

**Fernando Torres Pereira da Silva**

**Guilherme Silva Affonso**

**Uma aplicação de sistemas de redes de  
dados wireless: automação de  
estabelecimentos comerciais da área de  
alimentação**

Monografia de Conclusão de Curso  
apresentada à Escola Politécnica da  
Universidade de São Paulo para obten-  
ção do Título de Engenheiro

**Fernando Torres Pereira da Silva**

**Guilherme Silva Affonso**

**Uma aplicação de sistemas de redes de  
dados wireless: automação de  
estabelecimentos comerciais da área de  
alimentação**

Monografia de Conclusão de Curso  
apresentada à Escola Politécnica da  
Universidade de São Paulo para obten-  
ção do Título de Engenheiro

Área de concentração:  
Engenharia Mecatrônica

Orientador:  
Prof. Dr. Marcos de Sales Guerra  
Tsuzuki

## **Ficha Catalográfica**

Silva, Fernando Torres Pereira da  
Affonso, Guilherme Silva

Uma aplicação de sistemas de redes de dados wireless: automação de estabelecimentos comerciais da área de alimentação. São Paulo, 2008. 48 p.

Monografia de Conclusão de Curso (Graduação) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos.

1. SED. 2. Banco de Dados. 3. Redes Sem Fio. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos. II.



# Agradecimentos

Ao Prof. Marcos Tsuzuki, pelo incentivo e pelas idéias propostas desde o início deste projeto.

Às gerências dos estabelecimentos *Pizzaria São Pedro* e *Belmiro da Vila*, e ao departamento de marketing da rede de restaurantes *Habib's* pela colaboração ao desenvolvimento deste projeto.

A todos que, de maneira direta ou indireta, contribuíram para a realização deste trabalho.

# Resumo

O projeto consiste no desenvolvimento de uma rede de dados wireless com o intuito de dinamizar a logística de estabelecimentos comerciais da área de alimentação. Com o foco de aplicações wireless (celulares, palm tops, notebooks), a meta é viabilizar essa rede, obtendo uma solução de baixo custo e alta eficiência e desenvolver um sistema de dados e de interface homem-máquina em cada segmento do estabelecimento (administração, cliente, entre outros). Desta maneira, clientes podem realizar o pedido antes de chegar ao estabelecimento, ao chegar podem verificar o número de pessoas na fila ou mesmo verificar o seu programa de fidelidade.

# Abstract

The project consists of the development of a wireless network with the aim of making more dynamic the logistics of commercial establishments of the food area. Focused on wireless applications (cellphones, palm tops, notebooks), the aim is making viable this network, getting a low-cost and high-performance solution. A data system and human-machine interface will be developed in each area of the establishment (administration, client and so on). In that way, clients can make their requests before arriving the establishment and can view how many people is in the line before them, or even view their fidelity program.

# Conteúdo

## **Lista de Figuras**

<b>1 Motivação</b>	<b>12</b>
<b>2 Recursos Utilizados</b>	<b>15</b>
2.1 Introdução . . . . .	15
2.2 <i>Ruby on Rails</i> . . . . .	15
2.2.1 <i>Ruby</i> . . . . .	15
2.2.2 <i>Rails</i> . . . . .	16
2.3 <i>MySQL</i> . . . . .	17
2.4 Conclusão . . . . .	18
<b>3 Análise Dinâmica</b>	<b>19</b>
3.1 Introdução . . . . .	19
3.2 Redes de Petri . . . . .	19
3.3 Modelagem em Redes de Petri . . . . .	20
3.3.1 Procedimento Adotado . . . . .	20
3.3.2 Modelo Simplificado . . . . .	20
3.3.3 Estabelecimento Tradicional . . . . .	21
3.3.4 Estabelecimento Automatizado . . . . .	24
3.4 Pesquisa de Campo . . . . .	26
3.5 Conclusões . . . . .	27
<b>4 Implementação</b>	<b>29</b>
4.1 Introdução . . . . .	29

4.2	O Banco de Dados	32
4.3	Fluxo de Telas do Administrador	32
4.4	Fluxo de Telas do Cliente	40
<b>Apêndice A - Código-Fonte</b>		<b>45</b>
A.1	cliente_controller.rb	45
A.2	cardapio.rb	46
A.3	adicionar_ao_cart.rhtml	46
A.4	cardapio_exibir.rhtml	46
A.5	escolha_classif.rhtml	47
<b>Referências</b>		<b>48</b>

# Listas de Figuras

1.1	Terminais Cliente . . . . .	13
1.2	Visualização do Prato. . . . .	14
2.1	Arquitetura Modelo-Visualização-Controlador. . . . .	17
3.1	Rede de Petri do Modelo Simplificado. . . . .	21
3.2	Rede de Petri do Modelo com Atendimento sem Automação com 1 Mesa. . . . .	23
3.3	Rede de Petri do Modelo com Atendimento com Automação com 1 Mesa. . . . .	25
3.4	Tempo Total de Atendimento vs. Número de Mesas por Garçom.	28
4.1	Terminal Cliente. . . . .	30
4.2	Terminal Cozinha. . . . .	30
4.3	Terminal Recepção. . . . .	31
4.4	Estrutura do Banco de Dados. . . . .	33
4.5	Login Administrador. . . . .	35
4.6	Login Administrador - Mensagem de Erro. . . . .	35
4.7	Página Principal - Administrador. . . . .	35
4.8	Itens do Cardapio. . . . .	36
4.9	Exibição dos detalhes do Prato. . . . .	36
4.10	Edição do Prato. . . . .	37
4.11	Exclusão do Prato. . . . .	37
4.12	Novo Prato. . . . .	37
4.13	Lista de Usuários. . . . .	38
4.14	Esclusão de Usuário. . . . .	38
4.15	Novo Usuário. . . . .	38

4.16 Erro cadastrando Novo Usuário. . . . .	39
4.17 Lista de Administradores. . . . .	39
4.18 Alterar senha de administrador. . . . .	40
4.19 Cadastrar novo de administrador. . . . .	40
4.20 Logout. . . . .	40
4.21 Tela inicial da interface de clientes. . . . .	41
4.22 Macro divisões do cardápio. . . . .	41
4.23 Bebidas. . . . .	42
4.24 Sobremesas. . . . .	42
4.25 Pratos Quentes. . . . .	43
4.26 Pratos Frios. . . . .	43
4.27 Esvaziar Comanda. . . . .	44

# 1 Motivação

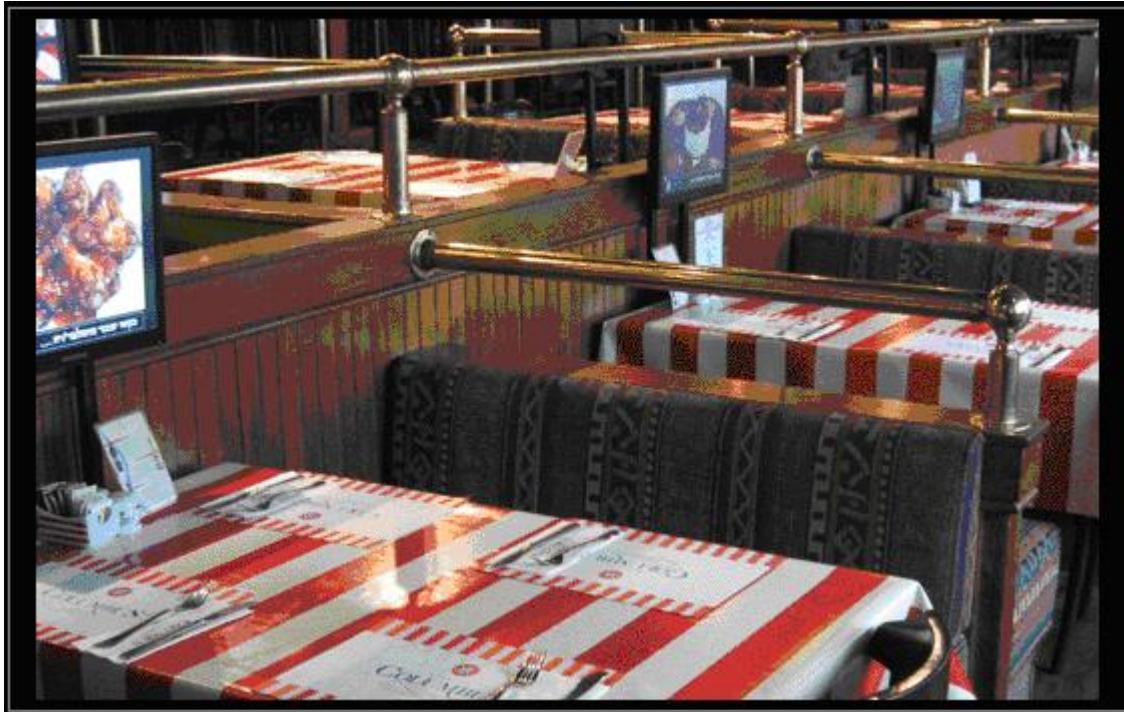
Diversas soluções estão disponíveis nesse ramo visando proporcionar uma melhora no desempenho de atendimento e administração desses locais. É exatamente nesse contexto que está fundamentado todo o propósito desse trabalho. Exemplos reais ilustram alguns problemas que rotineiramente ocorrem ou já ocorreram, com qualquer freqüentador desse tipo de estabelecimento. Quem nunca passou por dificuldades em encontrar um garçom para atendê-lo, naquelas confusas contas conjuntas em barzinhos, desorganizadas e infundáveis filas de espera na recepção, informações ou alterações sobre um determinado prato. Nisso, só estamos falando do lado do cliente.

Do lado do dono, podemos citar alguns questionamentos que assolam um proprietário desse tipo de comércio, tais como a dificuldade de obter informações mais precisas sobre um cliente, saber precisamente o intervalo de tempo de rotatividade das mesas, se aquela quantidade de garçons está suficiente ou superestimada, obter uma estimativa precisa do tempo de espera de uma mesa, saber direcionar o empreendimento para o perfil de consumo de sua clientela. Enfim, esses são somente alguns pontos que inviabilizam tomadas de decisões mais direcionadas, devido à falta de informações mais concretas.

Toda a idéia está fundamentada na instalação de monitores touch-screen em cada uma das mesas do estabelecimento. Nessas telas o cliente terá acesso a uma infinidade de possibilidades, desde solicitações de pedidos a visualizações dos pratos e do cardápio (vide Fig. 1.1).

Essa tela de acesso em cada uma das mesas será aqui chamada de terminal-cliente. Cada um desses terminais estará conectado aos demais terminais presentes na recepção, cozinha, bar e caixa, através de uma rede wireless presente no estabelecimento. Graças a essa rede, a mesma poderá ser acessada através de PDAs, celulares, notebooks dentro do estabelecimento, e acessada através da internet estando fora dela.

Cada pedido solicitado pelo cliente será enviado ao terminal correspondente.



**Figura 1.1:** Terminais Cliente.

Se for um prato quente ou sobremesa o terminal-cozinha o receberá, se for uma bebida será repassada para o terminal-bar. À medida que o pedido chega, uma fila de pedidos se forma na tela do terminal, informando a quantidade, a mesa, o item solicitado e a hora em que tal item foi pedido. Assim que o responsável tiver em mãos o prato ou a bebida, automaticamente um aviso é enviado ao cliente de que o prato solicitado já está pronto.

Vale destacar que, dependendo do tipo de estabelecimento, bares por exemplo, os terminais clientes vão poder conversar entre si, proporcionando uma interatividade maior no local, através da troca de mensagens (torpedos), ou visualização dos integrantes da mesa através de uma webcam.

O apelo visual também será uma forte marca do sistema. Através de fotos com alto grau de definição, será aguçado um forte desejo nos clientes de pedir aquele prato ou sobremesa, aumentando dessa forma as vendas do estabelecimento. Além disso, o cliente poderá ter uma visualização de como é o prato (vide Fig. 1.2), além de uma descrição do mesmo, com os ingredientes que o compõe, calorias, preço, quantidade de pedidos. Uma busca também poderá ser feita retirando do cardápio ingredientes que causem alergia ao cliente, tais como leite, glúten, pimenta, etc.

A qualidade no atendimento é um dos fatores que mais influenciam a volta de um cliente ao estabelecimento. Com esse sistema além de querermos propor-



**Figura 1.2:** Visualização do Prato.

cionar uma melhora nesse quesito, também melhoramos o canal de transmissão de informações e entretenimento, podendo ser adicionados um pacote de jogos, músicas, clipes, tornando a permanência do cliente muito mais agradável.

Haverá toda uma base de dados que vai permitir explorar, quanto um determinado cliente consome por refeição, quais pratos saem mais em determinado dia e horário, tempo de permanência de cada cliente, tempo de preparo dos pratos. Enfim uma gama de possibilidades que ajudará o proprietário a modelar melhor o seu negócio.

São todas essas inovações acima relatadas que motivam o desenvolvimento desse projeto. O mundo dos restaurantes, bares, pizzarias e churrascarias nunca mais será o mesmo depois dessas idéias.

## 2 Recursos Utilizados

### 2.1 Introdução

Para a realização de sistemas de automação que exijam robustez e agilidade no desenvolvimento são necessárias tecnologias que permitam o desenvolvimento, publicação e manutenção rápidos. Também é necessário que seja possível o acesso do usuário ao sistema, independente da plataforma que ele está usando.

Uma ferramenta básica que permite esta facilidade de acesso é o uso de aplicações web. E para isto, ferramentas de desenvolvimento que permitam uma integração rápida com bancos de dados são de grande importância

Foi optado pelo uso da tecnologia *Ruby on Rails* <sup>[5]</sup> para o desenvolvimento das aplicações. O software utilizado para a criação do banco de dados é o *MySQL*.

### 2.2 *Ruby on Rails*

O *Ruby on Rails* surgiu da integração de duas ferramentas: o *Ruby*, que é uma linguagem orientada a objetos, utilizada para a manipulação de dados dentro da aplicação web, e o *Rails*, que é uma ferramenta para o desenvolvimento da aplicação.

#### 2.2.1 *Ruby*

Com a praticidade do desenvolvimento em linguagens com orientação a objetos, o *Ruby* integra o processamento de dados ao código *HTML*, permitindo que o acesso a partes do modelos que acessem o banco de dados, que gerem a visualização da interface com o usuário, ou que façam o processamento dos dados e comandos recebidos sejam separados de forma lógica e funcional dentro da codificação <sup>[6]</sup>.

O uso dessa linguagem representa uma vantagem, pois interage fortemente

com o modelo de processamento utilizado pelo *Rails*, que será explicado a seguir.

### 2.2.2 *Rails*

*Rails* é uma estrutura para o desenvolvimento de aplicações-web orientadas a bancos de dados, baseado na junção de quatro *frameworks*. O *Rails* permite um fácil acoplamento com o Ajax <sup>[4]</sup>.

#### 2.2.2.1 *Active Record*

O *Active Record* é a camada de mapeamento objeto-relacional (ORM, em inglês) fornecida pelo *Rails*, responsável por atividades como a conexão com o banco de dados, o mapeamento de tabelas e a manipulação de dados.

O padrão de mapeamento objeto-relacional, em que as tabelas do banco de dados são mapeadas como classes e as linhas dessa tabela, como objetos, é utilizado pelo *Rails* de forma que a quantidade de configurações executadas pelo desenvolvedor é minimizada.

#### 2.2.2.2 *Action Pack*

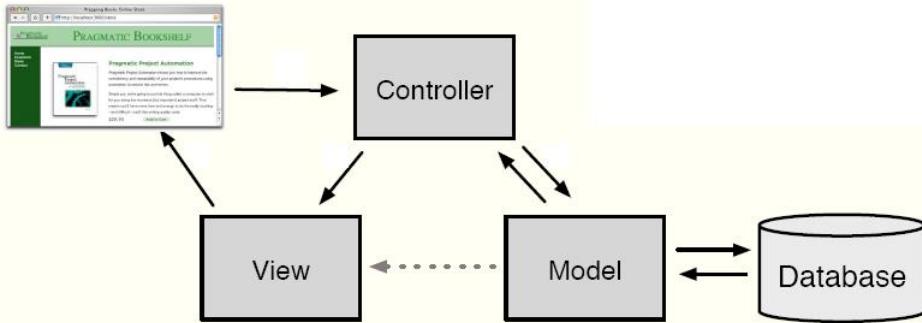
O *Rails* utiliza uma arquitetura para o desenvolvimento de aplicações interativas desenvolvida em 1979, que quebra a aplicação em três componentes distintos: o *modelo*, a *visualização* e o *controlador* <sup>[5]</sup>.

O *modelo* é responsável pela manutenção do estado da aplicação, seja ele transitório, quando está disponível ao usuário por alguns instantes, seja ele permanente, quando interage com banco de dados para a armazenagem de informações. Já o *controlador* recebe os eventos externos, interage com o modelo e renderiza uma visualização ao usuário.

A *visualização*, que está compreendida no *framework Action View*, que encapsula as funcionalidades necessárias para a renderização ao usuário, é responsável por gerar uma interface, normalmente baseada nos dados do modelo.

O *Action Controller* é responsável pela especificação de que modelo, ou controlador, uma requisição deve acessar.

A Fig. 2.1 ilustra o modo de funcionamento da arquitetura de *Modelo-Visualização-Controlador*, também chamada de MVC, em que o navegador da internet envia uma requisição, o controlador interage com o modelo, o controlador chama uma visualização e a visualização disponibiliza uma nova tela para o



**Figura 2.1:** Arquitetura Modelo-Visualização-Controlador.

navegador de internet.

#### 2.2.2.3 *Action Mailer*

*Action Mailer* é um componente simples do *Rails* que permite às aplicações enviar e receber e-mails.

#### 2.2.2.4 *Active Support*

*Active Support* é um conjunto de bibliotecas compartilhadas por todos os componentes do *Rails*. Muitos desses componentes são de uso interno. Porém o *Active Support* estende algumas classes interessantes e úteis do *Ruby* para o desenvolvedor.

## 2.3 MySQL

O *MySQL*<sup>[2]</sup> é um sistema de gerenciamento de banco de dados<sup>[1]</sup> que utiliza a linguagem SQL como interface.

Algumas de suas vantagens são a sua portabilidade, que possibilita a sua utilização em diversos sistemas operacionais, a sua compatibilidade com diversas linguagens de programação, o que inclui o *Ruby*, e o fato de ser um software livre. Possui também uma alta confiabilidade e performance.

## 2.4 Conclusão

As ferramentas aqui apresentadas têm sido cada vez mais difundidas entre desenvolvedores de *web applications* pela facilidade de uso e manutenção, bem como a agilidade que essas ferramentas proporcionam.

Com isso, é possível o desenvolvimento de sistemas com alto nível de eficiência.

# 3 Análise Dinâmica

## 3.1 Introdução

O sistema de pedidos em estabelecimentos da área de alimentação podem ser caracterizados como sistemas de eventos discretos pela sequencialidade e dependência entre os eventos.

Dessa forma, com o intuito de obter um modelo que fosse representativo para a obtenção de informações sobre o comportamento do sistema, foi construída uma rede de Petri contendo os eventos essenciais do funcionamento de um restaurante, bem como seus personagens principais (garçom, cozinha e recepção).

Com isso, foi possível fazer a identificação de pontos de *gargalo* no funcionamento do estabelecimento, bem como notar a atuação do sistema de automação no processo de execução de pedidos.

## 3.2 Redes de Petri

Rede de Petri <sup>[3]</sup> é uma técnica de modelagem que permite a representação de sistemas, utilizando como alicerce uma forte base matemática. Essa técnica possui a particularidade de permitir modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos.

A representação gráfica de uma rede de Petri básica é formada por dois componentes: um ativo chamado de transição (barra) e outro passivo denominado lugar (círculo). Os lugares equivalem às variáveis de estado e as transições correspondem às ações realizadas pelo sistema. Esses dois componentes são ligados entre si através de arcos dirigidos. Os arcos podem ser únicos ou múltiplos.

### 3.3 Modelagem em Redes de Petri

#### 3.3.1 Procedimento Adotado

Foram feitas algumas modelagens em redes de Petri para o funcionamento de um restaurante. Uma delas foi construída para a representação esquemática do funcionamento do estabelecimento e, assim, conseguir bases para o aprofundamento do detalhamento na modelagem.

Nos outros modelos, desejou-se um grau de detalhamento suficiente para a obtenção de comportamentos dinâmicos correspondentes à realidade durante a simulação, e que não podiam ser notados no modelo esquemático simplificado.

Ainda assim, como é intrínseco do processo de modelagem, algumas dinâmicas foram desprezadas, tendo em vista que não adicionariam informações com maior relevância.

As hipóteses adotadas para a simulação foram que:

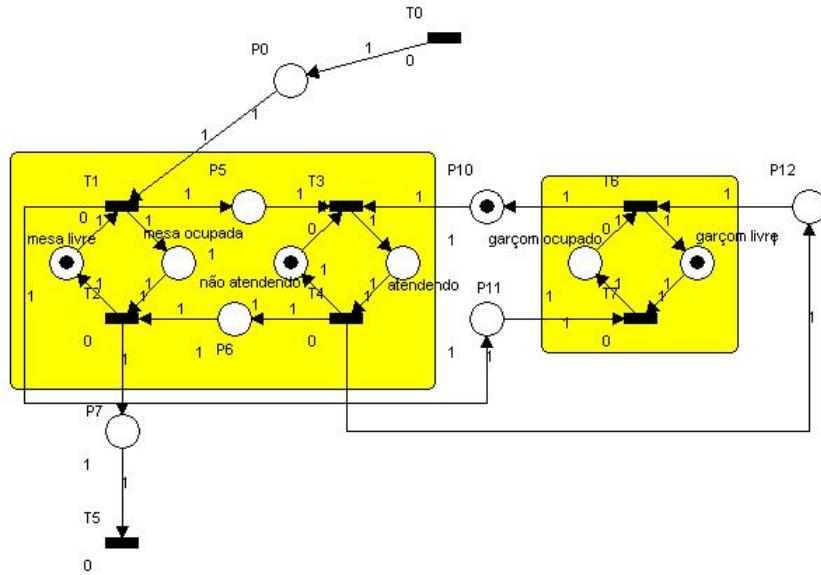
- o garçom atende uma certa quantidade fixa de mesas por vez;
- o processo de pedidos foi simplificado a um único evento;
- todas as transições de estado foram supostas imediatas, uma vez que não há dados estatísticos para a imposição de tempos de transição;
- a obtenção de dados de nível de ocupação das personagens do sistema foram feitas para o restaurante com uma demanda de mil clientes.

#### 3.3.2 Modelo Simplificado

O modelo simplificado tem em vista apenas mostrar os graus de dependência entre cada elemento do sistema representado na rede de Petri.

Assim, o funcionamento do restaurante pode ser descrito da seguinte forma:

1. O cliente chega ao estabelecimento;
2. Se a mesa estiver livre, ele a ocupa;
3. Tendo ocupado a mesa, se o garçom estiver livre, o cliente é atendido;
4. Após ser atendido, o garçom é liberado;
5. O cliente deixa o estabelecimento, liberando a mesa.



**Figura 3.1:** Rede de Petri do Modelo Simplificado.

Com estes requisitos, foi construída a rede de Petri mostrada na Fig. 3.1

Nessa simulação não são descritas atividades referentes ao atendimento do cliente, como o pedido, refeição e pagamento.

Foram distinguidas as entidades em dois blocos diferentes, representados na cor amarela (Fig. 3.1): um representando os estados da mesa e as atividades do cliente, e outro referente ao estado do garçom.

A partir desse modelo é possível que se extraia a modelagem necessária para se obter um maior nível de detalhes. No caso, a parte de atendimento foi mais bem descrita.

### 3.3.3 Estabelecimento Tradicional

Como pode se visto no modelo simplificado, para aumentar o número de mesas dentro da modelagem, era necessário que fossem colocados outros blocos semelhantes àquele referente à mesa na Fig. 3.1.

A modelagem do restaurante convencional, sem automação dos pedidos, foi feita descrevendo melhor as atividades referentes ao processo de atendimento do cliente e incluindo outras mesas no modelo.

Ao todo foram modeladas três mesas por garçom, e com isso foi retirado o grau de ocupação do garçom, de acordo com as hipóteses assumidas no Item 3.3.1.

A ordem de acontecimento dos eventos é:

1. O cliente chega ao estabelecimento;
2. Se a mesa estiver livre, ele a ocupa;
3. Tendo ocupado a mesa, se o garçom estiver livre, o cliente faz o pedido;
4. O garçom leva o pedido para a cozinha e, tornando-se livre para outras requisições;
5. Quando o prato está pronto, se o garçom estiver livre, ele leva o pedido à mesa do cliente;
6. O cliente recebe o prato e o garçom é liberado;
7. Ao terminar a refeição, o cliente pede a conta para o garçom;
8. Após pagar a conta, o garçom é liberado;
9. O cliente deixa o estabelecimento, liberando a mesa.

A Fig. 3.2 representa o modelo que simula as atividades descritas neste Item para a capacidade de atendimento de 1 mesa por garçom.

Neste caso, além de serem mais bem descritas as atividades de atendimento do cliente, foi incluído também, o modelo de atividades que ocorrem na cozinha.

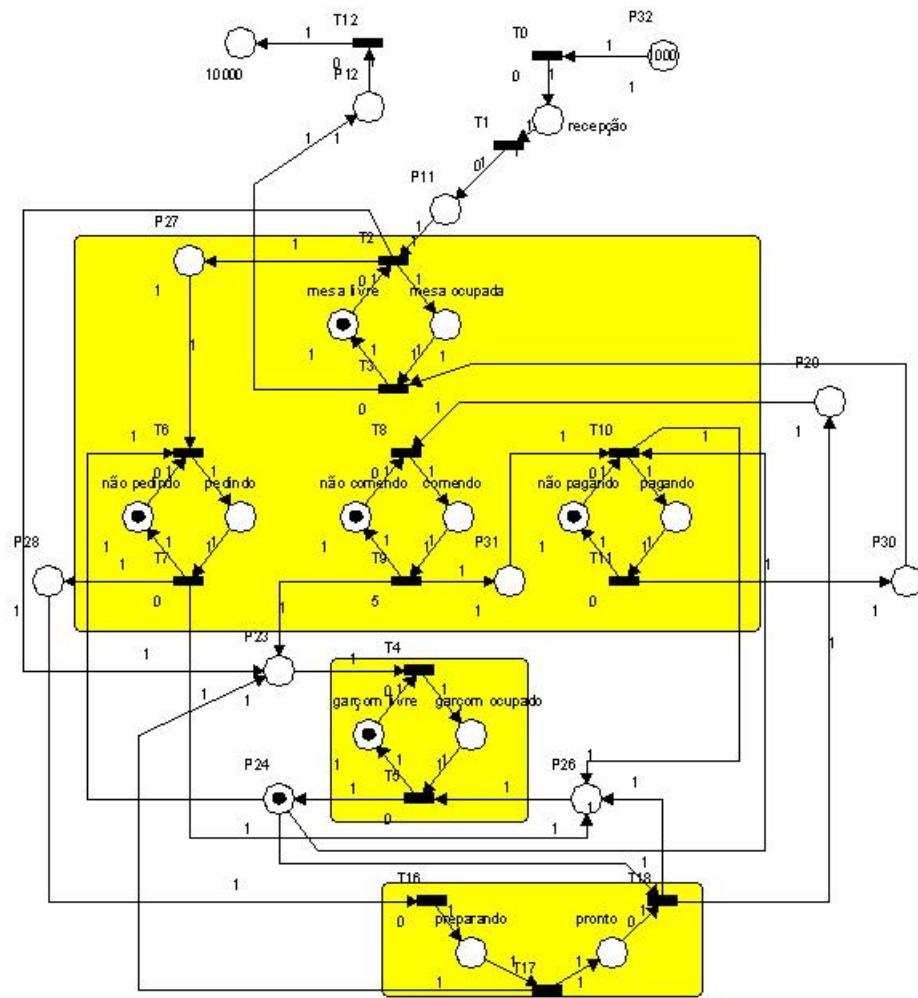
Para se medir a eficiência do sistema, foram monitoradas duas variáveis: o número de *steps* necessários para o atendimento dos 1000 clientes que compõem a demanda do restaurante, e a razão entre o número de *steps* em que o garçom está ocupado e o número total de *steps* para o atendimento da demanda.

Por meio dessas duas variáveis consegue-se a noção do tempo total de atendimento e do valor percentual de tempo em que o garçom está ocupado que otimiza o tempo total de atendimento.

Tendo conseguido o modelo da Fig. 3.2 que representa corretamente o comportamento descrito neste item, foram inseridas mais mesas no modelo, conseguindo-se os resultados apresentados na Tab. 3.1

Nota-se que para o número de uma ou duas mesas por garçom, o tempo de ocupação dele para o atendimento da demanda do restaurante permanece constante, porém sua porcentagem do tempo total em que fica ocupado aumenta, por conta do aumento da velocidade de atendimento.

O caso em que cada garçom deve atender duas mesas é aquele em que é minimizado o tempo de atendimento dos clientes. Ao se aumentar o número de



**Figura 3.2:** Rede de Petri do Modelo com Atendimento sem Automação com 1 Mesa.

**Tabela 3.1:** Dados obtidos pela simulação do Sistema Tradicional variando o número de mesas

número de mesas por garçom	tempo total em steps	steps em que o garçom esteve ocupado	porcentagem de ocupação do garçom
1	11004	4000	36,35
2	7006	4000	57,09
3	7238	4232	58,47

mesas por garçom para três, o tempo de atendimento total subiu em 232 steps, que é a mesma quantidade de tempo em que subiu o número de steps em que o garçom esteve ocupado, o que significa que o garçom foi o responsável pelo aumento do tempo total de atendimento.

De acordo com as hipóteses assumidas no Item 3.3.1, os dados obtidos com essa simulação não necessariamente correspondem fielmente à realidade, porém eles são proporcionais ao comportamento real, uma vez que não estão modelados os tempos de cada transição.

Com o intuito de maximizar o rendimento bruto por unidade de tempo, o dono do restaurante sem automação de pedidos deveria optar por uma configuração de um garçom para cada duas mesas.

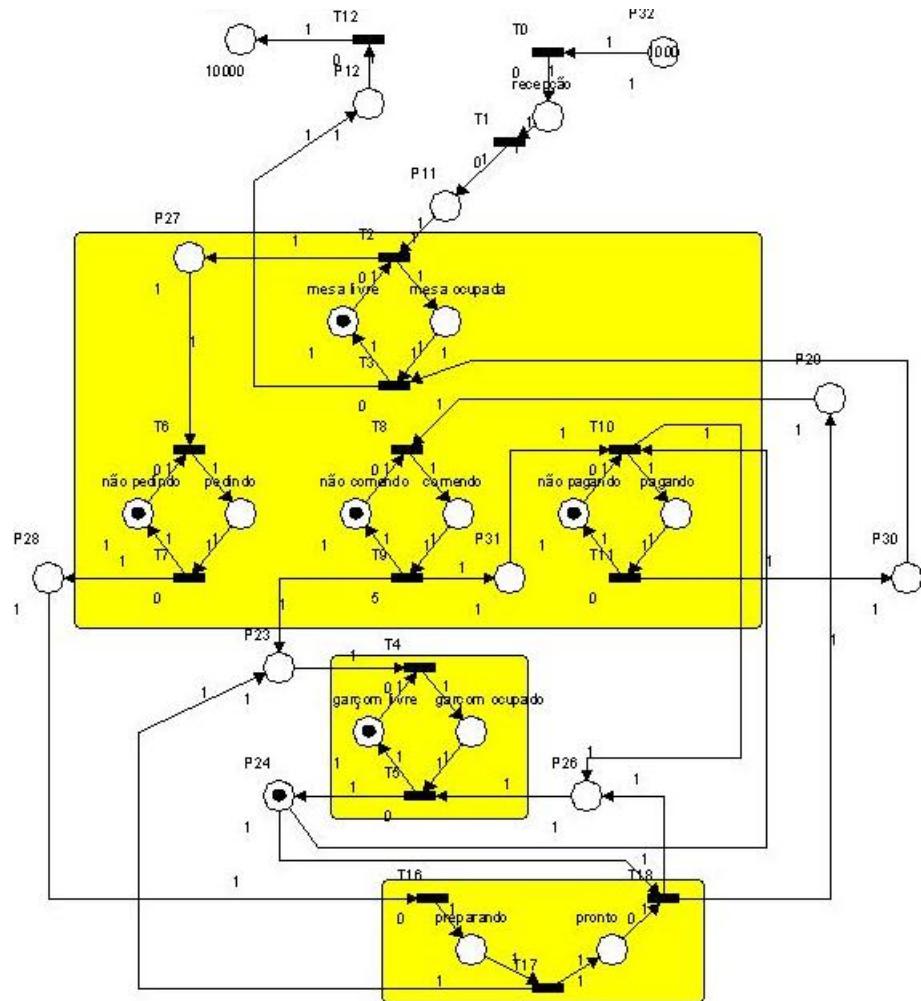
### 3.3.4 Estabelecimento Automatizado

Pôde-se perceber no Item 3.3.3 que a partir de um certo número de mesas que cada garçom deve atender, o funcionário começa a atrasar o funcionamento de todo o sistema. Isto se dá pelo conflito de chamadas recebidas pelo garçom, uma vez que, dado que o garçom é necessário para a execução de um pedido, os clientes de mesas diferentes se tornam dependentes entre si, o que significa que o atendimento de uma mesa fica prejudicado em razão do atendimento de outra.

Para a resolução desse problema é proposta a automação dos pedidos em estabelecimentos da área de alimentação.

De acordo com o que foi discutido, a automação será aplicada ao processo de pedidos, eliminando a dependência do garçom para a execução do mesmo. Assim:

1. O cliente chega ao estabelecimento;
2. Se a mesa estiver livre, ele a ocupa;
3. Tendo ocupado a mesa, o cliente faz o pedido;



**Figura 3.3:** Rede de Petri do Modelo com Atendimento com Automação com 1 Mesa.

4. Quando o prato está pronto, se o garçom estiver livre, ele leva o pedido à mesa do cliente;
5. O cliente recebe o prato e o garçom é liberado;
6. Ao terminar a refeição, o cliente pede a conta para o garçom;
7. Após pagar a conta, o garçom é liberado;
8. O cliente deixa o estabelecimento, liberando a mesa.

A rede de Petri que representa este funcionamento, para uma mesa, pode ser vista na Fig. 3.3.

Para a eliminação da dependência do garçom para a realização do pedido, na rede de Petri da Fig. 3.3 foram retirados três arcos orientados em relação a Fig. 3.2.

Dessa forma, foram obtidos os desempenhos mostrados na Tab. 3.2 durante a simulação, ao se variar o número de mesas que cada garçom deveria atender.

**Tabela 3.2:** Dados obtidos pela simulação do Sistema Automatizado variando o número de mesas

número de mesas por garçom	tempo total em steps	steps em que o garçom esteve ocupado	porcentagem de ocupação do garçom
1	11004	2000	18,16
2	6006	2000	33,30
3	4298	2000	46,53
4	4011	2000	49,86

Foram simuladas, ao todo, as situações com até 4 mesas por garçom, situação para a qual a rede de Petri tornou-se muito complexa.

Pode-se notar que, de acordo com as hipóteses assumidas (Item 3.3.1), o número de steps em que o garçom está ocupado durante a simulação caiu pela metade em relação ao sistema tradicional. Isto significa dizer que o processo de pedidos corresponderia a 50% do tempo total em que o garçom estaria ocupado. Isso mostra números que explicam o porquê da automação deste evento.

Com a situação em que o garçom atende uma só mesa por vez, o tempo total necessário para o atendimento não é reduzido (ver Tab. 3.1), porém o grau de ocupação do garçom cai pela metade.

Uma simulação com um maior número de mesas por garçom seria necessária para a identificação da situação que minimiza o tempo de atendimento da demanda.

## 3.4 Pesquisa de Campo

Para uma análise dos dados obtidos por meio da simulação em redes de Petri, foi realizada uma pesquisa em restaurantes da cidade de São Paulo, de porte médio a grande, onde haveria a possibilidade de implementação da solução de automação proposta neste trabalho.

Foi elaborado um questionário tendo como objetivo fornecer maior base de conhecimento do funcionamento interno de um estabelecimento, bem como verificar se resultados apontados pela rede de Petri, como locais de maior gasto de tempo e nível de ocupação de unidades do sistema, correspondem ao observado na realidade.

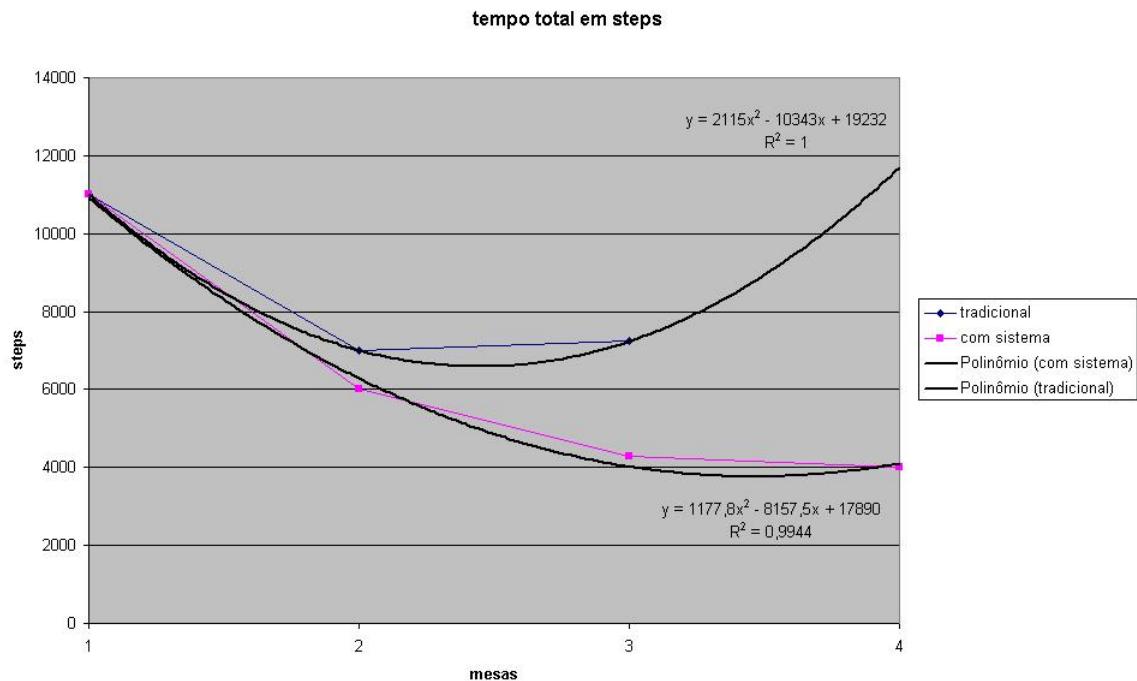
Dentre os locais procurados, foram obtidas respostas dos estabelecimentos citados na Tab. 3.3.

**Tabela 3.3:** Questionário e respostas obtidas da pesquisa de campo

Questão	Pizzaria São Pedro	Belmiro da Vila	Habib's Vila Prudente	Habib's Tucuruvi
Qual a relação (em %) entre o tempo gasto pelo garçom no atendimento de uma mesa e o tempo total de ocupação da mesa?	15% - 20%	17% - 22%	Em torno de 10%	Em torno de 10%
Há ocorrência de fila de espera? Se sim, qual o tempo médio de espera?	Sim, de 40 minutos - 45 minutos	Sim, 25 minutos	Sim. 15 minutos	Sim. 17 minutos
Qual a quantidade de garçons do horário de pico do restaurante?	20 garçons	4 garçons	15 garçons	13 garçons
Com relação à cozinha, qual seria o número de funcionários e o tempo médio de preparo dos pratos?	20 funcionários. 20 minutos — 22 minutos	4 funcionários. 15 Minutos	10 funcionários. 10 minutos	12 funcionários. 10 minutos
Qual seria a % de ocupação do restaurante fora dos horários de pico?	85% - 90%	65%	45% - 50%	35% - 45%
Por quanto tempo dura o horário de pico do estabelecimento?	2,5 horas	1,5 hora	1 hora	1 hora e 20 minutos
Quanto tempo uma mesa leva para ser fechada (logística do pagamento)?	5 a 10 minutos no horário de pico	5 minutos - 7 minutos	3,5 minutos	3,5 minutos
Qual o tempo médio de ocupação de uma mesa?	1 hora - 1,5 hora	60 minutos	20 — 25 minutos	20 minutos

## 3.5 Conclusões

Ao serem comparadas as tabelas obtidas com a simulação das redes de Petri pode-se concluir que a execução de pedidos é uma das partes do processo de atendimento de clientes que mais demandam tempo para serem realizadas.



**Figura 3.4:** Tempo Total de Atendimento vs. Número de Mesas por Garçom.

Pode-se notar que o número de mesas ótimo que cada garçom deve atender por vez que minimiza o tempo de atendimento da demanda de clientes aumenta, o que significa o aumento do rendimento bruto do estabelecimento por unidade de tempo. Ou seja, o número de funcionários necessários para a realização do atendimento passa a ser menor (ver Tab. 3.1 e Tab. 3.2) e, como o tempo de atendimento da demanda do estabelecimento também diminui, a automação do restaurante possibilita uma reorganização do quadro de atividades dos funcionários e que se tenha um maior retorno devido a diminuição do tempo de atendimento.

A Fig. 3.4 mostra a tendência de evolução do tempo de atendimento dos clientes em função do número de mesas que cada garçom atende.

O processo de automação se voltou à dinamização em cima de um recurso compartilhado do sistema, que é o garçom em relação às mesas. Com o alívio de carga sobre este recurso e com o aumento da velocidade de pedidos, outro local que, da mesma forma, é compartilhado (a cozinha) poderá ter que ser ampliada para satisfazer o aumento da demanda.

Pelos fatos observados neste capítulo, é demonstrada a viabilidade da automação do processo de pedidos, tanto em termos de dinâmica, como de retorno financeiro para o dono do estabelecimento.

# 4 Implementação

## 4.1 Introdução

Antes de começar a falar sobre os itens implementados nessa etapa inicial, é primordial entender como será o funcionamento desse projeto como um todo, para então ser detalhado cada um dos módulos que o compõem.

A idéia parte do princípio da instalação de monitores de touch screen em cada mesa do estabelecimento. Esses touchs serão aqui chamados de terminais-cliente (Fig. 4.1). Espalhados pelo estabelecimento, terão outros tipos de terminais que receberão as informações que a ele lhe interessam. Por exemplo, na cozinha todos os pratos e sobremesas solicitados serão transmitidos para um terminal-cozinha (Fig. 4.2). As bebidas para um terminal-bar.

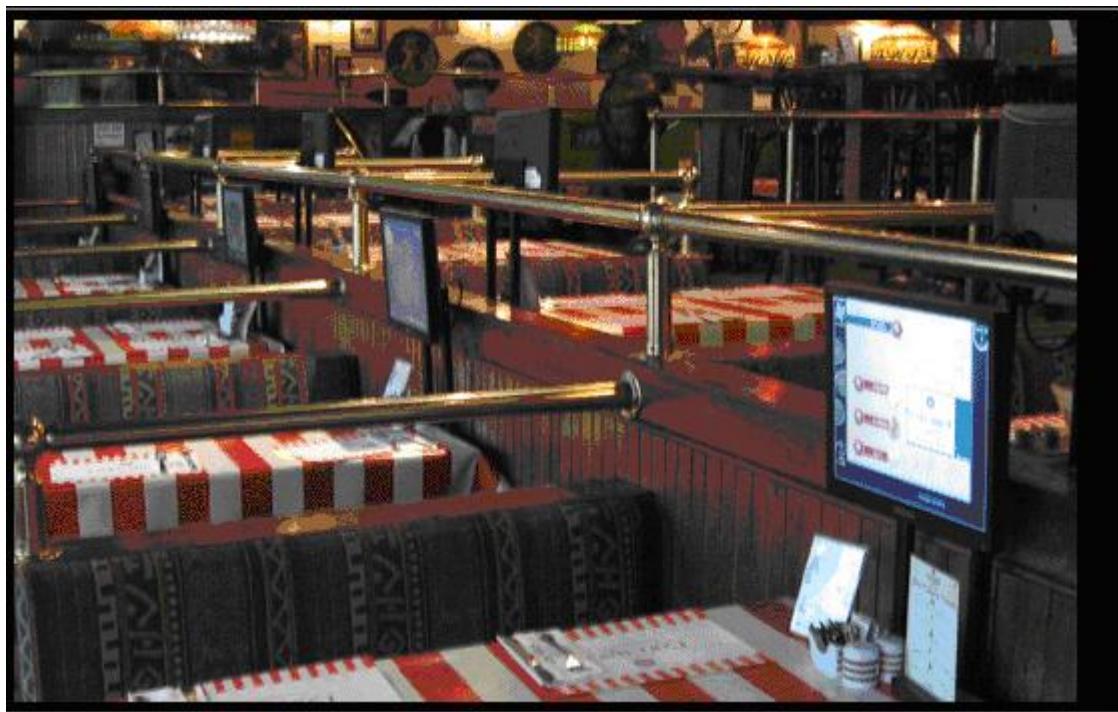
E como os garçons saberão quando os pratos, bebidas e sobremesas estão disponíveis para serem entregues na mesa? Para isso existirão os terminais-garçom, que receberá um indicativo que determinado pedido de uma mesa está pronto.

Além disso, haverá também um terminal-recepção (Fig. 4.3), que receberá a informação de ocupação ou não da mesa. Com isso pode ser feito um mapa do restaurante, sabendo quais mesas estão ocupadas e propiciar uma melhor alocação dos clientes na fila de espera da recepção.

Até agora foram apresentados todos os terminais, que são:

- Terminal cliente;
- Terminal cozinha;
- Terminal bar;
- Terminal recepção.

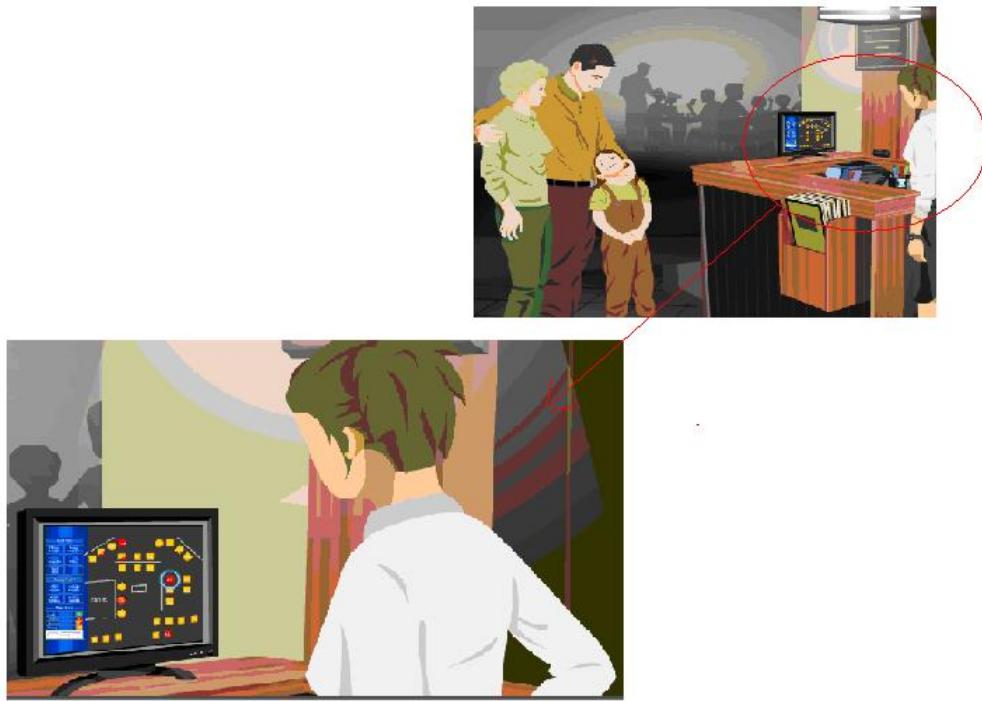
Todos esses terminais vão se comunicar entre si através de uma rede wireless presente no restaurante. Essa mesma rede permitirá que o cliente acesse do celular



**Figura 4.1:** Terminal Cliente.



**Figura 4.2:** Terminal Cozinha.



**Figura 4.3:** Terminal Recepção.

dele, por exemplo, todas as informações que ele teria visualizando pela tela touch screen. Uma rede web possibilitará que, para clientes classificados como confiáveis e que estejam cadastrados no sistema, sejam feitos pedidos remotos.

Por trás de tudo isso vai existir um banco de dados, que cadastrará os usuários, armazenará os pedidos, exibirá detalhes da mesa ocupada e da fila da recepção, informações do cardápio, dos garçons, da conta final. Enfim, será a fonte de estatísticas e constatações do empreendimento. Todas essas informações serão enviadas também para um computador servidor, que fará um backup e rodará os scripts dos terminais.

Quanto à interface, a mesma será feita em *Ruby on Rails*, com integração com o banco de dados através do *MySQL*. É uma das tecnologias mais atuais de aplicações web.

Portanto, o usuário vai visualizar o cardápio, faz o pedido automaticamente sem a ajuda do garçom, e ele é enviado para o terminal especificado. O garçom recebe a confirmação de que o pedido está pronto e ele é entregue na mesa solicitante. Esse seria um fluxo de processo simples com a implementação do sistema.

Muitas outras funcionalidades podem ser adicionadas, como entretenimento, integração entre os clientes (em bares, por exemplo) mas que não serão contem-

plados nesse trabalho.

Nessa primeira fase do projeto, os itens de implementação que serão contemplados serão as atividades de estruturação do banco de dados e esquematização do fluxo de telas da parte do usuário *administração*, portanto vamos detalhar cada uma das partes em dois tópicos.

O primeiro tópico diz respeito ao banco de dados, o qual permitirá ter acesso ao todos os tipos de informações cadastradas pelos usuários e como aproveitar essas informações para o crescimento do negócio. Serão listadas todas as entidades, os relacionamentos e os celeiros que fazem parte da estrutura. No segundo tópico será apresentado o fluxo de telas do usuário *administração*, em tese, o usuário que terá o maior número de responsabilidades.

## 4.2 O Banco de Dados

A tarefa de estruturar um banco de dados é complexa, visto que será o esqueleto e fonte de todos os dados reais e estatísticos que serão coletados pelo sistema. É através dele que vão ser tiradas constatações importantes sobre o estabelecimento, e servirão de base para tomadas de decisões relevantes sobre o negócio.

Para dar suporte a essa teoria, primeiro será apresentado e detalhado o banco de dados e logo em seguida serão apontadas algumas das aplicações que serão possíveis tendo esse conjunto de informações em mãos e como trabalhar com ele.

Como pode ser visto na Fig. 4.4, as entidades são: usuário, administrador, cliente, garçom, recepção, alternativa de cardápio, comanda, cardápio, ingrediente, mesa e classificação. Os celeiros vão ser: pratos diários, bebidas pedidas, estatísticas mesa, fila de espera e reserva.

As colunas essenciais de cada entidade estão especificadas na Tab. 4.1.

Os celeiros irão possuir as informações contidas na Tab. 4.2

## 4.3 Fluxo de Telas do Administrador

O usuário administrador englobará as responsabilidades de todos os outros usuários do sistema. Uma tela inicial vai possibilitar a escolha do tipo do usuário. Para ter acesso a tela de administração, será necessário login e senha. Essa tela não ficará habilitada nos terminais clientes, pois será necessário saber a url, o login

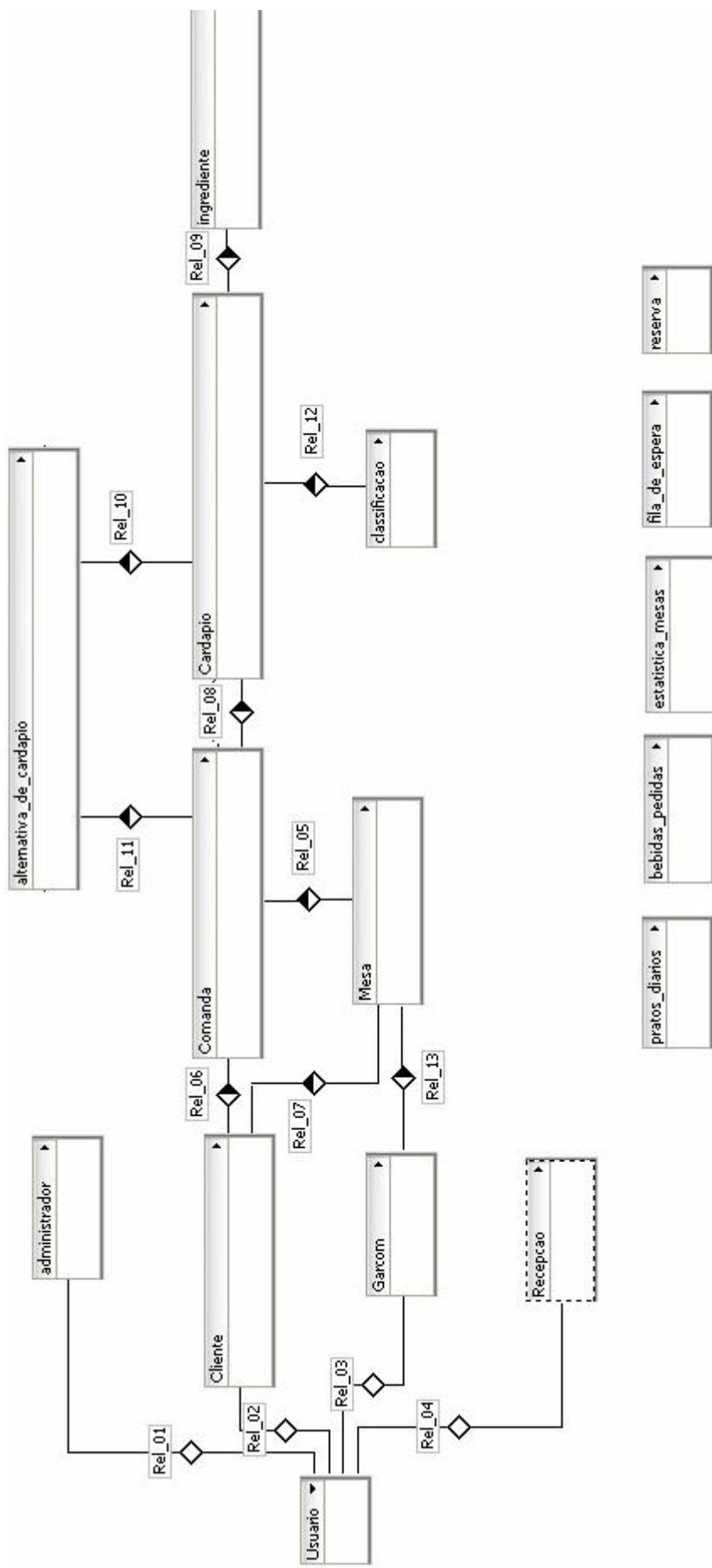


Figura 4.4: Estrutura do Banco de Dados.

**Tabela 4.1:** Colunas Essenciais das Entidades do Banco de Dados

Entidade	Colunas				
Usuário	ID				
Administrador	login	senha			
Cliente	RG	CPF			
Garçom	Nome	Turno			
Recepção	login	senha			
Comanda	ID	Pedidos	Status		
Cardápio	Nome	Preço	Calorias	Descrição	Foto
Ingrediente	Nome	Componente Alérgico			
Mesa	ID	Capacidade	Status	Área	
Alternativa de Cardápio	Nome				
Classificação	Tipo de Prato				

**Tabela 4.2:** Colunas dos Celeiros do Banco de Dados

Celeiro	Colunas				
Pratos Diá- rios	Tempo de Preparo	Procura	Contabilidade		
Bebidas Pe- dididas	Procura	Contabilidade			
Estatística Mesa	número de pessoas	Gastos	Tempo de Permanência	Número de Pedidos Feitos	
Fila de Es- pera	ID Cliente	ID Fila de Espera	Número de Acompa- nhantes		
Reserva	ID Cliente	Número de Acompa- nhantes	Senha	Horário	

e senha do usuário, ou seja, ao cliente não será repassada essa responsabilidade.

Optando pela opção administrador na tela inicial, uma segunda tela se abre, solicitando ação a ser efetuada (Cadastro Cardápio, Cadastro de Garçom, Cadastro Mesa, Cadastro de Clientes, Cadastro de Recepção, Cadastro de Comanda, Estatísticas). Esses itens são visíveis à esquerda da tela.

Os menus que foram implementados até o momento são: Login, Itens do Cardápio, Usuários, Administradores e Logout. Agora será detalhado cada um dos fluxos dos menus listados.

1. Login: Aqui o usuário com a responsabilidade de administrador e que já



**Figura 4.5:** Login Administrador.



**Figura 4.6:** Login Administrador - Mensagem de Erro.

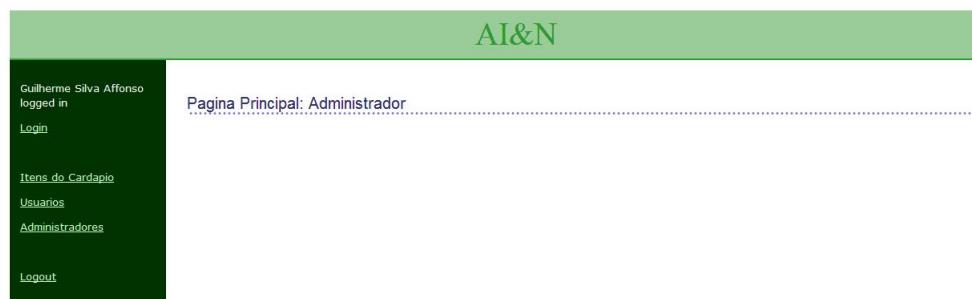
está cadastrado no sistema, entra com o login e senha (Fig. 4.5).

Caso o login e a senha não sejam os cadastrados no banco de dados, uma mensagem de erro aparece na tela (Fig. 4.6).

Entrando com os dados corretos, seu nome de usuário surge no campo superior esquerdo, confirmando o acesso como administrador (Fig. 4.7).

2. Itens do Cardápio: Nesse link, é realizado o cadastro dos itens do cardápio. Estando logado no sistema, o usuário consegue ter acesso a esse item. Ao clicar em "Itens do cardápio", visualiza-se a tela da Fig. 4.8.

Na Fig. 4.8 observa-se que à esquerda surge a foto do prato, no centro a descrição dele e à direta um menu com as opções de "Exibir", "Editar" e "Destruir".



**Figura 4.7:** Página Principal - Administrador.

**Figura 4.8:** Itens do Cardapio.

**Figura 4.9:** Exibição dos detalhes do Prato.

Clicando em exibir, surge outra tela dando as informações do prato em questão (Nome do prato, Descrição do prato, Preço do prato, a url da foto e o valor calórico)(figura 4.9).

Clicando em "Voltar", retorna-se a pagina da Fig. 4.8. Agora clicando em Editar, acessa-se a tela abaixo (Fig. 4.10), onde se pode editar qualquer um dos itens do prato.

Na opção "Destruir" é apagado o registro em questão, aparecendo uma mensagem de confirmação de exclusão (Fig. 4.11). Clicando em "Ok" o registro é excluído.

Para cadastrar um novo prato, clica-se no link "Novo item" presente na Fig. 4.8, na parte inferior esquerda. Surge então um formulário em branco (Fig. 4.12) para que sejam inseridas as informações referentes ao novo prato. Clicando em "Criar" o novo prato é adicionado a listagem.

3. Usuários: Nesse item é feita uma listagem de todos os usuários do sistema,

Guilherme Silva Affonso  
logged in

Login

Itens do Cardapio

Usuarios

Administradores

Logout

Edicao de Item do Cardapio

Prato  
Brigadeiro

Descricao  
Doce de chocolate recheado com chocolate granulado

Preco  
3.50

Foto url  
/images/brigadeiro.jpg

Valor calorico  
150.00

Editar | Exibir | Voltar

Figura 4.10: Edição do Prato.

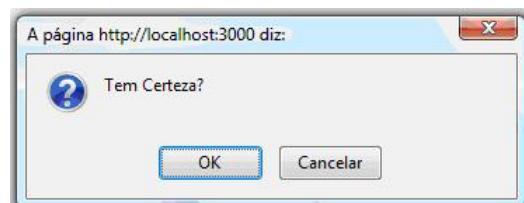


Figura 4.11: Exclusão do Prato.

Guilherme Silva Affonso  
logged in

Login

Itens do Cardapio

Usuarios

Administradores

Logout

Novo Item do Cardapio

Prato

Descricao

Preco  
0.0

Foto url

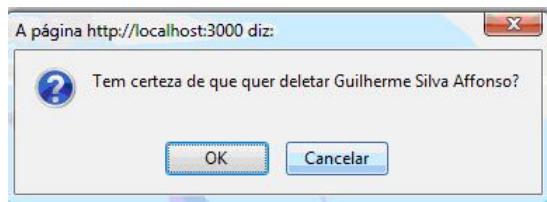
Valor calorico  
0.0

Gravar | Voltar

Figura 4.12: Novo Prato.



**Figura 4.13:** Lista de Usuários.



**Figura 4.14:** Exclusão de Usuário.

cadastrados por um administrador. Logo que se clica nesse link, é exibida a seguinte tela inicial (Fig. 4.13), mostrando que, nesse caso, existem cinco usuários cadastrados no sistema.

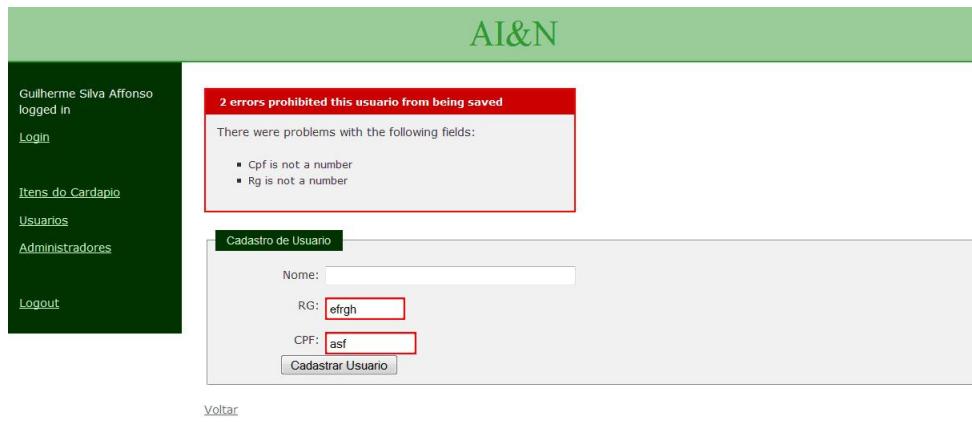
Para se apagar algum usuário, basta clicar no "X" localizado à esquerda do nome, surgindo uma mensagem de alerta para a exclusão (Fig. 4.14), com o nome do usuário a ser excluído. Clicando em "Ok" o usuário é apagado do sistema e aparece uma mensagem de confirmação da ação efetuada.

Para se cadastrar um novo usuário, acessa-se o link "Novo usuário". Um formulário em branco surge, sendo necessário adentrar com o Nome, o CPF e o RG do novo usuário (Fig. 4.15).

Caso o CPF e o RG digitados não sejam números, é emitida uma mensagem de erro para correção dos dados (Fig. 4.16). Efetuado o cadastro, uma



**Figura 4.15:** Novo Usuário.



The screenshot shows a user interface for 'AI&N'. The top navigation bar is green with the text 'AI&N'. The left sidebar is dark green with the following menu items: 'Guilherme Silva Affonso logged in', 'Login', 'Itens do Cardápio', 'Usuários', 'Administradores', and 'Logout'. The main content area has a white background. A red-bordered box displays an error message: '2 errors prohibited this user from being saved' and 'There were problems with the following fields: Cpf is not a number, Rg is not a number'. Below this, a form titled 'Cadastro de Usuário' is shown with fields for 'Nome' (empty), 'RG' (containing 'efrgh'), and 'CPF' (containing 'asf'). A 'Cadastrar Usuário' button is at the bottom of the form. At the bottom left of the main area is a 'Voltar' link.

**Figura 4.16:** Erro cadastrando Novo Usuário.



The screenshot shows a user interface for 'AI&N'. The top navigation bar is green with the text 'AI&N'. The left sidebar is dark green with the following menu items: 'Guilherme Silva Affonso logged in', 'Login', 'Itens do Cardápio', 'Usuários', 'Administradores', and 'Logout'. The main content area has a white background. A section titled 'Administradores' is shown with two entries: 'Fernando Torres Pereira da Silva' and 'Guilherme Silva Affonso', each preceded by a checkbox. Below this, there are links: 'Alterar Senha' and 'Novo Administrador'. At the bottom left of the main area is a 'Voltar' link.

**Figura 4.17:** Lista de Administradores.

mensagem de confirmação é exibida.

4. Administradores: Aqui são cadastrados, dentro os usuários do sistema, aqueles que terão como responsabilidade a tarefa de administrador (Fig. 4.17). Clicando no "X" localizado à esquerda do nome, o administrador é excluído, aparecendo uma mensagem de alerta da ação efetuada.

Algumas outras tarefas podem ser realizadas, como alteração de senha e cadastro de novo administrador.

Clicando em "Alterar senha" , um formulário em branco surge, sendo necessário inserir a senha antiga, a nova senha e confirmação da nova senha desejada (Fig. 4.18). Ao confirmar, a nova senha do usuário é cadastrada.

Clicando em "Novo administrador" , um formulário surge para que seja escolhido, dentro os usuários cadastrados, aquele na qual será atribuída a responsabilidade de administrador (Fig. 4.19). Escolhido o usuário, é necessário ainda cadastrar um login, uma senha, e confirmar a senha digitada. Efetuando o cadastro uma mensagem de confirmação de cadastro aparece.

5. Logout. Nesse item é realizado o "logout" do administrador conectado ao

**Figura 4.18:** Alterar senha de administrador.

**Figura 4.19:** Cadastrar novo de administrador.

sistema. Ao clicar no ícone "logout" a desconexão é realizada automaticamente, surgindo uma mensagem de que o usuário foi desconectado e surgindo os campos para um próximo "login" (Fig. 4.20).

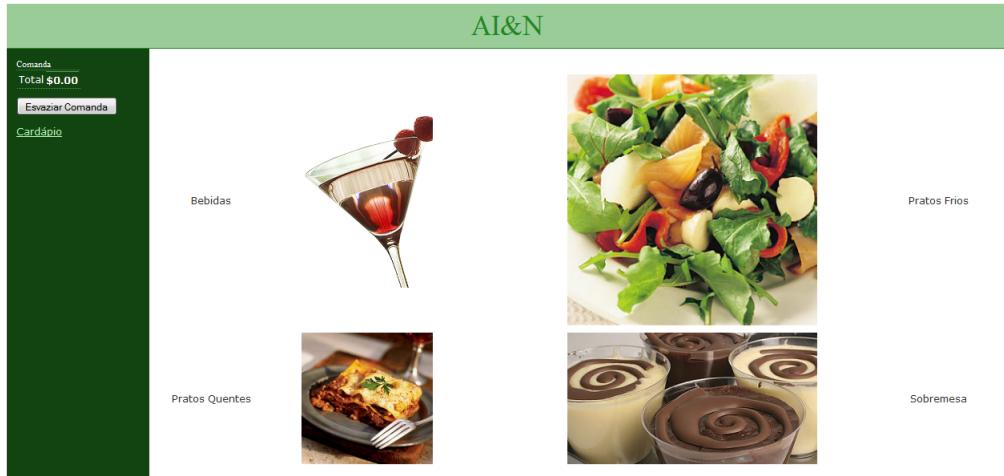
## 4.4 Fluxo de Telas do Cliente

Quando o cliente adentrar o restaurante e sentar-se a mesa, terá a disposição um terminal para solicitação de pedidos. A seguir será detalhada uma possibilidade de interface para esse auto-atendimento, onde o cliente pode visualizar o cardápio, os itens pedidos e as comandas.

**Figura 4.20:** Logout.



**Figura 4.21:** Tela inicial da interface de clientes.



**Figura 4.22:** Macro divisões do cardápio.

A Fig. 4.21 ilustra um protótipo da tela inicial que o cliente visualizaria. Nela temos, no menu à esquerda, o ícone cardápio e a enumeração dos itens pedidos (nome, quantidade e o preço). Abaixo há um botão para esvaziar a comanda. Como se percebe, ao se adentrar no sistema, a comanda está vazia (sem itens solicitados).

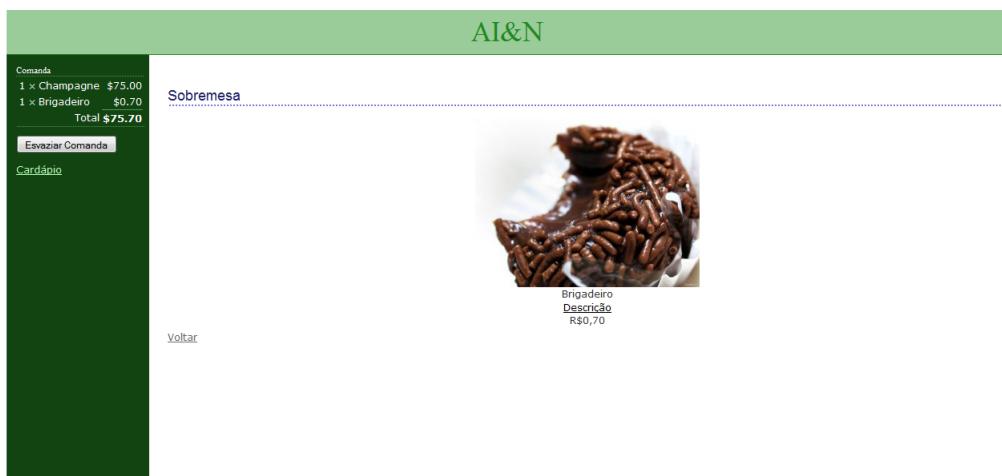
Optando pelo item cardápio, surgem as opções dos tipos de pratos cadastrados. Na Fig. 4.22, pode-se ver as macro divisões utilizadas (pratos quentes, pratos frios, bebidas e sobremesas).

Clicando na opção "Bebidas", são visualizadas as bebidas cadastradas no cardápio. Na figura abaixo (Fig. 4.23), vê-se a bebida champagne sendo inserida na comanda da mesa, mostrando o nome, a quantidade de itens pedidos, o preço do item escolhido e o total da conta.

Voltando ao cardápio e escolhendo agora o item "Sobremesas", mais uma vez



**Figura 4.23:** Bebidas.



**Figura 4.24:** Sobremesas.

são mostrados os itens cadastrados como sobremesas (Fig. 4.24). Optando por algum das opções disponíveis, ela automaticamente é inserida a comanda. Note que o total geral é atualizado.

O mesmo acontece clicando na opção "Pratos Quentes" do cardápio. Nela são visualizados os pratos classificados como quentes. Optando pelo prato, ele é adicionado a comanda e a valor total a ser pago é atualizado.

E por último temos a seção de "Pratos Frios" (Fig. 4.26). Optando por ela se visualiza os pratos que se encaixam nesse tipo de classificação. Escolhendo um determinado prato, ele é automaticamente adicionado a comanda e o valor total da conta é atualizado.

Nesse exemplo simulado, foi escolhido um prato constante de cada tipo, ilustrando o comportamento dos itens sendo inseridos na comanda da mesa e o valor total sendo atualizado. Há a possibilidade ainda de esvaziar a comanda, bastante

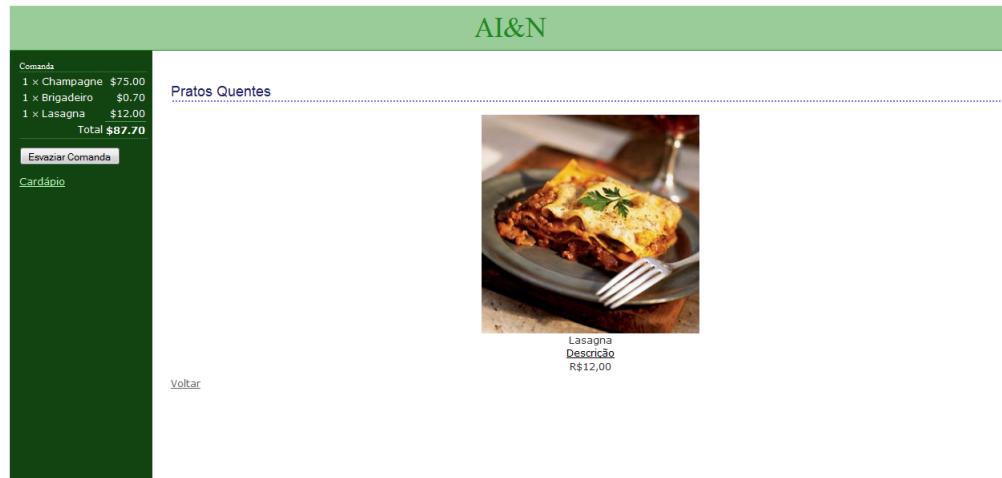
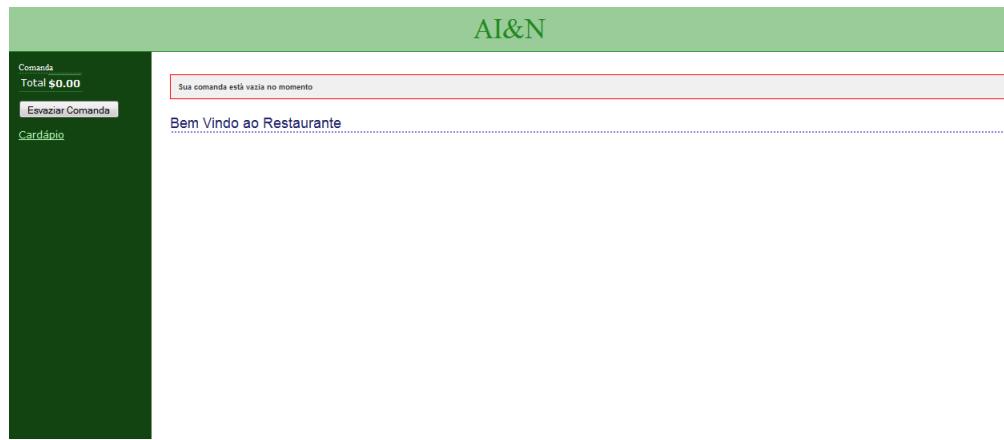


Figura 4.25: Pratos Quentes.



Figura 4.26: Pratos Frios.



**Figura 4.27:** Esvaziar Comanda.

para isso clicar no botão "Esvaziar Comanda" e os itens são apagados (Fig. 4.27).

# Apêndice A – Código-Fonte

## A.1 cliente\_controller.rb

```

class ClienteController < ApplicationController

  def index
    @cart = achar_cart
  end

  def escolha_classif
    @cart=achar_cart
    @all_classif=ClassificacaoCardapio.find(:all,
      :order=>'classificacao')
  end

  def cardapio_exibir
    @cart=achar_cart
    @cardapios = Cardapio.find(:all,
      :conditions=>["classificacao_id=?",params[:id]])
  end

  def cardapio_detalhe
    @cart=achar_cart
    @cardapio=Cardapio.find(params[:id])
  end

  def adicionar_ao_cart
    begin
      cardapio = Cardapio.find(params[:id])
      rescue ActiveRecord::RecordNotFound
        logger.error("Tentativa de acesso a produto inválido #{params[:id]}")
        flash[:notice] = "Produto Inválido"
        redirect_to :action=>:index
      else
        @cart = achar_cart
        @item_atual = @cart.adiciona_prato(cardapio)
      end
    end

    def esvaziar_cart
      session[:cart] = nil
      flash[:notice] = "Sua comanda está vazia no momento"
      redirect_to :action=>:index
    end
  end

```

```

private

def achar_cart
  session[:cart] ||= Cart.new
end

end

```

## A.2 cardapio.rb

```

class Cardapio < ActiveRecord::Base

  has_many :item_pedidos
  has_many :alternativa_de_pratos
  has_many :ingrediente_cardapios
  belongs_to :classificacao_cardapio

  validates_presence_of :prato, :descricao,
    :preco, :foto_url,
    :valor_calorico
  validates_numericality_of :preco
  validates_uniqueness_of :prato
  validates_format_of :foto_url,
    :with => %r{\.(gif|jpg|png)$}i,
    :message => "deve ser uma URL com formato de imagem GIF, JPG, ou PNG"

end

```

## A.3 adicionar\_ao\_cart.rhtml

```

<div class="cart-title" >Sua Comanda</div>
<table>
<% for cart_item in @cart.items %>
<tr>
<td><%= cart_item.quantidade %>&times;</td>
<td><%= h(cart_item.prato) %></td>
<td class="item-price" ><%= number_to_currency(cart_item.preco) %></td>
</tr>
<% end %>
<tr class="total-line" >
<td colspan="2" >Total</td>
<td class="total-cell" ><%= number_to_currency(@cart.preco_total) %></td>
</tr>
</table>
<%= button_to "Esvaziar Comanda" , :action => :esvaziar_cart %>

```

## A.4 cardapio\_exibir.rhtml

```

<h1><%= ClassificacaoCardapio.find_by_id(params[:id]).classificacao%></h1>

<table cellpadding="5" cellspacing="0" align="center">
<%i=1 -%>
<%for cardapio in @cardapios%>

```

```

<%if i==1 -%>
<tr valign="center">
<%end -%>
<td align="center">
  <%=link_to image_tag("#{cardapio.foto_url}", :border=>0),
  #
  :url=>{
    :action=>:adicionar_ao_cart, :id=>cardapio #}
  %><br/>
  <%=h(cardapio.prato)%><br/>
  <%=link_to "Descrição", :action=>:cardapio_detalhe, :id=>cardapio%><br/>
  <%=number_to_currency(cardapio.preco, :unit=>"R$",,
  :separator=>",",
  :delimiter=> "")%>
</td>
<%i=i+1 -%>
<%if i==5 -%>
</tr>
<%i=i-4 %>
<%end -%>
<%end%>
<%if i!=1 -%>
</tr>
<%end -%>
</table>
<%=link_to "Voltar", :action=>:escolha_classif%>

```

## A.5 escolha\_classif.rhtml

```

<%i = 1-%>
<table cellpadding="5" cellspacing="0">
<%for classif in @all_classif -%>
<%if i==1 -%>
<tr align="center">
  <td><%=classif.classificacao-%></td>
  <td><%=link_to image_tag("#{classif.foto_url}",,
  :border=>0, :width=>"60%"),
  :action=>'cardapio_exibir',
  :id=>classif -%></td>
<%i=2-%>
<%else -%>
  <td><%=link_to image_tag("#{classif.foto_url}",,
  :border=>0, :width=>"60%"),
  :action=>'cardapio_exibir',
  :id=>classif -%></td>
  <td><%=classif.classificacao-%></td>
  <%i=1-%>
</tr>
<%end -%>
<%end -%>
<%if i==2 -%>
</tr>
<%end -%>
</table>

```

# Referências

- 1 ELMASRI, R.; NAVATHE, S. B.. **Fundamentals of Database Systems**. 3<sup>rd</sup> Ed.. Vancouver: Addison-Wesley Publishers, 2000.
- 2 GILFILLAN, I.. **La Biblia de MySQL**. 2<sup>a</sup> Ed.. Madrid: Anaya Multimedia, 2003.
- 3 MIYAGI, P. E.. **Controle Programável - Fundamentos do Controle de Sistemas a Eventos Discretos**. 1<sup>a</sup> Ed.. São Paulo: Editora Edgard Blücher Ltda, 1996.
- 4 RAYMOND, S.. **Ajax on Rails**. 1<sup>st</sup> Ed.. Sebastopol, CA: O'Reilly Media, 2006.
- 5 THOMAS, D.; HANSSON, D. H.. **Agile Web Development with Rails**. 2<sup>nd</sup> Ed.. Dallas: The Pragmatic Bookshelf, 2007.
- 6 THOMAS, D.. **Programming Ruby**. 2<sup>nd</sup> Ed.. Dallas: The Pragmatic Bookshelf, 2005.