

**UNIVERSIDADE POSITIVO  
NÚCLEO DE CIÊNCIAS EXATAS E TECNOLÓGICAS  
ENGENHARIA DA COMPUTAÇÃO**

**TERMINAL DE CONSULTA PREÇOS REMOTO**

**Maximiliano Alves Brungari**

**Monografia apresentada à disciplina de Projeto Final como requisito parcial à conclusão  
do Curso de Engenharia da Computação, orientada pelo Prof. Marcelo Mikozs**

**UP/NCET**

**Curitiba**

**2008**

## **TERMO DE APROVAÇÃO**

Maximiliano Alves Brungari

Terminal de Consulta de Preços Remotos

Monografia aprovada como requisito parcial à conclusão do curso de Engenharia da Computação da Universidade Positivo, pela seguinte banca examinadora:

Prof. Marcelo Mikosz Gonçalves(Orientador)

Prof. Mauricio Perretto

Prof. Alessandro Brawerman

Curitiba, 15 de Dezembro de 2008.

## **AGRADECIMENTOS**

Pela minha opção profissional e pela escolha do curso de Engenharia da Computação agradeço ao meu pai Jacinto, que infelizmente não pode acompanhar esta batalha, mas certamente serviu como maior exemplo para que eu nunca desistisse do meu sonho. Agradeço também a minha família, minha mãe Maria que sempre esteve ao meu lado no dia a dia durante todos os anos de curso me ajudando a encontrar forças e agüentado meus momentos de stress, e minhas irmãs Juliana e Luciana, pela troca de experiências e apoio durante todos esses anos. E também a minha namora Dé por me acompanhar durante todo o projeto, me incentivar e abrir mão de todo o tempo que dispensei ao seu lado para realização deste sonho. Agradeço também ao meu orientador, Professor Marcelo Mikosz, pelas cobranças e ajuda durante toda a execução do projeto deste a proposta até a entrega.

## RESUMO

O objetivo do projeto é desenvolver um dispositivo computacional que auxilie o comerciante brasileiro a cumprir o decreto de lei No. 5.903, de 20 de setembro de 2006, que obriga todo tipo de comércio disponibilizar alguma consulta de preços ao seu consumidor final. Além disto, poucas empresas oferecem equipamentos semelhantes no Brasil para pequenos comerciantes.

A idéia é ter um sistema completo, desde o software de banco de dados até o hardware de tratamento e interface com o usuário, utilizando apenas o leitor já pronto.

O consumidor apresenta, ao leitor de código de barras do equipamento, o código de barras do produto a ser consultado, então através de comunicação serial o hardware trata o código e o envia para um servidor via rede Wi-Fi utilizando protocolo TCP/IP.

O servidor do sistema possui o banco de dados dos itens a serem consultados. Após o envio do código para o servidor da rede, localizado em um PC com uma interface amigável para usuários leigos, o servidor busca no banco de dados, através da chave do código de barras, a descrição e o preço de determinado produto.

Novamente, através da rede local usando protocolo TCP/IP o servidor devolve ao hardware a descrição e o preço do produto em questão, para assim a CPU do hardware interpretar e disponibilizar as informações em um display.

### Palavras chave:

Preços , TCP/IP, Wireless Lan, banco de dados

## **Remote Terminal to Consult Prices**

### **ABSTRACT**

This project's objective is develop a electronic device that help brazilians companys to follow the law number 5.903, of September, 20 , 2006. This law says that all ratail and shops must show the product price with a computer system. And just a few companys manufacturing same equipment doing a big prices to little costumers.

The idea is a complete system ( hardware, firmware and software ) that connected a serial barcode scanner, read the products barcode, review, find the barcode in a server database through wi-fi /802.11b and show the result (price) on a LCD Display.

Customers pass the barcode in scanner, before, through serial communication the system process data and send to net server. The server had a data base with all items to consult. The product key is used to consult database.

Finally, net server sends to hardware product description and product price. And hardware show this information through LCD display.

### **Key words:**

**Prices, TCP/IP, wireless lan, database**

## SUMÁRIO

<u>CAPÍTULO 1 - INTRODUÇÃO .....</u>	<u>9</u>
<u>CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA .....</u>	<u>11</u>
2.1 - Automação Comercial e Leis Brasileiras .....	11
2.2 – Código de Barras.....	12
2.3 – Wireless Lan .....	15
2.4 – Pontos de Acesso ( Access Point ).....	17
2.5 – Protocolo de Comunicação TCP/IP .....	17
2.6 – Microcontrolador Rabbit RCM4400W Core Module.....	18
2.7 – Banco de Dados (InterBase) .....	19
2.8 – Linguagem de Programação C++ .....	19
<u>CAPÍTULO 3 – ESPECIFICAÇÃO DO PROJETO .....</u>	<u>20</u>
3.1 – Aspectos Gerais .....	20
3.2 – Especificação do Hardware.....	21
3.3 – Especificação do Software .....	24
<u>CAPÍTULO 4 – DESENVOLVIMENTO E IMPLEMENTAÇÃO .....</u>	<u>27</u>
4.1 – Hardware.....	27
4.2 – Software .....	29
<u>CAPÍTULO 5 – VALIDAÇÃO E RESULTADOS.....</u>	<u>36</u>
5.1 – Software Gerenciador .....	36
5.1.1 – Banco da Dados .....	36
5.1.2 – Interface Socket .....	37
5.2 – Hardware e Firmware .....	37
5.2.1 – Interface de rede.....	37
5.2.2 – Display LCD .....	38
5.1.2 – Leitor de Código de Barras .....	38
<u>CAPÍTULO 6 – VIABILIDADE SÓCIO-ECONÔMICA.....</u>	<u>39</u>
<u>CAPÍTULO 7 - CONCLUSÃO .....</u>	<u>40</u>
<u>CAPÍTULO 8 - REFERÊNCIAS BIBLIOGRÁFICAS.....</u>	<u>41</u>

## LISTA DE FIGURAS

Fig. 1.1 – Diagrama de blocos dos sistema .....	10
Fig. 2.2 – Código de Barras EAN-13 .....	13
Fig. 2.3 – Leitura baseada em feixe laser .....	14
Fig. 2.4 – Leitor de Código de Barras CCD .....	14
Fig. 2.5 – Modelo de rede <i>wireless</i> .....	16
Fig. 2.6 – Microcontrolador <i>Rabbit 4400W</i> .....	18
Fig. 3.7 – Sistema do Terminal de Consulta Preço Remoto .....	20
Fig. 3.8 – Módulo de Hardware .....	21
Fig. 3.9 – Comunicação Serial do Leitor de Código de Barras.....	22
Fig. 3.10 – Comunicação Serial do <i>Display</i> LCD.....	23
Fig. 3.11 – Tabela de Formação de Caracteres do Display LCD.....	23
Fig. 3.12 – Diagrama de <i>software</i> .....	25
Fig. 3.13 – Plataforma de Desenvolvimento Dinamic C.....	<b>Erro! Indicador não definido.</b>
Fig. 4.14 – Conector J1 – E/S do kit RCM4400W.....	29
Fig. 4.15 – Casos de Uso do Sistema Gerenciador .....	30
Fig. 4.16 – Diagrama de seqüência – Cadastro de Mensagem.....	31
Fig. 4.17 – Diagrama de seqüência – Cadastro de Produtos .....	32
Fig. 4.18 – Diagrama de seqüência – Cadastro de Terminais .....	32
Fig. 4.19 – Diagrama de seqüência – Envia Configuração .....	33
Fig. 4.20 – Diagrama de seqüência – Consulta Banco de Dados.....	33
Fig. 4.21 – Fluxograma de funcionamento do <i>Firmware</i> .....	34
Fig. 4.22 – Modelo de relacionamento das Tabelas Mensagens e Produtos .....	35
Fig. 4.23 – Tabela Terminal .....	35
Fig. 4.24 – Tela de Cadastramento de Produtos.....	36

**LISTA DE TABELAS**

Tabela 2.1 – Características RCM4400W.....	21
Tabela 3.1 – Sinais do Display LCD.....	25
Tabela 4.1 – Conexão do Leitor de Código de Barras.....	30
Tabela 4.2 – Conexão do Display LCD.....	31
Tabela 6.1 – Viabilidade Sócio-Econômica do projeto.....	41
Tabela 6.2 – Custos de Desenvolvimento do Produto.....	42



**LISTA DE SIGLAS**

**NCET**- Núcleo de Ciências Exatas e Tecnológicas

**UP** – Universidade Positivo

**IP** – Internet Protocol

**TCP** – Transmission Control Protocol

**WiFi** – Wireless Fidelity

**LAN** – Local Area Network

**WLAN** – Wireless LAN

## CAPÍTULO 1 - INTRODUÇÃO

A legislação que passou a vigorar recentemente no Brasil obriga todos os comerciantes que usam sistemas de automação com código de barras a utilizar algum sistema auxiliar para informação do preço ao consumidor.

Segundo decreto de lei No. 5.903, de 20 de setembro de 2006 que regulamentada a Lei nº 10.962, de 11 de outubro de 2004, e a Lei nº 8.078, de 11 de setembro de 1990 todo o comércio que utilizar código de barras para apreçamento deverá disponibilizar para seu consumidor equipamentos para consulta de preços com leitura ótica.

O projeto visa desenvolver um equipamento para a consulta de preços de produtos com códigos de barras padrão EAN13 e EAN8 para uso remoto.

A idéia consiste em um sistema completo (hardware, software básico e software de interface ) que, devidamente ligado a um leitor de código de barras serial, leia o código de barras de produtos, interprete, consulte uma base de dados em um servidor via rede *Wireless Lan Ethernet* 802.11g e mostre graficamente em um visor digital acoplado ao seu hardware o preço e algumas outras informações a respeito do produto, como uma mensagem promocional que pode ser atribuída ao produto e mostrada no visor logo após a consulta. O digrama de blocos demonstrado na Figura 1.1 apresenta o fluxo das informações no sistema.

O hardware do sistema será composto por um leitor de código de barras padrão com interface serial, um microcontrolador, uma interface *Ethernet* 802.11g, uma bateria e um visor. O software de interação que é basicamente uma interface no qual o usuário possa cadastrar os itens pré-definidos e alguns outros parâmetros de mensagens promocionais, e um servidor de banco de dados para todo o sistema. O *firmware* desenvolvido em C estruturado fará a interface entre todos os periféricos do sistema, o *software* de interface e a resposta no *display* de LCD.

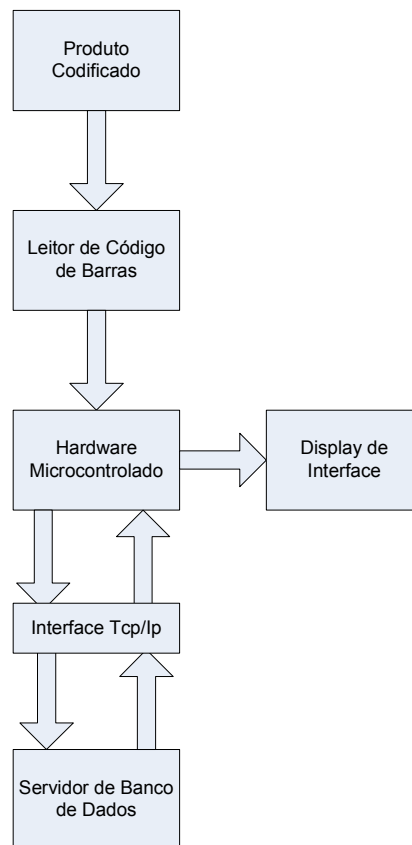


Fig. 1.1 – Diagrama de blocos dos sistema

Mesmo tendo um grande mercado, existem poucos fabricantes de dispositivos semelhantes no Brasil, fazendo assim que o seu custo não seja acessível a comércios de qualquer porte e a implementação deste equipamento com interface *wireless* torna-o muito atraente para o mercado aumentando a sua mobilidade e diminuindo o seu custo de instalação.

Os capítulos a seguir apresentam toda a fundamentação teórica necessária para o desenvolvimento do projeto, toda especificação técnica, a descrição de todo o projeto desenvolvido e as conclusões finais.

## CAPÍTULO 2 – FUNDAMENTAÇÃO TEÓRICA

Neste capítulo estão descritas todas as teorias necessárias para o desenvolvimento do projeto, desde as teorias ligadas ao meio em que o produto será inserido, até as teorias mais técnicas como redes e microprocessadores.

### **2.1 - Automação Comercial e Leis Brasileiras**

O Brasil é um país com uma série de leis que regulamentam equipamentos para o comércio gerando grandes oportunidades como o mercado de automação comercial que é baseado principalmente em equipamentos e softwares regulamentados para automação completa do comércio.

Na década de oitenta o comércio brasileiro tinha quase nível zero de automação, usando em alguns estabelecimentos de maior porte máquinas registradoras digitais e de menor porte máquinas registradoras mecânicas ou até mesmo calculadoras.

Chegada a década de noventa, no dia 11 de setembro de 1990 o governo federal decreta a Lei 8.078 regulamentando os direitos do consumidor e as obrigações dos agentes de oferta como indústrias e comércios, e segundo o artigo 31 deste decreto de lei:

*Art. 31. A oferta e apresentação de produtos ou serviços devem assegurar informações corretas, claras, precisas, ostensivas e em língua portuguesa sobre suas características, qualidades, quantidade, composição, preço, garantia, prazos de validade e origem, entre outros dados, bem como sobre os riscos que apresentam à saúde e segurança dos consumidores*

Após decretada a obrigatoriedade de apreçamento de mercadorias o governo federal brasileiro decreta a Lei de Convênio de ICMS 156/94 regulamentando equipamentos para automação comercial, surgindo assim o Emissor de Cupom Fiscal e a obrigatoriedade do comércio em ter sistemas de automação.

Depois do ano de 1994 a automação comercial começou a se expandir no Brasil. Vários tipos de comércios e de diversos tamanhos estavam automatizando suas lojas. Os fabricantes cada vez mais aderiam ao padrão EAN (regulamento pela empresa GS1 que antes se chamava EAN) de codificação de produtos, levando suas mercadorias até os pontos de vendas já com código de barras.

Com a evolução dos softwares e equipamentos para automação, os comerciantes passaram a armazenar os dados de suas mercadorias em banco de dados, assim além de cumprir algumas disposições legais, poderiam também gerenciar seus negócios. Armazenando todas as informações dos produtos como código de barras e preços em um banco de dados, não era mais

preciso etiquetar os produtos um a um. Porém era preciso manter alguma forma de apresentação de dados das mercadorias para o consumidor. A solução imediata foi a colocação de etiquetas nas gôndulas alinhadas aos seus respectivos produtos.

Com o passar dos anos, a grande maioria dos comerciantes não respeitou as formas de fixação de preços em seus estabelecimentos fazendo com que o governo federal também regulamentasse as condições de oferta e apreçamento através da lei de número 5.903 da data de 20 de setembro de 2006, que regulamenta as formas de fixação de preços e a obrigatoriedade de equipamentos para consulta de preços. Segundo o artigo 4 deste mesmo decreto:

*Art. 4º Nos estabelecimentos que utilizem código de barras para apreçamento, deverão ser oferecidos equipamentos de leitura ótica para consulta de preço pelo consumidor, localizados na área de vendas e em outras de fácil acesso.*

No ano de 2006 o Governo Federal através do decreto 5.903, de 20 de setembro de 2006, regulamenta a lei 10.962, de 11 de setembro de 2004. Com o seguinte texto sobre equipamentos de consulta preço:

*Art. 7º Na hipótese de utilização do código de barras para apreçamento, os fornecedores deverão disponibilizar, na área de vendas, para consulta de preços pelo consumidor, equipamentos de leitura ótica em perfeito estado de funcionamento.*

*§ 1º Os leitores óticos deverão ser indicados por cartazes suspensos que informem a sua localização.*

*§ 2º Os leitores óticos deverão ser dispostos na área de vendas, observada a distância máxima de quinze metros entre qualquer produto e a leitora ótica mais próxima.*

## **2.2 – Código de Barras**

O código de barras é uma representação gráfica de dados que, em uma seqüência de barras pretas e brancas, podem indicar valores numéricos ou alfanuméricos.

### **2.2.1 - História**

Em 1948 Bernard Silver e Norman Joseph Woodland começaram a desenvolver um dispositivo leitor de padrões de tintas usando luz ultravioleta. Após alguns testes foram detectados problemas em relação a estabilidade das tintas e ao alto custo da impressão.

No dia 20 de outubro de 1949, baseado no código morse, Woodland e Silver patentaram o primeiro sistema de código de barras. Formado por quatro linhas brancas colocadas sobre um fundo preto em um círculo concêntrico. Assim descrito no manual do código de barras encontrado no site da GS1.

### 2.2.2 – Tipos de Código de Barras

Existem diversos tipos de código de barras, portanto os formatos das barras variam de acordo com o código utilizado.

Os códigos mais usados comercialmente no mundo todo são os padrões EAN/UPC. A identificação de itens comerciais é feita através dos códigos EAN-13 e EAN-8 com um padrão mundial, não havendo repetições de códigos em lugar algum do mundo.

O código EAN 13 é composto de 4 agrupamentos conforme a Figura 2.2, o primeiro (*Number System*) composto por 2 ou 3 números, representa o país de fabricação do produto de acordo com uma tabela. O Brasil é representado pelo número 789. O segundo (*Mfg Code*) agrupamento, composto de 4 ou 5 números, corresponde ao código da empresa fabricante do produto. O terceiro agrupamento (*Product Code*) é composto por 5 dígitos que informam o código do item comercial. E o último agrupamento (*Check Digit*) é composto por apenas um dígito que é o dígito verificador, obtido por meio de um algoritmo.



Fig. 2.2 – Código de Barras EAN-13

O código EAN-8 é uma versão reduzida do EAN-13, modificado apenas para se adaptar a embalagens menores. Este código apenas inclui o número de identificação do país de fabricação, o código do item comercial e o dígito verificador.

### 2.2.2 – Leitores de Código de Barras

A decodificação dos dados representados em um código de barras é feita através de leitores óticos.

Os leitores óticos baseados em caneta laser ou à varredura laser usam basicamente um feixe simples de luz vermelha como verificado na Figura 2.3, no caso das canetas, ou diversos feixes de luz vermelha com o emprego de espelhos, que através da reflexão das barras claras e escuras convertem a representação gráfica em uma seqüência de bits, no caso dos leitores à varredura.

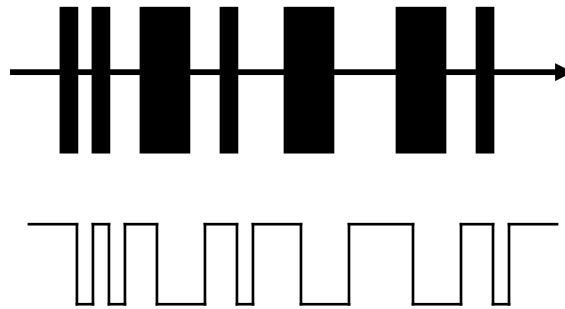


Fig. 2.3 – Leitura baseada em feixe laser

Leitores com tecnologia CCD (*Charge Coupled Devices*), mostrado na Figura 2.4, usam um vetor com centenas de sensores de luz minúsculos alinhados que recebem a resposta de reflexão simultaneamente. Estes tipos de leitores normalmente são os mais baratos por usarem a tecnologia mais simples.



Fig. 2.4 – Leitor de Código de Barras CCD

Os mais sofisticados e modernos são os leitores baseados em câmeras que utilizam uma pequena câmera de vídeo para capturar a imagem do código de barras e poderosos algoritmos de processamento de imagens para decodificá-los.

### 2.2.2 – Cálculo do Dígito Verificador

Para calcular o dígito verificador de um código de barras EAN são separados os dígitos em pares e ímpares da direita para esquerda. Somam-se os números de posição par. Multiplica-se os

ímpares por 3 e depois soma-se os números de posição ímpar. Depois somam-se os dois resultados. O dígito verificador é a diferença para o próximo múltiplo de dez.

Segue um exemplo de cálculo:

Código de Barras: 789105206230x

Pares: 3 6 2 0 9 7

Soma dos pares:  $3 + 6 + 2 + 0 + 9 + 7 = 27$

Ímpares: 0 2 0 5 1 8

Ímpares multiplicados por 3: 0 6 0 15 3 24

Soma dos Ímpares:  $0 + 6 + 0 + 15 + 3 + 24 = 48$

Soma de Pares e Ímpares:  $27 + 48 = 75$

Diferença para o próximo múltiplo de 10:  $80 - 75 = 5$

Dígito Verificador (x) = 5

### 2.3 – Wireless Lan

As redes sem-fio estão sendo cada vez mais utilizadas na comunicação entre dispositivos dos mais variados tipos e tamanhos (computadores pessoais, portáteis e de mão, telefones, eletrodomésticos, máquinas industriais, sensores e outros), como mostra a Figura 2.5, em diferentes ambientes, como residências, edifícios, cidades, florestas e campos de batalha. Por permitirem a mobilidade, estas redes facilitam a ubiquidade do poder computacional, tornando transparente a disseminação da informação e a cooperação dos dispositivos na realização das mais variadas tarefas.

Diferentes padrões e tecnologias de rede sem-fio surgiram nos últimos anos para acomodar esta vasta gama de aplicações e coberturas. Foram concebidas das redes celulares de larga cobertura, passando pelas redes locais (*WLANs*), até as redes PANs (*Personal Area Networks*) usadas na comunicação de equipamentos pessoais, como PDAs (*Personal Digital Assistants*), câmeras digitais, computadores e celulares. [RUBINSTEIN, M. G. ; REZENDE, J.R]



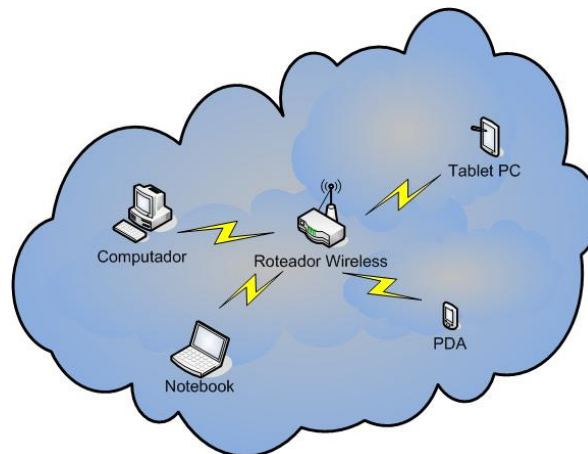


Fig. 2.5 – Modelo de rede *wireless*

As redes sem fio *Wi-Fi* (*Wireless Fidelity*) foram umas das grandes novidades tecnológicas dos últimos anos. A comunicação entre os dispositivos são feitas através de ondas de rádio não necessitando de cabeamento como mostra a figura 2.4 Após sete anos de pesquisa e desenvolvimento, no ano de 1997, o comitê de padronização da IEEE aprovou o padrão IEEE 802.11, que em uma versão inicial, atingia as taxas de transmissão nominais de 1 e 2 Mbps.

Em setembro de 1999, o IEEE ratificou uma revisão do padrão 802.11, chamada de 802.11 *High Rate* (HR/DSSS), ou 802.11b, que fornece taxa de transferência de dados muito superior, mantendo o padrão inicial. A arquitetura básica, assim como características e serviços definidos no padrão original foram totalmente mantidos, já que a nova especificação afetou somente a camada física. As modificações no modo em que os chips são codificados e um novo método de modulação chamado de QPSK possibilitaram esse aumento nas taxas de transferências.

Sua principal inovação é o suporte para duas novas velocidades: 5,5 e 11Mbps. Para isso, o DSSS foi escolhido como único modo de transmissão, já que o FHSS não suporta tráfego a tais velocidades. A idéia é que este novo padrão tenha interoperabilidade com sistemas 802.11 DSSS, mas não com FHSS. A transição 802.11/802.11b é semelhante à transição *Ethernet/Fast Ethernet*. As máquinas com rede *Fast Ethernet* podem diminuir suas velocidades para conversar com portas *Ethernet*, mas jamais ao contrário.

As *WLANs* (*Wireless LAN*) utilizam *Dynamic Rate Shifting*, permitindo que as taxas de transmissão sejam automaticamente modificadas dependendo da qualidade do sinal recebido pelo cliente. Este ajuste de velocidade acontece em função da qualidade do sinal recebido pelo cliente. A tecnologias de *Spread Spectrum* mais recentes são criadas para pular de forma transparente entre velocidades como 1, 2, 5.5 e 11Mbps. Quando uma estação se afasta de seu

*AP (Access Point)*, o sinal diminui de potência e as taxas máximas não podem ser mais atingidas. A estação irá então automaticamente deixar cair a velocidade da conexão [BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.]

## **2.4 – Pontos de Acesso ( Access Point )**

Um ponto de acesso (AP) é o ponto central das redes *wireless* ou é a conexão entre redes estruturadas e sem fio. Os pontos de acessos podem funcionar de três formas: *Root*, Repetidor e *Bridge*. [BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.]

### **2.4.1 – Modo Root**

No modo root o ponto de acesso é conectado a uma rede cabeada funcionando como ligação física entre os dispositivos sem fio e dispositivos padrão *Ethernet*. [BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.]

### **2.4.2 – Modo Repetidor**

O ponto de acesso apenas repete o sinal enviando por um outro ponto de acesso, ele funciona em modo *root*. Neste modo a queda de qualidade das conexões e taxas de transferência é facilmente detectada. [BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.]

### **2.4.3 – Modo Bridge**

Funcionando como *bridge*, normalmente, dois pontos de acesso fazem um link *wireless* entre duas redes cabeadas. Na maioria das vezes os pontos de acessos usam toda a sua banda para essa conexão, sem nenhum cliente direto. [BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.]

## **2.5 – Protocolo de Comunicação TCP/IP**

Para que os computadores de uma rede possam trocar informações é necessário que todos adotem as mesmas regras para o envio e o recebimento de informações. Este conjunto de regras é conhecido como Protocolo de comunicação. Falando de outra maneira pode-se afirmar: "Para que os computadores de uma rede possam trocar informações entre si é necessário que todos estejam utilizando o mesmo protocolo". No protocolo de comunicação estão definidas todas as regras necessárias para que o computador de destino, "entenda" as informações no formato que foram enviadas pelo computador de origem. Dois computadores com protocolos diferentes instalados, não são capazes de estabelecer uma comunicação e trocar informações.

O TCP (*Transmission Control Protocol*) é o protocolo de transporte destinado a aplicações que necessitam de uma entrega confiável e em seqüência. O protocolo utiliza o estabelecimento de uma conexão entre o transmissor e o receptor [TANENBAUM]

Uma conexão TCP é *full-duplex* e ponto a ponto. Consiste em um fluxo de bytes e cada byte em uma conexão TCP tem seu próprio número de seqüência de 32 bits. As entidades transmissoras e receptoras trocam dados na forma de segmentos, sendo um segmento TCP composto por um cabeçalho de tamanho fixo de 20 bytes que pode ser seguido por opções de cabeçalho, seguido por um conjunto de bytes de dados [TANENBAUM].

## 2.6 – Microcontrolador Rabbit RCM4400W Core Module

O microcontrolador *RCM4400W RabbitCore*, Figura 2.6, acrescenta um módulo de rede *Wi-Fi* 802.11b aos processadores *Rabbit* 4000. Esta característica permite que se crie soluções de comunicação e controle *wireless* de baixo custo e consumo.[RABBIT]



Fig. 2.6 – Microcontrolador *Rabbit* 4400W

Fonte: *RabbitCore User's Manual*

As características do *Rabbit* 4000 incluem *hardware* DMA, velocidade de *clock* de até 60 MHz, linhas E/S compartilhadas com até 6 portas e 4 níveis alternados de funções de pinos, E/S auxiliar, decodificador de quadratura, captura de entrada, Relógio de Tempo Real (RTC), monitor *watchdog* e outras características descritas na Tabela 2.1. [RABBIT]

TABELA 2.1 - Características RCM4400W

Característica	RCM4400W
Microprocessador	Rabbit 4000 de 58,98 MHz
Memória <i>Flash</i>	512K
Memória de dados SRAM	512K
Memória de Programa SRAM	512K
Portas Seriais	6 portas de alta velocidade compartilhadas 6 são configuráveis como portas seriais assíncronas; 4 são configuráveis como portas seriais síncronas (SPI); 2 são configuráveis como portas seriais SDLC/HDLC; 1 porta serial assíncrona usada para programação;
<i>Wi-Fi</i>	Padrão 802.11b, ISM 4 GHz

FONTE: *Rabbit Core User's Manual*.

O RCM 4400W é programado usando o *Software de Dinamic C* que é baseado em linguagem de programação ANSI C e também aceita comandos *Assembly*.

## 2.7 – Banco de Dados (InterBase)

InterBase é um sistema gerenciador de banco de dados relacionais da Borland. O Interbase é uma alternativa aos bancos de dados tradicionais como Access e SQL tem a vantagens de ser grátis. Além de não ser pesado é relativamente rápido e suporta bancos de dados de grande tamanho.

## 2.8 – Linguagem de Programação C++

A linguagem de programação C foi desenvolvida por Kernighan e Ritchie no *Bell Laboratories* em 1978. O propósito era desenvolver uma linguagem de programação portátil, pequena, simples e rápida que não impedisse o programador de fazer o que era necessário ser feito. Em 1988, comitê X3J11 da ANSI aprovou uma norma ANSI para essa linguagem. [LEE, R.C. ; TEPFENHART]

Em 1989, foi criado o comitê X3J16 da ANSI para padronização C++. O C++ é a linguagem mais utilizada à programação orientada a objeto.

## CAPÍTULO 3 – ESPECIFICAÇÃO DO PROJETO

### 3.1 – Aspectos Gerais

O sistema é constituído por um módulo de hardware e pelo módulo de software. O módulo de hardware tem como componente principal o microcontrolador *Rabbit 4000* com suporte a *Wi-Fi* 802.11b que gerencia a comunicação TCP/IP via *Wi-Fi* e a comunicação serial com o visor LCD e o Leitor de Código de Barras. E o módulo de *software* é responsável pelo banco de dados dos produtos, interface de comunicação com o operador do sistema, e gerenciamento da rede. Este processo está exemplificado na Figura 3.7.

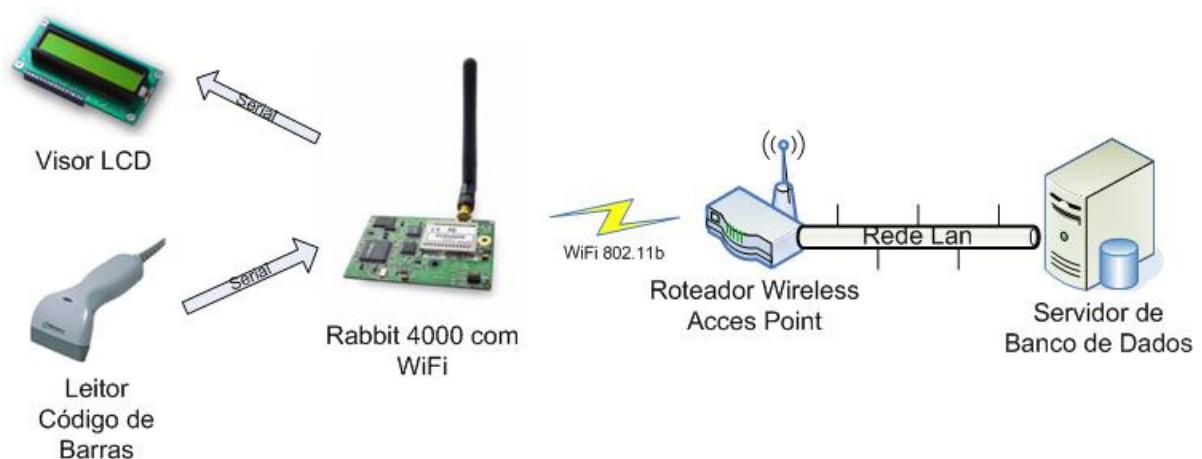


Fig. 3.7 – Sistema do Terminal de Consulta Preço Remoto

O leitor interpreta o código de barras apresentado pelo usuário e o envia para o microcontrolador via interface serial RS-232. O microcontrolador interpreta os dados recebidos pela serial e envia o código do produto pela interface *Wi-Fi* 802.11b, através da rede até o ponto de acesso *wireless*, que é um componente externo do sistema, ligado ao servidor via rede *Ethernet*. O software de interface com o operador e gerenciamento de rede detecta a tentativa de comunicação via TCP/IP do módulo de hardware e recebe os pacotes com o código de barras do produto. Recebendo o pacote, o software acessa o banco de dados dos produtos e envia a resposta com o preço, descrição e mensagem promocional para o módulo de hardware, via pacotes TCP/IP. O módulo de hardware, ao receber a resposta do software, envia os dados ao visor de LCD de forma ordenada mostrando primeiro a descrição do produto e o preço por um tempo pré-determinado, e logo após a mensagem promocional que também permanece disponível no visor LCD por tempo pré-determinado.

### 3.2 – Especificação do Hardware

O módulo de hardware é responsável pela aquisição e tratamento do sinal serial enviado pelo leitor de código de barras, pelo armazenamento das configurações de rede, pela interface *Wi-Fi* 802.11b e por disponibilizar no display os resultados obtidos como demonstrado no diagrama da Figura 3.8. Este módulo é composto pelo processador *Rabbit 4000 Core Module* e seu kit de desenvolvimento *RCM4400 Development Kit*, pelo leitor de código de barras e pelo visor de LCD 16 colunas x 2 linhas.

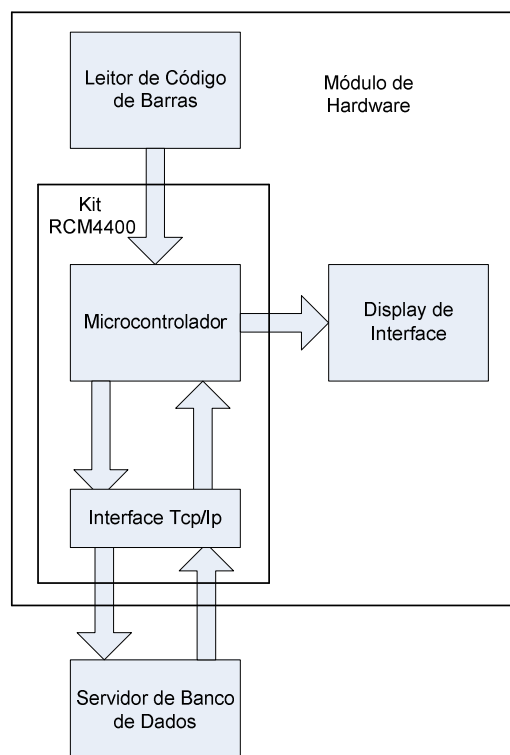


Fig. 3.8 – Módulo de Hardware

As interfaces com usuário do sistema são o leitor de código de barras, que recebe os dados, e o *display* LCD, que disponibiliza a resposta do sistema. Tanto o leitor de código de barras quanto o *display* LCD comunicam-se com o circuito de comunicação serial do RCM 4400W *Development Kit*.

#### 3.2.1 – Leitor de Código de Barras

O leitor de código de barras pode ser modificado de acordo com a necessidade de custo, pois, todos os tipos de leitores fabricados hoje têm configurações seriais e configuração do formato das palavras, sendo adaptáveis ao sistema. A comunicação entre o kit RCM4400 e o

leitor será feita através de protocolo serial, como o diagrama da Figura 3.9 demonstra e o leitor tem que estar habilitado para leituras de código de barras EAN-13 e EAN-8 com envio do dígito verificador.

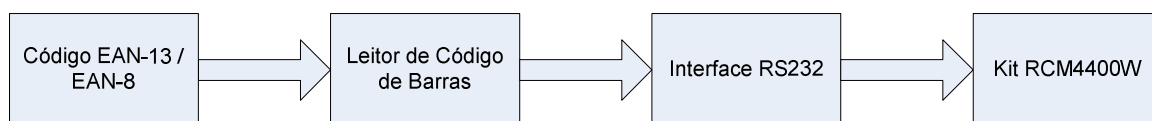


Fig. 3.9 – Comunicação Serial do Leitor de Código de Barras

### 3.2.2 – Display LCD

O *Display LCD* é baseado no controlador Hitachi HD44780A, alfanumérico e com 16 colunas e 2 linhas. A estrutura de formação de caracteres e configuração é de 8 bits configurados através dos sinais de dados descritos na Tabela 3.1

TABELA 3.1 - Sinais do Display LCD

Pino	Sinal	Descrição
1	GND	Terra
2	Vcc	Alimentação
3	Vee	Ajuste de Contraste
4	RS	Registrador
5	R/W	Escrita/Leitura
6	E	Habilitação
7	D0	Linha de Dados 0
8	D1	Linha de Dados 1
9	D2	Linha de Dados 2
10	D3	Linha de Dados 3
11	D4	Linha de Dados 4
12	D5	Linha de Dados 5
13	D6	Linha de Dados 6
14	D7	Linha de Dados 7
15	LED A	Iluminação +
16	LED K	Iluminação -

FONTE: *Datasheet* HD44780A

A comunicação com o microcontrolador é serial, de maneira que após receber a resposta do servidor o kit RCM4400W monta as *strings* de acordo com a tabela de caracteres, alocada na memória e apresentada na Figura 3.10. Logo após, através da interface serial habilita e envia as

strings ao *display*, contendo o preço do produto, o valor e a mensagem promocional. Nos períodos ociosos do terminal o display receberá do microcontrolador uma mensagem padrão.

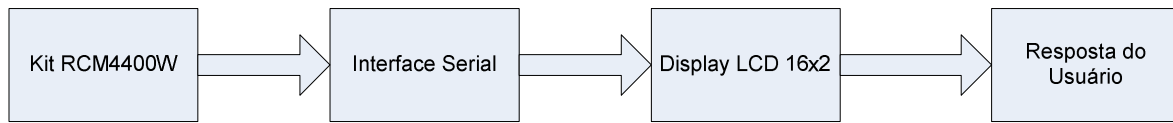


Fig. 3.10 – Comunicação Serial do *Display LCD*

Lower 3 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CIS FRAZ (1)			0 @ P ` P								- 9 3 α p				
xxxx0001	(2)			! 1 A Q a 9								。 7 7 4 ä q				
xxxx0010	(3)			" 2 B R b r								「 イ ツ ｽ ̢ θ				
xxxx0011	(4)			# 3 C S c s								」 ウ ㄗ ㄗ ε ∞				
xxxx0100	(5)			\$ 4 D T d t								、 イ ト ㄗ μ Ω				
xxxx0101	(6)			% 5 E U e u								・ オ ナ 1 ㄗ Ü				
xxxx0110	(7)			& 6 F U f v								ヲ カ ニ ヨ ρ Σ				
xxxx0111	(8)			' 7 G W g w								ㄗ ㄗ ㄗ ㄗ ρ π				
xxxx1000	(1)			< 8 H X h x								イ ク ネ リ ㄗ ㄗ				
xxxx1001	(2)			> 9 I Y i y								ウ ケ ㄗ ㄗ ㄗ ㄗ ㄗ				
xxxx1010	(3)			* : J Z j z								エ コ ㄗ ㄗ ㄗ ㄗ ㄗ				
xxxx1011	(4)			+ ; K [ k <								オ サ ヒ ㄗ ㄗ ㄗ ㄗ				
xxxx1100	(5)			, < L ¥ 1								ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ				
xxxx1101	(6)			- = M ] m }								ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ				
xxxx1110	(7)			. > N ^ n ÷								ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ				
xxxx1111	(8)			/ ? 0 _ o €								ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ ㄗ				

Fig. 3.11 – Tabela de Formação de Caracteres do Display LCD

### 3.2.3 – Kit RabbitCore RCM4400W

O Kit de desenvolvimento usado no projeto é o *Kit Rabbit Core RCM4400W*, da *Rabbit Inc.*. A principal característica deste kit é a adição do módulo *Wi-Fi* 802.11b ao módulo do processador *Rabbit 4000*. O kit tem um baixo custo e um baixo consumo de energia quando comparado a outros processadores com módulos *Wi-Fi*. A sua memória flash de 512 Kb possibilita a gravação de todas as variáveis necessárias ao sistema. O sistema de interface serial com dispositivos seriais é todo implementado no próprio kit.



O *firmware* do RCM4400W é desenvolvido no programa *Dinamic C* 10.2 ou inferior que tem suporte a linguagem C e linguagem *Assembly*, inclusive permitindo o uso de ambas no mesmo código de programa.

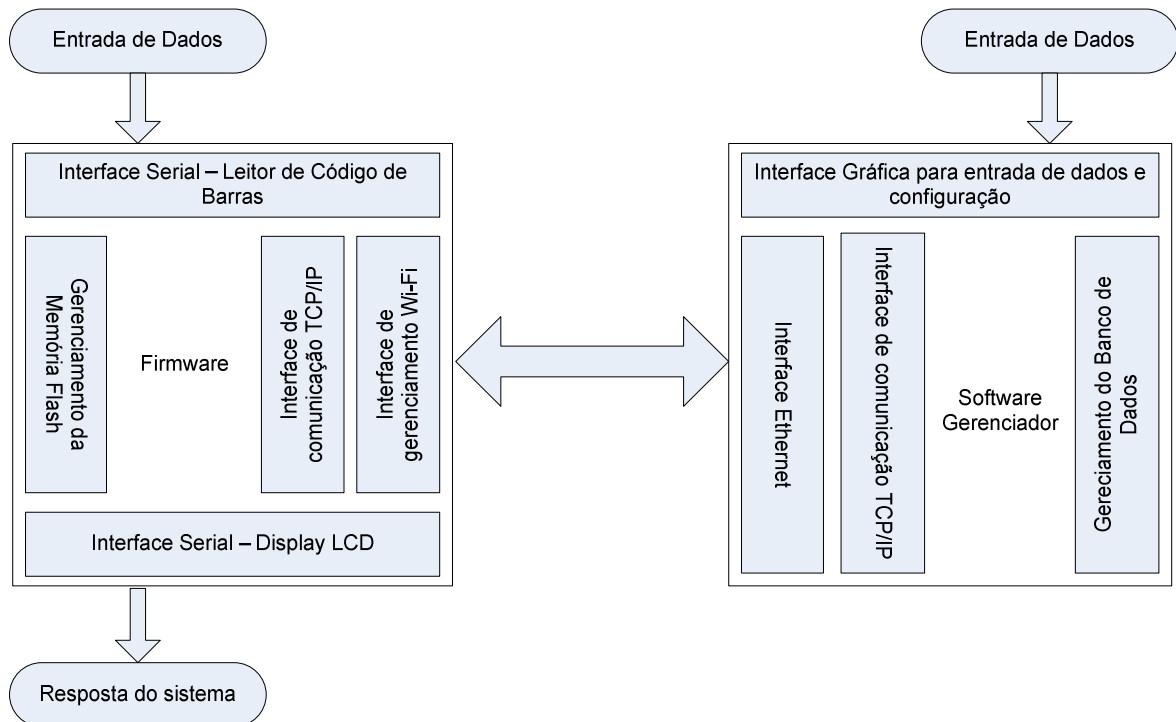
O software embarcado no RCM4400W será o responsável por todo o gerenciamento dos dispositivos seriais ( *display* LCD e leitor de código de barras), comunicação *Wi-Fi* com o servidor e acesso a memória *flash*.

### 3.3 – Especificação do Software

O sistema todo é composto por dois softwares principais: 1) *Firmware*, sistema embarcado no microcontrolador e desenvolvido em *Dinamic C*, que gerenciara todo o hardware, desde as comunicações com dispositivos seriais a gerenciamento dos protocolos de rede. 2) Software Gerenciador, alocado no servidor e desenvolvido em C++, é responsável pelo gerenciamento do banco de dados, pela interface do operador do sistema e por algumas configurações remotas do hardware assim como a comunicação TCP/IP com o hardware.

A Figura 3.12 demonstra todo o diagrama de blocos do sistema. Os software são divididos em blocos internos para facilitar o desenvolvimento de funções específicas para cada necessidade. Como mostrado na Figura 3.12 a entrada de dados do firmware dá-se através da interface que controla a porta serial conectada ao leitor de código de barras. A saída do firmware é controlada pela interface serial que controla o display. No software gerenciador todas as entradas de dados são feitas através da interface gráfica com o operador do sistema. A conexão entre os dois softwares também é feita através dos blocos de gerenciamento de rede internos dos sistemas.

Em geral, o *firmware* recebe o sinal serial enviado pelo leitor e o envia no formato de rede, neste caso TCP através de IP. Após o envio dos dados via rede *Wi-Fi* 802.11b o software gerenciador recebe o código e consulta o banco de dados. A resposta do sistema que é a descrição, preço e mensagem promocional do produto é enviada ao terminal de maneira que o *firmware* do controlador interprete as mensagens e as disponibilize no *display* LCD.

Fig. 3.12 – Diagrama de *software*

### 3.3.1 – Especificação do Software Gerenciador

O *software* de interface com o operador é o responsável pela entrada do banco de dados e pelo servidor de dados de todo o sistema. É desenvolvido em C++ utilizando a plataforma *Borland C++ Builder 6* e funciona em ambiente *Windows 32*. Funciona como um servidor de dados para o terminal. Ele trata o código recebido dos terminais e envia novamente a resposta, neste caso a descrição do produto, o preço e a mensagem promocional. Também é o responsável pela configuração de alguns parâmetros do terminal. A entrada dos dados de cadastro e configuração no sistema dá-se manualmente em uma interface amigável para o usuário.

### 3.3.2 – Especificação do Firmware

O software básico, ou *firmware*, é desenvolvido em linguagem C, utilizando o compilador *Dinamic C 10.1*, disponibilizado pela *Rabbit Inc.*, fabricante do kit RCM4400W utilizado no projeto. Este software é responsável por todo o gerenciamento do hardware do sistema.

### **3.3.3 – Especificação do Banco de Dados**

O banco de dados utilizado no projeto é o InterBase 6.0, que acompanha a plataforma de desenvolvimento Borland C++ Builder 6. O banco de dados selecionado tem uma série de rotinas padrões utilizadas pela plataforma de desenvolvimento do software e é o suficiente para as necessidades do sistema que utiliza um banco de dados simples com apenas 3 tabelas.

As tabelas são: 1) Tabela de Configuração dos Terminais , 2) Tabela de Mensagem e 3) Tabela de Produtos. A tabela de configuração dos terminais armazena as configurações de cada terminal remoto na rede. A tabela de mensagem armazena as mensagens promocionais que são cadastradas no sistema para a associação aos produtos, para serem mostradas no display do terminal após a consulta do preço do produto. Por fim, a tabela de produtos armazena os dados referentes ao produto, como preço, descrição e código e recebe também o código da mensagem que é atribuída a ele.

## CAPÍTULO 4 – DESENVOLVIMENTO E IMPLEMENTAÇÃO

Este capítulo apresenta todas as etapas do desenvolvimento do projeto demonstrando os métodos de desenvolvimento.

### **4.1 – Hardware**

O projeto do hardware é baseado todo no módulo *RabbitCore* 4400W e seu kit de desenvolvimento RCM4400W ligado ao leitor de código de barras serial e ao display LCD 16x2.

#### **4.1.1 – Conexão do código de barras**

O leitor de código de barras é conectado ao kit pela porta serial *Port C*, ligando os sinais TX e RX do leitor aos sinais RXC e TXC, respectivamente, do kit e o sinal GND do leitor é ligado diretamente ao terra do sistema. Na tabela 4.1 representa as ligações necessárias do leitor de código de barras ao kit RCM4400W.

TABELA 4.1 - Conexão do Leitor de Código de Barras

Leitor de Código de Barras		Kit RCM 4400W		
Pino	Sinal	Sinal	Nome	Pino
2	TX	RXC	PC3	27
3	RX	TXC	PC2	26
5	GND	GND		50

FONTE: Autor, 2008

#### **4.1.2 – Conexão do display LCD**

O *display* LCD é ligado ao kit pelo *Port A* para os sinais de dados e pelo *Port B* para endereçamento e habilitação. Os sinais de dados, D[0:7], do *display* são ligados as saídas de dados PA[0:7] disponíveis no kit. O sinal GND é ligado ao terra assim como sinal Vee de ajuste de contraste. Os sinais de controle RS e E são ligados ao *port B* para configuração e seleção do estado do *display*.

TABELA 4.2 - Conexão do Display LCD

Display LCD		Kit RCM 4400W	
Pino	Sinal	Sinal	Nome
1	GND	GND	GND
2	Vcc	Vcc	Vcc
3	Vee	GND	GND
4	RS	Endereço	PA6
5	RW	GND	GND
6	E	Endereço	PA7
7	D0	Dados	GND
8	D1	Dados	GND
9	D2	Dados	GND
10	D3	Dados	GND
11	D4	Dados	PA0
12	D5	Dados	PA1
13	D6	Dados	PA2
14	D7	Dados	PA3
15	A	GND	GND
16	K	Vc	Vcc

FONTE: Autor . 2008

#### 4.1.3 – Kit RCM4400W

O kit RCM4400W é fornecido montado pelo fabricante, não sendo necessária a montagem. Tirando a interface de gravação o único conector de acesso as ferramentas e funções do kit é o conector J1 como demonstra a Figura 4.14.

Os módulos de acesso a memória *flash* e ao dispositivo *Wi-Fi* 802.11b são todos internos ao kit de desenvolvimento sendo gerenciados via software.

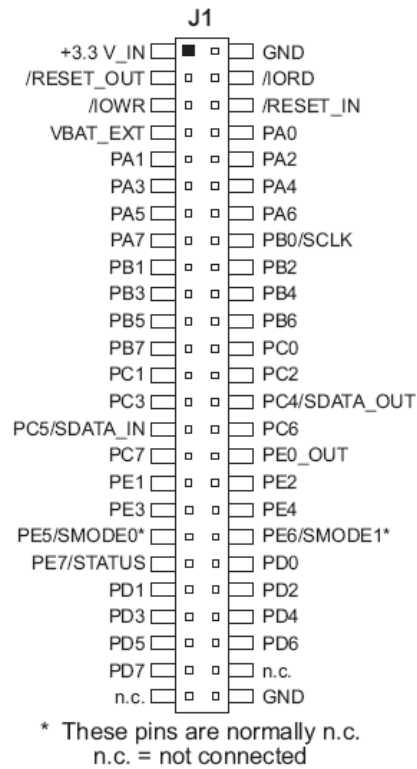


Fig. 4.13 – Conector J1 – E/S do kit RCM4400W

## 4.2 – Software

O projeto do software é baseado no software de gerenciamento, que controla o servidor de dados para o sistema e as funções do operador de sistema, e no firmware que controla todas as funções do hardware.

### 4.2.1 – Software Gerenciador

Através de suas funções o software gerenciador recebe os dados de cadastros necessários para o funcionamento do sistema, funciona como uma espécie de servidor para os terminais, atendendo a todas as solicitações dos terminais previamente cadastrados via rede, usando protocolo TCP/IP e também acessa o banco de dados. As principais funções do software gerenciador estão descritas a seguir nos casos de uso, figura 4.15, e diagramas de seqüência, mostrados entre as figuras 4.16 e 4.20.

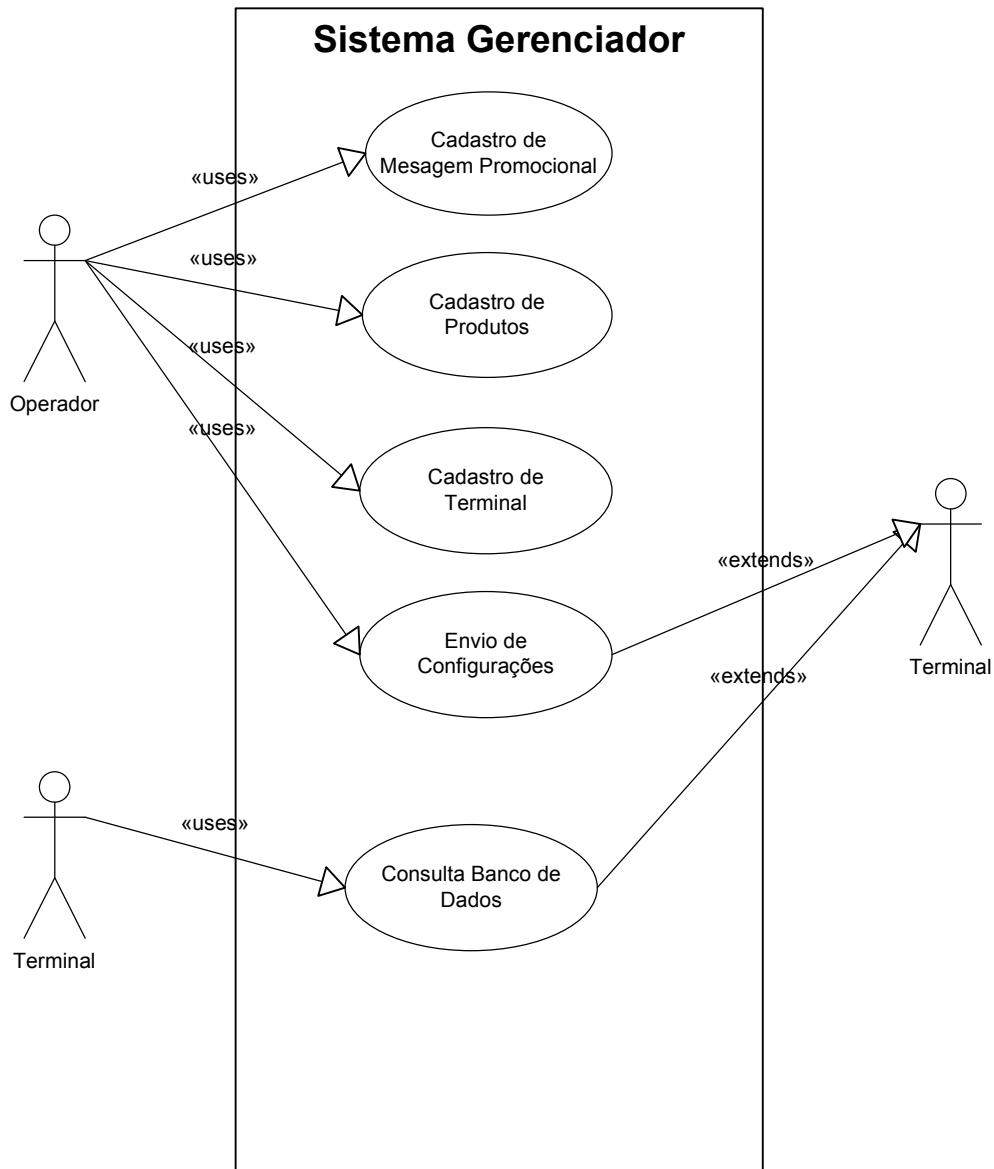


Fig. 4.14 – Casos de Uso do Sistema Gerenciador

Descrição dos casos de uso do sistema gerenciador:

**Caso de Uso:** Cadastro de mensagem promocional

**Ator:** Operador

**Descrição:** O ator Operador entra com os dados no sistema para cadastro da mensagem promocional

**Caso de Uso:** Cadastro de produto

**Ator:** Operador

**Descrição:** O ator Operador entra com os dados no sistema para cadastro do produto. Os dados são o código de barras, descrição do produto, preço e o código da mensagem promocional.

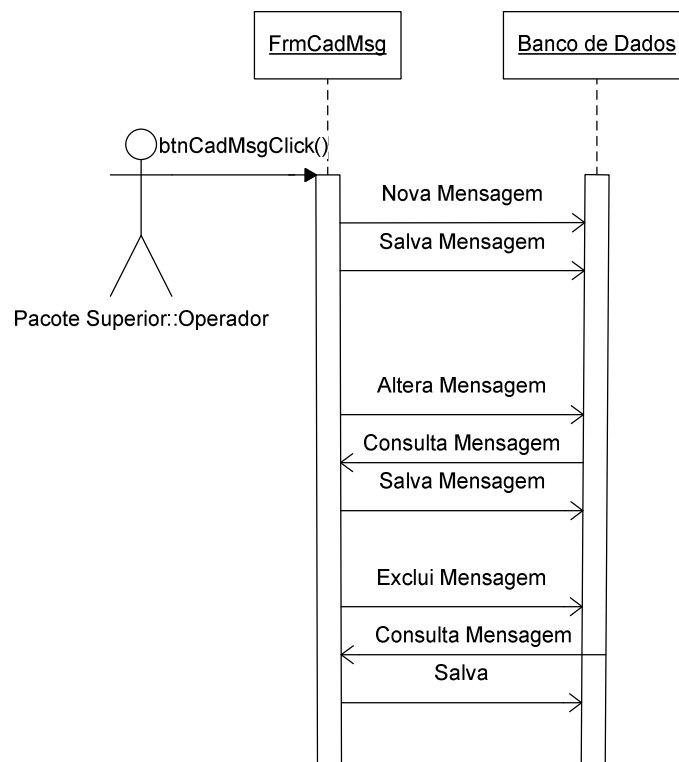
**Caso de Uso:** Cadastro de terminal**Ator:** Operador**Descrição:** O ator Operador entra com os dados do terminal. Os dados do terminal são, o número IP e a mensagem padrão.**Caso de Uso:** Envio de Configurações**Ator:** Operador**Descrição:** O ator Operador envia as configurações para o terminal. As configurações são, a mensagem padrão e data/hora.**Caso de Uso:** Consulta Banco de Dados**Ator:** Operador**Descrição:** O ator Terminal solicita uma consulta ao banco de dados usando como indexador o código de barras do produto. No caso de, produto ou terminal não cadastrados no banco de dados, o sistema retorna uma mensagem de erro padrão para cada situação.

Fig. 4.15 – Diagrama de sequência – Cadastro de Mensagem



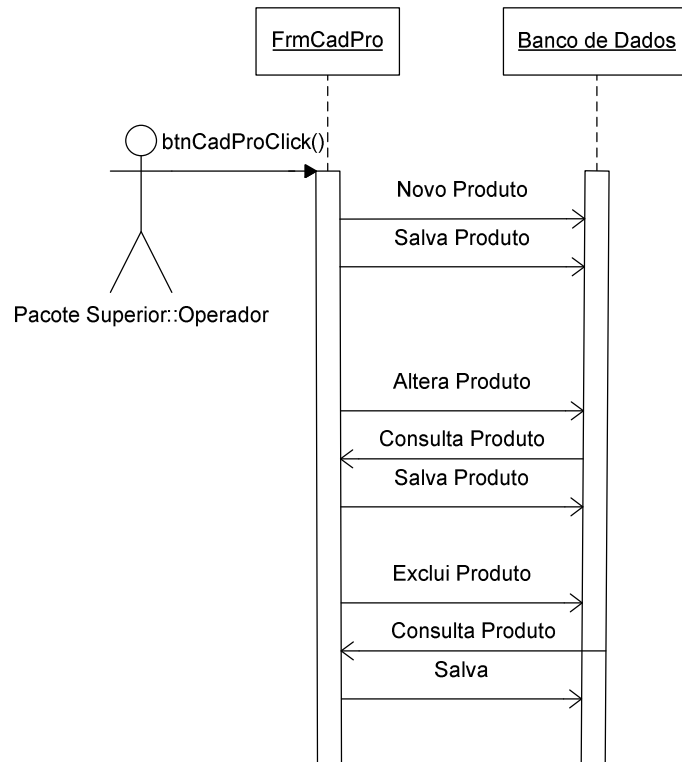


Fig. 4.16 – Diagrama de seqüência – Cadastro de Produtos

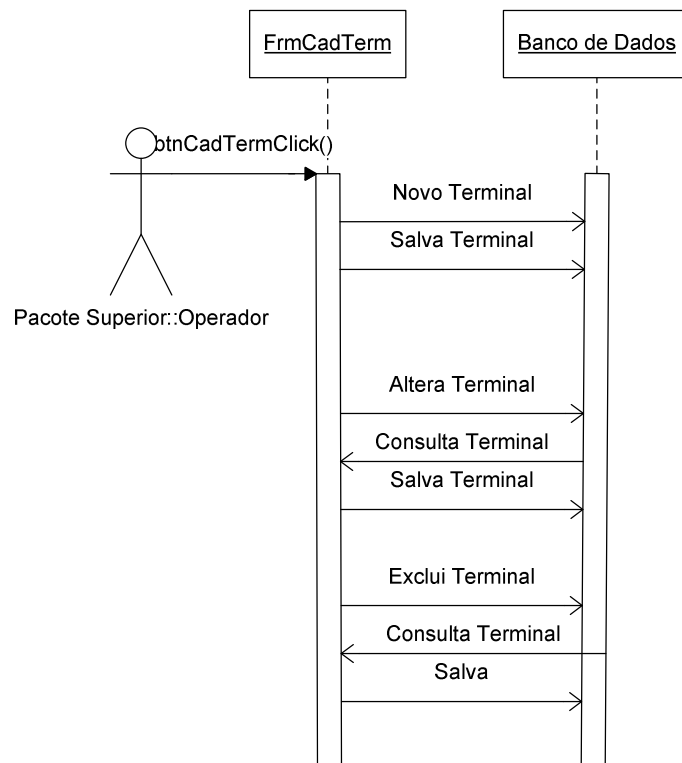


Fig. 4.17 – Diagrama de seqüência – Cadastro de Terminais

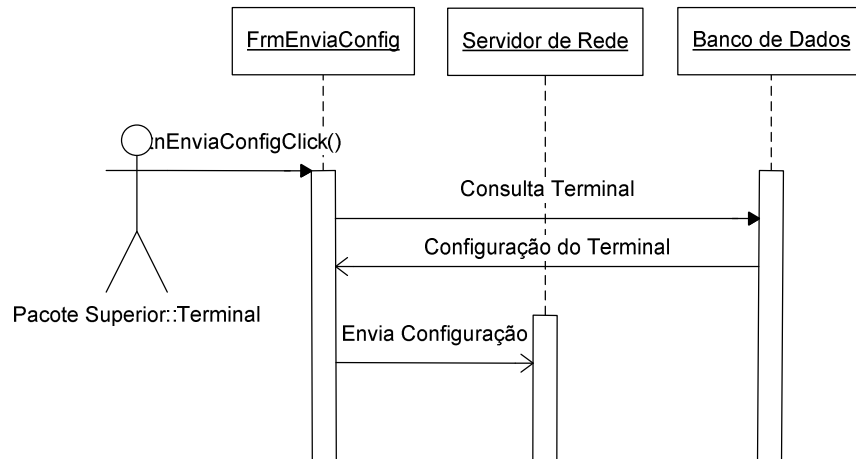


Fig. 4.18 – Diagrama de seqüência – Envia Configuração

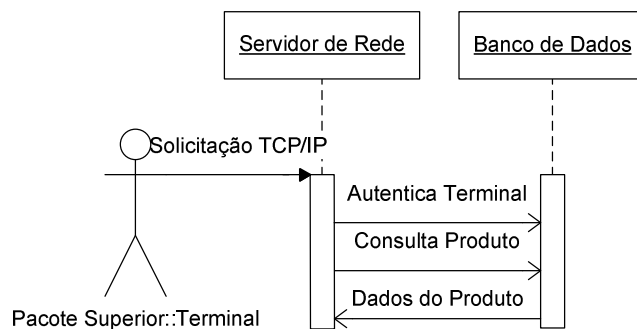


Fig. 4.19 – Diagrama de seqüência – Consulta Banco de Dados

#### 4.2.2 – Firmware

O sistema, após a inicialização das variáveis padrões alocadas na memória *flash*, entra em uma função de espera da leitura do código de barras, nesta função ele disponibiliza no display LCD a mensagem padrão e a hora.

Após a leitura do código de barras pelo leitor serial, o software envia o código via rede e aguarda o retorno do software de gerenciamento. Caso o terminal não esteja cadastrado ou a rede esteja desconectada, o terminal retorna uma mensagem de erro no display.

Caso o produto não esteja cadastrado o software também envia para o display uma mensagem de erro de cadastro. No caso normal onde o software gerenciador responde via rede a descrição, o preço e a mensagem promocional do produto o terminal disponibiliza no display

todos os dados, na seqüência: descrição e preço e logo após mensagem promocional. O *software* é responsável por todo o gerenciamento do hardware do sistema. Todo o funcionamento do sistema está descrito na Figura 4.21.

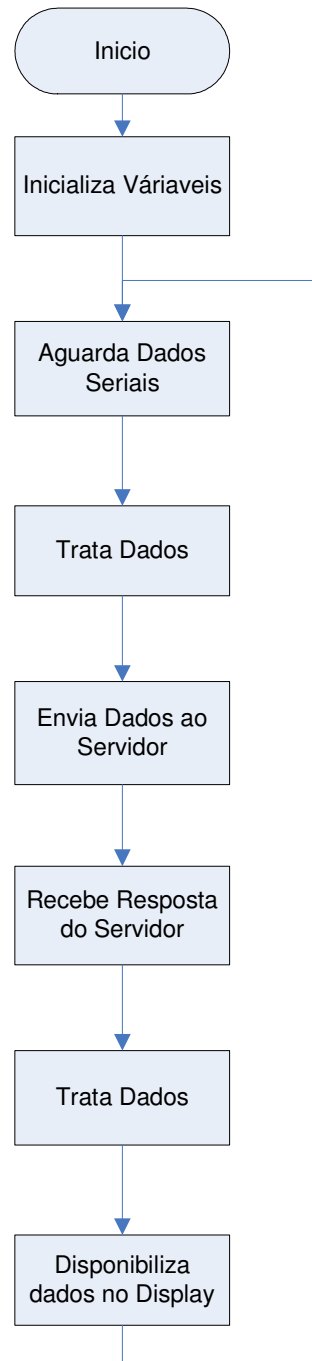


Fig. 4.20 – Fluxograma de funcionamento do *Firmware*

### 4.2.3 –Banco de Dados

Os modelos do banco de dados do sistema estão descritos nas Figuras 4.22 e 4.23.

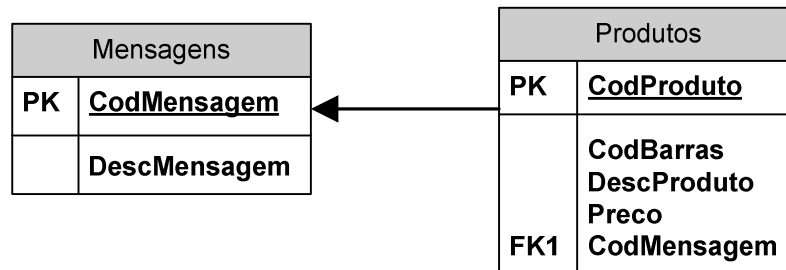


Fig. 4.21 – Modelo de relacionamento das Tabelas Mensagens e Produtos

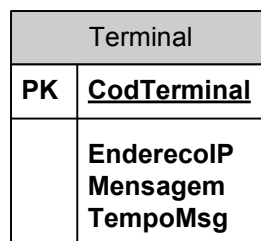


Fig. 4.22 – Tabela Terminal

A Figura 4.22 demonstra o modelo das tabelas Mensagens e Produtos. A tabela de Produtos contém todas as informações referentes ao cadastro dos produtos para serem mostradas no terminal, como a descrição do produto (DescProduto), o preço (Preço) e um relacionamento com a tabela Mensagens através do código da mesma (CodMensagem). A chave primária do produto é um código de seqüência gerado pelo sistema (CodProduto), porém a consulta ao banco que o terminal faz é através do código de barras, campo CódBarras. A tabela Mensagens contém apenas a informação da mensagem promocional que é apresentado pelo display do terminal (DescMensagem) e é indexada por um código seqüencial criado (CodMensagem) pelo sistema.

A Figura 4.23 é referente ao modelo da tabela Terminal que armazena os dados de configuração dos terminais remotos. A tabela tem as seguintes entradas: CódTerminal, que é o código seqüencial e indexador dos terminais. O EnderecoIP que armazena o número IP do terminal. O campo Mensagem que é a mensagem padrão de cada terminal. E o TempoMsg que armazena o tempo, em segundos, que as mensagens aparecem no display do terminal.

## CAPÍTULO 5 – VALIDAÇÃO E RESULTADOS

Para a validação do projeto primeiramente foram testados todos os módulos separadamente, com blocos de códigos independentes e depois o projeto como um todo. Também foram executados testes com um terminal de rede implementado em hardware e cinco outros terminais, simulados via software, conectados em um único servidor.

### 5.1 – Software Gerenciador

O software gerenciador recebeu a conexão simultânea de 6 terminais, com fluxo de informação para testar um situação real de todo os sistema.

#### 5.1.1 – Banco da Dados

A primeira etapa do desenvolvimento foi toda a estrutura da base de dados que foi testada e validada com o desenvolvimento de uma parte do Software Gerenciador que é responsável pelo cadastramento dos produtos, Figura 4.24, e das mensagens promocionais dos terminais de consulta.

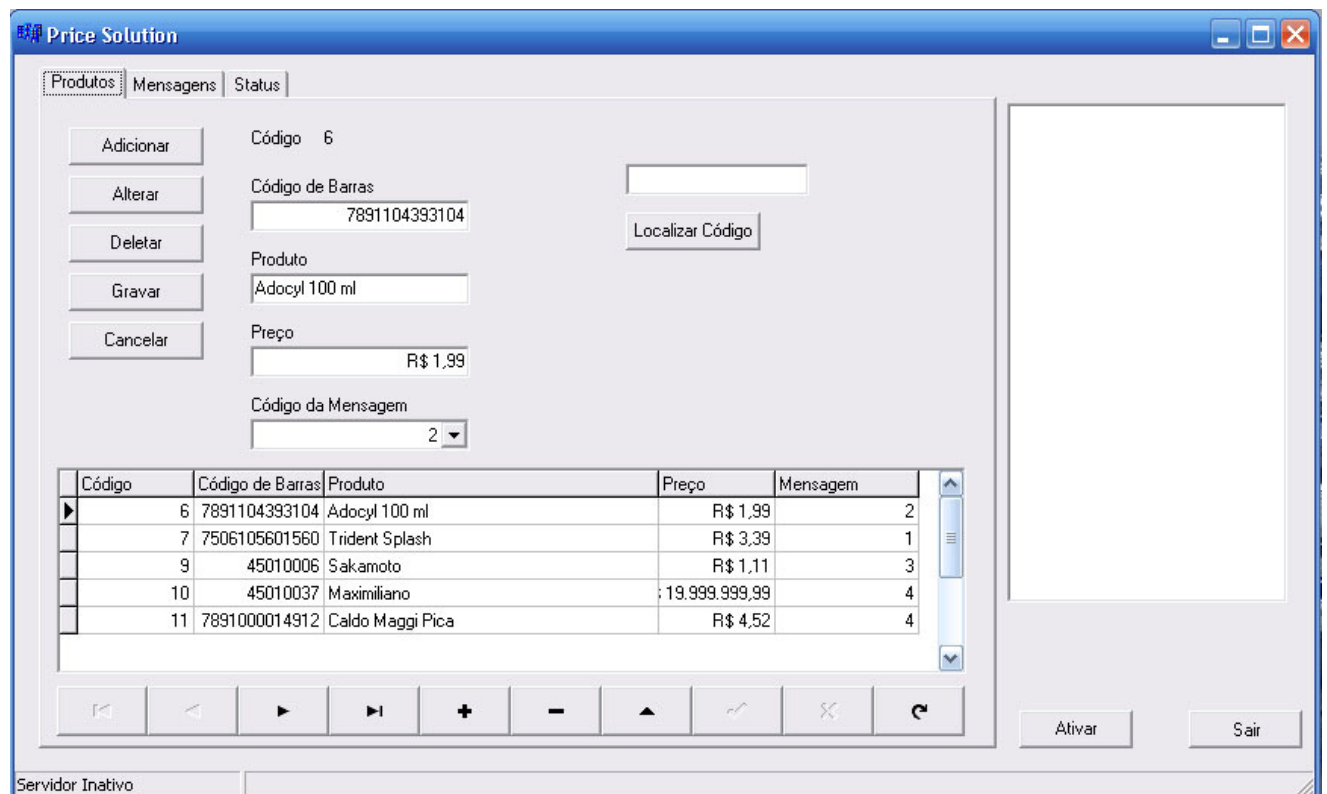


Fig. 4.23 – Tela de Cadastramento de Produtos

A validação foi o cadastramento de produtos e mensagens pelo software gerenciador e depois a verificação direto nas tabelas através do software *DataBase Desktop* fornecido pela

Borland junto com o pacote do *C++ Builder*. Também foram testadas todas rotinas de consulta na base de dados usando como índice de pesquisa o código de barras, já que o terminal envia apenas este dado para o software gerenciador. Todos os resultados obtidos na interface do projeto com a inserção e consulta de dados foram com várias inserções e deleções no banco de dados validando todo o processo sem erros.

### **5.1.2 – Interface Socket**

O software gerenciador trabalha como servidor de *socket* para o sistema. Todas as rotinas de desenvolvimento da interface de rede foram testadas em um primeiro momento com um software que simulava o terminal de consulta preços remoto e depois com o próprio terminal rodando direto no microprocessador.

Na aba “status”, do software gerenciador, existe uma caixa de diálogo que armazena todo o log de comunicação entre o servidor e o terminal apontando o IP, a mensagem trocada e a direção. Este log foi utilizado para testes durante o desenvolvimento do projeto e também através dele é possível o funcionamento perfeito do sistema e da comunicação socket.

Para solucionar possíveis problemas de sincronismo de dados entre o servidor e os terminais foram selecionadas mensagens pré-determinadas enviadas pelo terminal para o servidor identificando a rotina à ser executada pelo mesmo.

Todas as rotinas de comunicação possível foram executadas no sistema não apresentando nenhuma falha, inclusive o teste com 6 terminais simultaneamente.

## **5.2 – Hardware e Firmware**

O hardware e o firmware foram testados com leituras de dados e simulações de conexão e desconexões de rede.

### **5.2.1 – Interface de rede**

A interface de rede utilizada pelo módulo Rabbit RCM4400W é do tipo Wireless Lan. Através de bibliotecas disponibilizadas pelo fabricante do microcontrolador foi possível executar todos os comandos necessários a utilização da interface de rede, desde os comandos referentes a inicialização da interface Ethernet baseada em protocolo TCP/IP até a comunicação socket com o servidor.

Para solucionar erros de sincronismo entre o servidor e o terminal, foi implementado no firmware rotinas que enviam mensagens pré-determinadas ao servidor para solicitar dados diferentes.

No desenvolvimento do projeto a identificação da rede Wi-Fi (SSID) e o IP do servidor devem ser fixos.

### **5.2.2 – Display LCD**

A comunicação com o display LCD foi escrita com base em exemplos do próprio fabricante. Foram desenvolvidas rotinas para escrever os bytes em sequência formando os caracteres no display e depois rotinas que enviam diretamente strings e que selecionam a linha para escrita no visor.

O display funcionou perfeitamente durante os testes apresentando para os usuários do sistema as mensagens corretas.

### **5.1.2 – Leitor de Código de Barras**

O leitor de código de barras utilizado pelo sistema tem comunicação serial e o microcontrolador Rabbit 4400W trata a interface serial via buffers. Através de comandos podemos inicializar os parâmetros da comunicação serial e depois ler byte a byte os caracteres em formato ASCII disponibilizados pela pilha contida no buffer.

No firmware foi desenvolvida uma rotina para identificar a leitura completa do código de barras independente do tamanho, contanto que o último caracter enviado pelo leitor seja um “*enter*”, ou pela tabela ASCII o número 13.

## CAPÍTULO 6 – VIABILIDADE SÓCIO-ECONÔMICA

O projeto é completamente viável economicamente como demonstra a tabela 6.1 e realmente pode ser desenvolvido como uma alternativa de baixo custo para os produtos existentes no mercado. O custo de desenvolvimento do produto foi dentro do esperado como apresentado na tabela 6.2 e o custo que um produto deste nível chegaria ao consumidor final também é o esperado. A grande vantagem é a rede Wireless facilitando a instalação do produto e diminuindo o custo de infra-estrutura, já que o valor de mercado de um roteador de rede sem fio é muito baixo.

**TABELA 6.1 - Viabilidade Sócio-Econômica do projeto**

<b>Descrição</b>	<b>Valor</b>	<b>Quantidade</b>	<b>Encargos</b>	<b>Total S/ Impostos</b>	<b>Total C/ Impostos</b>
Mão de obra direta	R\$ 16,00	400	35%	R\$ 6.400,00	R\$ 8.640,00
Mão de obra do orientador	R\$ 5,90	48	35%	R\$ 283,20	R\$ 382,32
Mão de obra do coordenador do curso	R\$ 0,95	48	35%	R\$ 45,60	R\$ 61,56
Softwares Computador	R\$ 0,42	200	-	R\$ 83,33	R\$ 83,33
Software Proteus	R\$ 1,14	140	-	R\$ 159,09	R\$ 159,09
Fonte de alimentação simples	R\$ 0,06	100	-	R\$ 5,68	R\$ 5,68
Mesa Digital	R\$ 0,30	40	-	R\$ 12,12	R\$ 12,12
Multímetro	R\$ 0,03	20	-	R\$ 0,57	R\$ 0,57
Notebook	R\$ 3.000,00	1	-	R\$ 3.000,00	R\$ 3.000,00
Roteador Wi-Fi	R\$ 300,00	1	-	R\$ 300,00	R\$ 300,00
Microncontrolador	R\$ 500,00	1	-	R\$ 500,00	R\$ 500,00
Kit de Desenvolvimento	R\$ 500,00	1	-	R\$ 500,00	R\$ 500,00
Leitor de Código de Barras	R\$ 80,00	1	-	R\$ 80,00	R\$ 80,00
Display de Cristal Líquido	R\$ 35,00	1	-	R\$ 35,00	R\$ 35,00
Cristal frequência	R\$ 1,00	1	-	R\$ 1,00	R\$ 1,00
ProtoBoard	R\$ 70,00	1	-	R\$ 70,00	R\$ 70,00
Componentes Diversos	R\$ 50,00	1	-	R\$ 50,00	R\$ 50,00
Instalação Física	R\$ 8,00	4	-	R\$ 32,00	R\$ 32,00
<b>Total do Projeto</b>					<b>R\$ 13.912,68</b>

Fonte: Autor, 2008

**TABELA 6.2 - Custos de desenvolvimento do produto**

<b>Descrição</b>	<b>Valor</b>	<b>Quantidade</b>	<b>Encargos</b>	<b>Total S/ Impostos</b>	<b>Total C/ Impostos</b>
Mão de obra direta	R\$ 16,00	10	25%	R\$ 160,00	R\$ 200,00
Microncontrolador	R\$ 400,00	1	25%	R\$ 400,00	R\$ 500,00
Placa de Conexões	R\$ 100,00	1	25%	R\$ 100,00	R\$ 125,00
Display de Cristal Líquido	R\$ 25,00	1	25%	R\$ 25,00	R\$ 31,25
Leitor de Código de Barras	R\$ 60,00	1	25%	R\$ 60,00	R\$ 75,00
Fonte de alimentação simples	R\$ 15,00	1	25%	R\$ 15,00	R\$ 18,75
Componentes Diversos	R\$ 100,00	1	25%	R\$ 100,00	R\$ 125,00
Gabinete	R\$ 50,00	1	25%	R\$ 50,00	R\$ 62,50
Cd com software e manuais	R\$ 5,00	1	25%	R\$ 5,00	R\$ 6,25
<b>Custo Total</b>					<b>R\$ 1.143,75</b>

Fonte: Autor, 2008.



## CAPÍTULO 7 - CONCLUSÃO

O terminal de consulta de preços remoto, foi desenvolvido com um módulo de hardware composto por um leitor de código de barras, um display de LCD e o kit do microcontrolador Rabbit, e o módulo de software composto pelo servidor de rede e o módulo de cadastro.

O desenvolvimento do projeto foi simples. As principais rotinas do software gerenciador puderam ser implementadas com bibliotecas e classes padrões do Borland C++ Builder 6. E o firmware teve um desenvolvimento sem grandes problemas facilitado pela agradável estrutura de desenvolvimento do microcontrolador Rabbit e da intuitiva linguagem de programação e bibliotecas úteis do Dinamic C.

O kit de desenvolvimento Rabbit RCM4400W foi uma grata surpresa pela facilidade de desenvolvimento, pela quantidade de bibliotecas úteis e pela forma de tratamento de entradas e saídas através dos *ports*.

O terminal de consulta de preços apresentado cumpre todas as especificações apresentadas e identificadas na proposta do projeto. Porém durante o desenvolvimento surgiram algumas formas de melhoria que acabaram não sendo implementadas. A principal é a configuração local e via servidor dos parâmetros de rede do terminal , como *SSID* da rede, IP local e IP do servidor.

## CAPÍTULO 8 - REFERÊNCIAS BIBLIOGRÁFICAS

- [1] RUBINSTEIN, M. G. ; REZENDE, J. R. Qualidade de Serviço em Redes 802.11. Universidade Federal do Rio da Janeiro. Rio de Janeiro.
- [2] BARCELOS, J.P.M.; GONÇALVEZ, R.G. ; ALVES Jr., N.. Nota Técnica - O padrão 802.11. Centro Brasileiro de Pesquisas Físicas – CBPF.
- [3] TANENBAUM, ANDREW S. Redes de Computadores. 4ª ed. São Paulo: Campus, 2003.
- [4] RABBIT, Semiconductor Inc. RabbitCore RCM4400W OEM User's Manual.
- [5] LEE, R.C. ; TEPFENHART, W.M. UML e C++ - Guia Prático de Desenvolvimento Orientado a Objeto. 1ª. Ed. São Paulo. Makron Books, 2001
- [6] UNIVERSIDADE FEDERAL DO PARANÁ. Normas para apresentação de documentos científicos. Editora UFPR, volume 6, 2003.
- [7] COMER, DOUGLAS E. Redes de Computadores e Internet. 2ª. Ed. Porto Alegre. Bookman, 2001
- [8] SCHMIDT, DOUGLAS C.; HUSTON, STEPHEN D. C++ network programming. Volume 1. Boston , 2002.
- [9] SCHILDT, HERBET; GUNTLE, GREG; Borland C++ Builder, A referencia completa. 1ª. Ed. Rio de Janeiro. Editora Campus, 2001
- [10] AXELSON, JAN. Embedded Ethernet and Internet Complete 1ª. Ed., Madison, WI, Ed. LVR 2003.