilita estabelecer uma comunicação via rede e receber todas as informações geradas pelo *hardware*. Como consequência dessa escolha, foi criado um cliente *socket* (*socket client*) que “lê” os “pacotes” do *socket server* e procura os dados de longitude, latitude e velocidade que posteriormente são utilizados pelo *software* de análise de contexto. Uma característica importante do GPSD é que os dados do *hardware* são publicados utilizando o formato JSON, facilitando e uniformizando o processo de decodificação da informação (*parser*) graças ao ordenamento das informações que esse formato dá ao texto transmitido. Como o GPSD, o **PPPD** é um programa *daemon* que se encarrega de abstrair os detalhes de *hardware* e facilitar o estabelecimento de conexões ponto a ponto (PPP) com a provedora do serviço de internet via rede celular (ISP) (HART, 2012). O PPPD executa todas as tarefas de discagem e negociação da conexão com a ISP (como adquirir um endereço IP válido, por exemplo) ocultado ao usuário detalhes relacionados com o *hardware* do modem utilizado. Para operar, o PPPD deve ser configurado com diversas informações como o nome da operadora e o servidor de nomes de domínio de internet (DNS) que será utilizado na comunicação; tal configuração pode ser feita utilizando o programa **PPP-config** que oferece uma interface gráfica que guia passo a passo o

² Daemon, acrônimo de Disk And Execution MONitor (Monitor de Execução e de Disco), é um programa de computador que roda de forma independente em background, ao invés de ser controlado diretamente por um usuário.

desenvolvedor na criação dos arquivos de configuração (GERONIMO 2008). No Apêndice C, apresentam-se os passos necessários para configurar o PPPD e para testar o GPSPD.

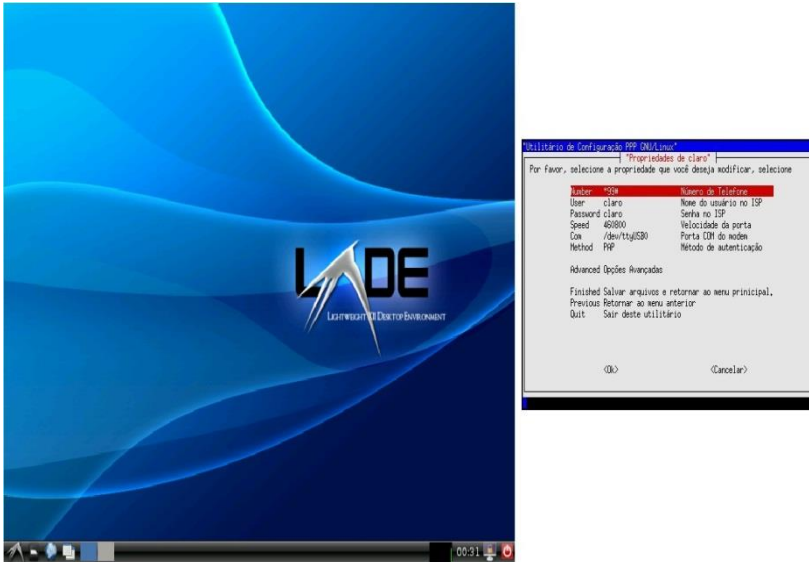


Figura 12. Janelas do X11 e PPP-Config.

3.2.1 Conceitos e definições do software gerenciador

Sendo coerente com a metodologia empregada no desenvolvimento deste projeto, foi preciso criar um software que administrasse e gerenciasse todas as interfaces e conexões físicas e ao mesmo tempo tomasse decisões rapidamente baseado nos conceitos fundamentais dos LBS descritos no Capítulo 1. Assim, usa-se uma arquitetura *multithread* para obter maior velocidade de processamento e organização durante o desenvolvimento, separando o software em várias partes, segundo suas funcionalidades.

Na Figura 13 apresenta-se o diagrama de fluxo do software desenvolvido. Cada uma das etapas do software é explanada a seguir.

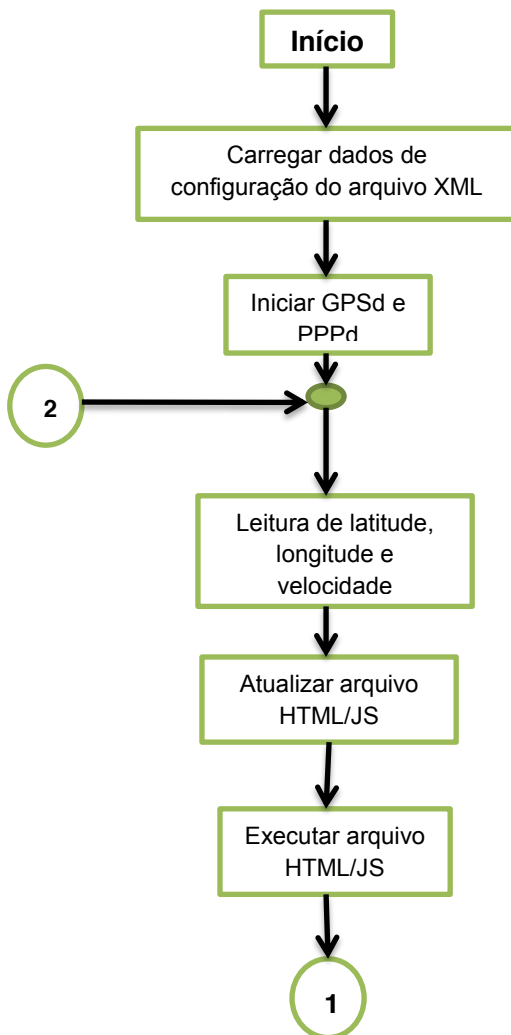


Figura 13 Diagrama de fluxo do software gerenciador

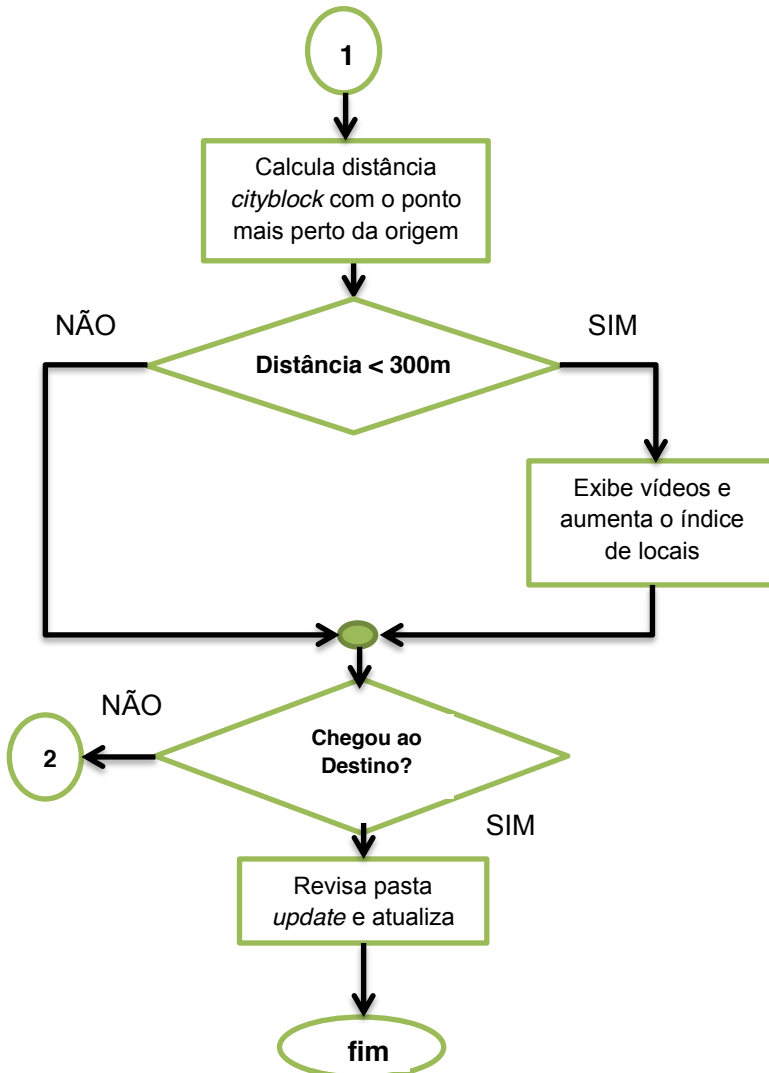


Figura 13 Diagrama de fluxo do software gerenciador (continuação).

3.2.2 Pré-Configuração do sistema e arquivo XML

Aproveitando que as rotas do sistema de transporte da cidade estão identificadas por números, neste projeto foram criados arquivos de configuração (cujo nome é o número da rota do veículo que será monitorado) com o objetivo de pré-configurar o sistema com informação conhecida. Nesses arquivos de configuração estão condensadas (em formato XML) algumas informações básicas da rota como o nome, origem, destino, pontos de passagem obrigatórias, horários de saída e informações dos locais comerciais que tem que ser publicada no trajeto (longitude e latitude do local, nome do vídeo, etc.). Como exemplo, neste texto será utilizada a rota TICEN-TITRI (que cobre grande parte da região centro) para apresentar de forma detalhada os resultados de cada uma das etapas do software. O Apêndice D apresenta o modelo do arquivo de configuração para tal rota. Cabe destacar que as informações da rota de exemplo foram obtidas da página web da prefeitura de Florianópolis. Esse arquivo XML é utilizado pelo software gerenciador para criar um arquivo com *tags* HTML/JavaScript que permitem ao sistema de visualização apresentar as informações de rota e de posição atual do veículo monitorado.

3.2.3 Determinação da localização, análise de contexto e exibição de informação

Seguindo os conceitos explanados na seção 3.2.1, o software gerenciador obtém os dados de longitude e latitude a partir de uma mensagem (*string*) recebida da comunicação *socket* TCP (**localhost:2947**) estabelecida com o **gpsd**. O formato da “*string*” alvo é o seguinte:

```
{"class":"TPV","tag":"MID2","time":"2010-04-30T11:48:20.10Z","ept":0.005,"lat":46.498204497,"lon":7.568061439,"alt":1327.689,"epx":15.319,"epy":17.054,"epv":124.484,"track":10.3797,"speed":0.091,"climb":-0.085,"eps":34.11,"mode":3}
```

Os dados de longitude, latitude e velocidade estão destacados em negrito. Com esses dados são efetuadas duas operações principais: a primeira operação é a de atualizar o arquivo *HTML/JavaScript* (criado na etapa de pré-configuração) para apresentar a posição atual do veículo monitorado em relação a sua rota e a segunda operação é a de comparar a posição atual do veículo com a posição dos locais comerciais a apresentar (e também com os próprios pontos de referência) para saber quão próximo está o móvel de uma região comercial de interesse (ver Figura 13). A leitura dos dados do GPS é feita a cada segundo, mas as tarefas de atualização e comparação mencionadas só são efetuadas a cada quinze segundos para diminuir a quantidade de processamento requerida por tais processos. Uma vez atualizado o arquivo *HTML/JavaScript*, o browser recarrega o arquivo e executa o código contido nele. Esse código possui diretivas para utilizar o serviço de decodificação geográfica (**geocoding**) e exibição de mapas do Google Maps (API). Quando o processo do browser termina, é exibido no monitor a rota que o ônibus irá percorrer e também um marcador (*marker*) indicando a posição atual do ônibus. Na Figura 14 é apresentada uma imagem do resultado da execução do código *HTML/JS* pelo browser, note-se em azul a rota que tem que ser percorrida e o *marker* indicando a posição atual do veículo.

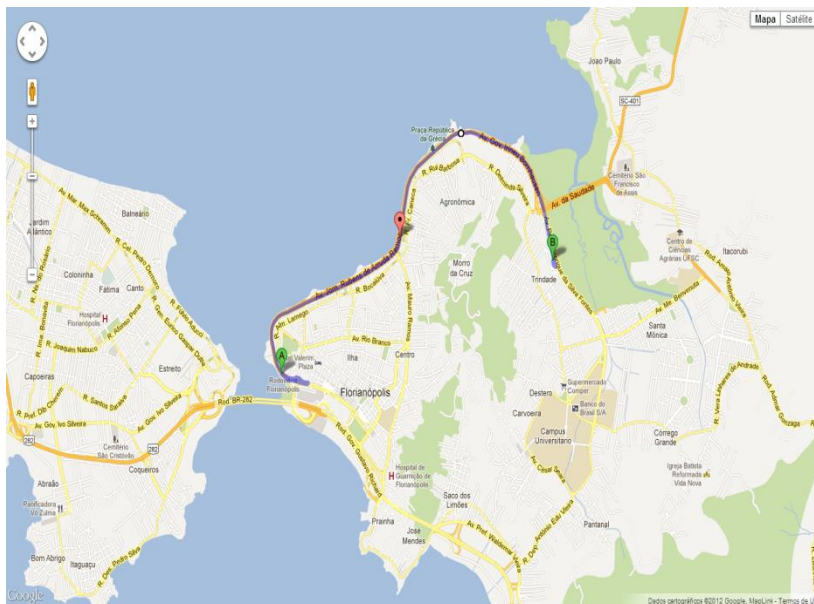


Figura 14. Execução do arquivo HTML/JS no *chromium* apresentando a rota e a posição de um ônibus TICEN-TITRI.

A API Google Maps utilizada neste projeto pode ser utilizada gratuitamente em sites de uso comercial que tenham menos do que 25 mil acessos por dia. Além disso, a API é escrita na popular linguagem para programação web *JavaScript (JS)* e possui uma excelente documentação, facilitando a utilização dos recursos desta tecnologia e reduzindo o tempo e custo de desenvolvimento. Embora existam alternativas para aproveitar este tipo de tecnologia (Yahoo! Maps API, Bing Maps Platform, plataforma de desenvolvimento MapQuest e OpenLayers), várias razões levaram à escolha do produto da Google: a primeira é a familiaridade que a grande maioria de pessoas tem com a sua interface, a segunda é a extensa e detalhada documentação encontrada em diversas fontes e, por último, a possibilidade de incorporar no sistema outros serviços da Google como o do **geocoding** que permite localizar um endereço (na sua

representação absoluta) partindo da posição dada em termos da latitude e da longitude (GOOGLE, 2012).

Com as informações do GPS e o serviço de *geocoding* da API Google Maps é possível criar um subsistema de análise de contexto como o explicado na Seção 2.4. O sistema de análise de contexto tem por objetivo determinar a importância da localização atual do dispositivo de informação perante objetos, locais ou pessoas que se encontram ao seu redor. Neste projeto foi aproveitado o fato de se ter prévio conhecimento da localização (em termos de longitude e latitude) dos locais comerciais a serem publicados pela plataforma para utilizar a métrica *city block* e determinar a proximidade do veículo a qualquer um desses pontos de interesse. A distância entre dois pontos utilizando a métrica *city block* é:

$$d_1(p, q) = ||p - q|| = \sum_{i=1}^n |p_i - q_i|$$

Onde \mathbf{p} é o par ordenado que corresponde à posição atual do veículo monitorado e \mathbf{q} é o par ordenado correspondente à posição do local comercial de interesse. Ambos, \mathbf{p} e \mathbf{q} , estão dados em termos de latitude e longitude (WIKIPEDIA, 2012).

Para este trabalho, considera-se que a posição atual do veículo monitorado se encontra dentro de uma região de interesse se a distância entre os dois pontos tem um valor menor do que 0,005 unidades, o que equivale a 300 m, aproximadamente. Com isso os vídeos e as demais informações dos locais de interesse são apresentados quando o veículo está numa região de 300 metros ao redor desse ponto.

Assim como existem outros tipos de métricas (por exemplo, métrica euclidiana) para avaliar a importância da posição atual do veículo, a empresa Google disponibiliza um serviço que pode ser

utilizado conjuntamente com a sua API Google Maps para determinar não só a distância em metros entre dois pontos, mas também o tempo estimado de viagem considerando as condições do trânsito da cidade. Este serviço não foi utilizado devido ao tempo de resposta elevado quando rodando na plataforma Beagle Board com uma conexão de internet de relativa baixa velocidade.

Sabendo se o veículo monitorado está (ou não) numa região de interesse, o software gerenciador seleciona um vídeo e o exibe executando o programa VLC. Os vídeos devem estar num formato que o programa VLC possa reconhecer e que se caracterizam por ter alta taxa de compressão como, por exemplo, o formato H264. Os vídeos são exibidos em tela cheia sobrepostos ao mapa onde se apresenta a informação de localização do veículo. Quando os vídeos terminam, o programa VLC é encerrado imediatamente fazendo como este tenha que ser carregado novamente a cada novo vídeo que precise ser exibido. Uma forma de impedir o processo de fechar e abrir o VLC constantemente é utilizar uma lista de reprodução (*playlist*) para apresentar sequencialmente todos os vídeos dos locais comerciais.

Caso exista a necessidade de carregar ou atualizar um vídeo (ou até mesmo atualizar o software gerenciador), a plataforma conta com um servidor FTP que possibilita a transferências de arquivos desde uma estação remota só com o conhecimento do endereço IP válido da plataforma (designado pela ISP). O servidor FTP da BB está endereçado a uma pasta de nome *update* no sistema de arquivos do OS. A cada chegada ao ponto de destino, o *software* gerenciador verifica se existe um novo vídeo (ou nova versão do aplicativo) para ser carregado ao sistema de arquivos. Caso exista algum, o software copia o vídeo na pasta correta (pasta do número da rota que o veículo está cobrindo) deixando a pasta *update* liberada para futuras atualizações. No início do dia (0 h)

também é feita uma checagem da pasta *update*. Caso algum arquivo tenha sido inserido em um período onde não é feito o procedimento de verificação de atualização. Contudo, a política de atualização de vídeo (e aplicativos) pode ser redefinida para obter uma resposta em tempo real e sob demanda do cliente.

3.2.3 Resultados da etapa de desenvolvimento de software

O resultado desta etapa permite o gerenciamento de cada um dos módulos de *hardware* do sistema e a execução de tarefas que interligam esses módulos, em especial as operações relacionadas com análise de contexto. A arquitetura empregada junto com os “*daemons*” (*gpsd* e *pppd*) auxiliares utilizados neste projeto, proporciona grande flexibilidade ao produto e possibilitam o intercâmbio de *hardware* sem precisar de quase nenhum retrabalho na parte do *software*. Por se tratar de um protótipo e uma prova de conceito, muitos recursos de software não foram utilizados e futuramente podem ser avaliados para dar melhor interação entre o sistema e o cliente (ou entre o sistema e os passageiros).

3.3 Resultados dos testes e simulações

Durante a etapa de desenvolvimento foram realizados alguns testes e procedimentos que ajudaram na visualização do comportamento do sistema em geral. Um dos procedimentos realizados foi a medição do consumo de corrente nas diferentes situações de uso do protótipo desenvolvido. No Quadro 6 se apresentam os resultados das medições efetuadas. Os cenários testados foram os seguintes:

- Beagle Board ligada e transmitindo sinal de vídeo via interface HDMI.

- Beagle Board ligada e só recebendo dados do GPS.
- Beagle Board ligada, recebendo dados do GPS e com conexão à internet via modem 3G.
- Beagle Board e demais periféricos ligados e executando o aplicativo do LBS.

Cabe destacar que o consumo de corrente do monitor não foi considerado nos cálculos por este não estar dimensionado desde o ponto de vista elétrico para este tipo de aplicações.

Quadro 6 - Consumo de corrente do sistema em diferentes cenários.

CENÁRIO	CONSUMO DE CORRENTE
Beagle Board + Vídeo	500 mA
Beagle Board + GPS	950 mA
Beagle Board + GPS+ 3G	1700 mA
LBS	2500 mA

Uma simulação do comportamento do sistema foi efetuada tomando dados reais (adquiridos do site da prefeitura de Florianópolis) da linha de ônibus que cobre a rota TICEN-TITRI. O teste consistiu em preencher uma variável interna do software gerenciador com coordenadas de latitude e longitude de 30 lugares pelos quais a rota exemplo passa (incluídos destino e origem). Esses dados, sequencialmente são “lidos” pelo sistema com intervalo de 15 segundos, forçando a execução das tarefas de análise de contexto, exibição de vídeos e atualização do arquivo HTML/JS. Junto com os 30 dados de latitude e longitude,

foi utilizado o arquivo de configuração em formato XML apresentado no Apêndice D. Nesse arquivo foram cadastrados três lugares de interesse comercial (Beira Mar Shopping, Koxixos bar, Angeloni Trindade) onde se pretende que o sistema exiba vídeos. O procedimento da simulação consistiu então em que a cada quinze segundos um par ordenado de latitude e longitude fosse lido pelo sistema representando a posição atual do veículo monitorado. A cada dado lido executava-se a operação de cálculo de distância entre o ponto de interesse mais próximo da origem (que ainda não tinha sido passado) e a posição simulada do veículo. Caso a distância entre esses dois pontos seja menor do que aproximadamente trezentos metros, o vídeo especificado pelo arquivo de configuração era exibido pela plataforma conseguindo assim comprovar o correto funcionamento do software gerenciador. Na Figura 14 se aprecia um momento específico da simulação com o qual se demonstra não só o funcionamento da plataforma desenvolvida, mas também a sua aplicação no sistema de transporte da cidade de Florianópolis.

4. CONCLUSÕES

O projeto apresentado nesta monografia conta com todos os recursos necessários para a prestação de serviços baseados na localização (LBS), uma área interdisciplinar que está ganhando grande interesse por parte da comunidade científica e comercial em geral. O “*know how*” adquirido durante o processo de construção deste projeto serve como plataforma para os novos desafios que sejam propostos nas áreas de sistemas embarcados, telecomunicações, ergonomia e design de produto; áreas de concentração que são tratadas ao longo do programa de pós-graduação do IFSC, no qual esta dissertação se enquadra.

Embora o projeto tenha uma componente de desenvolvimento de hardware, não foi necessário construir novos módulos dado que bastou com utilizar produtos já disponíveis no mercado para atingir os objetivos descritos no Capítulo 1. No entanto, alguns dispositivos apresentados e avaliados neste documento podem ser utilizados como base de projetos de *hardware* customizado (especialmente para o subsistema de comunicação 3G e para o subsistema de GPS) e criar assim um verdadeiro produto eletrônico original. Espera-se que no futuro próximo apareçam iniciativas de estudantes dos cursos técnicos, superior, e de pós-graduação do IFSC visando o desenvolvimento deste tipo de módulos que ajudem no complemento do projeto.

Outro dos pontos de interesse futuro é melhorar a eficiência e aparência da interface desenvolvida em HTML/JavaScript. Um ponto fraco do sistema é que, a cada novo ponto de localização adquirido, o código HTML/JS tem que ser atualizado. Isso exige que o browser recarregue novamente todo o arquivo HTML para o usuário possa apreciar as mudanças. Desse modo, o usuário percebe uma modificação muito grande na tela de tempos em tempos, criando desconforto visual e estético.

Por último, pode-se dizer que todos os objetivos definidos no início do projeto foram atingidos. Uma plataforma de publicidade baseada na posição (*LBA*) para a divulgação dinâmica de conteúdo publicitário em veículos do sistema de transporte de Florianópolis foi construída seguindo todos os conceitos e características básicas dos LBS comerciais. Além disso, o projeto foi desenvolvido utilizando como base a plataforma de hardware Beagle Board e utilizando módulos de hardware de baixo custo e de baixo consumo de energia quando comparados com outros produtos similares como foi demonstrado no Capítulo 3. A componente social do projeto também é atendida dado que o sistema, embora tenha sido projetado para fins comerciais e turísticos também pode ser utilizado para apresentar informações de interesse social como a localização de hospitais, escolas, parques, etc. Outro dos aportes sociais que se faz com o desenvolvimento deste projeto é a melhoria da infraestrutura do sistema de transporte da cidade de Florianópolis, considerando que atualmente os passageiros só têm informações dos horários de saída e chegada de cada uma das rotas obtida em cartazes em alguns lugares dos Terminais Integrados de Transporte.

Este projeto foi desenvolvido utilizando a ferramenta web GIT para manter um controle da evolução do projeto. Os códigos fontes, comentários e documentos estão disponíveis para ser acessados pela comunidade científica e para qualquer um que sinta interesse pelo tema³.

Nos próximos anos se espera que o crescimento da cidade fomente a inovação tecnológica ao redor do sistema público de transporte e com este projeto se dá um passo importante, desde um enfoque acadêmico, para alcançar a concretização dos sistemas de transporte inteligentes.

³ O repositório do projeto pode ser acessado no endereço https://github.com/hectordelahoz/lba_IFSC

REFERÊNCIAS

- [1] BAND. News, “**MUNDO: ORELHÕES DE NOVA YORK TÊM TELA TOUCHSCREEN E WI-FI,**” 2012. Disponível em: <http://bandnewstv.band.uol.com.br/noticias/conteudo.asp?ID=634633&tc=mundo-orelhões-de-novc-york-têm-telc-touchscreen-e-wi-fi>. Acesso em: 03 dez 2012.
- [2] COMPANY, Yelp, “**YELP HOME PAGE.**” Disponível em: <http://www.yelp.com/>. Acesso em: 03 dez 2012.
- [3] VAN GROVE, Jennifer, “**Top 5 Location-Based Services,**” 2010. Disponível em: <http://mashable.com/2010/10/27/best-location-based-services/>. Acesso em: 03 dez 2012.
- [4] COLIN. Johnson, “**EE Times 40th Anniversary: From 3-D chips to cognitive computing.**” Disponível em: <http://www.eetimes.com/electronicsnews/4400608/Slideshow-Emerging-Technologies-changing-the-world-in-next-ten-years?pageNumber=6>. Acesso em: 03 dez 2012.
- [5] PRASAD. Maneesh, “**Location based services**” Disponível em: <http://www.gisdevelopment.net/technology/lbs/techlbs003pf.htm>, Acesso em: 03 dez 2012.
- [6] FUNDAÇÃO Floripa amanhã. “**Floripa 2030, Agenda Estratégica de Desenvolvimento sustentável de Florianópolis na região**” 2 ed. Florianópolis. 2010. 60p .
- [7] OTAEGUI Oihana, LINAZA Maria Teresa. “**Position Paper on Intelligent Transport Systems**” Espanha: Vicomtech, 2010. 8p.
- [8] GROSSMAN, Uwe. “**Location Based Services,**” Disponível em: <http://www.piramk.fi/mobilemarit/presentations/GER1.pdf>. Acesso em: 03 dez 2012.

- [9] WIKIPEDIA, “**Location-based advertising.**” Disponível em: http://en.wikipedia.org/wiki/Location-based_advertising. Acesso em: 03 dez 2012.
- [10] RAMAPRASAD, Unni. HARMON, Robert, “**PERCEIVED EFFECTIVENESS OF PUSH VS . PULL MOBILE LOCATION- BASED ADVERTISING**” vol. 7, no. 2, pp. 28–40, 2010.
- [11] WIKIPEDIA, “**Geotargeting**”. Disponível em: <http://en.wikipedia.org/Geotargeting>. Acesso em: 03 dez 2012.
- [12] HONGYING. Yin, “**Location Based Service**” Disponível em: <http://www.comlab.hut.fi/opetus/333/slides2003/l38.pdf>. Acesso em: 03 dez 2012.
- [13] PARETAS Gerard. “**Introduction to BeagleBoard**” Disponível em: <http://www.teslabs.com/wpontent/uploads/2010/02/beagleboard.pdf>. Acesso em: 03 dez 2012.
- [14] ORGANITATION. BeagleBoard, “**BeagleBoard System Reference Manual Rev C4**”. Disponível em: http://beagleboard.org/static/BBSRM_latest.pdf. Acesso em: 03 dez 2012.
- [15] COMPANY. Trimble, “**CONDOR GPS MODULE**”. Disponível em: http://www.microcomputersystems.com/CONDORGPSModules_UG_2B.pdf. Acesso em: 03 dez 2012.
- [16] COMPANY, Globalsat. “**bu353 Datasheet.**”. Disponível em: http://www.usglobalsat.com/store/download/62/bu353_ds_ug.pdf. Acesso em: 03 dez 2012.
- [17] TELIT, Wireless Solution “**G24 Datasheet**”. Disponível em: http://www.m2mgsm.com/download/G24LITE/manuals/Telit_G24-LITE_Hardware_User_Guide_r0.pdf. Acesso em: 03 dez 2012.

- [18] COMPANY Huawei Technologies, “**E1750 Datasheet**”. Disponível em:http://www.mtn.co.za/Support/DeviceDownloads/Documents/Huawei%20E1750%2011_126_10_00_00%20Release%20Notes.pdf. Acesso em: 03 dez 2012.
- [19] GUIA Floripa, “**Sistema Integrado De Transporte**”. Disponível em:<http://www.guiafloripa.com.br/utilidades/onibus/sit.php3>. Acesso em: 03 dez 2012.
- [20] WIKI Embedded Linux, “**BeagleBoard Ubuntu**” Disponível em : <http://elinux.org/BeagleBoardUbuntu>. Acesso: 01 fev 2012.
- [21] CARLSON Jay. “**Beagle Online**”. Disponível em: http://code.google.com/p/creation-station/wiki/Beagle_Online. Acesso em: 01 Fev 2012.
- [22] LXDE, “**Lightweight X11 Desktop Environment**”. Disponível em: <http://lxde.org/>. Acesso em: 02 out 2012.
- [23] GOOGLE, “**The Chromium Projects**”. Disponível em: <http://www.chromium.org/>. Acesso: 02 out 2012.
- [24] VideoLAN, “**VLC video player**”. Disponível em: <http://www.videolan.org/vlc/>. Acesso em: 02 out 2012.
- [25] RAYMOND Eric, “**GPSD Client HOWTO**” Disponível em: <http://catb.org/gpsd/client-howto.html>. Acesso em: 02 out 2012.
- [26] R. Hart, “**Setting up the PPP connection manually**” Disponível em:<http://stommel.tamu.edu/~baum/linux/LDP/HOWTO/PPP-HOWTO-14.html>. Acesso em: 02 out 2012.
- [27] GARCIA. Geronimo, “**Conexão automática com a internet usando pppconfig**”. Disponível em: <http://www.vivaolinux.com.br/dica/Linux-conexao-automatica-com-a-internet-usando-pppconfig>. Acesso em: 02 out 2012.

- [28] GOOGLE, “**Google Maps JavaScript API v3**”. Disponível em: <https://developers.google.com/maps/documentation/javascript/tutorial>. Acesso em: 02 out 2012.
- [29] WIKIPEDIA, “**Taxicab geometry**”. Disponível em: http://en.wikipedia.org/wiki/Taxicab_geometry. Acesso em: 03 dez 2012.

APÊNDICE A - Instalação Ubuntu Linux 11.10 na Beagle Board rev. C3

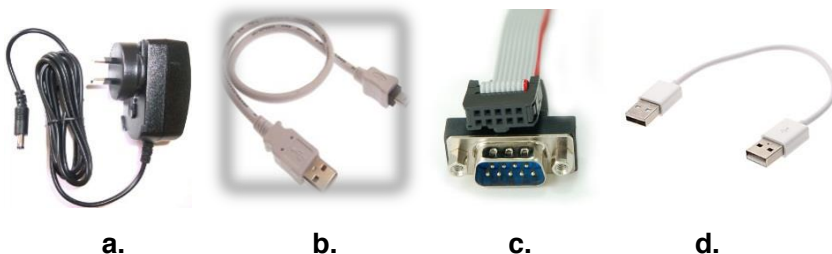


Foto dos cabos básicos para interagir com a Beagle Board. **a** Fonte 5V, **b**. USB-USB-OTG, **c**. IDC10-serial, **d**. USB-USB.

Para a instalação do Ubuntu na BB é preciso contar com um computador (chamado de host) com sistema operacional Linux, também é preciso contar com um cartão SD (de pelo menos dois giga bytes (2GB) de capacidade de armazenamento) que serve como “disco rígido” (HD) na Beagle Board. Neste projeto foi utilizada uma versão precompilada para o micro controlador OMAP3530 disponível na internet. Uma vez que a versão do sistema operacional (OS) desejada é baixada no host, seguem uma série de passos para preparar o cartão SD e hospedar nele a imagem do OS (ver item 3.2.1). Os passos utilizados para instalar Ubuntu-Oneric na BB são:

1. Desde um terminal ou *prompt* no host, executar o comando ***wget <http://rcne.net/deb/rootfs/oneri/ubuntu-11.10-r14-minimal-armel-2012-09-27.tar.xz*** para baixar a imagem comprimida do sistema operacional.
2. Verificar a integridade do pacote baixado executando o comando ***md5sum ubuntu-11.10-r14-minimal-armel-2012-09-27.tar.xz***. A resposta deste comando tem que

ser um *string* parecido com **31be6761a37af98906c5c1e892601e85 ubuntu-11.10-r14-minimal-armel-2012-09-27.tar.xz**.

3. Descompactar o arquivo utilizando o comando **tar xJf ubuntu-11.10-r14-minimal-armel-2012-09-27.tar.xz**. O resultado deste comando é a criação de uma pasta com o nome **ubuntu-11.10-r14-minimal-armel-2012-09-27**. Deve-se entrar nessa pasta utilizando o comando **cd ubuntu-11.10-r14-minimal-armel-2012-09-27** para ter acesso a vários *scripts* que ajudam na instalação da imagem do OS na SD.
4. Com o cartão SD inserido no leitor de cartões do host, executar o script **sudo ./setup_sdc card.sh --probe-mmc**. O script procura a posição de todos os discos rígidos instalados no sistema de arquivos do computador. Isso serve para identificar como é enxergado o cartão SD pelo sistema e o seu endereço será utilizado nos passos seguintes. O usuário deve procurar por uma *string* indicando o tamanho do cartão inserido. O seguinte é um exemplo de resposta para um cartão de quatro giga bytes (4Gb): **Disk /dev/mmcblk0: 3957 MB, 3957325824 bytes**.
5. Executar o script **sudo ./setup_sdc card.sh --mmc /dev/sdX --uboot board** onde **/dev/sdX** corresponde ao endereço do cartão SD no sistema de arquivos do host (achado no ponto anterior) e **board** corresponde ao nome da placa que utilizara o OS. No nosso exemplo, o comando para compilar o Ubuntu-Oneric para a BB versão C3 ficaria da seguinte forma: **sudo ./setup_sdc card.sh --mmc /dev/mmcblk0 --uboot beagle_cx**. Ao finalizar o processo de instalação o cartão apresentará duas partições uma para boot e outra para o

sistema de arquivos (Root File System- RFS). O cartão pode ser desmontado e inserido na BB, assim a próxima vez que a BB seja ligada esta descompactará a imagem do OS.

APÊNDICE B - Comunicação entre a Beagle Board e uma rede de computadores com acesso à Internet

Conectar a Beagle Board numa rede de computadores é muito conveniente na etapa de desenvolvimento já que isso possibilita a utilização de *cross-compilers*. Assim o projetista desenvolve no *host* (com muito mais poder de processamento do que a BB) e deixa a BB como plataforma de teste e de execução. A versão da BB utilizada neste projeto (versão C3) não tem interface ethernet como acontece com revisões mais recentes do hardware (versão XM) e é por isso que para esta tarefa é utilizada a interface USB-OTG. Segue o passo a passo para comunicar a BB com o *host*.

Para fazer esta conexão é preciso do cabo serial (ICD10-DB9, ver Figura 3c) para conectar a BB no *host* e um terminal de emulação de porta serial instalado no *host* (ex. minicom). A porta serial no host deve ser configurada com *baudrate* em 115200, 8-N-1 (8 bits de dados, sem bit de paridade e 1 bit de parada) e sem controle de fluxo.

- Depois de configurar a porta serial do host se deve conectar a BB utilizando o cabo da Figura 3c. A BB deve ser alimentada com a fonte externa de 5V. A sequência de boot deve iniciar e as informações desse processo devem aparecer na tela do *host*. O processo termina com a sequência de *login* no sistema operacional.
- Uma vez logado, o usuário deve ganhar privilégios de super usuário (root) utilizando o comando ***sudo -i***, depois se deve abrir o arquivo de interfaces de rede utilizando o comando ***vi /etc/network/interfaces***. Nesse arquivo devem ser adicionadas as seguintes linhas:

- **auto usb0**
- **iface usb0 inet static**
- **address xxx** (*onde xxx é o endereço IP que vai levar a BB*)
- **netmask 255.255.255.0**
- **broadcast 192.168.0.255 (opcional)**
- **gateway 192.168.0.1 (opcional)**

Ao finalizar a edição do arquivo, este deve ser salvo e para isso se deve pressionar a tecla **ESC** e digitar **:wq** (incluído os dois pontos). Outra opção para configurar a interface de rede é utilizar o comando ***ifconfig usb0 192.168.0.2 netmask 255.255.255.0 up***, mas este comando teria que ser digitado cada que a BB fosse ligada, repetindo todo o processo descrito até agora.

- Com o arquivo salvo e ainda com privilégios de *root*, a interface de rede deve ser reiniciada utilizando o comando ***/etc/init.d/networking restart*** ou simplesmente resetando a BB.
- No lado do host, o arquivo ***/etc/network/interfaces*** também tem que ser alterado utilizando as mesmas linhas descritas anteriormente só que com um endereço IP diferente. Ao final, as interfaces de redes do host também tem que ser resetadas.
- Depois deste procedimento, o cabo USB apresentado na Figura 3b pode ser utilizado para ligar a BB e o host, utilizando esse cabo é possível alimentar a BB. Se tudo foi bem configurado, o host deve reconhecer a BB conectada na interface de rede usb0 (utilizar o comando ***ifconfig*** para verificar se a interface de rede foi habilitada), caso a interface seja reconhecida é possível utilizar o comando ***ping xxx*** para validar a conexão com

a BB. Caso tudo esteja certo, será possível utilizar o **ssh** e o **scp** desde o host para compartilhar informação com a BB sem precisar do cabo serial nem do terminal de emulação serial (minicom).

- Para conectar a BB na internet, no host deve ser instalado um programa chamado **firestarter** utilizando o comando **sudo apt-get install firestarter**. Uma vez instalado, o **firestarter** deve ser configurado para que a interface padrão do host (eth0 ou wlan0) compartilhe internet com a interface usb0.
- Com o **firestarter** configurado e rodando no host, devem ser configurados na BB o **Domain Name System** (DNS) e a rota padrão de saída. O DNS é configurado editando o arquivo **/etc/resolv.conf** adicionando as linhas **nameserver 8.8.8.8** e **nameserver 8.8.4.4** que são endereços de DNS abertos. A rota padrão de saída deve ser configurada utilizando o comando **sudo route add default gw xxx**, onde xxx é o endereço IP da interface padrão do *host* (eth0 ou wlan0). Para validar a conexão pode ser executado na BB o comando **ping www.google.com**.

APÊNDICE C - Configuração de comunicações GSM/3G com o PPP-Config no Linux

Conhecendo a operadora que será utilizada para estabelecer a conexão GPRS/3G é possível executar o comando `pppconfig` para configurar os parâmetros da conexão. No prompt da BB se deve executar o comando ***sudo pppconfig***. Com isso uma interface é aberta onde deve ser selecionada a opção: ***criar uma conexão***. Logo depois se deve dar um **nome à conexão (por exemplo claro)**. Quando a opção do **DNS** for requisitado, deve ser escolhida a opção de DNS estático e deve ser utilizado o mesmo DNS que foi configurado na interface de rede do sistema (8.8.8.8 e 8.8.4.4). O usuário e a senha da operadora são configurados a seguir. Neste projeto foi utilizada a empresa CLARO como operadora prestadora do serviço de internet (ISP). O usuário padrão deste ISP é **claro** e a senha é **claro**. O número que deve ser discado é o ***99#** e por último deve ser especificada a localização do modem no sistema operacional (em geral, o modem é reconhecido no **/dev/ttyUSB1**). Com todas as configurações feitas, se deve escolher a opção ***salvar e retornar ao menu principal***. Com isso, dois arquivos de script são criados. O primeiro, chamado com o nome dado na tela de configuração (claro), se encontra na pasta ***/etc/ppp/peers*** (***/etc/ppp/peers/claro***) e contem as opções que serão utilizadas para rodar o `pppd`. Esse arquivo tem que ser complementado com algumas instruções. O segundo arquivo criado pelo `pppconfig` é o chamado ***chatscript*** que contem as instruções de discagem e permite estabelecer a conexão com o ISP. O arquivo de discagem tem o mesmo nome dado na tela de configuração da conexão (claro) e se encontra na pasta ***/etc/chatscript*** (***/etc/chatscript/claro***). Assim como com o arquivo de configuração do `pppd`, é necessário fazer algumas alterações.

Com todas as configurações feitas, é possível executar o comando ***pon*** para iniciar a comunicação do modem com a operadora (o comando ***poff*** termina a conexão do modem com o ISP). O módulo *hardware* utilizado neste projeto sinaliza uma conexão bem sucedida quando o seu LED fica de cor azul escuro. Quando o modem se encontra conectado com a operadora, a interface `ppp0` deve fazer parte das interfaces de rede da BB. Para verificar a existência dessa interface pode ser utilizado o comando ***ifconfig***. Se a interface `ppp0` estiver disponível é porque o modem ganhou um IP válido. Esse IP é utilizado para modificar a tabela de roteamento principal (*main*) da BB para que esta utilize a interface `ppp0` como saída padrão de internet e não a interface `usb0`. Com isso se ganha independência do host permitindo a BB se tornar num minicomputador com o cem por cento das funcionalidades dum PC padrão. Neste projeto foram criados dois scripts que fazem o trabalho de editar a tabela *main* da BB partindo do endereço IP ganho pelo modem.

APÊNDICE D - Arquivo XML que condensa as informações da rota de exemplo TICEN-TITRI

```
<?xml version="1.0" encoding="UTF-8"?>
<rota>
  <numero>110</numero>
  <nome>TICEN_TITRI</nome>
  <origem>
    <alias>TICEN</alias>
    <latitude>-27.59836</latitude>
    <longitude>-48.55370</longitude>
  </origem>
  <destino>
    <alias>TITRI</alias>
    <latitude>-27.58388</latitude>
    <longitude>-48.52284</longitude>
  </destino>
  <regiao>CENTRO</regiao>
  <horarios>
    <dias_uteis>
      <saida>
        <horario>06:26</horario>
      </saida>
      <saida>
        <horario>06:46</horario>
      </saida>
      <saida>
        <horario>07:04</horario>
      </saida>
      <saida>
        <horario>07:20</horario>
      </saida>
      <saida>
        <horario>07:37</horario>
      </saida>
      <saida>
        <horario>07:54</horario>
      </saida>
    </dias_uteis>
  </horarios>
</rota>
```

<saida>
 <horario>08:10</horario>
</saida>
<saida>
 <horario>08:27</horario>
</saida>
<saida>
 <horario>11:56</horario>
</saida>
<saida>
 <horario>12:16</horario>
</saida>
<saida>
 <horario>12:32</horario>
</saida>
<saida>
 <horario>12:50</horario>
</saida>
<saida>
 <horario>13:09</horario>
</saida>
<saida>
 <horario>13:28</horario>
</saida>
<saida>
 <horario>13:46</horario>
</saida>
<saida>
 <horario>14:05</horario>
</saida>
<saida>
 <horario>14:25</horario>
</saida>
<saida>
 <horario>14:46</horario>
</saida>
<saida>
 <horario>15:06</horario>
</saida>

<saida>
 <horario>15:26</horario>
</saida>
<saida>
 <horario>15:44</horario>
</saida>
<saida>
 <horario>16:03</horario>
</saida>
<saida>
 <horario>16:22</horario>
</saida>
<saida>
 <horario>16:38</horario>
</saida>
<saida>
 <horario>16:55</horario>
</saida>
<saida>
 <horario>17:11</horario>
</saida>
<saida>
 <horario>17:27</horario>
</saida>
<saida>
 <horario>17:43</horario>
</saida>
<saida>
 <horario>17:59</horario>
</saida>
<saida>
 <horario>18:15</horario>
</saida>
<saida>
 <horario>18:32</horario>
</saida>
<saida>
 <horario>18:47</horario>
</saida>

```

        <saida>
            <horario>19:03</horario>
        </saida>
    </dias_uteis>
    <domingos_feriados>
        <saida>
            <horario>
                nao_operar
            </horario>
        </saida>
    </domingos_feriados>
</horarios>
<referencias>
    <waypoint>
        <alias>
            PontehercilioLuz
        </alias>
        <latitude>
            -27.59298
        </latitude>
        <longitude>
            -48.56268
        </longitude>
    </waypoint>
</referencias>
<pautas>
    <local>
        <alias>
            BeiraMarShopping
        </alias>
        <latitude>
            -27.58472
        </latitude>
        <longitude>
            -48.54479
        </longitude>
        <video>
            <nome>
                /usr/dat/centro/110/beiramarshopping.avi
            </nome>
        </video>
    </local>
</pautas>

```

```

        </nome>
        <duracao>
            20
        </duracao>
    </video>
</local>
<local>
    <alias>
        KoxixosBar
    </alias>
    <latitude>
        -27.57326
    </latitude>
    <longitude>
        -48.53936
    </longitude>
    <video>
        <nome>
/usr/dat/centro/110/koxixosbar.avi
        </nome>
        <duracao>
            15
        </duracao>
    </video>
</local>
<local>
    <alias>
        AngeloniTrindade
    </alias>
    <latitude>
        -27.57461
    </latitude>
    <longitude>
        -48.52693
    </longitude>
    <video>
        <nome>
/usr/dat/centro/110/angelonitrindade.avi
        </nome>
```

```
<duracao>
    15
</duracao>
</video>
</local>
</pautas>
</rota>
```