

RESUMO

O projeto Zéfiros tem como principal objetivo responder à crescente necessidade de melhor eficiência no atendimento de restaurantes. Focando na automação de processos que mais demandam tempo para serem realizados, o sistema desenvolvido por seus idealizadores auxilia em 3 processos em especial: requisição de atendimento por parte de uma mesa, tomada de pedido pelo garçom e envio do mesmo à cozinha.

O sistema completo é composto por transmissores e um receptor de rádio frequência, de um circuito lógico com micro-controlador e outros componentes eletrônicos, e também de uma tela de LEDs. Adicionalmente, integram-se ao sistema um servidor central de pedidos e um software para PDAs, desenvolvidos de forma otimizada para suprir os requisitos do projeto.

A implantação do sistema proposto permite não só maior rapidez no atendimento mas também redução de custos.

ABSTRACT

The main objective of the Zéfiro Project is to answer to the growing needs of efficiency in restaurant services. Its focus is directed to 3 important time-consuming processes, namely: Calling the waiter, order-taking and order sending to the kitchen.

The whole system consists of transmitters and a Radio Frequency Receiver, a logic circuit having a microcontroller and others electronic components and also a LED Display. Additionally, these components are integrated with a dedicated server and PDAs, both powered by software developed for this project.

The system proposed allows both a faster service and a reduction of the Total Cost of Ownership (TCO).

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama esquemático do sistema desenvolvido.....	13
Figura 2 – Estrutura simplificada do hardware	17
Figura 3 – Receptor Matrix®.....	18
Figura 4 – Botões para programação do Receptor Matrix®.....	19
Figura 5 – Conjunto de 4 <i>displays</i> do Receptor Matrix®.....	19
Figura 6 – Modulação ASK	21
Figura 7 – Esquema do bloco Decodificador e de Controle de Comunicação ..	22
Figura 8 – Diagrama de blocos da arquitetura interna do LM7805	24
Figura 9 – Configuração do LM7805 para atuar como regulador de tensão	24
Figura 10 – Mapeamento dos segmentos do display do Receptor Matrix®	25
Figura 11 – Chave usando transistor bipolar NPN.....	26
Figura 12 – Circuito completo para conformação do sinal.....	29
Figura 13 – MAX232CPE: encapsulamento e arquitetura	31
Figura 14 – Pinagem típica de um microcontrolador 8051	32
Figura 15 – Divisão da memória RAM do 8051	33
Figura 16 – Estrutura interna da RAM de um 8051	33
Figura 17 – Ligações entre o 8051 e pinos dos displays do receptor.....	39
Figura 18 – Fluxo geral do programa do 8051	40
Figura 19 – Mapeamento do display de 7 segmentos.....	44
Figura 20 – Conexões dos portos com os sinais conformados	44
Figura 21 – Exemplo de transmissão serial assíncrona (RS232).....	46
Figura 22 – Leiaute do módulo completo de decodificação e comunicação	48
Figura 23 – Circuito esquemático do módulo de decodificação e comunicação.....	49
Figura 24 – Diagrama de blocos do Controle de Tela.....	50
Figura 25 – Estrutura interna do DM74LS164	51
Figura 26 – Encapsulamento do DM74LS164	51
Figura 27 – Estrutura interna do DM74LS47	52
Figura 28 – Relações de conversão do DM74LS47	52
Figura 29 – Encapsulamento do DM74LS47	53
Figura 30 – Gráfico da variação do sinal recebido em função da distância.....	90
Figura 31 – Plantas do local dos testes e posicionamento dos transmissores e do receptor.....	91
Figura 32 – Diagrama de classes do software de gerenciamento instalado no servidor do sistema	94
Figura 33 – Diagrama de navegação do usuário	95
Figura 34 – Diagrama de interface	95
Figura 35 – Diagrama de organização de pacotes	96
Figura 36 – Screenshots do programa implantado nos PDAs	97

LISTA DE TABELAS

Tabela 1 – Relação pinos/segmentos para o display do Receptor Matrix®	25
Tabela 2 – Registradores de funções especiais	34
Tabela 3 – Estrutura do registrador PSW	36
Tabela 4 – Seleção de bancos de registradores em função de RS0 e RS1	37
Tabela 5 – Tabela para decodificação dos displays do receptor	45
Tabela 6 – Potência do transmissor em função da distância	89
Tabela 7 – Valores calculados para α	89
Tabela 8 – Valores calculados para a constante A	89

LISTA DE SIGLAS

ASK	Amplitude Shift Keying
CAD	Computer Auxiliated Design
CI	Circuito Integrado
CU	Caso de Uso
CMOS	Complementary Metal Oxide Silicon
CRUD	Create, Read, Update and Delete
EAGLE	Easily Applicable Graphical Layout Editor
EIA	Electronic Industry Association
FIFO	First In First Out
I/O	Input/Output
ISM	Industrial, Scientific and Medical
LAN	Local Area Network
LED	Light Emissor Diode
PC	Personal Computer
PDA	Portable Device Advanced
RAM	Randomic Acess Memory
RF	Rádio Freqüência
ROM	Read Only Memory
RXD	Reception
SPI	Serial Peripheral Interface Bus
SQL	Structured Query Language
TCP	Transmission Control Protocol
TIA	Telecommunications Industry Association
TTL	Transistor-Transistor Logic
TXD	Transmission
ULA	Unidade Lógica Aritmética
UML	Unified Modeling Language
XML	Extensible Markup Language
Wi-Fi	Wireless Fidelity

SUMÁRIO

RESUMO	1
ABSTRACT	2
LISTA DE ILUSTRAÇÕES	3
LISTA DE TABELAS	4
LISTA DE SIGLAS	5
SUMÁRIO.....	7
1 Introdução.....	9
1.1 Objetivos.....	10
1.2 Justificativas	10
1.3 Organização do Trabalho.....	11
2 Descrição da Proposta	13
3 Hardware	17
3.1 Transmissor e Receptor RF	17
3.2 Decodificador e Controle de Comunicação.....	21
3.2.1 Regulador de Tensão 12V – 5V	23
3.2.2 Conformação de Sinal	24
3.2.3 Regulador de Tensão para Comunicação Serial	29
3.2.4 Microcontrolador.....	31
b. Memória e Registradores.....	32
c. Projeto Lógico do Firmware.....	38
d. Entradas e Saídas	38
e. Fluxo Geral.....	39
f. Estrutura de dados	40
g. Configuração dos Registradores.....	42
h. Varredura do Display.....	43
i. Transmissão Serial – RS232.....	45
3.2.5 EAGLE.....	46
3.2.6 Circuito Completo	47
3.3 Controle de Tela/Display.....	50
3.3.1 Registradores de deslocamento	50
3.3.2 Decodificadores BCD para 7 segmentos.....	52
3.3.3 Displays de 7 segmentos	53
4 Software.....	54
4.1 Tecnologias Utilizadas	54
4.2 Descrição completa dos sistemas	55
4.2.1 Sistema Servidor.....	55
4.2.2 Sistema Cliente.....	59
4.3 Descrição formal do Software	63
4.3.1 Atores	63
4.3.2 Glossário.....	63
4.3.3 Relatório de Revisão dos Casos de Uso	65
4.3.4 Casos de Uso	65
I. Sistema Cliente: Palm.....	65
II. Sistema Servidor : PC.....	66
4.3.5 Detalhamento dos casos de uso	67
a. Sincronização dos dados.....	67

b.	Edição de preferências – Sistema Cliente	67
c.	<CANCELADO> Edição de sua ordem preferencial de exibição da lista de itens	68
d.	Visualizar detalhes do Produto/Pedido/Mensagem	68
e.	<CANCELADO> Adição de Produto a grupo personalizado	68
f.	<CANCELADO> Remoção de Produto de grupo personalizado	70
g.	Criação de Pedido e adição de Item a ele	71
h.	Edição de Itens em Pedido	71
i.	Envio de pedido	72
j.	<REMOVIDO: VER CU-4> Visualização de detalhes dos últimos pedidos enviados	73
k.	Requisição da próxima mesa da fila	74
l.	Incl: Validação e Interpretação de mensagem recebida por Sistema Servidor	75
m.	Edição de preferências – Sistema Servidor	75
n.	Inclusão/Edição/Exclusão de usuário autorizado	76
o.	Inclusão/Edição/Exclusão de dados de produtos	76
p.	Inclusão/Edição/Exclusão de dados de performance	76
q.	Análise/Exportação de dados de estoque	77
r.	Análise/Exportação de dados de performance	77
s.	Informar que pedido está pronto para ser servido	79
t.	Mover pedido de “Grupo Aberto” a “Grupo Pronto” e vice-versa	80
u.	Envio de mensagem a garçom	80
v.	Criação de pedido e adição de itens a ele	80
w.	Visualização e Edição de pedido	80
x.	Executar manutenção da base de dados	82
y.	Colocar mesa na fila FIFO	82
5	Metodologia	84
5.1	Hardware	84
5.2	Pedidos Wi-Fi	85
6	Resultados	88
6.1	Transmissor e Receptor RF	88
6.2	Módulos de hardware	91
6.3	Software	92
7	Discussão	98
8	Conclusão	100
9	Bibliografia	101

1 Introdução

Cada vez mais bares e restaurantes buscam maneiras de otimizar o atendimento de seus clientes, com o uso de sistemas cada vez mais sofisticados. A busca por fatores de qualidade e por maior produtividade torna-se cada vez mais importante dentro destes estabelecimentos. Estes objetivos, por sua vez, podem ser conflitantes, uma vez que, a agilidade do atendimento e a qualidade do mesmo muitas vezes não são compatíveis.

Estes conflitos refletem a complexidade envolvida na solução de um sistema que vise otimizar o atendimento e ao mesmo tempo melhorar o índice de satisfação dos clientes. Comumente restaurantes que usam sistemas de automatização do atendimento tem uma redução na satisfação do cliente, causada pela falta de interação do atendente com os clientes, pois na implantação do sistema busca-se apenas agilizar o atendimento para reduzir o tempo de permanência dos clientes e também reduzir o quadro de funcionários.

A partir de 2001 uma série de sistemas foi implantada em bares e restaurantes do Brasil. Todos estes sistemas baseiam-se em redes sem fio e PDA's, que reduzem o tempo de processamento dos pedidos. Durante o decorrer dos anos esse modelo baseado apenas em PDA's não sofreu nenhuma evolução, o que fez com que o peso da vantagem tecnológica destes sistemas decaísse.

Faz-se necessário reformular este tipo de sistema e buscar uma solução que baseada em soluções anteriormente desenvolvidas possa melhorar a qualidade do atendimento e, se possível, otimizar ainda mais o atendimento. Neste contexto imaginamos um sistema em que possamos saber onde estão as pessoas que necessitam ser atendidas e não mais esperar que os atendentes encontrem-nas visualmente e também saber a ordem em que estas pessoas necessitam de atendimento. Com uma posição e uma prioridade podemos estabelecer uma logística de atendimento ótima, que visa principalmente a satisfação dos clientes.

1.1 Objetivos

O objetivo principal deste trabalho é a implementação de um Sistema de Automatização do Atendimento em ambientes como bares e restaurantes. Este sistema será composto por duas redes sem fio, a primeira para sustentar a comunicação entre PDA's e um servidor, e a segunda para dar substrato à sinalização de pedidos de atendimento. Todas as informações serão processadas de forma centralizada por um servidor de dados.

Os objetivos secundários são: consolidação do aprendizado de formulação de sistemas eletrônicos, desenvolvimento das habilidades em engenharia de software e desenvolvimento de um algoritmo de logística de alto nível para o atendimento.

1.2 Justificativas

Segundo estatísticas da Prefeitura Municipal de São Paulo, existem na cidade cerca de 12500 restaurantes e 15000 bares. Considerando 10% dos restaurantes e 5% dos bares como mercado potencial, temos cerca de 1250 restaurantes e 750 bares. Isso apenas na cidade de São Paulo. Temos ainda outras cidades da Grande São Paulo, como Guarulhos, Osasco, ABC, e ainda outras cidades do interior e litoral de São Paulo, como a Região Metropolitana de Campinas, Sorocaba, Santos, que tem um comércio muito ativo. Devemos considerar também as cidades turísticas como Campos do Jordão, onde os restaurantes fazem parte do roteiro turístico. Temos, portanto, um mercado muito grande que ainda não foi explorado.

Percebemos que cada vez mais restaurantes estão começando a utilizar o sistema com PDA. Isso se deve, principalmente, a uma cultura que visa, cada vez mais, otimizar os processos por meio da automação.

Através de pesquisas com alguns donos de restaurantes, vimos que muitos já pesquisaram ou já pensaram em colocar algum sistema de automação.

Em meio a esse contexto, nosso projeto visa, além de oferecer este sistema de PDA, automatizar mais um processo, que é o chamado do garçom pelo cliente. Isso permitirá, por exemplo, que num restaurante *fast-food*, o garçom não precise ficar esperando que o cliente escolha um prato, podendo atender outras mesas enquanto isso. E quando o cliente escolher o prato, aperta o botão que fica em cima da mesa e o garçom é chamado. Considerando que a escolha de um prato demora, em média, 2 a 3 minutos, no caso de um restaurante que atenda 100 grupos de clientes por dia, o mesmo economizará cerca de 200 a 300 minutos-garçom por dia. O que pode ser mais significativo se houver clientes esperando por mesa.

Estes dados indicam que este projeto tem grandes chances econômicas de se suceder bem, caso bem trabalhado.

1.3 Organização do Trabalho

O capítulo 3 apresenta o sistema desenvolvido, apresentando sua arquitetura e suas divisões fundamentais que nortearam a divisão do estudo em duas frentes, sistemas eletrônicos (*hardware*) e programas (*software*). Neste capítulo também é apresentado o cronograma das atividades do projeto.

No capítulo 4, é abordado o sistema sobre uma visão de sistemas eletrônicos, apresentando a teoria pertinente, os circuitos elaborados e eventuais testes realizados para os mesmos.

Em seguida, no capítulo 5, as especificações dos programas são apresentadas sobre a visão da Engenharia de Software. Neste capítulo também são mostrados alguns resultados.

A metodologia de desenvolvimento usada ao longo do projeto é apresentada no capítulo 6, onde são justificadas as escolhas tecnológicas envolvidas no mesmo, bem como os recursos e infra-estrutura necessários

para o seu desenvolvimento. Por fim, no capítulo 7, são apresentadas as conclusões finais do sistema desenvolvido.

2 Descrição da Proposta

O sistema aqui proposto deverá ser capaz de controlar o fluxo de informações entre PDA's, dispositivo sinalizador de pedidos, servidor e periféricos. Este deve fornecer a indicação de qual, dentre as mesas e/ou clientes, deverá ser atendido primeiramente, de acordo com a ordem de solicitação ou outro critério de prioridade que deve ser estabelecido no programa de controle.

O sistema simplificado pode ser visto na figura 1:

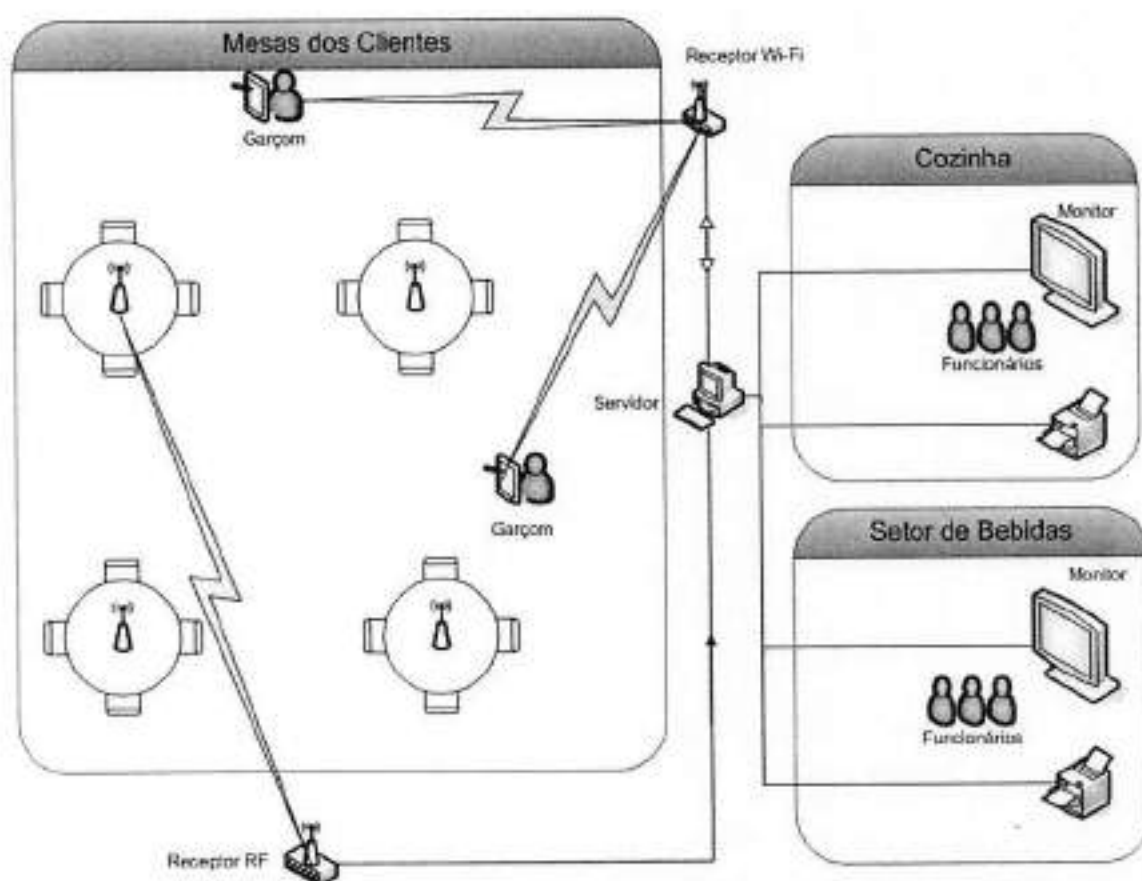


Figura 1 – Diagrama esquemático do sistema desenvolvido

A figura 1 nos fornece uma visão geral do sistema, até mesmo de uma forma simplificada, cabendo explicações sobre cada elemento de forma a

elucidar e deixar claro o que o sistema deverá fazer. A descrição básica de cada elemento segue abaixo:

1. PDA's: Utilizados pelos funcionários do estabelecimento para comunicação sem fio com o servidor (envio de pedidos e mensagens do tipo "Qual a próxima mesa da fila?", recepção de mensagens do tipo "Pedido pronto para ser servido!" e "Esta é a próxima mesa na fila!"). Para utilizá-los o garçom deve fornecer um nome de usuário e uma senha, que serão utilizados em cada mensagem enviada ao servidor. Desta maneira, cada usuário poderá ser facilmente identificado no momento da comunicação, permitindo análise de performance dos mesmos e também garantindo maior segurança ao sistema. Além disto, os *Palms* devem conectar-se ao servidor sempre que o Banco de Dados dos produtos for atualizado, de forma a garantir a integridade do sistema.
2. Roteador Sem Fio: Responsável pela comunicação entre os PDA's e o Servidor Central da LAN *Wi-fi*.
3. Emissor RF: Aparelho localizado no centro de cada mesa, ativado ao pressionar-se um botão. Estes dispositivos apenas enviam um sinal que identifica de maneira única a mesa, solicitando atendimento: não existe recepção por parte deste aparelho. Existem diversas soluções baratas e suficientemente robustas que implementam funcionalidades semelhantes: como exemplo, podemos citar aparelhos controles de acesso a edifícios (como controles remotos de portões eletrônicos). A nomenclatura do emissor RF foi utilizada apenas para não utilizar uma nomenclatura semelhante ao da roteador, já que tanto o roteador quanto este emissor são aparelhos que se utilizam de rádio freqüências para fazer suas comunicações.
4. Receptor RF: Responsável pela recepção dos sinais enviados pelos emissores RF, ou seja, dos chamados dos clientes.

5. Servidor: *Desktop* responsável pelo processamento das mensagens enviadas pelos PDAs e também pela gestão da fila de atendimento das mesas. Além disto, deve gerir a base de dados do sistema, contendo os itens servidos pelo restaurante e também o histórico de pedidos (assim, podem ser feitas avaliações de performance de garçons, identificação de produtos com grande saída, combinações comuns, etc.). O software presente no mesmo será detalhado mais adiante.
6. Display: Indicará as mesas presentes na fila, por ordem de chegada (FIFO).
7. Monitores/Impressoras: Os pedidos enviados à cozinha e ao setor de bebidas são exibidos em um display de leds, um monitor comum ou através da impressão do pedido em uma pequena impressora, ou até mesmo uma solução híbrida entre as anteriores (o objetivo é contemplar as três implementações, dando prioridade à que se mostrar mais popular entre os possíveis compradores do sistema).

A interface do Palm e do usuário devem ser semelhantes, porém possuir algumas funcionalidades distintas.

1. Palms:

Sendo cada PDA destinado a um único garçom, o sistema de tomada de pedidos só pode ser acessado através do login do garçom no aplicativo. Ao mesmo tempo, é estabelecida uma conexão com o servidor central, que permite acesso ao Palm com o nome de usuário em questão. Algumas listas padrão de produtos (por exemplo, uma divisão entre pratos quentes e bebidas) e cliques na tela são a base para criar um pedido.

Além desta funcionalidade, o garçom pode obter do servidor, sob demanda, o número da próxima mesa presente na fila (o servidor a retira da fila em seguida, não permitindo que o Palm em questão requisiute uma nova mesa pelos próximos segundos)

A ligação unívoca entre Palm e usuário, discutida anteriormente, trás algumas vantagens associadas à personalização do ambiente e da

aplicação (além de criar uma clara necessidade por parte do usuário de cuidar bem da sua máquina)

- O garçom pode definir como prefere visualizar os dados da sua base no seu Palm através do estabelecimento de uma "ordem preferida" de produtos, ou seja, uma tabela de prioridades que é armazenada em seu próprio Palm.
- O garçom pode também criar e editar uma lista de produtos personalizada, onde são mostrados quaisquer produtos que o garçom desejar (desde que presentes na base de dados, obviamente).

Métodos como clicar em ou arrastar ícones serão possíveis para a execução da maioria das tarefas no aparelho, de forma a facilitar o uso da ferramenta.

2. Servidor:

Além das funcionalidades automáticas (sob demanda), como as respostas às mensagens enviadas pelos Palms ou pelo emissor RF, o servidor também possui uma interface gráfica com o usuário do desktop. Este último pode ser um administrador do sistema, que pode acessar todas as funcionalidades, ou então um usuário com menores direitos de acesso.

Um administrador pode, após login, editar as bases de dados de produtos (incluindo a ordem de exibição padrão, válida para todo Palm que não a tiver editado localmente), usuários e performance. Os dados desta última podem ser analisados de forma amigável, através gráficos e cálculos estatísticos automáticos.

Além destas funcionalidades administrativas, temos também outras mais funcionais, como o envio de mensagens do tipo "Pedido da mesa XX pronto para ser servido!" e também outras mensagens para chamar o garçom à cozinha em caso de necessidade. Adicionalmente, será possível gerar um pedido diretamente da cozinha, uma funcionalidade que pode ser útil como backup dos Palms (provavelmente não utilizada em operação normal).

3 Hardware

Este capítulo tem como objetivo detalhar a implementação de hardware e seus principais blocos estruturais e componentes.

A estrutura do hardware é basicamente composta por 3 módulos, Receptor RF, Transmissor RF, Controle de Decodificação e Comunicação, e Controle de Tela/Display como mostra a figura 2.

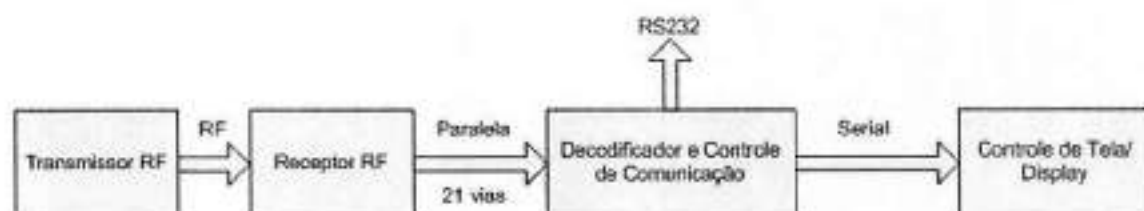


Figura 2 – Estrutura simplificada do hardware

O diagrama acima mostra de forma simplificada cada bloco do sistema. Cada um dos blocos será detalhado a seguir para elucidar completamente o hardware e seus componentes.

3.1 Transmissor e Receptor RF

Este bloco funcional é composto por um Receptor de RF e diversos transmissores, um para cada mesa.

O receptor deve ser capaz de receber os sinais emitidos pelos transmissores a distâncias comparáveis às dimensões físicas de bares e restaurantes. Este também deve ser eficaz mesmo na presença de diversos obstáculos, como, por exemplo, pessoas, paredes, janelas, entre outros.

Os transmissores devem possuir um baixo custo, pois o número de transmissores pode chegar a uma centena, e baixo consumo de potência, de forma que as baterias tenham uma durabilidade considerável. A aplicação que sugerimos não tem uma dependência fundamental de segurança na

transmissão e recepção de dados, sendo assim, consideramos este como um requisito secundário.

Muitos candidatos à solução foram pesquisados e descartados durante o andamento deste projeto, como foi o caso de uma solução que seria implementada usando a tecnologia ZigBee® [8][9]. Esta por ser uma tecnologia emergente, ainda apresenta um custo muito alto para os padrões deste projeto e de outros projetos que procuram algum tipo de comunicação sem fio para seus dados, como é o caso da automação residencial e predial.

Dentre todas as soluções pesquisadas e sugeridas a que atendeu os requisitos expostos anteriormente foi o Receptor Matrix, da PPA Automação, figura 3.

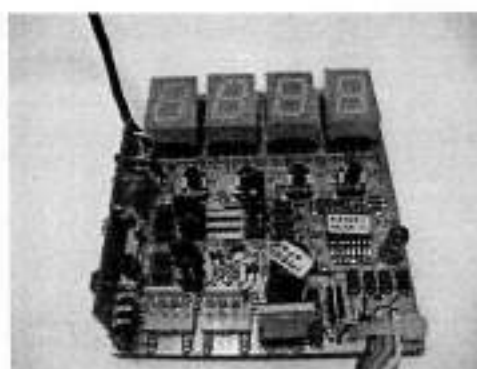


Figura 3 – Receptor Matrix®

O Receptor Matrix ® pode trabalhar até com 1008 transmissores, sendo que para que estes sejam recebidos corretamente deve haver uma pré-programação do receptor, que chamaremos daqui em diante de cadastro.

O cadastro é feito na forma de captura, onde o código que é recebido pela antena fica armazenado temporariamente e um número de identificação deve ser atribuído a este código, antes que outro código seja recebido pelo receptor. Todo este processo é feito através de uma interface de quatro botões, figura 4. Os botões têm as funções de incremento, decremento, confirmação e cancelamento de operações de programação e cadastramento.

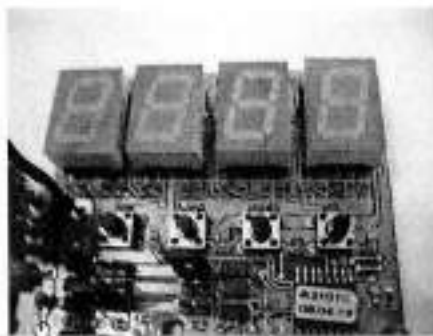


Figura 4 – Botões para programação do Receptor Matrix®

Quando um transmissor cadastrado é acionado e seu código recebido corretamente pelo receptor, seu número de identificação é exibido no *display*, que é formado por 4 *displays* de 8 segmentos¹, CTK-D168BK. Dois portos deste receptor também são acionados quando um controle cadastrado efetua a transmissão, sendo que estes portos têm como finalidade acionar ou desativar outros circuitos. Este tipo de interface é muito útil em sistemas de automação predial, onde esta saída é usada para acionar motores ou circuitos de alarmes, porém para nossos objetivos esta funcionalidade não tem um uso que possa ser visualizado neste ponto de desenvolvimento do sistema proposto.



Figura 5 – Conjunto de 4 *displays* do Receptor Matrix®

Uma grande desvantagem deste receptor é a ausência de interface de comunicação com outros periféricos, por onde o número do transmissor acionado ou outras informações fosse enviado. Sem este tipo de comunicação não podemos obter a informação que necessitamos. Tendo em vista esta deficiência, foi necessária a criação de um módulo que pudesse extrair

¹ Displays convencionais possuem 7 segmentos que mostram os números e dão a possibilidade de formação de algumas letras. Um display com 8 segmentos possui também um ponto, que pode estar à direita ou esquerda do número exibido. Este tipo de display é chamado mais comumente de display de 7 segmentos com ponto.

informações do Receptor Matrix® e converte-las para um formato que pudesse ser enviado tanto para o computador, assim como para outros periféricos de interesse.

A descrição precisa do funcionamento e outras questões relevantes serão mais bem explanadas no tópico que aborda a Decodificação e Controle de Comunicação.

A transmissão para este aparelho é feita com controles remotos, figura 6 (a), como os usados para acionar alarmes de casas e carros, ou abrir portas e portões. Entretanto, aproveitamos apenas o hardware, figura 6 (b), pois a montagem convencional dele não possuiu um design compatível com o ramo em que desejamos aplicar esse sistema.



Figura 6 – (a) Transmissor Completo, (b) Hardware do Transmissor.

Este controle remoto possui uma capacidade de transmissão de códigos de 10 bits para identificação e mais um bit que indica qual dos dois botões foi pressionado, totalizando um código de 11 bits.

A transmissão de dados entre estes aparelhos é feita usando a frequência de 433,92 MHz, através da modulação ASK² [1] [2] [3] [4]. A idéia básica deste tipo de modulação é multiplicar o sinal por uma portadora de maior frequência, neste caso 433,92 MHz. Para um sinal binário, este tipo de modulação possui apenas dois estados, ligado e desligado, que representam os bits "1" e "0", respectivamente. A figura 6 mostra um exemplo de modulação ASK binária.

² ASK – Amplitude Shift Keying

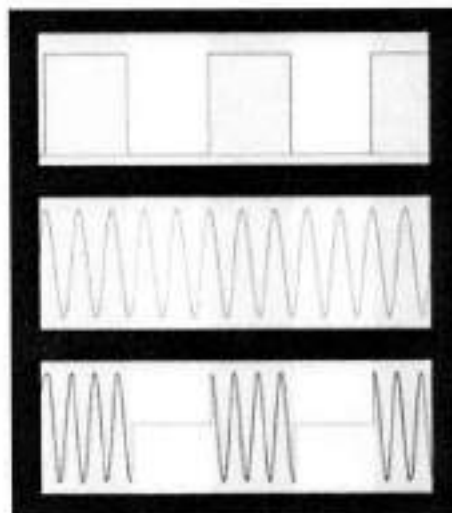


Figura 6 – Modulação ASK

A modulação ASK é uma das melhores escolhas quando o baixo custo é essencial. Esta qualidade decorre do fato da modulação e demodulação serem muito simples, tornando tanto a modulação como a demodulação simplificadas, conseqüentemente, reduzindo os custos dos circuitos envolvidos no processo.

A grande deficiência neste tipo de modulação reside no fato de possuir uma baixa imunidade a ruídos. Esta característica pode ser um problema ao bom funcionamento do sistema, principalmente em ambientes onde existam outros aparelhos operando na mesma faixa de frequência. A frequência usada pertence à banda ISM³, onde a utilização não necessita de autorizações dos órgãos controladores para operação.

3.2 Decodificador e Controle de Comunicação

Este bloco funcional é responsável pela decodificação das informações do Receptor Matrix®, tratamento destas informações, envio para o servidor de dados e envio para da fila para o display de exibição. Este bloco é constituído por diversos outros pequenos blocos, que em conjunto poderão desempenhar a função que descrevemos anteriormente. A figura 7 mostra o esquema desta

³ ISM – Industrial, Scientific and Medical

estrutura, destrinchada em pequenas parcelas que serão melhor explicadas à frente.

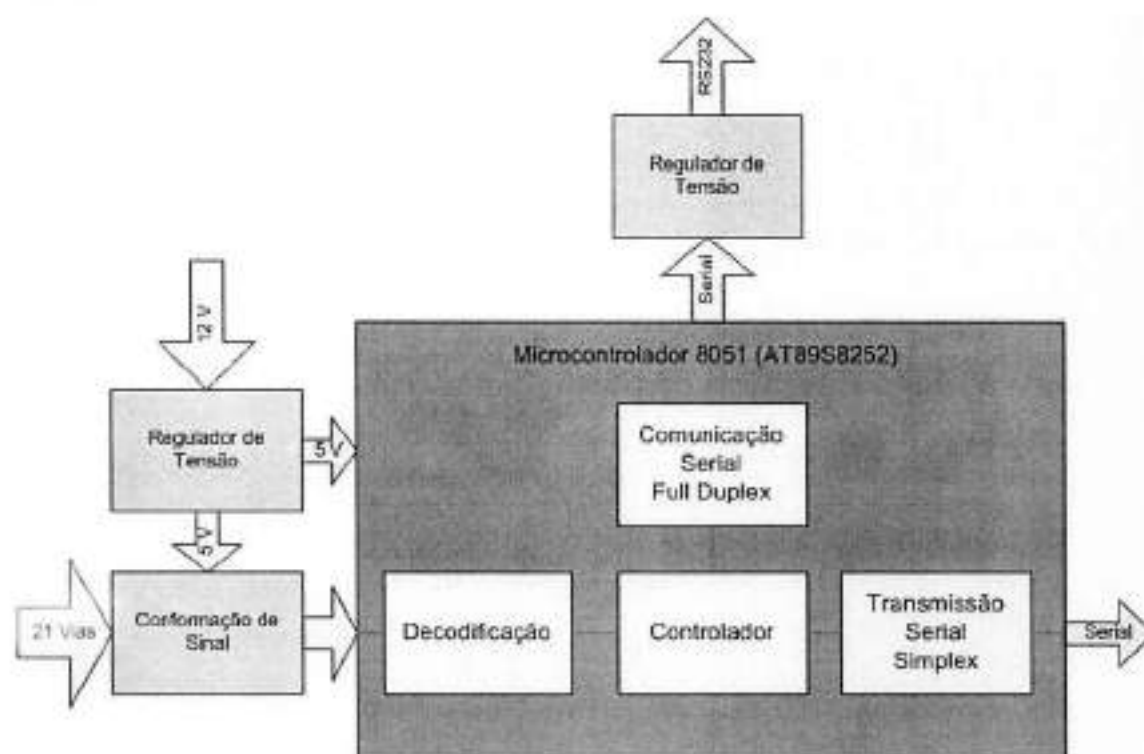


Figura 7 – Esquema do bloco Decodificador e de Controle de Comunicação

O maior desafio em criar o bloco decodificador foi encontrar o ponto para tomada de sinais dentro da placa do receptor, pois esta é confeccionada usando componentes SMD, o que torna difícil criar conexões sem a aparelhagem necessária. Desta forma os únicos componentes que não apresentavam estas características eram os displays de 8 segmentos, sendo que apenas os sinais que constituem os algarismos são de importância para tratamento, isto é, o sinal relativo ao ponto pode ser descartado.

Para a composição do sinal do sinal a ser decodificado utilizamos os sinais provenientes de 3 displays. O número de mesas pode chegar a uma centena, porém dificilmente a um milhar, por isso, não utilizamos os sinais provenientes do 4º display. Com isso o número de sinais a ser tratado e decodificado totaliza 21.

A figura 7 tem esta divisão particular de funções, pois no decorrer do nosso projeto enxergamos cada um dos blocos como um problema a ser resolvido para integrar o sistema como um todo. Podemos ver na figura que 4 blocos estão dentro do micro-controlador, esta foi uma métrica que utilizamos:

usar todos os recursos possíveis do micro-controlador e diminuir espaço físico do hardware, isto é, implementar o máximo possível de funções por *software* para diminuir o espaço físico ocupado pelo *hardware*.

A partir deste ponto cada um dos blocos será explicado com mais detalhes, aprofundando o conhecimento do leitor sobre as tecnologias utilizadas e alguns dos motivos que levaram às determinadas escolhas. Um tópico também é usado para falar sobre o programa de CAD utilizado para desenhar o circuito completo e seu leiaute. Por fim, será apresentado o esquema completo deste circuito.

3.2.1 Regulador de Tensão 12V – 5V

O Receptor Matrix® é um aparelho que funciona com uma alimentação de +12 V, porém a maioria dos micro-controladores exige uma tensão de alimentação de +5V, como é o caso do AT89S8252 da Atmel™. Como a placa de decodificação será colocada junto ao receptor e utilizará a mesma fonte de alimentação que este, foi necessário utilizar um regulador de tensão que pudesse fornecer a tensão adequada de alimentação para os componentes da placa de decodificação.

Para a finalidade descrita anteriormente, escolhemos o regulador de tensão LM7805, da Fairchild Semiconductors™ [5]. Este dispositivo pode fornecer tensões entre 4,75 V e 5,25 V, quando alimentado com tensões que variam entre 7 V e 20 V. Para esta configuração, ainda, ele é possibilita um fluxo de corrente de até 1 A e uma potência de saída de até 15 W. Este dispositivo pode ser empregado como regulador de carga, para rejeição de efeito *Ripple*⁴, como regulador de corrente, dentre outras, além da aplicação de regulador de tensão já mencionada.

Este dispositivo emprega um limitador de corrente interno, proteção para sobrecarga térmica e proteção contra curto-circuito, tornando este componente praticamente indestrutível. Se um dissipador de calor adequado for acoplado

⁴ **Ripple** – Nome dado à oscilação de potenciais contínuos

ao dissipador já existente, este dispositivo pode fornecer uma corrente de até 1 A. O esquema com a configuração interna do dispositivo pode ser vista na figura 8 (extraída do datasheet do componente).

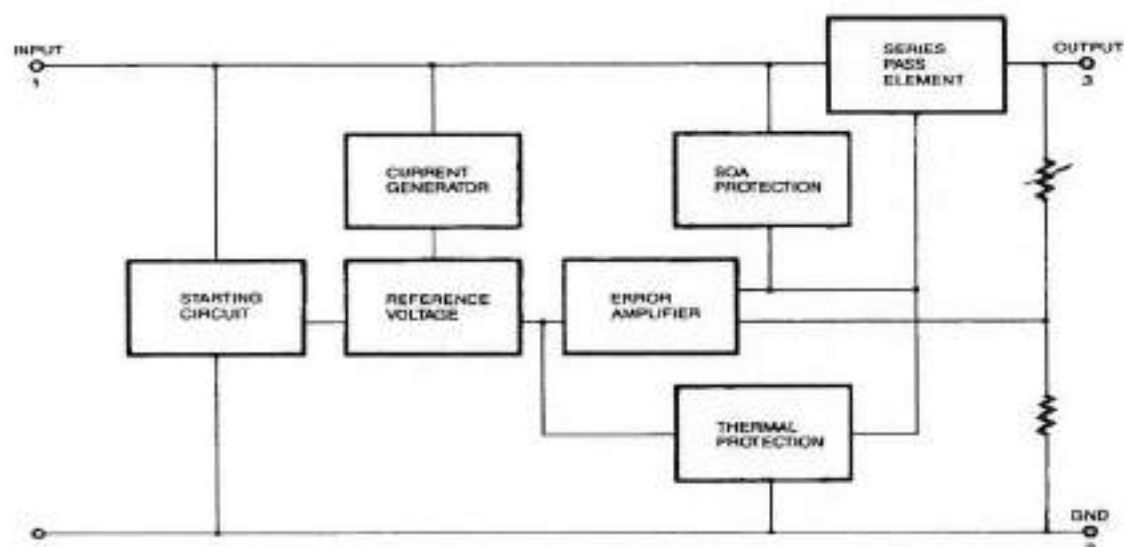


Figura 8 – Diagrama de blocos da arquitetura interna do LM7805

Para criar o circuito regulador de tensão baseado neste dispositivo basta acoplarmos 2 capacitores entre seus terminais. A configuração utilizada no projeto está apresentada abaixo.

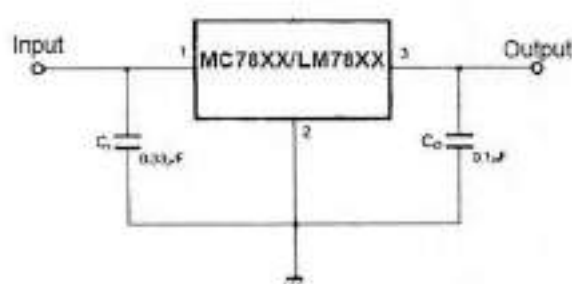


Figura 9 – Configuração do LM7805 para atuar como regulador de tensão

3.2.2 Conformação de Sinal

Ao escolhermos os displays para retirar os sinais de referência nos deparamos com dois problemas: o mapeamento dos sinais e o nível de tensão dos displays.

Para mapear os sinais primeiramente deveríamos descobrir qual dos pinos ativa cada um dos segmentos, pois não conseguimos encontrar o datasheet deste display em particular e não podemos confiar em datasheets de outros fabricantes. Esta foi uma tarefa fácil, que consistiu em medir cada um dos pinos para cada um dos valores numéricos possíveis, de 0 a 9. A figura 10 mostra a figura de um dos displays e a atribuição de letras que usamos para o mapeamento.

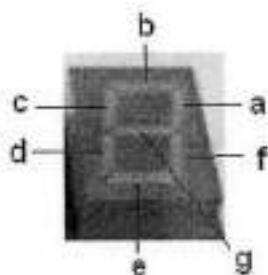


Figura 10 – Mapeamento dos segmentos do display do Receptor Matrix®

Neste display existem 10 pinos, 5 na parte superior e 5 na parte inferior. A numeração que utilizei para o mapeamento foi a seguinte de 1 a 5 nos pinos superiores da esquerda pra direita (vendo por baixo) e de 6 a 10 nos pinos inferiores da direita pra esquerda (vendo por baixo). Os resultados obtidos são mostrados na Tabela 1.

Pino	Segmento	Pino	Segmento
1	a	6	d
2	b	7	e
3	GND	8	GND
4	c	9	f
5	g	10	Sem função

Tabela 1 – Relação pinos/segmentos para o display do Receptor Matrix®

Esta relação é necessária para que a decodificação seja efetuada. Todo este processo pode ser chamado de engenharia reversa, que é era um processo muito comum antigamente quando não existiam especificações claras e a grande maioria dos sistemas de transmissão de dados eram proprietários, fazendo com que a integração entre aparelhos de fabricantes diferentes necessitasse de um hardware adicional para conversão de dados de um sistema para o outro.

O segundo problema que enfrentamos para esta decodificação é o nível de tensão que alimenta os displays do receptor, que esta entre 1,6 V e 1,7 V. Este nível de tensão não é adequado para a maioria dos dispositivos lógicos disponíveis no mercado, tanto na família TTL como na família CMOS. Existem dispositivos lógicos que trabalham com níveis de tensão mais baixos, como 3 V por exemplo, este tipo de dispositivo reconheceria o sinal 1,7 V como sendo um "1", porém teríamos outro problema nas mãos: o micro-controlador trabalha com tensões de 0 V e 5 V, e o valor 3 V cai na faixa indeterminada de reconhecimento de valores, e , desta forma, ainda teríamos de conformar o sinal de saída destes dispositivos para a entrada do micro-controlador.

Tendo em vista todas as dificuldades expostas acima, preferimos conformar o sinal de 1,7 V para 5 V. Para esta tarefa utilizamos uma montagem simples e bastante conhecida: um transistor bipolar funcionando como chave. Uma chave simples a transistor bipolar NPN é vista na figura 11.

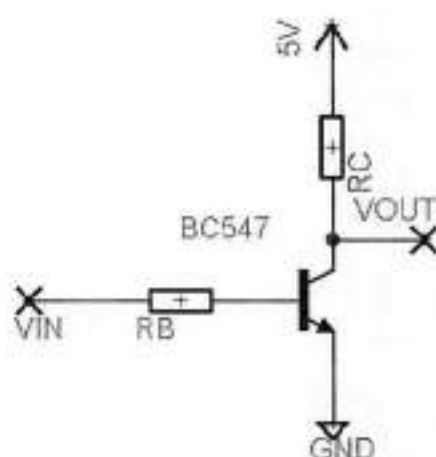


Figura 11 – Chave usando transistor bipolar NPN

Nesta configuração quando temos como V_{IN} uma tensão maior que a tensão de polarização V_{BE} existirá uma corrente fluindo através da base. Ao aumentarmos a tensão da base a corrente da I_B aumenta, levando à uma diminuição da tensão entre o coletor e o emissor, até alcançar o valor V_{CEsat} . A partir daí um aumento de I_B não implica diminuição sensível de V_{CE} . Diz-se que o transistor bipolar está saturado, apresentando resistência entre coletor e emissor relativamente pequena (R_S). Seria o caso de compararmos o transistor bipolar a uma chave fechada.

Ao diminuirmos o valor da corrente de base, a corrente de coletor irá diminuir, tendendo a I_{C0} . Nessa condição, a tensão V_{CE} é bastante grande (aproximadamente V_{CC}) enquanto a corrente I_C é muito pequena. Diz-se que o transistor bipolar está cortado, apresentando resistência entre coletor e emissor relativamente alta.

Uma análise neste circuito segundo o modelo de Ebers-Moll⁵ mostra que, muito embora excelente, o desempenho do transistor bipolar como chave não é ideal pelas seguintes razões:

- a. Existe no estado fechado da chave uma pequena queda de tensão no transistor.
- b. O circuito de entrada (controle da chave) dissipa uma pequena potência.
- c. Mesmo no estado de chave aberta há passagem de corrente pelo transistor.

Tendo em vista as características explicadas acima podemos concluir que o transistor bipolar atuando como chave funciona como uma fonte de tensão controlada por corrente. Outro ponto que pode ser observado é que existe uma inversão entre o sinal de entrada e o sinal de saída, pois quando temos uma tensão suficientemente grande na entrada o transistor conduz e V_{OUT} tem valor de V_{CESat} , que na maioria dos transistores tem valor próximo de 0,3 V (idealmente nulo).

A escolha do modelo de transistor foi feita observando algumas curvas de parâmetros de alguns modelos. As diversas análises feitas nos levaram a um modelo bastante conhecido, o BC547, que possui diversos fabricantes. Este modelo possui um ganho (h_{FE}) de 250, aproximadamente, funcionando bem com correntes de base que variam entre 0,5 μA a 100 μA , que é muito pertinente para nossa aplicação de não carregamento do circuito de onde retiramos os dados. Outra característica importante é a velocidade de chaveamento que pode chegar a 100 MHz, quando a corrente de coletor é de 1 mA, segundo o *datasheet*. Uma velocidade de chaveamento compatível com a velocidade do micro-controlador (24 MHz) é desejável neste projeto.

Nossa grande preocupação era de não carregar o circuito do qual estávamos retirando o sinal e com isso tivemos de dimensionar nosso circuito chave para trabalhar com uma corrente de base baixa o suficiente para atingir o objetivo citado. Com isso escolhemos uma resistência bastante alta, 1 MΩ, com isso a corrente será de cerca de 1 μA, que pode ser calculada segunda a fórmula:

$$I_B = \frac{V_{IN} - V_{BE}}{R_B}$$

A corrente de coletor, I_C , pode ser obtida considerando que o transistor está no limiar entre a saturação e a região linear, desta forma podemos considerar o ganho de corrente entre a base e o coletor como sendo h_{FE} , ou simplesmente β . Para o cálculo da corrente I_C devemos considerar a menor corrente I_B , pois se devemos lembrar que o aumento de corrente de base não aumenta significativamente a corrente de coletor, quando estamos no estado de saturação. A resistência de coletor pode ser obtida através da diferença entre a tensão de alimentação, V_{CC} , e a tensão entre o coletor e o emissor, V_{CE} , dividida pela corrente I_C . Logo temos as seguintes expressões:

$$I_C = \beta \cdot I_B \qquad R_C = \frac{V_{CC} - V_{CE}}{I_C}$$

Fazendo os cálculos com os seguintes valores: $V_{IN} = 1,65$ V, $V_{BE} = 0,62$ V, $V_{CESat} = 0,15$ V, $I_B = 1$ μA e $\beta = 250$, obtemos: $R_B = 1$ MΩ, $I_C = 300$ μA, $R_C = 19,4$ kΩ. Sendo que utilizamos um RC de 22 kΩ, por ser um valor comercial.

Um fato deve ser observado com atenção no circuito do transistor npn funcionando como chave, ele possui um caráter de inversor, isto é, quando um valor "1" é colocado em V_{IN} , obtemos um valor "0" em V_{OUT} . Desta forma, foi necessário inverter novamente o sinal de forma que este fosse usado com entrada do microcontrolador. Para esta tarefa utilizamos diretamente inversores CMOS, LS74HC04, da NationalTM [7]. Esta nova configuração nos leva ao novo circuito que será aplicado a cada um dos 21 sinais que devem ser conformados:

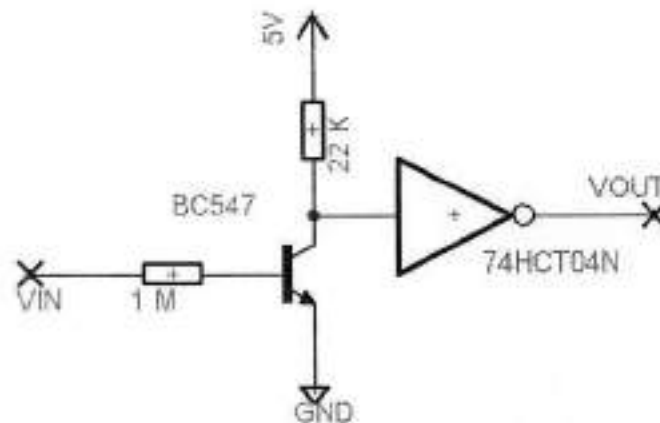


Figura 12 – Circuito completo para conformação do sinal

As expressões que relacionam os tempos de pertinentes ao chaveamento (subida, descida, entre outros) não foram considerados, pois a frequência de operação desejada se encontra muito distante da frequência de operação máxima, e com isso consideramos estes tempos por acreditarmos que eles terão pouca influencia no processo de chaveamento.

Uma outra configuração foi testada, não utilizando o inversor, e usando lógica reversa para decodificação no microcontrolador, porém a arquitetura interna das entradas do microcontrolador causa variações no sinal da chave, fazendo com que este não funcione corretamente. Portanto, o uso do inversor CMOS é fundamental para isolar estas duas partes do circuito e garantir o funcionamento correto do circuito.

3.2.3 Regulador de Tensão para Comunicação Serial

Já foi dito que tensão de alimentação do microcontrolador é de 5 V, porém as tensões para a comunicação serial RS-232 são de -12 V e +12 V, portanto devemos também usar um regulador de tensão nesta interface para conformar os sinais para valores de tensão adequados para a comunicação com nosso servidor.

Depois de pesquisar por alguns componentes que pudessem fazer esta transformação de potenciais optamos por um CI bastante conhecido para esta finalidade: o MAX232CPE, da MaximTM [7], figura 13.

O MAX232CPE é um *driver/receiver* usado em interfaces de comunicação dos tipos EIA/TIA-232E⁶ e V.28⁷/V.24⁸, particularmente em aplicações onde fontes de tensão de +12 V e -12 V não estão disponíveis.

Este componente é especialmente útil em sistemas onde a alimentação é feita por baterias. O uso de apenas 4 capacitores para montagem do circuito torna este CI recomendado para aplicações onde o espaço no circuito impresso é um fator crítico.

Internamente este dispositivo é constituído por dobradores de tensão, que é um tipo de circuito constituído a partir da associação dos capacitores colocados externamente. Desta forma, o dispositivo consegue gerar sinais de +12 V e -12 V a partir de uma alimentação de +5 V. A conformação dos dados é feita por circuitos *buffers* inversores, alimentados pelas tensões geradas pelos capacitores. Estes tem em suas entradas circuitos *pull-up*, isto é resistores ligados ao V_{CC} de +5V, assim, quando o sinal provindo do microcontrolador for de 5 V, a tensão na entrada do inversor será 0 e em sua saída teremos "1", que em níveis de tensão corresponde à tensão positiva de alimentação. No caso contrário, quando temos 0 V, o sinal na entrada do inversor será de 5 V, e na saída do inversor teremos "0" que corresponde a tensão de alimentação negativa. Esta lógica vale para as entradas de transmissão, TX, do MAX232CPE. Para as entradas de recepção temos resistores fazendo a função *pull-down*, isto é resistores ligados ao terra, e a lógica pode ser aplicada assim como no caso anterior para determinar a saída, porém neste caso as tensões de alimentação dos *buffers* inversores são 5 V e 0 V. V.24 is a specification for single-ended communications that includes the definition of connector pin allocations. It is used together with V.28 to define a specification for serial asynchronous or synchronous communications.

⁶ **EIA/TIA-232E** – é o nome da norma que define o padrão RS232, sendo que o nome oficial para a norma é *Interface Between Data Terminal Equipment and Data Circuit-Termination Equipment Employing Serial Binary Data Interchange*. A sigla EIA/TIA refere-se aos grupos que fundamentaram a norma em 1962, *Electronic Industry Association and the Telecommunications Industry Association*.

⁷ **V.28** – é a especificação que define a comunicação serial, incluindo as características elétricas. É usada juntamente com a V.24 para definir as especificações para comunicação serial síncrona e assíncrona.

⁸ **V.24** – é a especificação que define a comunicação serial, incluindo a alocação de pinos. É usada juntamente com a V.28 para definir as especificações para comunicação serial síncrona e assíncrona.

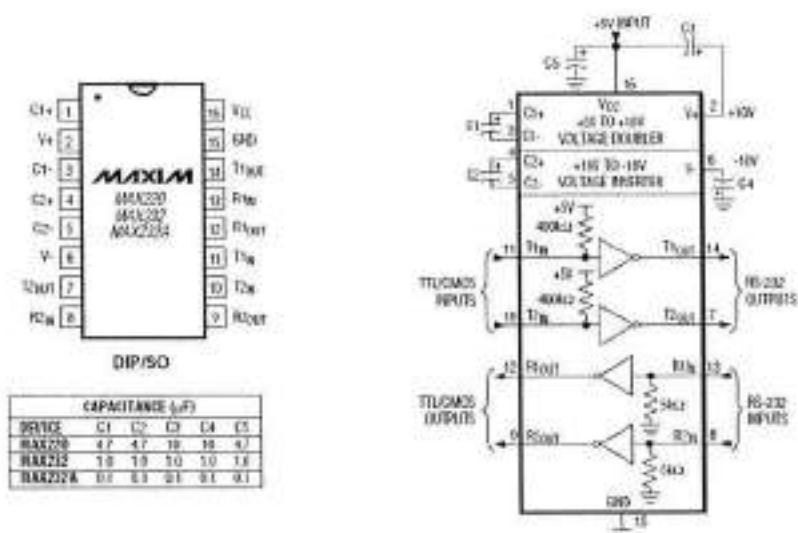


Figura 13 – MAX232CPE: encapsulamento e arquitetura

3.2.4 Microcontrolador

Este é o componente mais importante de todo o hardware de decodificação e controle de comunicação, já que lê é responsável pelas funções que dão nome ao bloco. Dada sua importância para o circuito iremos nos ater um pouco mais nas explicações sobre este, inicialmente introduzindo pequenos conceitos que devem ser entendidos antes de prosseguirmos com as soluções e implementações realizadas sobre este CI.

O microprocessador escolhido para este projeto foi o AT89S8252, da fabricante AtmelTM. Este é um dispositivo que CMOS de 8 bits, com 8k bytes de memória FLASH (programas) e 2 k bytes de memória EEPROM (dados), internas. Este dispositivo possui ainda um baixo consumo, devido a sua arquitetura CMOS.

Este componente é fabricado segundo o padrão de instruções e pinagem da família de microcontroladores 8051, mostrado na figura 14.

A memória Flash integrada no chip permite que o sistema seja reprogramado através de uma interface serial SPI ou por um programador de memória não-volátil convencional.

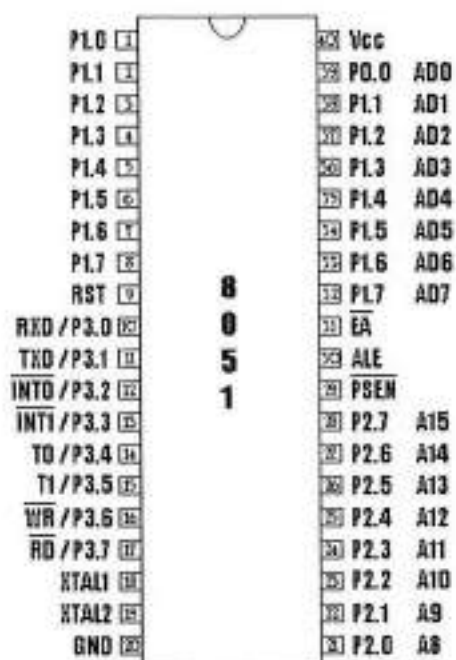


Figura 14 – Pinagem típica de um microcontrolador 8051

O AT89S8252 possui todas as funcionalidades do padrão 8051, além de outras adicionais, tais como 8k bytes de memória Flash, 2k bytes de EEPROM, 256 bytes de RAM, 32 linhas de entrada e saída, temporizador programável, dois ponteiros de dados, três contadores/temporizadores de 16 bits, uma arquitetura de interrupção de dois níveis, uma porta serial *full duplex*.

Além disso, o CI foi projetado para operar em dois modos de economia de energia que podem ser selecionados via software. O modo *Idle* para a UCP, que mantém a memória RAM, temporizadores, contadores, porta serial e sistema de interrupção funcionando. O modo *Power-Down* salva o conteúdo da RAM, porém congela o oscilador, desabilitando outras funções do CI até que aconteça uma outra interrupção ou um *reset* de hardware.

b. Memória e Registradores

O 8051 dispõe de 128 bytes de memória RAM interna, utilizada também para a pilha interna, e 128 bytes de registradores de funções especiais, como indica a figura abaixo.

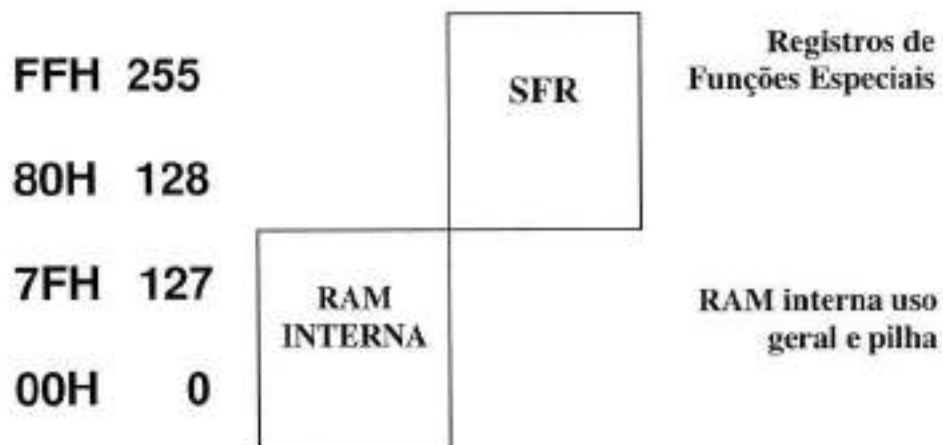


Figura 15 – Divisão da memória RAM do 8051

A estrutura da RAM interna segue a seguinte ilustração.

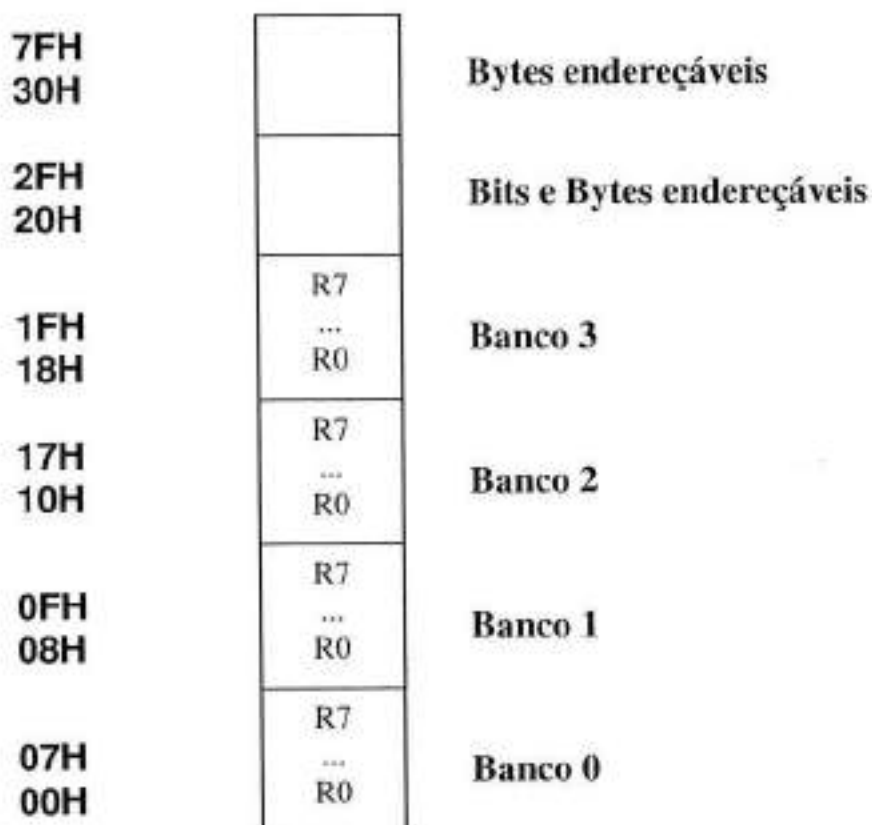


Figura 16 – Estrutura interna da RAM de um 8051

Os bancos 0 a 3 são selecionados pelos dois bits do registro PSW e podem ser utilizados como registradores de índice. Por exemplo, interrupções usam banco 1, enquanto programas usam o banco 0.

A seguir temos o esquema dos registradores de funções especiais.

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8	PSW								DF
D0									D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A0
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH	PCON				87

Tabela 2 – Registradores de funções especiais

Os vários registradores que fazem parte dos registradores de funções especiais incluem desde posições da RAM das portas de entrada e saída até os registros para o uso dos temporizadores. Abaixo segue um breve explicativo de cada registrador especial ao lado de seu endereço na memória RAM.

- **P0 (80H), P1 (90H), P2 (A0H) e P3 (B0):** São as posições na RAM que contém os dados das 4 portas de I/O do microcontrolador, caso as mesmas sejam utilizadas com esse fim. Se for feita uma escrita em um desses registros, o conteúdo nos pinos correspondentes à porta será automaticamente alterado. Já uma leitura dos mesmos verifica o estado dos pinos.

- **SP (81H):** É o apontador de pilha (*Stack Pointer*), que aponta para o alto da pilha.
- **DPL (82H) e DPH (83H):** Em conjunto formam o DPTR, utilizado para endereçamento indireto de 16 bits.
- **PCON (87H):** O registro PCON (*Power Control*) permite adaptar o microcontrolador para redução de consumo com segurança.
- **TCON (88H) e TMOD (89H):** Registros dos Temporizadores/Contadores. Permitem a programação dos mesmos.
- **TL0 (8AH), TH0 (8CH), TL1 (8BH) e TH1 (8DH):** São os registros dos dados dos dois Temporizadores/Contadores (T0 e T1).
- **SCON (98H) e SBUF (99H):** Registros para uso da porta de comunicação serial.
- **IE (A8H) e IP (B8H):** Registro para programação (habilitação/desabilitação, prioridade etc.) das interrupções.
- **PSW (D0H):** O PSW (*Program Status Word* - palavra de status do programa) é o registro dos *Flags* do 8051;
- **ACC (E0H):** É o acumulador.
- **B (F0H):** Registro auxiliar B.

Vamos analisar agora alguns destes registradores com maiores detalhes

PSW – Palavra de Status de Programa

O registrador Palavra de Status de Programa (PSW) contém vários bits que indicam o estado atual da Unidade Central de Processamento. O acesso a esse registrador pode ser feito através do endereço D0h, mas também é um bit endereçável.

A tabela abaixo apresenta a estrutura do registrador PSW, que reside no espaço de endereçamento de registros de função especial.

D7h	D6h	D5h	D4h	D3h	D2h	D1h	D0h
CY	AC	F0	RS1	RS0	OV	-	P
PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PWS.1	PSW.0

Tabela 3 – Estrutura do registrador PSW

- PSW.7 - *Flag de carry* para bit 7 da ULA;
- PSW.6 - *Flag de carry auxiliar* para bit 3 da ULA (para operações BCD);
- PSW.5 - Flag 0 - Flag de status de propósito geral (definido pelo usuário);
- PSW.4 - *Bit 1 de seleção de banco*
- PSW.3 - *Bit 0 de seleção de banco*
- PSW.2 - *Flag de overflow* para operações aritméticas;
- PSW.1 - (sem nome)
- PSW.0 - *Flag de paridade do acumulador* (1=ímpar, 0=par).

Este registro contém o bit de *carry*, *carry auxiliar*, para operações BCD, dois bits de seleção de banco de registros, flag de overflow, bit de paridade e dois flags de status definidos pelo usuário.

O bit de *carry* é usado em operações aritméticas e, também, serve como "acumulador" para operações booleanas.

Os bits RS0 e RS1 são usados para selecionar um dos quatro bancos de registros mostrado na tabela 4. As instruções referem-se a estas localizações de RAM como registros de R0 a R7. A seleção de qual dos quatro bancos estará sendo utilizado é feita com base aos conteúdos de RS0 e RS1, durante o tempo de execução do programa.

O bit de paridade reflete o número de 1's do acumulador. $P = 1$ se o acumulador tiver um número ímpar de 1's e $P = 0$, se tiver um número par. Assim o número de 1's do acumulador mais P será sempre par.

O flag de overflow é afetado por operações aritméticas.

RS1	RS0	BANCO	Endereço na RAM interna
0	0	0	00 h - 07 h
0	1	1	08 h - 0F h
1	0	2	10 h - 17 h
1	1	3	18 h - 1F h

Tabela 4 – Seleção de bancos de registradores em função de RS0 e RS1

Além do flags usuais, como CY, AC , OV, P, tem dois flags de propósito geral que não estão associados com qualquer estado ou função da CPU. O bit PSW.5 (F0) e PSW.1 (sem nome). O programador pode usar F0 (PSW.5) como um bit flag para aplicação definida pelo usuário. Pode ser setado ou ressetado pelo programa como uma função de alguma condição especial ou ser lido de um pino de alguma porta. Documentos da INTELTM indicam que PSW.1 esta reservado para uma utilização futura e é recomendado que não seja utilizado em programas

SP – Registro Apontador de Pilha

O registro SP é um registro de tamanho 8 bits. Este será incrementado antes que um dado seja armazenado na pilha, durante a execução de uma instrução PUSH, ou CALL.

Assim, a região de pilha (stack) irá residir no espaço de RAM interna do 8051. O SP é inicializado com o valor 07H após ocorrer o RESET. Isto faz com que a pilha inicie na localização 08H. Caso sejam usados os bancos de registradores 1, 2 e/ou 3, e seja utilizada a pilha, o SP deverá ser alterado para iniciar a pilha em outra região da RAM que não cause problemas. A instrução MOV IE,#novo endereço deverá aparecer no programa, preferencialmente no início.

Este é acessado pelo endereço 81H e, portanto, não é bit endereçável.

DPTR – Registro Apontador de Dados

O registro apontador de dados é formado por um byte alto (DPH) acessado pelo endereço 83H e por um byte baixo (DPL), acessado pelo endereço 82H. Este registro pode ser manipulado como um registro de 16 bits (DPTR), ou como dois registros independentes de 8 bits.

A sua função é de manter um endereço de 16 bits, usado para acessar memórias externas.

c. Projeto Lógico do Firmware

O projeto foi desenvolvido na linguagem de programação de alto nível C. Para isso foi utilizado o software *WordPad*, já incluído no pacote *Windows*, e o compilador SDCC, que é um software código aberto, que transforma um código escrito na linguagem C para código objeto no formato da família 8051.

d. Entradas e Saídas

As entradas que o microcontrolador receberá são os números das mesas, pelas portas P0 e P1. Um processo de varredura fica sempre verificando se há algum número de mesa novo, que não está cadastrado na fila.

Uma das saídas é a da tela de LEDs que mostrará a ordem de atendimento das mesas, ou seja, transcreve a fila que está registrada no microcontrolador. Esta saída consiste de dois pinos: P2.1, que servirá de CLOCK, e P2.2, que enviará os dados para a tela.

Outra saída é a comunicação com o PC, para o qual o microcontrolador envia os números das mesas utilizando o protocolo RS-232.

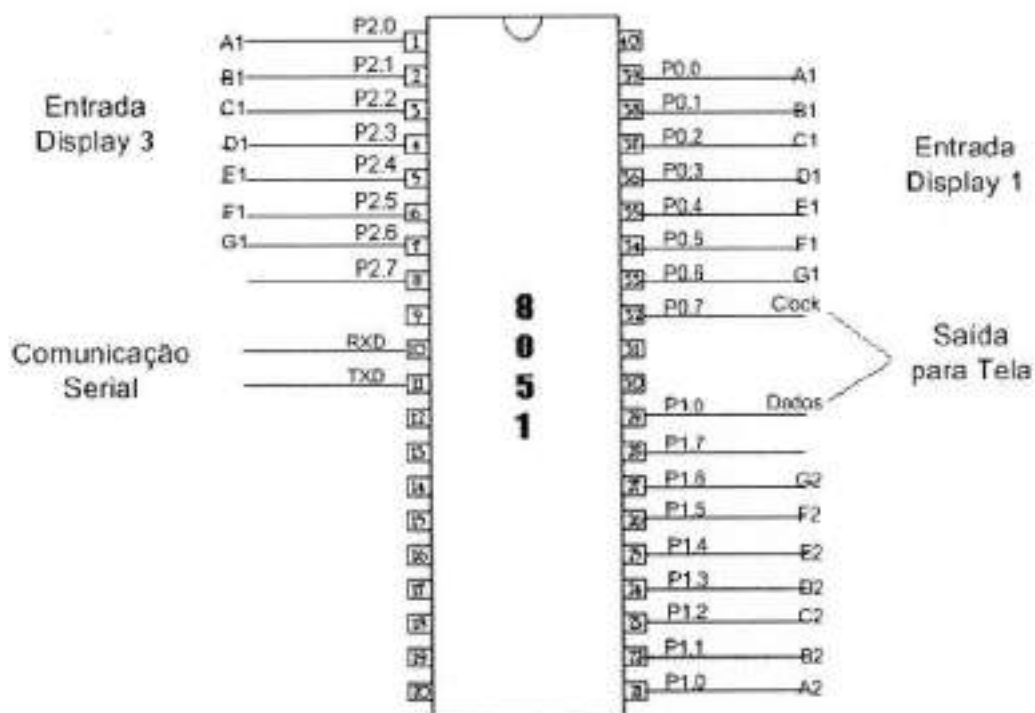


Figura 17 – Ligações entre o 8051 e pinos dos displays do receptor

e. Fluxo Geral

O Programa é iniciado com os valores de configuração inicial dos registradores de funções especiais e alguns outros valores, para configurar o temporizador e o seu modo, o tipo de comunicação serial e as interrupções.

Logo após a inicialização, é realizada a varredura do display de entrada, que verifica se há algum número de mesa novo na entrada.. Esta varredura sempre será realizada a cada interrupção causada pelo estouro do TIMER 1.

Caso haja um número novo, este é colocado na estrutura fila de mesas, uma função coloca este número na tela de LEDs, e outra envia através do canal serial para o PC.



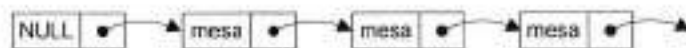
Figura 18 – Fluxo geral do programa do 8051

f. Estrutura de dados

Para atender as necessidades de se implementar uma fila, utilizamos as técnicas de implementação de estrutura de dados. O programa contém uma fila do tipo FIFO, no qual cada nó contém dois campos: um número inteiro que indica o número da mesa, e um ponteiro que aponta para a próxima mesa:



- **int mesa**, que contém o número da mesa;
- **struct fila prox**, que aponta para o próximo número da fila.



As funções que auxiliam na manipulação da fila são a:

- **int vazio(struct fila *p)**, que retorna 1 se a fila está vazia, e 0, caso contrário;
- **void insere(int x, struct fila *p)**, que insere o número de mesa x, na fila p. A inserção de um nó na fila sempre é feita colocando-o entre a cabeça da fila e o primeiro nó. O processo de inserção segue os seguintes passos:

- Aloca-se um espaço na memória do tamanho da estrutura fila;



- Registra-se o número da mesa nesse novo nó (no exemplo, o número 38);



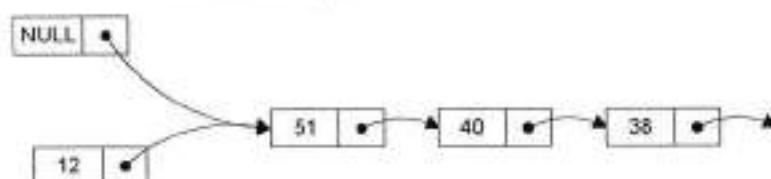
- Inserir o nó no final da fila;



- **void remove(struct fila *p)**, que remove o último número da fila.

Esse processo ocorre da seguinte maneira:

- Aponta-se o ponteiro prox da cabeça para o ponteiro prox do nó a ser retirado (queremos retirar a mesa 12, neste caso):



- Remove-se o nó com o número 12 da memória.



- **struct fila *busca(int x, struct fila *ini)**, que retorna o nó da fila, se o número existir, e NULL, caso contrário;
- **int size(struct fila *ini)**, que retorna o tamanho da fila;

g. Configuração dos Registradores

O processo de inicialização dos registradores determina o modo de funcionamento geral do microcontrolador. Descrevemos abaixo como a configuração é realizada.

PCON = 0x00

O modo de consumo foi configurado para funcionamento normal, sem ativar os modos especiais Idle e Power-down.

Outra configuração aqui implícita é o SMOD = 0, que no modo 1 a ser utilizado, divide o clock por 32.

SCON = 0x52

Esse registrador não é endereçável por bit e, portanto, deve ser registrado desta forma. Os bits 5 e 7 em zero e o bit 6 em um define uma comunicação serial no modo 1. O bit 4 (SCON.4) deste registrador indica que **REN = 1** (habilita a recepção). O bit 1 indica **TI = 1**, que é o flag de interrupção da transmissão. É setado pelo hardware depois da transmissão do oitavo bit de dados quando no modo 0, e, nos outros, ao início do Stop bit. Deve ser zerado pelo software da rotina de interrupção para permitir novas interrupções. O bit 0 indica **RI = 0**, que é o flag de requisição de interrupção da transmissão. É setado pelo hardware após a recepção do oitavo bit de dados quando no modo 0, e, nos outros, ao meio do tempo de recepção do Stop bit. Também deve ser zerado pelo software da rotina de interrupção para permitir novas interrupções.

TMOD = 0x20

Foi configurado o Timer 1 no modo 2, ou seja, como um temporizador de 8 bits e recarga automática. Neste modo de funcionamento, temos um sistema que não necessitamos reescrever por software o valor a ser contado. Em TL0 ou TL1 escrevemos os valores iniciais a serem contados, e, em TH0 ou TH1, os valores a serem carregados após a primeira contagem.

TH1 = 0xFA

A divisão foi feita para que sejam contados 4800 pulsos.

TR1 = 1

O Timer 1 é disparado aqui por este registro.

IE = 0x90

O registro IE.4 habilita a interrupção do canal serial e o IE.7 habilita interrupções, em geral.

h. Varredura do Display

A rotina de varredura do Display ocorre sempre quando há um estouro no Timer 1. Esta rotina coleta os dados de entrada em P0 e P1 e verifica se o número varrido já existe na tela.

O número é coletado de um sistema com dois displays de 7 segmentos, que transmite este sinal até a entrada do microcontrolador.

Cada display de 7 segmentos tem a seguinte configuração:

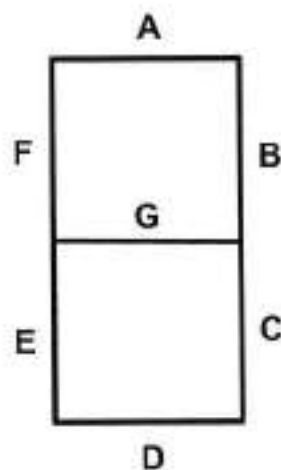


Figura 19 – Mapeamento do display de 7 segmentos

A conexão lógica entre o módulo de conformação de sinal e o 8051, seguindo uma nomenclatura da origem dos sinais, é apresentada abaixo. Como foi dito anteriormente 21 pinos são necessários para a implementação, desta forma devemos utilizar 3 dos 4 portos disponíveis para I/O com a finalidade de decodificar os sinais.

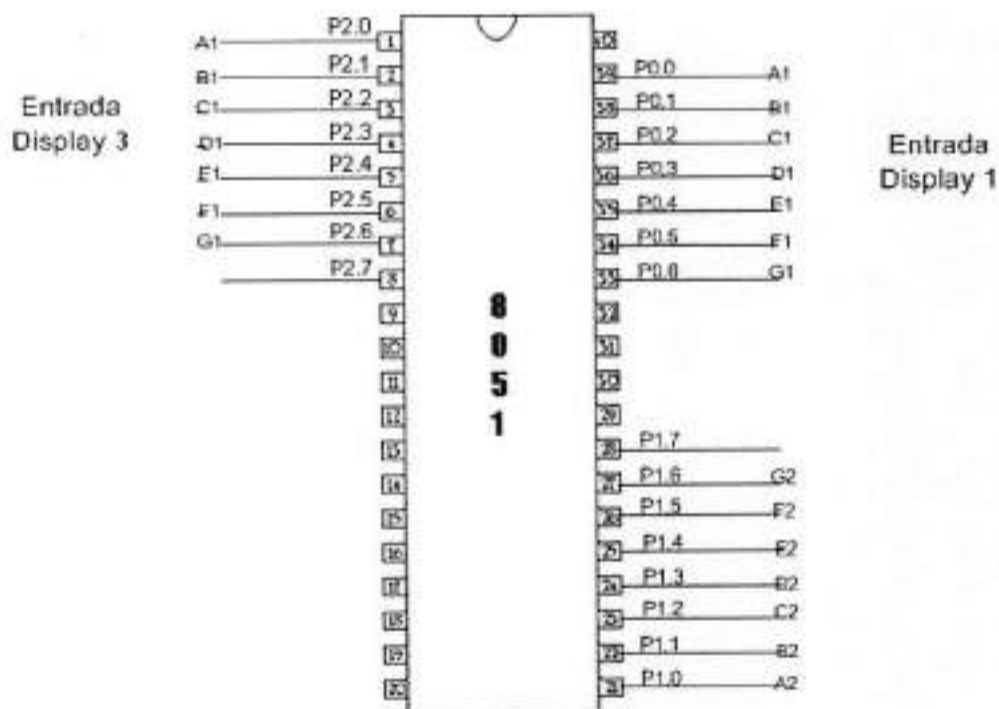


Figura 20 – Conexões dos portos com os sinais conformados

Com os sinais podendo ser adquiridos pelo microcontrolador basta agora decodificá-los internamente, usando uma estrutura de *switch...case*, muito utilizada para comparar um valor com diversas possibilidades em linguagens de programação de alto nível. Para esta decodificação apresentamos a tabela 5 com os correspondentes valores a serem identificados em função dos sinais provenientes do display.

a	b	c	d	e	f	g	Valor ₁₀
0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1
1	1	0	1	1	0	1	2
1	1	1	1	0	0	1	3
0	1	1	0	0	1	1	4
1	0	1	1	0	1	1	5
1	0	1	1	1	1	1	6
1	1	1	0	0	0	0	7
1	1	1	1	1	1	1	8
1	1	1	1	0	1	1	9

Tabela 5 – Tabela para decodificação dos displays do receptor

É realizada a verificação se este número de mesa já se encontra na fila. Caso positivo, o número é descartado, caso contrário, o programa insere a mesa na fila.

i. Transmissão Serial – RS232

O único dado a ser enviado pelo microcontrolador para o PC, através da porta serial, é o número da mesa que acabou de ser colocado na fila.

O modo de transmissão utilizado aqui é o modo 1, no qual o pino de recepção é o RXD e o de transmissão é o TXD. Em cada pacote são transmitidos 10 bits, sendo um o *Start* bit (nível 0) seguido de 8 bits de dados e o *Stop* bit (nível 1).

Na recepção, o *Stop* bit vai para o *bit* RB8 do registrador SCON.

Por exemplo, para enviar o número 38, primeiro transcrevemos este em hexadecimal 0x38. Portanto, o dado a ser transmitido é "00111000".

O dado a ser transmitido é invertido e se transforma em 00011100.

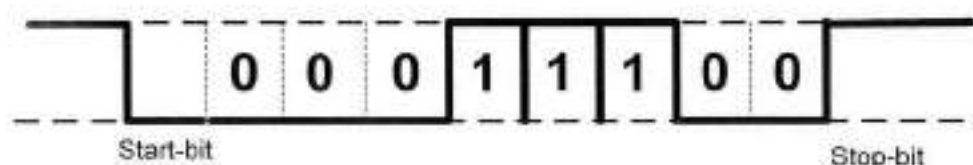


Figura 21 – Exemplo de transmissão serial assíncrona (RS232)

3.2.5 EAGLE

O software utilizado para desenhar o circuito e seu leiaute foi o EAGLE, da CadsoftTM, versão 4.16 Professional. Este software possui uma versão gratuita para estudantes e testes, porém a versão gratuita possui diversas limitações, como de tamanho do circuito e ferramentas desabilitadas, como o roteador automático de trilhas.

Este software foi escolhido por integrar tanto o editor de circuitos como o editor de leiautes. Outro motivo para sua utilização é a grande variedade sua biblioteca que possui a maioria dos componentes mais utilizados, além de possuir uma sessão de bibliotecas criadas pelos usuários em seu site, que facilita bastante o trabalho de desenvolvimento dos circuitos. Além disso, as bibliotecas podem ser editadas e parâmetros, como as distâncias entre os terminais dos componentes, podem ser alterados. A edição de bibliotecas foi um fator fundamental para efetuar o roteamento das trilhas na placa de decodificação que desenvolvemos.

Este programa possui 3 ambientes de desenvolvimento principais. O primeiro é o *Schematic*, onde os circuitos podem ser desenhados assim em programas de simulação de circuitos como o PSPICE, WORKBENCH, dentre outros. Neste modo de desenvolvimento é possível exportar os esquemas para o segundo ambiente de desenvolvimento, o *Board*, onde é possível criar e editar leiautes de placas, roteando as trilhas em até 16 camadas. O último

ambiente de desenvolvimento é um criador e editor das bibliotecas que são usados nos outros dois ambientes.

Os ambientes de desenvolvimento apresentam interfaces bastante amigáveis, porém teclas de atalho convencionalmente usadas em diversos programas não têm as mesmas funções neste, o que cria uma pequena dificuldade de adaptação no início. Outra deficiência deste programa é a atualização dos esquemas, que deve ser feita manualmente, pois resquícios de operações passadas permanecem na tela.

3.2.6 Circuito Completo

A montagem do circuito completo para decodificação e comunicação de dados foi elaborada utilizando os componentes dos blocos já descritos anteriormente.

A elaboração do circuito foi uma etapa rápida no processo de desenvolvimento deste hardware, porém o roteamento do leiaute foi umas das tarefas mais difíceis e demoradas na elaboração do circuito. A grande quantidade de componentes torna praticamente impossível não haver o cruzamento de uma ou mais trilhas. O roteamento não envolve apenas ligar as trilhas aos componentes, mas também posicionar os componentes de forma que a alocação das trilhas se torne mais simples.

O EAGLE converte automaticamente os circuitos desenhados no em leiautes, porém o posicionamento que este programa atribui não é o melhor possível, podendo dizer até que é péssimo, desta forma o roteamento se torna uma árdua tarefa para seu roteador automático. Os resultados com o roteador automático não foram satisfatórios e por isso motivo abandonamos esta funcionalidade e passamos a desenvolver o leiaute usando apenas nossas habilidades lógicas. O programa foi utilizado apenas para auxiliar no desenho do leiaute.

O microcontrolador que utilizamos no projeto não existia nas bibliotecas do EAGLE, porém existia um similar, da mesma família e fabricante, o

AT89S51. Por ser da mesma família ele possui a mesma pinagem do AT89S8252, que é o fator importante no desenho do esquema e do leiaute.

As figuras 14 e 15 mostram o leiaute do circuito desenvolvido e o esquema do circuito desenvolvido, respectivamente.

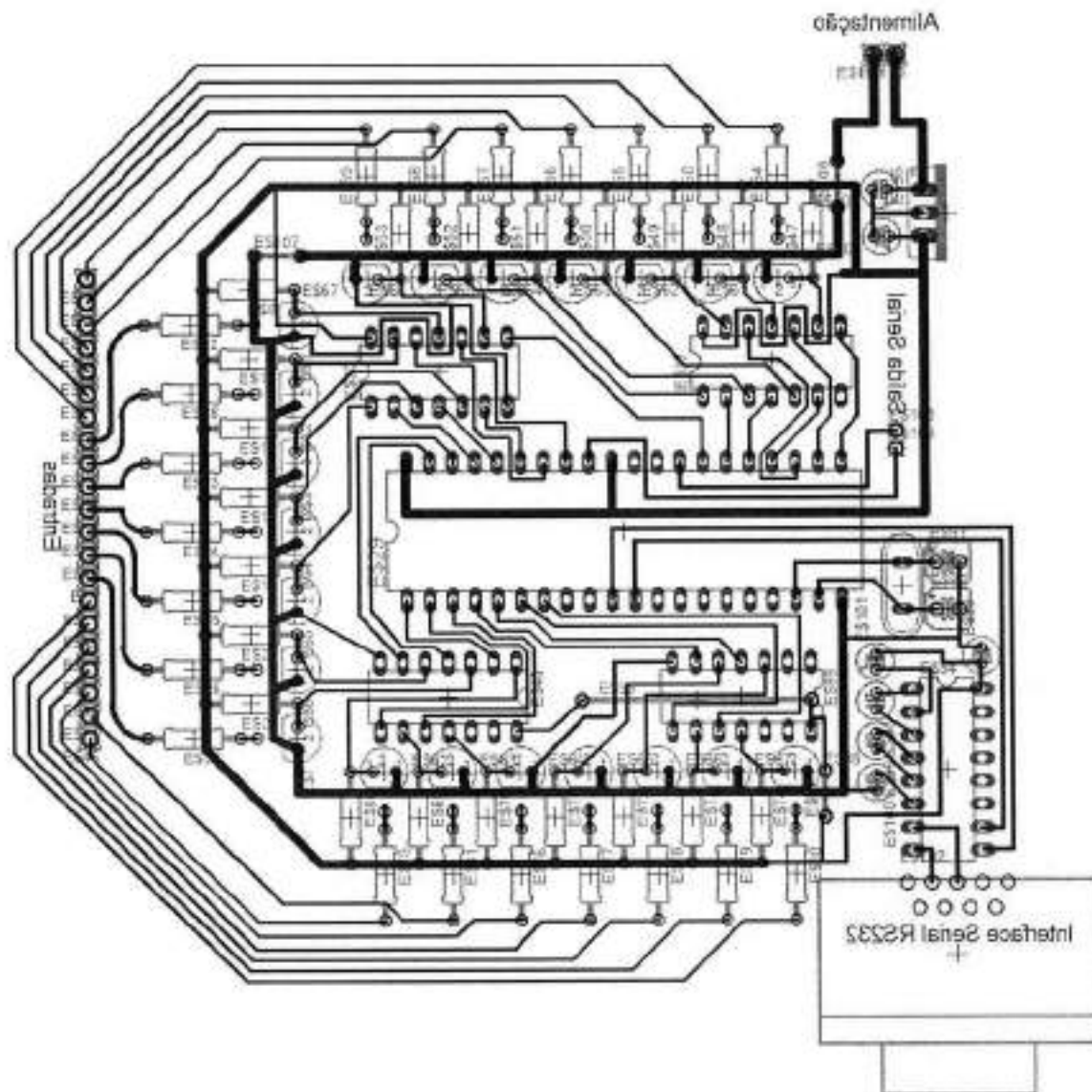


Figura 22 – Leiaute do módulo completo de decodificação e comunicação

3.3 Controle de Tela/Display

Este bloco irá controlar o funcionamento de um display que irá exibir as 4 primeiras posições da fila de mesas. A estrutura deste hardware também pode ser dividida e explicada através de diagramas de blocos, figura 24.

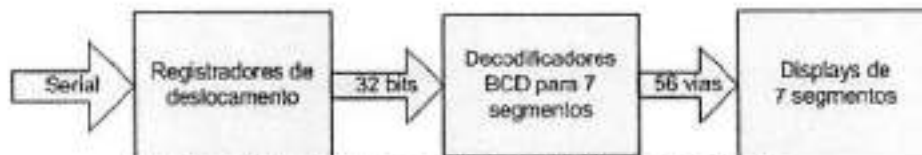


Figura 24 – Diagrama de blocos do Controle de Tela

Novamente neste ponto do trabalho irei dividir as explicações conforme a divisão de blocos, como havia feito anteriormente para o hardware de Decodificação e Controle de Comunicação.

3.3.1 Registradores de deslocamento

A recepção dos dados é feita através de registradores de deslocamento, onde a saída de um registrador serve como entrada para o registrador seguinte. Esta configuração é bastante utilizada em comunicação serial, sendo geralmente chamada de buffer de recepção, já que armazena os dados recebidos até que novos dados sejam fornecidos pela linha de comunicação.

O modelo de registrador de deslocamento escolhido foi o DM74LS164, da Fairchild SemiconductorsTM [5]. Este dispositivo é um registrador de deslocamento de 8 bits com entrada serial e saída paralela. Esta configuração é bastante útil para efetuar a ligação com os decodificadores que serão ligados aos displays de 7 segmentos. A frequência de operação do dispositivo é de 36 MHz, que supera em 50% o *clock* do microcontrolador.

A configuração interna deste dispositivo atende a idéia básica que tínhamos: *flip-flops* ligados em cadeia para formar uma linha de deslocamento de bits, figura 25. A encapsulamento pode ser visto na figura 26.

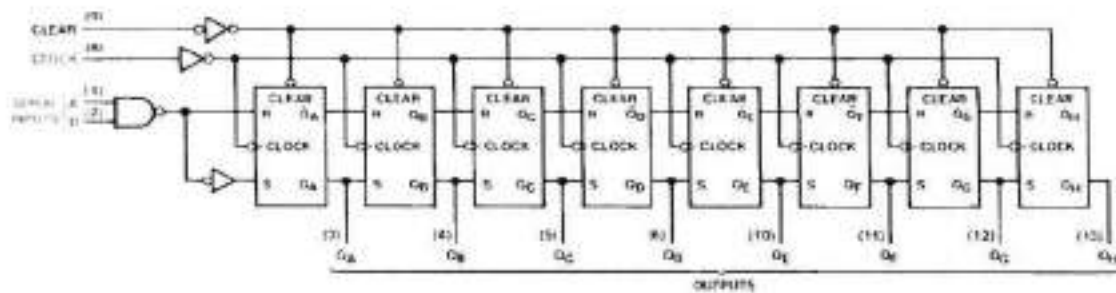


Figura 25 – Estrutura interna do DM74LS164

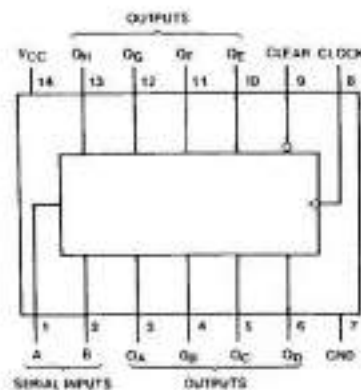


Figura 26 – Encapsulamento do DM74LS164

Com a configuração explicada, torna-se claro que cada vez que quisermos atualizar o display deveremos reenviar todos os dados, isto é, 4 bits para cada display de 7 segmentos. No primeiro caso citado, com 8 displays, necessitaríamos enviar 32 *bits* para fazer a atualização completa dos displays.

Este tipo de procedimento [15], atualizar todos os valores, poupa programação, pois de outra forma teríamos de criar um hardware que pudesse receber comandos e executá-los, como atualizar os valores de um único display. Apesar de ser uma solução que agrega maior inteligência ao módulo do display, não é uma solução de implementação tão simples, talvez podendo até usar um microcontrolador para executar este controle, o que aumentaria os custos do projeto.

A comunicação serial para o display será feita utilizando dois pinos do AT89S8252, sendo que um destes pinos irá gerar os sinais de dados e o outro um *clock* para o circuito receptor. Desta forma, criamos uma transmissão simplex síncrona, ao contrário da outra transmissão serial feita pelo microcontrolador, RS232, que é assíncrona.

3.3.2 Decodificadores BCD para 7 segmentos

Para este hardware também foram utilizados decodificadores de binário para 7 segmentos. Estes dispositivos além de fazer a conversão entre bases diferentes, servem como um *drivers* de corrente para os displays. O modelo de decodificador usado foi o DM74LS46, da Fairchild Semiconductors™ [5].

O decodificador DM74LS46 possui entre outras qualidades uma capacidade grande de fornecer corrente a cargas, isto é, existe a possibilidade de usar displays de tamanho maior, do que os que poderiam ser usados no caso de um acoplamento direto com os registradores.

Este dispositivo é constituído internamente por lógica booleana, isto é, as saídas no instante t dependem apenas das entradas no mesmo instante. A figura 27 mostra o esquema interno deste dispositivo.

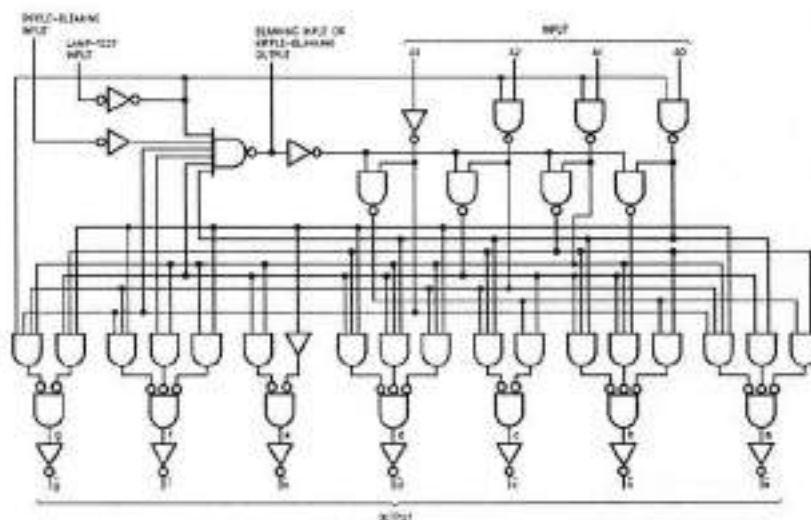


Figura 27 – Estrutura interna do DM74LS47

A tabela de decodificação do DM74LS46 é mostrada abaixo. Os valores são mostrados em base decimal, que representam a conversão dos 4 *bits* que são usados com entrada para este dispositivo.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	c	3	4	5	6	

Figura 28 – Relações de conversão do DM74LS47

O encapsulamento pode ser visto na figura 29.

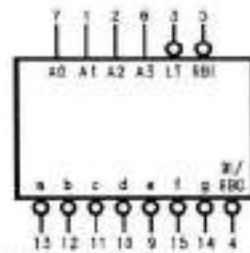


Figura 29 – Encapsulamento do DM74LS47

3.3.3 Displays de 7 segmentos

O display será formado por 8 displays de 7 segmentos, agrupados dois a dois para compor as dezenas. Com isto teremos a exibição das quatro primeiras posições da fila de espera. Caso exista necessidade de exibir os valores como centenas e não como dezenas o programa permite uma mudança rápida de estrutura, dando substrato ao funcionamento deste modo, desde que este esteja implementado em hardware corretamente, isto é, 3 displays de 7 segmentos para formar cada grupo de números.

4 Software

4.1 Tecnologias Utilizadas

Este tópico se refere à definição das tecnologias envolvidas no desenvolvimento do software. As seguintes decisões foram tomadas nestes sentido:

- **Linguagem de Programação:** Pela sua portabilidade, o Java foi escolhido como linguagem de programação para este projeto. Para aparelhos móveis, poder-se-ia utilizar usar a versão Java ME (*Mobile Edition*); entretanto, devido ao possível aumento de complicações técnicas com a utilização desta tecnologia, a versão J2SE (*Standard Edition*) foi utilizada durante o desenvolvimento.
- **Protocolos de Comunicação:** a troca de mensagem entre sistemas se dá por meio de arquivos no formato padrão XML, transportados via protocolo TCP. Tal decisão foi tomada de forma a padronizar as comunicações, facilitando integração com outros sistemas. Além disto
- **Ferramentas de Desenvolvimento:** como plataforma de desenvolvimento de Software, o IDE Netbeans foi escolhido. O mesmo preenche as necessidades do projeto e oferece uma interface de programação de interfaces de fácil uso, auxiliando enormemente no projeto. Para a geração de diagramas UML utilizados, foi escolhido o StarUML, ferramenta de uso livre (*OpenSource*).
- **Base de dados:** Inicialmente, foi-se decidido que o MySQL seria utilizado tanto para os sistemas cliente quanto servidor. Entretanto, devido à baixa complexidade da base de dados presente no dispositivo móvel, foi decidido que este último usaria arquivos-texto para armazenar dados sobre produtos, pedidos, etc. O MySQL foi mantido para o sistema servidor.

4.2 Descrição completa dos sistemas

Conforme especificação do Projeto, dois tipos distintos de sub-sistemas (aqui chamados de Sistemas, para a abreviar a notação) se comunicam de forma a fornecer os serviços propostos, os Sistemas Cliente e Servidor. A figura a seguir ilustra a relação entre eles:

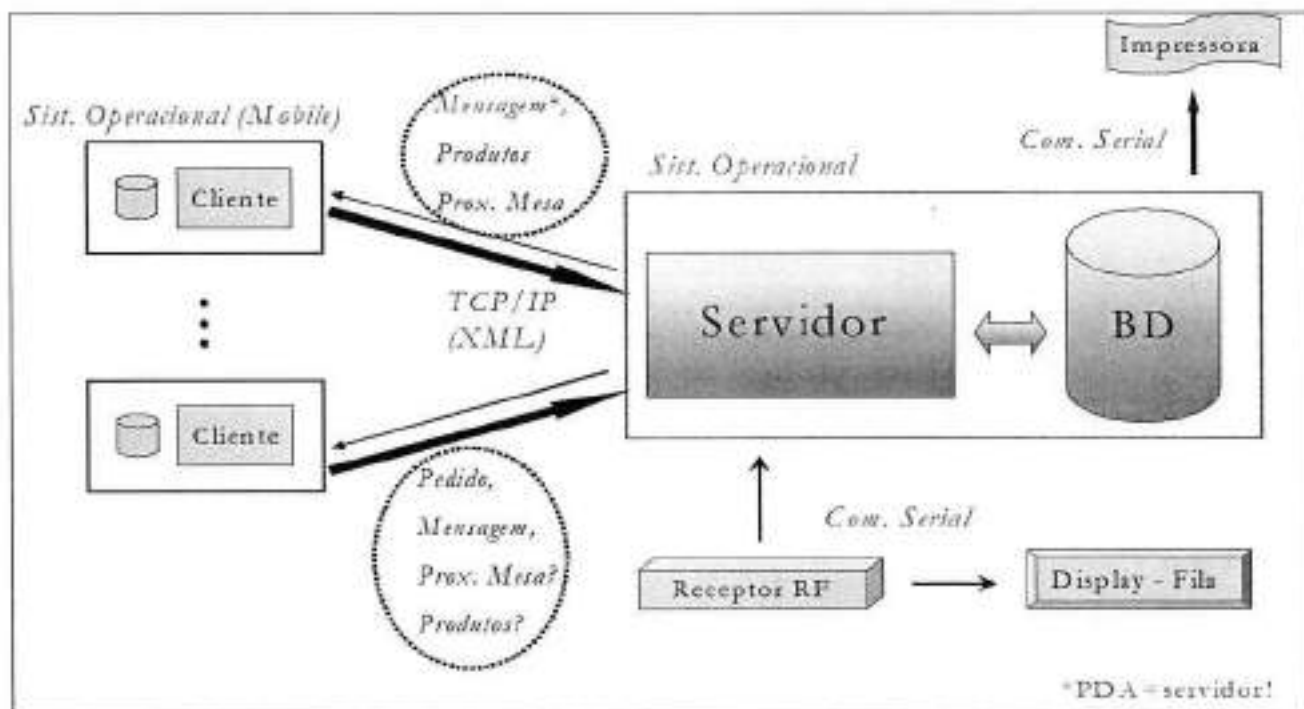


Figura 30 – Relacionamento entre blocos do sistema

4.2.1 Sistema Servidor

a. Funções

O software rodando no servidor é responsável pela integração das diferentes interfaces de comunicação (Aparelhos Móveis, Call Bell, Impressoras, Displays, etc.) e também pela gestão da base de dados central. Fisicamente, este sistema localiza-se em um computador pessoal, que deve ter a infra-estrutura básica para utilização do mesmo (Java Virtual Machine e MySQL, basicamente). Devido à portabilidade provida pelo Java, o aplicativo torna-se altamente independente da plataforma utilizada.

Este sistema deve ser capaz de receber os seguintes tipos de documentos provenientes de Sistemas Clientes:

- **Pedidos:** criados nos aplicativos móveis, os mesmos são enviados ao Servidor, que deve armazená-los no banco de dados, disponibilizá-los na interface gráfica e também imprimi-los em uma mini-impressora. Desta forma, fica fácil de avisar os garçons do momento em que um determinado pedido ficou pronto, além de permitir que análises estatísticas sejam realizadas nos mesmos.
- **Requisição das listas de Produtos e Extras:** como toda a administração da base de dados de produtos e extras fica a cargo do sistema servidor, o mesmo deve ser capaz de fornecer as listas atualizadas dos mesmos, sob demanda. Uma observação importante é que esta transferência de dados deve ser o mais transparente possível ao usuário do Sistema Servidor.
- **Requisição da próxima Mesa da fila:** conforme especificado, a fila de mesas esperando por atendimento fica disponível no Servidor, de forma os usuários do mesmo sejam capazes de alterá-la (adicionando ou removendo uma mesa, por exemplo). Além disto, os usuários do Sistema Cliente podem requisitar o número da próxima mesa da fila que deve ser atendida, removendo-a da fila (pois presume-se que o garçom que a requisitou irá atendê-la) e, desta forma, da interface gráfica.
- **Mensagem:** ambos os Sistemas são capazes de enviar mensagens de texto uns aos outros. Portanto, é necessário que o Servidor seja capaz de receber estas mensagens, disponibilizá-las na interface, armazená-las e também exibir uma notificação de recebimento de mensagem na tela, de forma que o usuário no Servidor fique rapidamente ciente da recepção.

Além dos Documentos XML que podem ser trocados, há também os números de mesa que devem ser adicionados à fila recebidos via Canal Serial (provenientes do Receptor RF). Neste caso, os números devem ser

adicionados ao fim da fila, de maneira semelhante ao que é feito pelo microcontrolador, e os números exibidos na interface devem ser atualizados.

b. Multi-threading

Devido ao grande número de usuários que podem tentar acessar o Sistema Servidor ao mesmo tempo, bem como à grande quantidade de Documentos diferentes que podem ser trocados, faz-se necessário um mecanismo eficiente de tratamento destes Documentos, ou a performance do Sistema seria muito prejudicada. A solução para este tipo de problema é o estabelecimento de um Sistema Servidor MultiThread, conforme ilustrado na figura seguinte:

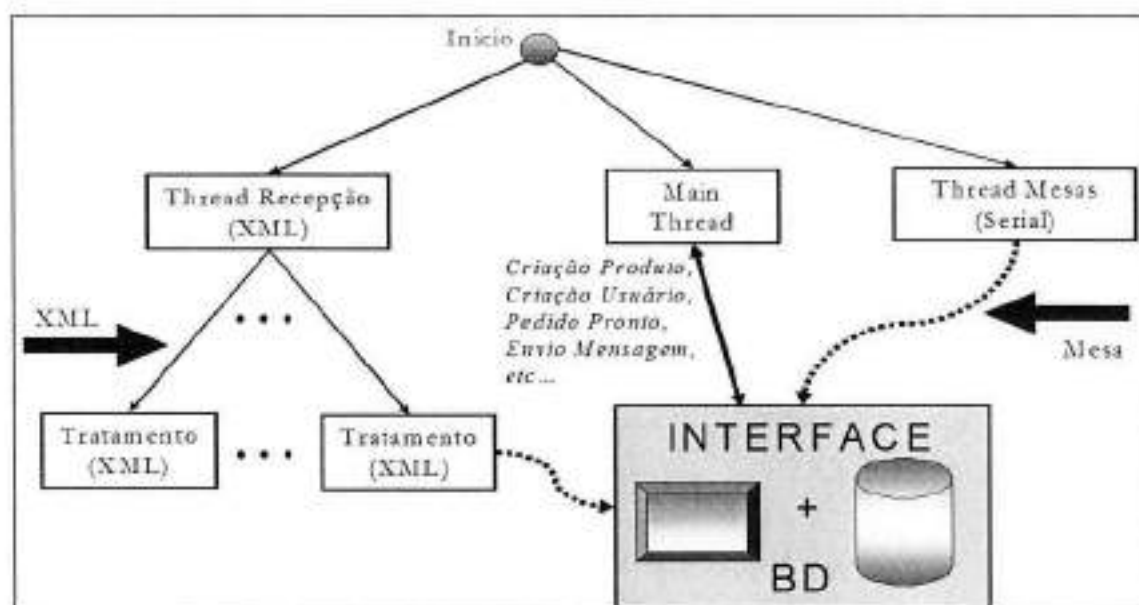


Figura 31 – Servidor MultiThread Implementado

O funcionamento do mesmo é o seguinte: na inicialização, três processos ("Threads") diferentes são disparados: o primeiro ("Main Thread") trata tanto da geração quanto do tratamento dos eventos da interface do sistema; o segundo ("Thread Recepção XML") se encarrega de escutar uma determinada porta do sistema por conexões; já o terceiro se encarrega de tratar as mesas recebidas via canal serial. A porta padrão para recepção de documentos XML é a 7070, enquanto o envio é feito pela porta 7071 (o

contrário ocorre no Sistema Cliente, conforme será explicado na descrição do mesmo mais adiante).

Quando uma nova conexão TCP é efetuada, um novo Thread é gerado ("Tratamento XML"), o qual se encarrega da recepção do Documento e tratamento do mesmo. O tratamento exato depende do tipo de Documento recebido, conforme descrito anteriormente: pode ser necessário instanciar um objeto a partir das informações contidas no mesmo (no caso de recepção de Pedido ou Mensagem), preparação e envio de resposta (no caso de requisições de lista de Produtos ou próxima mesa da fila), etc.

Graças ao esquema descrito, o Sistema Servidor amplia sua capacidade de recepção e envio de Documentos, tratando-os de forma a não "perturbar" o usuário quando não for necessário.

c. Interação com o Usuário

Além das funcionalidades automáticas de tratamento de eventos descritas anteriormente, o Sistema Servidor também possui uma interface gráfica para interação com o usuário. Este último pode ser um administrador do sistema possuindo, portanto, direitos de acesso a todas as funcionalidades, ou então um usuário com menor nível de acesso.

- **Administrador:** pode, após *login*, editar as bases de dados de produtos, usuários e também acessar dados de performance. Os dados desta última podem ser analisados de forma amigável, através de gráficos e cálculos estatísticos padronizados.
- **Usuário:** é responsável por atividades funcionais, como enviar mensagens aos sistemas cliente (as quais incluem avisos de que pedidos estão prontos, mensagens de texto, etc). Pode também criar e editar pedidos, além de ser responsável por informar quando os mesmos ficam prontos.

Obs.: Conforme descrito, será possível gerar um pedido diretamente da cozinha, uma funcionalidade que pode ser útil como backup dos Palms (provavelmente não utilizada em operação normal)

4.2.2 Sistema Cliente

a. Funções

O Sistema Cliente corresponde ao aplicativo rodando no dispositivo móvel (PDA), e que se comunica com um servidor através de uma rede sem fio.

Sendo seus usuários os garçons do restaurante, este sistema tem como principal finalidade criar e enviar ao Sistema Servidor os Pedidos efetuados, de uma maneira rápida e eficiente. Além disto, deve também ser capaz de requisitar ao Servidor as listas de Produtos e Extras atualizadas, de maneira a manter sua base de dados sincronizada com a base oficial.

Como funcionalidade adicional, é incluída a possibilidade de requisitar a próxima mesa da fila que deve ser atendida, aumentando a eficiência de seus usuários. Eles passam a estar no "lugar certo, na hora certa".

De forma semelhante ao Sistema Servidor, o Sistema Cliente é também capaz de enviar e receber mensagens de texto. É interessante notar que, em operações de recepção de Mensagens o nomeado Sistema Cliente passa a agir como "servidor" (de acordo com a nomenclatura de arquiteturas cliente-servidor), ou seja, é ele quem aceita uma conexão proveniente do Sistema Servidor. Portanto, é necessário que o Sistema Cliente também fique na escuta de uma porta e espere por conexões de maneira semelhante ao Sistema Servidor, sendo assim dotado de um thread específico para tratar as mesmas conforme mostra a figura a seguir:

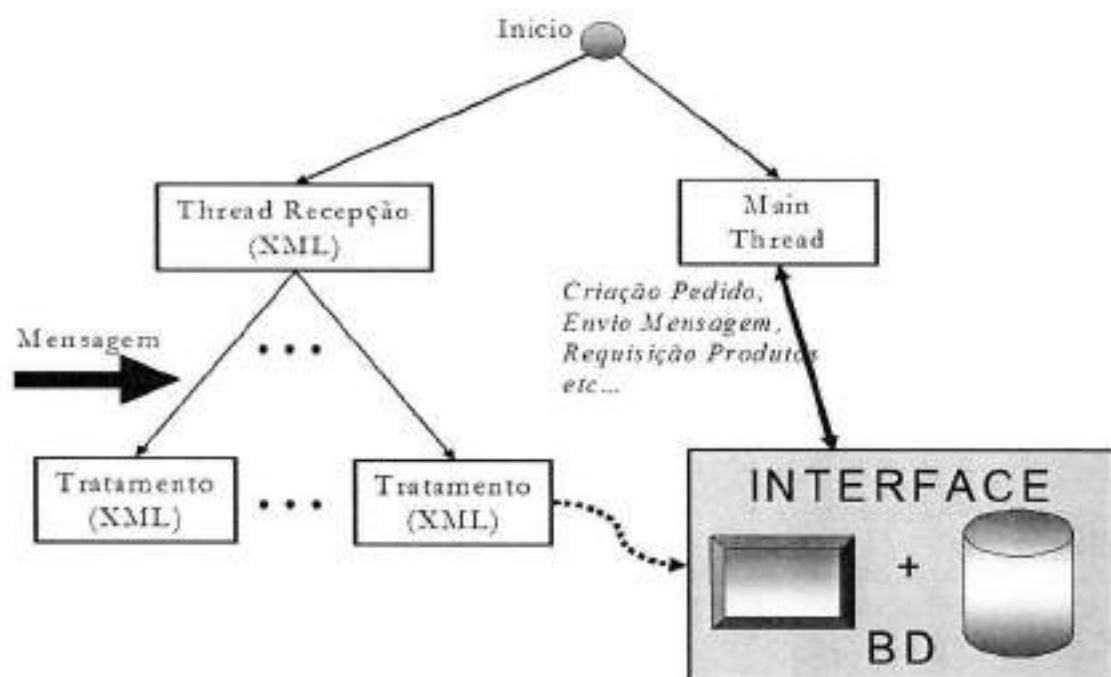


Figura 32 – Cliente MultiThread Implementado

Apesar deste detalhe de nomenclatura, ainda serão usados os termos "Sistema Servidor" e "Sistema Cliente".

b. Eficiência e Usabilidade

Um ponto importante e de grande preocupação ao longo do projeto é a facilidade de uso do aplicativo no momento de tomada de pedido. O número de clicks necessários para criar um Pedido, a facilidade de encontrar os produtos que devem ser adicionados ao mesmo, bem como a facilidade de entrada de dados (principalmente dígitos decimais) estão entre as maiores preocupações. Desta forma, três soluções de interface foram providas:

- **Montagem do Pedido:** a geração de um novo pedido é bem direta. Tem-se uma única tela na qual são entrados os números de mesa e/ou comanda e na qual os itens presentes no pedido sendo criado podem ser visualizados e editados. Na mesma tela é também possível requisitar a próxima mesa da fila (que automaticamente preenche o campo "Mesa" descrito) e iniciar a adição de um item ao pedido. Este último

exige mais alguns passos devido à possibilidade de adição ou remoção de ingredientes (chamados aqui de Extras). Pode-se observar na figura seguinte um modelo de interface:



Figura 33 – Modelo de Tela para criação de Pedidos

- Organização dos Produtos:** a primeira solução, simples porém altamente útil, é a organização dos mesmos por ordem alfabética. Além disto, no momento da criação do produto (ou em sua edição), é possível atribuir um ou vários "tipos". Desta forma, pode-se realizar um filtro usando o tipo de produto como critério, garantindo uma exibição mais limpa da lista de produtos disponíveis. A figura a seguir ilustra a idéia.



Figura 34 – Filtros por Tipos

- **Entrada de números:** Boa parte dos usuários (principalmente os menos experientes) de dispositivos móveis como PDAs têm uma certa dificuldade de utilização das ferramentas disponíveis para entrada de texto e números. Como o Sistema Cliente prevê um grande número de entradas numéricas (número de mesa, número de comanda, quantidade de um produto...), decidiu-se fornecer uma interface amigável e auto-explicativa ao usuário, que pode então digitar números utilizando a própria interface do aplicativo. A figura a seguir mostra a interface desenvolvida:



Figura 35 – Interface otimizada para entrada de valores numéricos

c. Questões de segurança e integridade

Para utilização do Sistema Cliente é exigido o fornecimento de nome de usuário e senha válidos, os quais são utilizados em cada mensagem enviada ao servidor. Desta maneira, cada usuário poderá ser facilmente identificado no momento da comunicação, permitindo análise de performance dos mesmos e também garantindo maior segurança ao sistema (evitando por exemplo mensagens enviadas por Palms que não sejam os usados pelos garçons no estabelecimento). A ligação unívoca entre Palm e usuário conforme discutido cria, além disto, uma clara necessidade por parte do usuário de cuidar bem da sua máquina.

Além disto, de maneira a garantir um Banco de Dados sempre atualizado, o ideal é que os Sistemas Clientes atualizem suas listas de produtos e extras sempre que forem iniciados.

Estas e outras opções (como número máximo de mensagens e pedidos armazenados no histórico para consulta) podem ser configuradas a qualquer momento, via interface gráfica.

4.3 Descrição formal do Software

4.3.1 Atores

- **Garçom:** Qualquer usuário utilizando o aplicativo cliente (no dispositivo móvel).
- **Administrador no Servidor:** Usuário com direitos de administrador, o qual deve estar logado como tal no Servidor.
- **Usuário no Servidor:** Qualquer usuário utilizando o aplicativo no servidor.
- **Sistema Cliente:** O aplicativo cliente.
- **Sistema Servidor:** O aplicativo servidor.

4.3.2 Glossário

- **Administrador no Servidor:** vide definição em "Atores"
- **Base de Dados:** Conjunto de informações sobre um determinado tipo de Classe tratado no sistema
- **Classe:** abstração utilizada para representar um grupo de elementos de um mesmo tipo (exemplo: Produtos, Mensagens, Pedidos, dentre outros).
- **Cliente:** cliente do Estabelecimento

- **Comanda:** O número que identifica certo Cliente dentro do Estabelecimento
- **Documento:** Qualquer conjunto de dados trocado entre Sistemas Cliente e Servidor (esta é uma nomenclatura corrente na definição para se referir a arquivos no formato XML)
- **Estabelecimento:** Estabelecimento comercial utilizando o aplicativo, podendo ser um bar, um restaurante, etc.
- **Extra:** tipo de produto que podem ser adicionado ou removido de um Produto. Exemplo: "Cebola"
- **Garçom:** vide definição em "Atores"
- **Interface:** usado para referir-se às diversas telas disponíveis para interação com os usuários do aplicativo
- **Item:** uma entrada em Pedido, consistindo de uma certa quantidade de Produtos após adição e/ou remoção de Extras
- **Mensagem:** Qualquer mensagem trocada entre o Sistema Cliente e o Sistema Servidor (pode-se tratar de mensagens de texto, mensagens de erro, etc.)
- **Mesa:** O número da mesa presente no Estabelecimento
- **Menu Admin:** Usado em geral para indicar a necessidade de login (identificação de usuário + senha) para poder acessar um funcionalidade do aplicativo
- **Objeto:** abstração utilizada para referir-se a um representante específico de uma Classe
- **Pedido:** conjunto de Itens associados a uma Mesa e/ou a um número de Comanda específico(s)
- **Pedido em Aberto:** Pedido que ainda não está pronto
- **Produto:** produtos disponíveis diretamente para a venda. Exemplo: "Pizza de Calabresa";
- **Sistema Cliente:** vide definição em "Atores"
- **Sistema Servidor:** vide definição em "Atores"
- **Usuário no Servidor:** vide definição em "Atores"

4.3.3 Relatório de Revisão dos Casos de Uso

- **14/09/2006:**
 - Revisão completa dos Casos de Uso;
 - Re-formatação do documento;
 - Inclusão do Caso de Uso 19: Mover pedido de "grupo aberto" a "grupo pronto" e vice-versa;
 - Alteração no nome do Caso de Uso 4 para explicitar as Classes que podem ser visualizadas

- **15/11/2006:**
 - Revisão completa dos Casos de Uso;
 - Cancelamento dos Casos de Uso 3, 5 e 6;
 - Alteração do Caso de Uso 4 para incluir Caso de Uso 10, o qual foi então cancelado;
 - Simplificação de Caso de Uso 8, de forma a cobrir apenas um aspecto da edição do Pedido: remoção de Itens. Considera-se que o cancelamento e visualização do mesmo não precisam ser descritos em detalhes (ações cobertas pelo CRUD⁹)

4.3.4 Casos de Uso

I. Sistema Cliente: Palm

- a. Sincronização dos dados
- b. Edição de preferências
- c. <CANCELADO> Edição de ordem preferencial de exibição da lista de itens
- d. Visualizar detalhes do Produto/Pedido/Mensagem
- e. <CANCELADO> Adição de Produto a grupo personalizado

⁹ CRUD - Create, Read, Update and Delete.

- f. <CANCELADO> Remoção de Produto de grupo personalizado
- g. Criação de Pedido e adição de Itens a ele
- h. Edição de Itens em pedido
- i. Envio de pedido
- j. <REMOVIDO: VER CU - 4>Visualização de detalhes dos últimos pedidos enviados
- k. Requisição da próxima mesa da fila

II. Sistema Servidor : PC

- **Funções de Admin (acessadas a partir de login):**

- l. Inc1: Validação e interpretação de mensagem recebida por Sistema Servidor
- m. Edição de preferências – Sistema Servidor
- n. Inclusão/Edição/Exclusão de usuário autorizado (Garçom)
- o. Inclusão/Edição/Exclusão de dados de produtos (*obs.: inclui edição de valores no estoque e também da ordem padrão para exibição de produtos*)
- p. Inclusão/Edição/Exclusão de dados de performance

- **Comandos via interface gráfica:**

- q. Análise/Exportação de dados de estoque
- r. Análise/Exportação de dados de performance
- s. Informar que pedido está pronto para ser servido
- t. Mover pedido de "grupo aberto" a "grupo pronto" e vice-versa
- u. Envio de mensagem a garçom
- v. Criação de pedido e adição de itens a ele
- w. Visualização e Edição de pedido
- x. Executar manutenção da base de dados

- **Resposta a RF**

y. Colocar mesa na fila FIFO

4.3.5 Detalhamento dos casos de uso

a. Sincronização dos dados

<i>Criado por:</i>	Marcos Simplicio
<i>Data de Criação:</i>	15/07/2006
<i>Última Atualização</i>	Marcos Simplicio
<i>Realizada por:</i>	
<i>Data da Última Atualização:</i>	15/11/2006
<i>Atores:</i>	Garçom, Sistema Servidor
<i>Descrição:</i>	Garçom requisita atualização da base de dados no Sistema Cliente
<i>Prioridade:</i>	1
<i>Estado</i>	Definido
<i>Pré-condições:</i>	<ol style="list-style-type: none">1. Ator conectado no Sistema Cliente2. Haver conectividade com o Sistema Servidor.3. Sistema Servidor está pronto para receber e enviar mensagens
<i>Pós-condições:</i>	<ol style="list-style-type: none">1. Sistema Cliente volta à tela inicial, mostrando resultado da ação

<i>Fluxo Normal:</i>	<ol style="list-style-type: none">1. Garçom requisita atualização da Base de Dados2. Sistema Cliente envia requisição a Sistema Servidor3. <u>Incl1</u>4. Sistema Servidor envia dados para atualizar base de dados do Sistema Cliente [A1]5. Ao final da recepção de dados, Sistema Cliente mostra mensagem de confirmação a Usuário
<i>Fluxo Alternativo:</i>	<ul style="list-style-type: none">• [A1]3.1 Sistema Servidor envia mensagem de erro, pois não conseguiu interpretar a mensagem3.2 Sistema Cliente recebe mensagem de erro e redireciona usuário à tela de login, na qual ela é exibida
<i>Casos de uso incluídos</i>	<ul style="list-style-type: none">• Incl1: Validação e Interpretação de mensagem recebida

b. Edição de preferências – Sistema Cliente

<i>Criado por:</i>	Marcos Simplicio
<i>Data de Criação:</i>	15/07/2006
<i>Última Atualização</i>	Marcos Simplicio
<i>Realizada por:</i>	

<i>Data da Última Atualização:</i>	15/11/2006
<i>Atores:</i>	Garçom
<i>Descrição:</i>	Ator acessa e edita as opções disponíveis, salvando-as no Sistema Cliente.
<i>Prioridade:</i>	3
<i>Pré-condições:</i>	1. Ator conectado no Sistema Cliente
<i>Pós-condições:</i>	1. Ator de volta à tela onde está a opção de edição

<i>Fluxo Normal:</i>	<ol style="list-style-type: none"> 1. Ator acessa opção de Edição de preferências 2. Sistema Cliente disponibiliza opções que podem ser modificadas e os comandos "confirmar" e "cancelar" 3. Ator seleciona e edita as opções desejadas: [Ext1] <ul style="list-style-type: none"> ▪ Dados de login no Sistema; ▪ Número máximo de pedidos que podem ser visualizados após envio; ▪ Outros 4. Ator confirma as alterações [A1] [E1] 5. Sistema Cliente aplica as preferências conforme selecionado por Ator 6. Sistema Cliente exibe mensagem de confirmação e retorna à Tela Principal
<i>Fluxo Alternativo:</i>	<ul style="list-style-type: none"> • [A1] <ol style="list-style-type: none"> 4.1 Ator cancela alterações 4.2 Retornar ao Passo 6 do Fluxo Principal
<i>Exceções:</i>	<ul style="list-style-type: none"> • [EI] – Opção não pode ser editada por alguma razão <ol style="list-style-type: none"> 4.1 Sistema Cliente exibe na tela razão da impossibilidade 4.2 Retornar ao Passo 2 do Fluxo Principal

c. <CANCELADO> Edição de sua ordem preferencial de exibição da lista de itens

<i>Criado por:</i>	Marcos Simplicio
<i>Data de Criação:</i>	15/07/2006
<i>Última Atualização</i>	Marcos Simplicio
<i>Realizada por:</i>	
<i>Data da Última Atualização:</i>	15/11/2006
<i>Atores:</i>	Garçom
<i>Descrição:</i>	Ator edita a ordem preferencial de apresentação dos itens na tela, sobrepondo ordenação padrão
<i>Prioridade:</i>	CANCELADO
<i>Pré-condições:</i>	1. Ator conectado no Sistema Cliente
<i>Pós-condições:</i>	1. Ordem de apresentação dos itens na tela alterada de acordo com preferência do ator

<i>Fluxo Normal:</i>	1. Ator seleciona ícone de um item
----------------------	------------------------------------

	<ol style="list-style-type: none"> 2. Sistema Cliente disponibiliza console de navegação, que permitem ao ator mover o ícone para esquerda, direita, baixo e cima, em um passo ou até o limite da tela (topo, fundo, limite esquerdo e limite direito) 3. Ator leva item à posição desejada dentro dos limites estabelecidos pelo sistema 4. Sistema Cliente salva a nova posição dos itens, que passa a ser a ordem mostrada ao ator deste momento em diante
Fluxo Alternativo:	<i>Não há</i>
Exceções:	<i>Não há</i>

d. Visualizar detalhes do Produto/Pedido/Mensagem

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator acessa os detalhes de um item no Sistema Cliente.
Prioridade:	2
Pré-condições:	<ol style="list-style-type: none"> 1. Ator conectado no Sistema Cliente 2. Ator seleciona opção de visualização de uma das Classes disponíveis via Interface
Pós-condições:	<ol style="list-style-type: none"> 1. Ator de volta à Interface de visualização de Classes

Fluxo Normal:	<ol style="list-style-type: none"> 1. Sistema Cliente disponibiliza Objetos da Classe selecionada, passíveis de serem visualizados 2. Ator seleciona Objeto desejado 3. Sistema Cliente exibe detalhes disponíveis para o Objeto selecionado e opção de retorno à tela anterior 4. Ator retorna à tela anterior
Fluxo Alternativo:	<i>Não há</i>
Exceções:	<i>Não há</i>

e. <CANCELADO> Adição de Produto a grupo personalizado

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator seleciona Produto e adiciona-o à lista personalizada de alimentos ou de bebidas, dependendo do tipo de Produto selecionado
Prioridade:	2

Pré-condições:	<ol style="list-style-type: none"> 1. Ator conectado no Sistema Cliente 2. Grupo de Produtos selecionado não é um dos grupos personalizados
Pós-condições:	<ol style="list-style-type: none"> 1. Produto selecionado adicionado à lista personalizada adequada a seu tipo

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator seleciona Produto via Interface 2. Sistema Cliente disponibiliza opção de adicionar Produto a grupo personalizado, de acordo com seu tipo (alimento ou bebida, por exemplo) 3. Ator requisita adição do Produto à lista personalizada 4. Sistema Cliente adiciona Produto selecionado à lista personalizada adequada (usando um campo no banco de dados para este fim) 5. Sistema Cliente envia mensagem de confirmação a Ator
Fluxo Alternativo:	<i>Não há</i>
Exceções:	<i>Não há</i>

f. <CANCELADO>Remoção de Produto de grupo personalizado

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator seleciona Produto e remove-o da lista personalizada de alimentos ou de bebidas, dependendo do grupo selecionado
Prioridade:	2
Pré-condições:	<ol style="list-style-type: none"> 1. Ator conectado no Sistema Cliente 2. Grupo de Produtos selecionado é um dos grupos personalizados
Pós-condições:	<ol style="list-style-type: none"> 1. Produto selecionado removido da lista personalizada (e apenas dela)

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator seleciona Produto via Interface 2. Sistema Cliente disponibiliza opção de remover Produto do grupo personalizado selecionado 3. Ator requisita remoção do Produto da lista personalizada 4. Sistema Cliente remove Produto selecionado da lista personalizada adequada (usando um campo no banco de dados para este fim) 5. Sistema Cliente envia mensagem de confirmação a Ator
Fluxo Alternativo:	<i>Não há</i>
Exceções:	<i>Não há</i>

g. Criação de Pedido e adição de Item a ele

Criado por:	Marcos Simplicio
Data de Criação:	30/05/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	21/06/2006
Atores:	Garçom
Descrição:	Ator escolhe Produtos em lista e os adiciona a Pedido com as quantidades desejadas, além da inclusão / exclusão de extras do Produto em questão
Prioridade:	1
Pré-condições:	1. Ator conectado no Sistema Cliente 2. Ator seleciona opção de criação de Pedido, via Interface (isto faz com que um novo Pedido seja gerado)
Pós-condições:	1. Ator de volta à Interface de adição de Itens a Pedido

Fluxo Normal:	<ol style="list-style-type: none">1. Ator requisita adição de Item ao Pedido atual, via interface.2. Sistema Cliente disponibiliza lista de Produtos, organizados de forma a facilitar visualização3. Ator seleciona um Produto4. Sistema disponibiliza interface para definição da quantidade desejada do produto e para inclusão / exclusão de extras5. Ator ajusta quantidade para o valor desejado e adiciona / exclui extras caso assim deseje.6. Ator confirma alterações [A1]7. Sistema Cliente adiciona Item criado ao pedido atual8. Sistema Cliente envia mensagem de confirmação de adição do Item
Fluxo Alternativo:	[A1] 5.1 Ator requisita o cancelamento da adição de Item 5.2 Fim do Caso de Uso
Exceções:	<i>Não há</i>

h. Edição de Itens em Pedido

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator edita os de Itens no Pedido atual
Prioridade:	1

Pré-condições:	1. Ator conectado no Sistema Cliente 2. Ator requisita edição do Pedido atual
Pós-condições:	1. Sistema Cliente na de edição do Pedido atual

Fluxo Normal:	<ol style="list-style-type: none"> 1. Sistema exibe tela na qual figuram os Itens presentes no pedido. Sistema Cliente disponibiliza opções para remoção individual de itens, salvamento ou cancelamento das alterações, bem como cancelamento do pedido. 2. Ator seleciona Item para edição [A1] 3. Sistema Cliente disponibiliza interface para edição do Item 4. Ator realiza as alterações desejadas 5. Ator requisita o salvamento das alterações efetuadas [A2], [E1] 6. Sistema Cliente salva as alterações feitas no pedido 7. Sistema Cliente exibe mensagem de confirmação 8. Sistema Cliente retorna à tela de edição de Pedido
Fluxo Alternativo:	<ul style="list-style-type: none"> • [A1] <ol style="list-style-type: none"> 2.1 Ator requisita remoção de um item específico do pedido, via interface 2.2 Sistema Cliente remove item do pedido e também da tela 2.3 Retornar ao passo 7 do fluxo principal • [A2] <ol style="list-style-type: none"> 5.1 Ator requisita cancelamento das alterações 5.2 Sistema Cliente não salva as alterações efetuadas no pedido 5.3 Retornar ao passo 7. do Fluxo Normal
Exceções:	<ul style="list-style-type: none"> • [E1] - <i>Ator remove todos os itens do pedido e então requisita salvamento das alterações</i> <ol style="list-style-type: none"> 5.4 Sistema avisa a remoção de todos os itens do pedido e informa que pedido será cancelado, oferecendo ao ator a possibilidade de cancelar as alterações (retornar a fluxo alternativo [A2]) ou confirmar cancelamento (Pedido é cancelado e chega-se ao fim do Caso de Uso)

i. Envio de pedido

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator envia Pedido atual ao Sistema Servidor
Prioridade:	1

Pré-condições:	<ol style="list-style-type: none"> 1. Ator conectado no Sistema Cliente 2. Existem Itens no pedido corrente
Pós-condições:	<ol style="list-style-type: none"> 1. Pedido enviado ao Sistema Servidor 2. Todos os Itens removidos do Pedido atual

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator requisita o envio do pedido, via interface 2. Sistema gera arquivo em formato padrão com os itens presentes no pedido, número da mesa e outras informações necessárias, enviando-o ao Sistema Servidor via protocolo de rede [E1] 3. Sistema Servidor recebe arquivo 4. <u>[Inc1]</u> 5. Sistema Servidor checa a validade dos dados (número de mesa correto, produtos presentes no estoque, etc) presentes no pedido 6. Caso o pedido esteja correto, Sistema Servidor disponibiliza pedido em sua interface ("pedido ativo"), imprime ticket com seus detalhes e envia mensagem de confirmação de recepção ao Sistema Cliente [A1] 7. Sistema Cliente recebe mensagem de confirmação [E2] 8. Sistema Cliente armazena pedido enviado em banco de dados, removendo-o do grupo "pedido corrente" 9. Caso haja mais pedidos armazenados que o número N máximo, Sistema Cliente apaga os mais antigos de forma a manter apenas os N mais recentes
Fluxo Alternativo:	<ul style="list-style-type: none"> • [A1] 6.1 Caso contrário, Sistema Servidor envia mensagem descrevendo erro encontrado 6.2 Sistema Cliente recebe mensagem de erro e exibe-a, não arquivando o pedido que acaba de ser enviado 6.3 Fim do Caso de Uso
Exceções:	<ul style="list-style-type: none"> • [E1] – <i>Algum dos campos presentes no Pedido possui um valor inválido (ambos número de Mesa e de Comanda em branco, por exemplo)</i> 2.1 Sistema Cliente exibe mensagem de erro e não envia Pedido ao Sistema Servidor 2.2 Fim do Caso de Uso • [E2] - <i>Sistema Cliente recebe mensagem de erro no envio do Pedido (protocolo TCP)</i> 7.1 Mensagem de problema na conexão é exibida 7.2 Fim do Caso de Uso
Casos de uso incluídos	<ul style="list-style-type: none"> • Inc1: Validação e Interpretação de mensagem recebida

j. <REMOVIDO: VER CU-4> Visualização de detalhes dos últimos pedidos enviados

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator
Prioridade:	2
Pré-condições:	1. Ator conectado no Sistema Cliente 2. Existir pedidos enviados armazenados no Sistema Cliente
Pós-condições:	1. Ator visualiza um dos N pedidos já enviados armazenados no sistema

Fluxo Normal:	1. Ator requisita visualização de pedidos já enviados 2. Sistema disponibiliza lista dos N pedidos já enviados 3. Ator seleciona pedido desejado e requisita visualização de detalhes 4. Detalhes do pedido selecionado são exibidos
Exceções:	<i>Não há</i>

k. Requisição da próxima mesa da fila

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Garçom
Descrição:	Ator envia mensagem a Sistema Servidor requisitando o número da próxima Mesa da fila. Ao receber a resposta, Sistema Cliente insere este número no campo correspondente à mesa do Pedido atual
Prioridade:	1
Pré-condições:	1. Haver conectividade com o servidor. 2. Sistema Servidor estar pronto para receber e enviar mensagens
Pós-condições:	1. Número da próxima mesa a ser atendida recebido, caso haja

Fluxo Normal:	1. Ator envia requisição da próxima mesa da fila, via interface gráfica. 2. <u>[Inc1]</u> 3. Sistema Servidor envia o número da mesa em questão [E1] 4. Sistema Servidor retira o número de mesa que acaba de ser enviado da fila FIFO 5. Sistema Cliente recebe o número da próxima Mesa da fila o coloca no campo correspondente à Mesa do Pedido atual
---------------	--

Exceções:	<ul style="list-style-type: none"> • [E1] - Não há mesas na fila
	2.1 Sistema Servidor envia mensagem de que não há Mesas a serem atendidas
Casos de uso incluídos	<ul style="list-style-type: none"> • Inc1: Validação e Interpretação de mensagem recebida

1. Inc1: Validação e Interpretação de mensagem recebida por Sistema Servidor

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Sistema Cliente
Descrição:	Sistema Servidor interpreta mensagem recebida de Sistema Cliente, validando-a. Caso não sejam encontrados erros, o Sistema Servidor executará a ação requerida pelo Sistema Cliente; caso contrário, uma mensagem de erro é enviada a este último
Prioridade:	1
Pré-condições:	1. Mensagem enviada a Sistema Servidor, partindo do Sistema Cliente
Pós-condições:	1. Mensagem interpretada e ação necessária tomada

Fluxo Normal:	<ol style="list-style-type: none"> 1. Sistema Servidor recebe mensagem enviada por Sistema Cliente 2. Sistema Servidor checa a validade das informações recebidas [A1] 3. Uma vez validadas, Sistema Servidor executa ação, de acordo com o requisitado na mensagem
Fluxo Alternativo:	<ul style="list-style-type: none"> • [A1] <ol style="list-style-type: none"> 2.1 Caso haja algum problema com a interpretação, uma mensagem descrevendo o erro é gerada 2.2 Sistema Servidor envia mensagem de erro a ator que enviou a mensagem

m. Edição de preferências – Sistema Servidor

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Administrador no Servidor
Descrição:	Ator acessa e edita as opções disponíveis, salvando-as no Sistema Servidor.
Prioridade:	3

<i>Pré-condições:</i>	1. Ator conectado no Sistema Servidor (login+senha exigidos)
<i>Pós-condições:</i>	1. Ator de volta à tela onde está a opção de edição de preferências

<i>Fluxo Normal:</i>	<ol style="list-style-type: none"> 1. Ator acessa opção de edição de preferências ("Menu Admin") 2. Sistema Servidor disponibiliza opções que podem ser modificadas e os comandos "confirmar" e "cancelar" 3. Ator seleciona e edita as opções desejadas: <ul style="list-style-type: none"> ▪ Dados de login no Sistema; ▪ Dados no banco de dados (usuários, produtos, etc.); ▪ Outros 4. Ator confirma as alterações [A1] [E1] 5. Sistema Servidor aplica as preferências conforme selecionado por ator 6. Sistema Servidor exibe mensagem de confirmação e retorna à tela anterior
<i>Fluxo Alternativo:</i>	<ul style="list-style-type: none"> • [A1] <ol style="list-style-type: none"> 4.1 Ator cancela alterações 4.2 Retornar ao Passo 6 do Fluxo Principal
<i>Exceções:</i>	<ul style="list-style-type: none"> • [E1] – Opção não pode ser editada por alguma razão <ol style="list-style-type: none"> 4.3 Sistema Servidor exibe na tela razão da impossibilidade 4.4 Retornar ao Passo 2 do Fluxo Principal

n. Inclusão/Edição/Exclusão de usuário autorizado

o. Inclusão/Edição/Exclusão de dados de produtos

p. Inclusão/Edição/Exclusão de dados de performance

Os Casos de Uso acima são análogos entre si, podendo ser classificados como CRUD ("Create, Read, Update and Delete"). Portanto, a descrição destes casos de uso visa apenas uniformizar seu tratamento.

Abaixo se encontra a descrição do Caso de Uso "Inclusão/Edição/Exclusão de usuário autorizado"

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006

Atores:	Administrador no Servidor
Descrição:	Ator edita grupo de informações de usuários na Base de Dados do Sistema Servidor.
Prioridade:	1
Pré-condições:	1. Ator conectado no Sistema Servidor (login+senha exigidos)
Pós-condições:	1. Usuário incluído, editado ou excluído da Base de Dados do Sistema Servidor

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator requisita ação de manutenção de usuários, via interface ("Menu Admin") 2. Sistema Servidor disponibiliza interface figurando a lista de todos os usuários cadastrados, bem como opção de adição de novo usuário ou retornar à tela anterior 3. Ator seleciona um usuário [A1] 4. Sistema disponibiliza opções de edição e remoção para o usuário selecionado 5. Ator seleciona opção edição de usuário [A2] 6. Sistema Servidor exibe detalhes do cadastro do usuário, que podem ser editados por ator, bem como opções de salvar e cancelar 7. Ator edita os campos desejados e requisita que os dados sejam salvos 8. Sistema Servidor requisita confirmação da ação anterior 9. Caso usuário confirme, sistema executa ação requisitada (atualizando campos modificados e também campos que dependem de alguma forma destes) e retorna ao Menu Usuários [A3] [A4] 10. Ator requisita saída do menu de edição de usuários 11. Sistema Servidor volta ao Menu Admin
Fluxo Alternativo:	<ul style="list-style-type: none"> • [A1] 3.1 Ator acessa opção de adição de usuário 3.2 Sistema abre Menu Edição de Usuários, criando novo usuário com dados em branco. 3.3 Retornar ao passo 7. do Fluxo Normal • [A2] 5.1 Ator seleciona opção de remoção de usuário 5.2 Sistema requisita confirmação da ação anterior 5.3 Caso usuário confirme, sistema executa ação requisitada [A2.1] 5.4 Retornar ao passo 10. do Fluxo Principal • [A2.1] 5.3.1 Caso contrário (cancelamento de ação por usuário), sistema cancela ação requisitada 5.3.2 Retornar ao passo 5.4 do Fluxo Alternativo [A2]

	<ul style="list-style-type: none"> • [A3] 9.1 Caso contrário (cancelamento de ação por usuário), sistema cancela ação e continua no menu atual 9.2 Retornar ao passo 6. do Fluxo Principal
	<ul style="list-style-type: none"> • [A4] 9.3 Caso alguma alteração feita possa causar inconsistência na base de dados, Sistema Servidor exibe mensagem de erro. 9.4 Retornar ao passo 9. do Fluxo Principal
Exceções:	<i>Não Há</i>

q. Análise/Exportação de dados de estoque

r. Análise/Exportação de dados de performance

Os Casos de Uso acima são análogos entre si, bastando apenas escolher o tipo de informação da base de dados que será analisado/exportado: estoque (campo de produto) e pedido, respectivamente.

Abaixo se encontra a descrição do Caso de Uso "Análise/Exportação de dados de estoque"

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Usuário no Servidor
Descrição:	Ator acessa ferramenta de análise de estoque no Sistema Servidor, podendo extrair informações que lhe pareçam úteis e mesmo exportá-las em formatos comumente usados
Prioridade:	3
Pré-condições:	1. Não há
Pós-condições:	1. Informações sobre o estoque dos produtos exibidas por Sistema Servidor e/ou exportadas por Ator

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator requisita ação de análise de dados de estoque, via interface 2. Sistema Servidor disponibiliza novo menu no qual figura a lista de todos os produtos com estoque cadastrado, bem como opção de retornar ao menu anterior 3. Ator seleciona o item (ou os itens) que deseja visualizar 4. Sistema ativa ferramenta de análise para os itens
---------------	---

	<p>selecionados, disponibilizando visualização gráfica dos estoques destes e opção de exportação.</p> <p>5. Ator requisita exportação dos dados, escolhendo nome e formato do arquivo exportado [E1]</p> <p>6. Sistema exporta os dados para o local desejado</p>
Exceções:	<i>Não há</i>

s. Informar que pedido está pronto para ser servido

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Usuário no Servidor
Descrição:	Ator informa Sistema Servidor e Cliente (apenas o garçom responsável pelo pedido) que um pedido enviado previamente está pronto para ser servido
Prioridade:	1
Pré-condições:	<ol style="list-style-type: none"> 1. Haver possibilidade de comunicação entre Sistemas Cliente e Servidor. 2. Sistema Cliente estar pronto para receber mensagens
Pós-condições:	<ol style="list-style-type: none"> 1. Garçom avisado que Pedido está pronto para ser servido através de mensagem recebida no Sistema Cliente e Pedido classificado como pronto no Sistema Servidor (que o remove então do grupo de Pedidos em Aberto)

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator seleciona Pedido em Aberto, via Interface 2. Sistema Servidor habilita opção possibilitando informar que o pedido está pronto para ser servido 3. Ator acessa opção acima 4. Sistema Servidor envia mensagem a Sistema Cliente responsável pelo pedido, informando que o Pedido está pronto, recebendo então resposta de confirmação em intervalo de tempo estipulado [A1] 5. Sistema Servidor classifica Pedido como pronto na Base de Dados 6. Sistema Servidor exclui Pedido em questão do grupo de Pedidos em Aberto, adicionando-o a grupo de Pedidos prontos
Exceções:	<ul style="list-style-type: none"> • [E1] - Sistema Servidor não recebe mensagem de confirmação dentro de intervalo de tempo estipulado <p>5.1 Mensagem de problema na conexão com o cliente é exibida</p>

t. Mover pedido de “Grupo Aberto” a “Grupo Pronto” e vice-versa

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Usuário no Servidor
Descrição:	Ator move Pedido de grupo de pedidos que ainda não estão prontos (“Grupo Aberto”) a grupo de pedidos que já estão prontos (“Grupo Pronto”) ou vice-versa, sem que um aviso ao Sistema Cliente seja enviado (útil caso haja problemas de conexão entre Sistemas).
Prioridade:	2
Pré-condições:	1. Haver pedidos no grupo de origem
Pós-condições:	1. Pedido movido de um grupo a outro

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator seleciona pedido do grupo de origem (“aberto” ou “pronto”), via interface gráfica. 2. Sistema Servidor habilita opção de mover pedido para o grupo destino 3. Ator ativa opção acima 4. Pedido é movido do grupo origem ao grupo destino
Exceções:	<i>Não há</i>

u. Envio de mensagem a garçom

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Usuário no Servidor
Descrição:	Ator envia Mensagem a Sistema Cliente, de forma a alertar garçom de algo. Existe uma lista de mensagens padrão que pode ser armazenada no Banco de Dados, a qual pode ser facilmente acessada no momento do envio
Prioridade:	2
Pré-condições:	1. Não há
Pós-condições:	1. Mensagem enviada a Sistema Cliente

Fluxo Normal:	<ol style="list-style-type: none"> 1. Ator preenche campo de mensagem [A1] 2. Ator preenche campo de remetente [A2] [A3] 3. Ator seleciona no campo destinatário o nome do usuário
---------------	---

	<p>ao qual deseja enviar a mensagem (sistema disponibiliza lista de todos os Usuários cadastrados no sistema que podem receber mensagens, ou seja, que têm acesso a Sistema Cliente)</p> <ol style="list-style-type: none"> 4. Ator acessa opção de envio de mensagem 5. Sistema Servidor envia mensagem ao usuário escolhido como destinatário[E1] 6. Sistema Servidor exibe mensagem informando o resultado da ação
Fluxo Alternativo	<ul style="list-style-type: none"> • [A1] <ol style="list-style-type: none"> 1.1 Ator pode acessar lista com mensagens padrão e selecionar a mensagem desejada 1.2 Sistema preenche o campo mensagem com o texto padrão escolhido 1.3 Retornar ao passo 2. do Fluxo Normal • [A2] <ol style="list-style-type: none"> 1.1 Ator pode acessar lista com Usuários e selecionar o remetente desejado 1.2 Sistema preenche o campo remetente com dado do usuário escolhido 1.3 Retornar ao passo 3. do Fluxo Normal • [A3] <ol style="list-style-type: none"> 1.4 Retornar ao passo 3. do Fluxo Normal (<i>campo remetente não é de preenchimento obrigatório</i>)
Exceções:	<ul style="list-style-type: none"> • [E1]- <i>Algum campo obrigatório deixado em branco</i> <ol style="list-style-type: none"> 5.1 Sistema envia mensagem de erro e pede que o campo em questão seja preenchido 5.2 Fim do Caso de Uso

v. Criação de pedido e adição de itens a ele

w. Visualização e Edição de pedido

O Sistema Servidor disponibiliza uma interface para criação e edição de pedidos, sem a necessidade da utilização de um sistema móvel. Isto pode ser útil caso haja algum problema que impossibilite o uso destes últimos para um ou mais usuários dos mesmos: fazendo com que os pedidos entrem no sistema, o mesmo pode continuar com as tarefas de controle de estoque e armazenamento de dados de performance.

O uso desta funcionalidade, entretanto não é recomendável em condições normais de funcionamento. Como tais casos de uso são iguais aos seus análogos no Sistema Cliente, eles não serão descritos novamente.

x. Executar manutenção da base de dados

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Usuário no Servidor
Descrição:	<p>Ator requisita que Sistema Servidor faça manutenção da base de dados, de forma a melhorar a performance do sistema. Esta é uma opção acessível não somente via interface gráfica mas também é oferecida ao usuário no momento do fechamento do programa</p> <ul style="list-style-type: none">• Pedidos no "grupo prontos" há um tempo maior do que um valor configurável são enviados a uma base de dados separada daquela que o sistema acessa quando o usuário visualiza o "grupo prontos"• Itens descontinuados há um tempo maior do que um valor previamente estabelecido são removidos da base de dados• Etc.
Prioridade:	1
Pré-condições:	1. Não há
Pós-condições:	1. Manutenção na base de dados realizada de acordo com opções pré-definidas por usuário

Fluxo Normal:	<ol style="list-style-type: none">1. Ator ativa opção de iniciar manutenção da base de dados2. Sistema dá início às operações de manutenção, respeitando as opções pré-estabelecidas.
Exceções:	<i>Não há</i>

y. Colocar mesa na fila FIFO

Criado por:	Marcos Simplicio
Data de Criação:	15/07/2006
Última Atualização	Marcos Simplicio
Realizada por:	
Data da Última Atualização:	15/11/2006
Atores:	Cliente (via Call Bell)
Descrição:	Ator envia requisição de atendimento via Call Bell e sua mesa é colocada no final da fila FIFO
Prioridade:	1
Pré-condições:	1. Haver conectividade entre sistema do Call Bell e o Sistema Servidor.

Pós-condições:	1. Mesa do Cliente colocada em fila FIFO
----------------	--

Fluxo Normal:	<ol style="list-style-type: none">1. Ator envia requisição de atendimento ao Sistema Servidor, via Call Bell2. Sistema Servidor associa pedido à mesa que enviou a requisição [E1], [E2]3. Sistema Servidor coloca código da mesa em questão na fila FIFO
Exceções:	<ul style="list-style-type: none">• <i>[E1] – Sistema Servidor incapaz de reconhecer a mesa que enviou a requisição</i> <ol style="list-style-type: none">2.1 Mensagem de erro exibida na Interface do Sistema Servidor <ul style="list-style-type: none">• <i>[E2] – Mesa em questão já se encontra na fila FIFO</i> <ol style="list-style-type: none">2.2 Sistema Servidor não coloca a Mesa requisitante na fila

5 Metodologia

Este projeto pode ser dividido em 2 subsistemas distintos, que possuem como interface comum o software presente no servidor central:

5.1 Hardware

- a) Especificação e avaliação dos detalhes de funcionamento de equipamentos necessários ao sistema proposto, basicamente os emissores e receptores RF e o Display que implementa a fila de mesas. Além disto, é necessária a definição de um protocolo de comunicação com o servidor.

Atividades contempladas:

- o Análise de catálogos e especificações técnicas detalhadas dos equipamentos envolvidos no sistema (descritos acima). Possível necessidade de engenharia reversa para adequação às necessidades do projeto.
- o Estudo dos protocolos de comunicação de cada equipamento para avaliar a possibilidade de integração de cada um na rede.
- o Estudo de casos de sistemas onde estes equipamentos já estejam presentes

Indicador de Conclusão:

- o Relatório com descritivo técnico dos equipamentos analisados e protocolos de comunicação correspondentes, bem como lista contendo os equipamentos que serão efetivamente utilizados.
- b) Implementação de um caso piloto para testes e validação do sistema descrito. Um módulo de comunicação com a RF deve ser desenvolvido para

o aplicativo presente no servidor central (o mesmo aplicativo que trata da LAN Wi-fi na qual estão os Palms)

Atividades contempladas:

- o Desenvolvimento da funcionalidade para tratamento de chamadas e implementação da fila FIFO no software instalado no servidor central.
- o Implementação da arquitetura proposta, com interligação entre os elementos escolhidos no item anterior e o software no servidor.
- o Teste e validação do sistema integrado.

Indicador de Conclusão:

- o Sistema testado e operante, apresentando as funcionalidades propostas.

5.2 Pedidos Wi-Fi

- a) Especificação e avaliação dos detalhes de funcionamento de equipamentos necessários ao sistema proposto, a saber: PDAs, roteadores Wi-Fi e impressoras/monitores. A relativa padronização destes tipos de equipamentos facilita esta parte do processo.

Atividades contempladas:

- o Análise de catálogos e especificações técnicas dos equipamentos envolvidos no sistema, de acordo com as necessidades do projeto (ponto principal: comunicação sem fio).
- o Definição dos softwares necessários para garantir as funcionalidades desejadas

Indicador de Conclusão:

- o Lista contendo os equipamentos que serão efetivamente utilizados, bem como suas configurações necessárias.
- o Softwares necessários instalados e funcionais

- b) Implementação dos aplicativos de aquisição e envio de dados, tanto do lado do servidor central quanto dos *Palms* (trata-se de aplicativos distintos, porém de interface semelhante), a partir de uma definição clara das funcionalidades que devem estar disponíveis nos mesmos para que as necessidades de um estabelecimento sejam atendidas a contento. Estas informações serão obtidas através dos estudos de casos.

Observação importante: o aplicativo que roda no servidor central deve se comunicar tanto com os Palms quanto com a rede RF, como já foi descrito.

Indicador de Conclusão:

- o Elaboração de relatório com descritivo técnico e fluxogramas dos módulos computacionais envolvidos, contendo um capítulo sobre os casos que levaram a definição dos módulos e dos fluxogramas. A versão preliminar (contendo os casos de uso a serem implementados) deste estudo encontra-se na seção Anexos nomeada como "Casos de Uso", podendo ser alterada caso sejam encontradas novas necessidades durante o estudo de mercado.
 - o Sistema testado e funcional.
 - o Interface com o usuário implementada e aprovada por equipe e potenciais clientes.
- c) Modelagem e estruturação do Banco de Dados para armazenamento de informações adquiridas e enviadas. A estrutura será determinada com base na definição dos requisitos funcionais e não-funcionais do sistema, bem como na determinação de quais informações, após análise, podem trazer valor agregado ao negócio de um possível cliente.

Indicador de Conclusão:

- o Relatório técnico contendo as especificações e descrições detalhadas dos módulos e do banco de dados.
- o Desenvolvimento da base de dados descrita e sua integração com o aplicativo de tratamento.
- o Sistema integrado testado e funcional.

6 Resultados

Ao fim do projeto podemos dizer que atingimos os objetivos estabelecidos na proposta inicial do mesmo. Desta forma, o produto de nosso trabalho ao longo do ano gerou os módulos de hardware e os softwares, descritos anteriormente, que integram o nosso sistema, levando-o ao funcionamento e, assim validando a proposta. Existem ainda algumas funcionalidades do software que podem ser expandidas e melhoradas (em especial com um maior feedback recebido de profissionais da área de restaurantes), porém as funcionalidades principais foram testadas e aprovadas.

A seguir apresentaremos alguns dos resultados obtidos, começando por testes de alcance entre nosso receptor e transmissor. Em seguida é dada uma explicação sobre o funcionamento dos módulos de hardware desenvolvidos. E por fim, são apresentados resultados da etapa de programação, contendo diagramas de classes e tomadas de tela dos aplicativos desenvolvidos.

6.1 *Transmissor e Receptor RF*

Nossa maior preocupação em utilizar um módulo pronto era a questão do alcance que este poderia ter. Sendo assim, efetuamos dois testes para avaliar esta questão: o primeiro deveria medir a atenuação e as perdas em espaço livre na transmissão dos dados, e o segundo para verificar o comportamento e alcance entre receptor e transmissor quando submetidos a obstáculos, como paredes, portas, janelas e até mesmo pessoas.

Alguns testes foram realizados em laboratório. Para estes testes foi utilizado o analisador de espectros HP8656A, para medição da potência do sinal emitido pelo controle em função da distância. O teste foi feito afastando-se o controle da antena do analisador de espectros e verificando a potência incidente na raia pertinente. Segue abaixo uma tabela com os valores encontrados.

z (m)	dBm	P(mW)	P(W)
1	-20	1,00E-02	1,00E-05
2	-25	3,16E-03	3,16E-06
3	-29	1,26E-03	1,26E-06

Tabela 6 – Potência do transmissor em função da distância

A partir dos valores encontrados é possível encontrar a constante de decaimento da potência através da seguinte expressão:

$$P(W) = A \cdot e^{-2\alpha z}$$

A constante α pode ser obtida da seguinte forma:

$$\alpha = \frac{\ln(P_1 / P_2)}{2 \cdot (z_2 - z_1)}$$

E a constante A por:

$$A = \frac{P_1}{e^{-2\alpha z_1}}$$

Para levantar estes valores utilizei 3 pontos e a média dos valores encontrados nestes para compor os valores de α e A.

	1 a 2 m	2 a 3 m	1 a 3 m	Média
α	0,575646273	0,460517019	0,518081646	0,518081646

Tabela 7 – Valores calculados para α

	1 m	2 m	3 m	Média
A	2,81838E-05	2,51189E-05	2,81838E-05	2,71622E-05

Tabela 8 – Valores calculados para a constante A

Com estes valores foi possível traçar o gráfico da potência em função da distância.



Figura 36 – Gráfico da variação do sinal recebido em função da distância

O segundo teste efetuado, como foi dito anteriormente, consistiu em verificar o recebimento de sinal diante de obstáculos no percurso entre o transmissor e o receptor. Este teste foi efetuado na residência de um dos integrantes do grupo e obteve resultados bastante animadores que serviram para confirmar nossa opção pelo Receptor Matrix®.

Neste teste o receptor foi colocado em um dos cômodos e o transmissor foi acionado em diversos outros cômodos, a fim de se levantar se os obstáculos e a distância poderiam atrapalhar no funcionamento do aparelho. A figura 31 mostra as plantas da residência com os locais dos onde foram posicionados os transmissores e o cômodo onde foi colocado o receptor. Os móveis não estão indicados para simplificar o esquema, mas estes podem contribuir com a atenuação do sinal.

Em todas as posições marcadas nas plantas houve um perfeito funcionamento e o receptor conseguiu identificar o controle que era acionado. Isso mostra que o sistema de transmissão e recepção funciona muito bem para distâncias da ordem de 16 m com paredes como obstáculos.

Para os testes todas as portas foram deixadas abertas, pois em um ambiente real não esperamos que existam portas fechadas entre os transmissores e o receptor. Uma observação importante a se fazer é a espessura das paredes, que variava entre 14 cm e 18 cm.

Outro teste efetuado para verificar o funcionamento do controle foi cobri-lo com a própria mão e tentar transmitir. Neste teste percebemos que a mão absorve completamente o sinal se não estiver a uma distância mínima de 1 cm da parte frontal do controle.

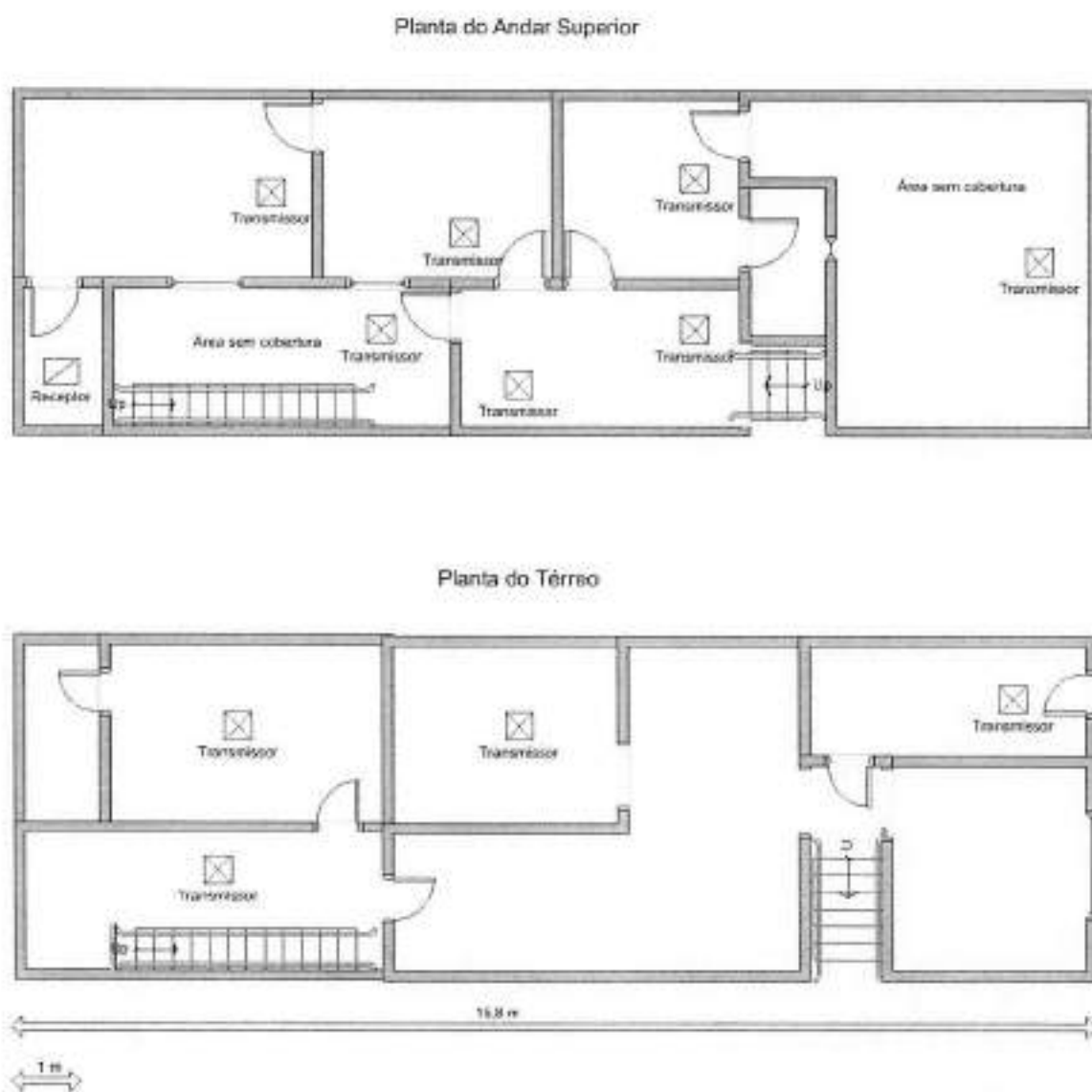


Figura 37 – Plantas do local dos testes e posicionamento dos transmissores e do receptor

6.2 Módulos de hardware

Nos módulos de hardware estão englobados tanto o Decodificador e Controle de Comunicação, como o Controle de Display.

Alguns dos resultados, como esquema das placas e leiautes já foram apresentados anteriormente e não serão novamente colocados aqui.

Os testes efetuados para este módulos foram feitos testando-se partes, já que cada um dos circuitos é composto subcircuitos que se repetem diversas vezes, como o caso do conjunto de chaves a transistor NPN. Sendo desta forma os testes foram feitos de uma forma reduzida, pois as placas definitivas não estavam prontas até o momento da edição deste trabalho. Acreditamos que o funcionamento total se dará corretamente, pois montar o circuito total nada mais é que repetir diversas vezes os circuitos testados. Para validação do sistema como um todo foi montado a interface para apenas um dos displays do Receptor Matrix®, e os testes confirmaram o funcionamento correto do sistema reduzido.

6.3 Software

Este tópico se destina a exibir e comentar os resultados obtidos nas implementações de software. Para demonstrar estes resultados, serão apresentados os diagramas UML que nortearam o desenvolvimento, bem como alguns *screenshots* do programa em uso.

A primeira etapa no processo de desenvolvimento do software foi a elaboração e validação dos casos de uso dos Sistemas Cliente e Servidor, descritos anteriormente. Em seguida, foram gerados os diagramas que determinam a arquitetura do software desenvolvido. Dentre eles, serão apresentados os mais importantes e mais intensamente utilizados e revisados durante o desenvolvimento: o diagrama de classes, o diagrama de navegação do usuário, o diagrama de interface e um diagrama organizando os diversos pacotes de classes.

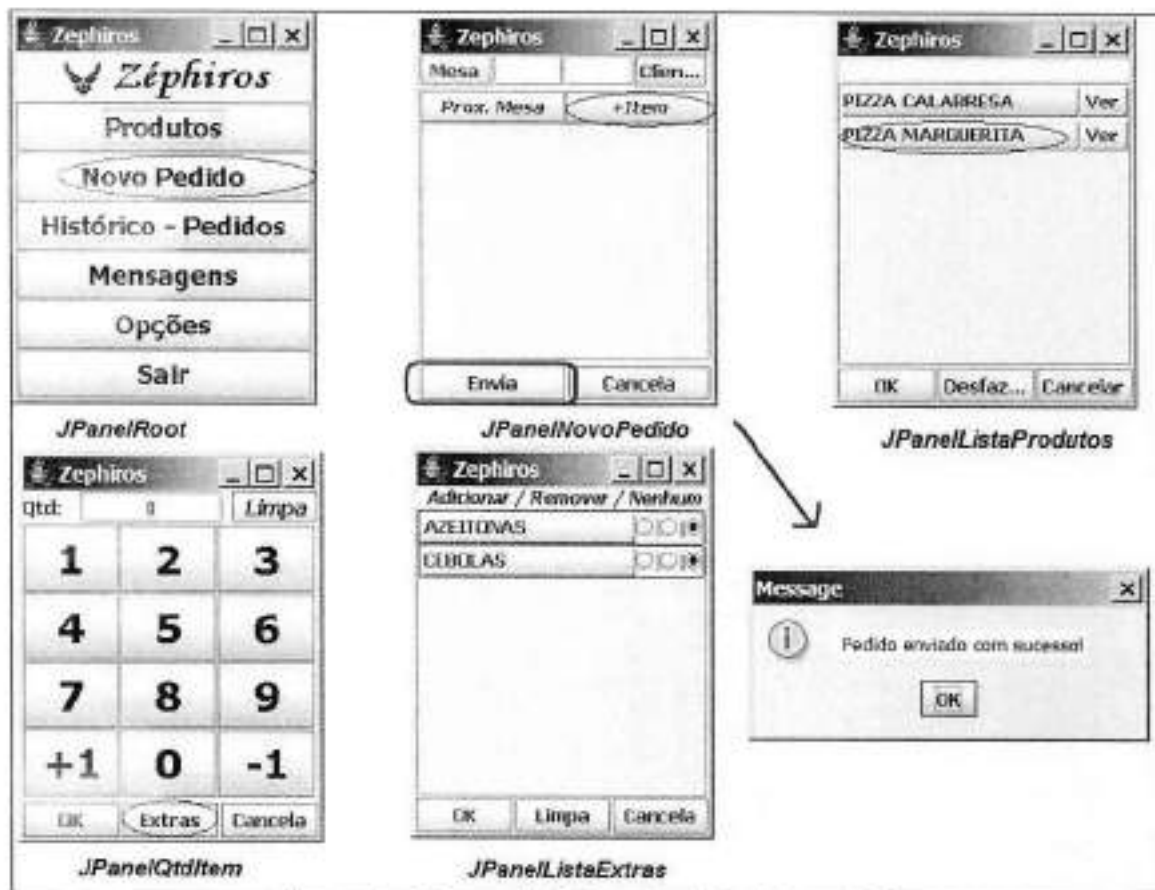


Figura 42 – Screenshot: Geração de Pedido em PDAs



Figura 43 – Screenshot: Outras funções: PDA

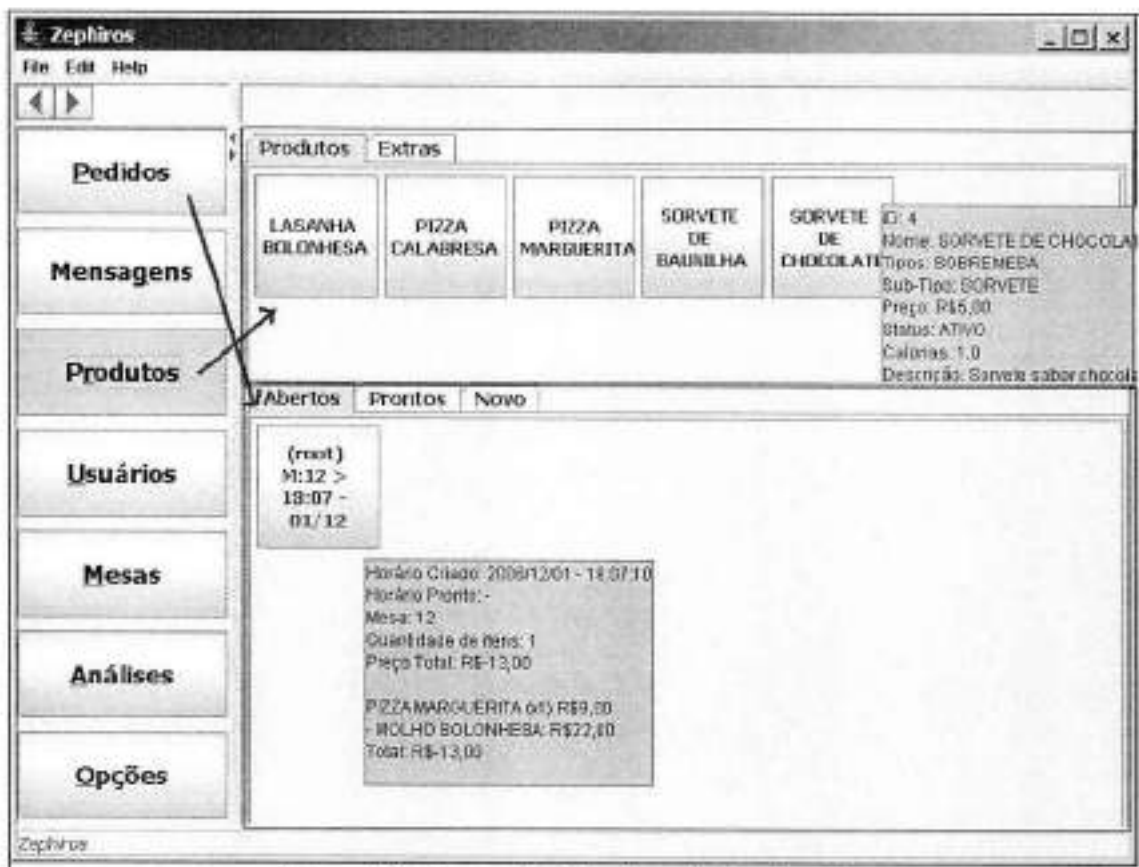


Figura 44 – Screenshot: Funcionalidades Servidor
(Obs.: Pedidos e Produtos foram trazidos à mesma tela)

7 Discussão

A partir da especificação do escopo inicial de trabalho criamos um protótipo que pode ser aplicado em condições reais de uso em um estabelecimento conveniente. Dentro deste protótipo final estão englobados os seguintes módulos:

- Transmissão e recepção em RF.
- Transmissão e Recepção Wireless.
- Módulos de exibição de informações (fila, pedidos, estoque, etc).
- Software de controle de rede de comunicação.
- Software de gerenciamento geral.

Cada etapa de criação dos módulos que compõem o sistema foi praticamente concluída, pois para darmos o trabalho como completamente concluído precisamos desenvolver manuais com especificações técnicas e

manuais de usuário. Desta forma teremos o projeto completamente documentado, bastando apenas revisões e integrações entre os manuais para gerar um manual geral. Esses relatórios são indicadores de conclusão do projeto, assim como descrito na metodologia.

Para validar nosso projeto tentaremos aplicá-lo em um estabelecimento real, oferecendo a este o sistema a preço de custo. Desta forma teremos uma forma de analisar o protótipo sob aspectos reais de funcionamento e não em testes pré-determinados, que geralmente não trabalham com o potencial máximo do sistema e não exploram as falhas que este possa ter.

O desenvolvimento dos módulos trabalhou geralmente sob o aspecto de custos, por isso determinadas soluções, que eram antes consideradas certas, foram abandonadas, como foi explicado ao longo deste trabalho.

A relação das tecnologias usadas pode estar longe de trabalhar com tecnologias de ponto, pois isso implica custos maiores, porém implementações com tecnologias emergentes podem adicionar uma característica de segurança ao sistema, que não era fator crítico para o projeto deste sistema. Talvez a segurança não justifique o custo, por isso modificar o sistema de transmissão apenas baseado neste quesito seria uma decisão arbitrária, a não ser que novas funcionalidades sejam implementadas, como transmissão e recepção para as mesas, dando substrato ao sistema de atendimento praticamente automatizado, baseado em menus eletrônicos.

8 Conclusão

Este trabalho veio suprir uma necessidade de modernização de um sistema que necessita cada vez mais de aperfeiçoamento e eficiência. O trabalho, fundamentado em dados estatísticos e em conversas com possíveis compradores, além de comparação com sistemas atuais, mostra o potencial comercial do produto no mercado.

Com base no trabalho desenvolvido, é ainda possível visualizar novas áreas de oportunidade dentro deste nicho, elaborando funcionalidades que potencialmente melhoraram de maneira significativa a logística de atendimento. Como exemplo, pode-se citar o uso de 3 receptores sem fio para os PDAs, possibilitando ao sistema determinar a posição do garçom e, assim, indicar a pessoa mais adequada para atender uma determinada mesa, reduzindo o tempo total de atendimento. Esta e outras idéias (como o menu eletrônico sugerida na discussão) poderão ser exploradas em implementações futuras.

Pode-se dizer que este projeto explorou bastante as habilidades específicas de seus membros, além de incitar o desenvolvimento de novas soluções a partir dos conhecimentos já adquiridos (como o roteamento de um leiaute melhor do que algoritmos computacionais comerciais).

A colaboração de integrantes com diferentes habilidades serviu também para colocá-los em contato com temas normalmente pouco explorados na graduação de cada ênfase, além de resultar em um gerenciamento do tipo "dividir para conquistar", muito útil para atingir objetivos que envolvam tarefas complexas.

Por fim, o grupo tentou fazer com que o presente projeto de formatura seja usado como ferramenta para gerar conhecimento aproveitável de forma produtiva à sociedade, não se restringindo a um trabalho teórico para mera validação da formação acadêmica dos seus membros. É por isto que o foco deste projeto foi direcionado para criação do que um dia pode se tornar um produto comercialmente interessante.

9 Bibliografia

- [1] RODEN, Martin S. Digital and Data Communication Systems. Prentice-Hall, 1982.
- [2] SIMON, Marvin K. Digital Communication Techniques: Signal Design and Detection. Prentice-Hall, 1995.
- [3] PAHLAVAN, Kaveh; LEVESQUE, Allen H. Wireless Information Networks. Wiley Inter-Science, 1995.
- [4] <http://www.inatel.br> – Modulações em Rádio Frequência.
- [5] Fairchild: datasheet de LM7805, 74HC04 – <http://www.fairchildsemi.com>
- [6] Atmel: datasheet do AT89S8252 – <http://www.atmel.com>
- [7] Maxim: datasheet do MAX232CPE – <http://www.maxim-ic.com>
- [8] http://www.projotoderedes.com.br/artigos/artigo_zigbee.php - Informações sobre ZigBee – Acessado em: 21/11/2006
- [9] ZigbeeAlliance – www.zigbee.org.
- [10] PRESSMAN, Roger S. Software Engineering: A Practitioner's Approach. McGrawHill, Sixth edition, 2005.
- [11] KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet. Addison Wesley, 3ª edição, 2006.
- [12] JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. The Unified Software Development Process. Addison-Wesley Professional, 1st Edition, 1999.
- [13] DEITEL, H.M ; DEITEL, P.J. Java: Como Programar. Bookman, 4ª edição, 2002.
- [14] SEDRA, Adel S.; SMITH, Kenneth, C. Microeletrônica. São Paulo: Pearson Education do Brasil, 2000.
- [15] TOCCL, R.J., "Sistemas Digitais: Princípios e Aplicações ", Pearson, 8ª ed. 2003.
- [16] FREGNI, E., SARAIVA, A. M. "Engenharia do Projeto Lógico Digital", Ed. Edgard Blücher, 1995.
- [17] SILVA JÚNIOR, Vidal Pereira da. Aplicações Práticas do Microcontrolador 8051. São Paulo: Érica, 1998.
- [18] CUGNASCA, Carlos Eduardo et al. Microprocessadores e Microcontroladores. São Paulo: POLI-USP, 2006. Disponível em

<<http://www.pcs.usp.br/~pcs2497/apontamentos.html>>. Acesso em: 10 de outubro de 2006.

[19] CUGNASCA, Carlos Eduardo et al. Programação de Microprocessadores. São Paulo: POLI-USP, 2006. Disponível em

<<http://www.pcs.usp.br/~pcs2497/apontamentos.html>>. Acesso em: 10 de outubro de 2006.

[20] CUGNASCA, Carlos Eduardo et al. Comunicação Serial no 80C51. São Paulo: POLI-USP, 2006. Disponível em

<<http://www.pcs.usp.br/~pcs2497/apontamentos.html>>. Acesso em: 10 de outubro de 2006.

[21] CUGNASCA, Carlos Eduardo et al. Família 80C51 - Resumo. São Paulo: POLI-USP, 2006. Disponível em <<http://www.pcs.usp.br/~pcs2497/apontamentos.html>>.

Acesso em: 10 de outubro de 2006.

[22] ARAKAKI, Reginaldo. Comunicação Serial. São Paulo: POLI-USP, 2006.

Disponível em <<http://www.pcs.usp.br/~pcs2497/apontamentos.html>>. Acesso em: 10 de outubro de 2006.