

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DESENVOLVIMENTO DE UM SOFTWARE DE
INTERAÇÃO WEB PARA RECEPÇÃO E
GERENCIAMENTO DE ENCOMENDAS EM UMA
PLATAFORMA E-COMMERCE**

TRABALHO DE GRADUAÇÃO

Fernando Giaretta

Santa Maria, RS, Brasil

2013

**DESENVOLVIMENTO DE UM SOFTWARE DE INTERAÇÃO
WEB PARA RECEPÇÃO E GERENCIAMENTO DE
ENCOMENDAS EM UMA PLATAFORMA E-COMMERCE**

Por

Fernando Giaretta

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da
Universidade Federal de Santa Maria (USFM, RS),
como requisito parcial para a obtenção de grau de
Bacharel em Ciência da Computação

Orientador: Prof^a. Dr^a. Márcia Pasin

**Trabalho de Graduação N° 361
Santa Maria, RS, Brasil
2013**

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**

A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**DESENVOLVIMENTO DE UM SOFTWARE DE INTERAÇÃO WEB
PARA RECEPÇÃO E GERENCIAMENTO DE ENCOMENDAS EM
UMA PLATAFORMA E-COMMERCE**

elaborado por
Fernando Giaretta

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:


Prof.^a. Dr.^a. Márcia Pasin
(Presidente/Orientador)


Prof.^a. Dr.^a. Patrícia Pitthan (UFSM)


Prof.^a. Dr.^a. Ana Trindade Winck (UFSM)

Santa Maria, 26 de julho de 2013.

“Happiness is only real when shared.”

— CHRISTOPHER MCCANDLESS

AGRADECIMENTOS

Primeiramente gostaria de agradecer aos meus pais e irmã pela confiança e constante apoio em todos os anos de graduação. A minha grande amiga e namorada Lisi, pelas incontáveis horas de incentivo e atenção.

Aos meus grandes amigos de Tucunduva e Santa Maria pela sempre presente parceria e amizade.

A todos os professores do curso de Ciência da Computação, em especial à professora Márcia por ter me orientado neste trabalho e às professoras Ana Winck e Patrícia Pitthan pela disponibilidade de fazerem parte da banca de avaliação.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

DESENVOLVIMENTO DE UM SOFTWARE DE INTERAÇÃO WEB PARA RECEPÇÃO E GERENCIAMENTO DE ENCOMENDAS EM UMA PLATAFORMA E-COMMERCE

Autor: Fernando Giaretta
Orientador: Prof^a. Dr^a. Márcia Pasin
Local e data da defesa: Santa Maria, 26 de julho de 2013.

Este trabalho tem como objetivo o desenvolvimento de um nova versão do *software* para recebimento de compras do sistema de *e-commerce Delivery Much*. O sistema *Delivery Much* consiste em uma central *online* que agrupa diversas empresas do ramo alimentício para o comércio de seus produtos. O serviço oferecido possibilita ao cliente a visualização do cardápio agregado de diferentes empresas e a realização de compras pelo meio digital. O *software* em questão permite, pelo administrador, a visualização e edição dos dados das empresas presentes no sistema, assim como o gerenciamento e manipulação das compras realizadas através do *website Delivery Much*. Desenvolvido para atuar somente na cidade de Santa Maria/RS, o projeto piloto não mais atende as necessidades de dinamismo necessárias para a expansão de mercado, dessa maneira o novo projeto visa autonomia no gerenciamento de informações pelas empresas clientes e menor intervenção da equipe *Delivery Much* no âmbito comercial.

Palavras-chave: Java, RMI, E-commerce, Sistemas Distribuídos.

ABSTRACT

Undergraduate Paper
Undergraduate program in Computer Science
Universidade Federal de Santa Maria

Developing an E-Commerce application to the management and integration of a delivery service

Author: Fernando Giaretta
Adviser: Prof^a. Dr^a. Márcia Pasin

This work describes the development of a new software version to manage purchases of e-commerce system called Delivery Much. This system consists of a central module that aggregates several online companies in the food sector to trade their products. A user interface allows the visualization of the menu of these companies and makes purchase through the digital way. The software in question also allows viewing and editing information of the existing companies in the system, as well as the management and handling of purchases made through the Delivery Much website. Developed to operate only in the city of Santa Maria / RS, the pilot project no longer meets the needs of dynamism necessary for the expansion of the market. Thus this new project version seeks autonomy in information management by the customers companies and less intervention in trade by Delivery Much team.

Keywords: Java, RMI, E-commerce, Distributed Systems

LISTA DE FIGURAS

Figura 1. Evolução do faturamento <i>e-commerce</i> (em bilhões) no mercado brasileiro, no período de 2008-2012 (GUASTI; ROTHER, 2013)	17
Figura 2. Página inicial <i>iFood</i> (http://www.ifood.com.br).....	18
Figura 3. Página inicial Peixe Urbano Delivery (http://delivery.peixurbano.com.br).....	19
Figura 4. <i>Screenshot</i> da página inicial do <i>website Delivery Much</i> (http://www.deliverymuch.com.br)	21
Figura 5. Visão de <i>hardware</i> e <i>software</i> do sistema <i>Delivery Much</i> , respectivamente.....	28
Figura 6. Interação entre módulos do <i>software</i> Azis	30
Figura 7. Notificação de novo pedido.	34
Figura 8. Pedidos nas listas “aguardando confirmação” e “preparando/entregando” no módulo principal.....	35
Figura 9. Módulo Relatórios com busca por período	37
Figura 10. Módulo Entrega listando bairros de entrega com respectivo valor.....	38
Figura 11. Módulo Produtos exibindo o cardápio da empresa	39
Figura 12. Tela de atualização	40
Figura 13. Falha durante processo de atualização	40
Figura 14. Tela de login antes e durante autenticação, respectivamente.....	41
Figura 15. Tela de login da antiga versão.....	42
Figura 16. Antiga tela principal.....	42
Figura 17. Tela do módulo principal na nova versão do Azis.....	43
Figura 18. Aviso de empresa <i>offline</i> na página <i>Delivery Much</i>	43
Figura 19. Descrição e tempo médio de espera no <i>website Delivery Much</i>	44
Figura 20. <i>Status</i> de acompanhamento de compra pelo usuário	45
Figura 21. Tela de compras da antiga versão	45
Figura 22. Tela de compras da nova versão	46

LISTA DE TABELAS

Tabela 1. Comparação entre *Frameworks* de comunicação (Performance Dirimi, 2010)27

LISTA DE ABREVIATURAS E SIGLAS

HTTP	<i>Hypertext Transfer Protocol</i>
IDE	<i>Integrated Development Environment</i>
PHP	<i>Hypertext Preprocessor</i>
RMI	Remote Method Invocation
TCP	<i>Transmission Control Protocol</i>

SUMÁRIO

1	INTRODUÇÃO.....	13
1.1	Contextualização e motivação	13
1.2	Objetivos e metodologia	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	14
1.3	Estrutura do Texto	15
2	TRABALHOS RELACIONADOS	16
2.1	<i>E-Commerce</i>	16
2.2	<i>iFood</i>	17
2.3	Peixe Urbano Delivery.....	18
2.4	Conclusões parciais.....	19
3	CONCEITUALIZAÇÃO DO SISTEMA E ARQUITETURA.....	21
3.1	<i>Delivery Much Online</i>	21
3.2	Ferramentas utilizadas	24
3.2.1	Java.....	24
3.2.2	RMI.....	25
3.3	Arquitetura do sistema <i>Delivery Much</i>	27
3.4	<i>Software</i> de recebimento de encomendas (<i>Azis</i>)	29
4	DESENVOLVIMENTO.....	31
4.1.1	Módulo de Atualização.....	31
4.2	Módulo de Comunicação	32

4.2.1	Módulo de Login	33
4.2.2	Módulo Principal	34
4.2.3	Módulo de Compras	35
4.2.4	Módulo de Relatórios	36
4.2.5	Módulo de Entrega	37
4.2.6	Módulo de Produtos	38
5	COMPARAÇÃO ENTRE A ANTIGA E A NOVA VERSÃO	40
5.1	Tela de atualização.....	40
5.2	Tela de login	41
5.3	Tela principal	42
5.4	Tela de compras	45
5.5	Avaliação	46
6	CONSIDERAÇÕES FINAIS	48
6.1	Conclusões	48
6.2	Trabalhos futuros	49
7	BIBLIOGRAFIA	51

1 INTRODUÇÃO

1.1 Contextualização e motivação

A informatização de processos juntamente com a Internet está dando um novo rumo ao nosso cotidiano. Soluções computacionais são desenvolvidas com a finalidade de centralizar informações e facilitar tarefas rotineiras. Atualmente, existe uma grande demanda por praticidade e comodidade, ou seja, as pessoas procuram por produtos que propiciem rapidez, confiabilidade e funcionalidade.

A imersão social no mundo tecnológico vem sendo parte marcante na economia. Não se pode dizer que existe um problema, mas sim uma oportunidade de negócio que visa atender aos clientes advindos da explosão digital provocada pela evolução tecnológica das últimas décadas. O acesso à Internet no Brasil chegou a 83,4 milhões de pessoas no segundo trimestre do último ano (2012) de acordo com dados do Ibope Nielsen Online (Acesso à Internet no Brasil chega a 84,3 milhões de pessoas, 2013).

O *e-commerce*, ou comércio eletrônico, é caracterizado por qualquer tipo de transação comercial através de um meio digital, abrangendo uma gama de negócios, passando por comércio de bens e prestação de serviços entre organizações (O que é e-Commerce, 2013). O comércio digital juntamente com a expansão exorbitante da venda de produtos alimentícios através desse meio criou um nicho de mercado que está começando a se popularizar no Brasil (GAZZONI, 2013). A partir disso, surgiu a *Delivery Much Online* (localizada na Incubadora Tecnológica de Santa Maria no Rio Grande do Sul) que consiste em uma central de compras de alimentos através da Internet, atuando como intermediária entre as empresas alimentícias e os consumidores desses.

O foco da *Delivery Much Online* é realizar o encontro de consumidores e produtores de uma forma alternativa, utilizando como ferramenta os serviços da Internet, com simplicidade e confiabilidade, acarretando a necessidade de um sistema robusto e eficaz para a realização da comunicação entre ambas as partes. O processo de transformação de insumos e matérias primas em produto é realizado pelas empresas contratantes, assim, os pedidos realizados pelos usuários chegam até os restaurantes através de um *software*, juntamente com todas as informações necessárias para a elaboração e entrega do mesmo.

O serviço *Delivery Much* em seu projeto piloto foi desenvolvido para atuar somente na cidade de Santa Maria/RS. Recentemente a empresa *Delivery Much Online* decidiu expandir o seu mercado de atuação a outras cidades do estado, porém o então atual sistema não disponha do dinamismo necessário que essa expansão necessitaria, acarretando a necessidade de um sistema mais flexível e dinâmico.

Após análise da arquitetura do sistema *Delivery Much* foi constatado que um novo sistema era necessário, pois a utilização ou modificação da antiga arquitetura envolveria maiores esforços da equipe e não traria os resultados esperados. Assim, o sistema foi abandonado e foi dado início a um novo projeto, que visa escalabilidade e flexibilidade de manutenção. Dessa reestruturação surgiu a necessidade do desenvolvimento de uma nova versão para o *software* de controle e recebimento de pedidos no lado dos clientes da *Delivery Much Online*, o qual deve ser compatível com a arquitetura do novo sistema e deve visar a escalabilidade e flexibilidade de manutenção, assim como o restante do projeto. Este *software* é proposto a ser desenvolvido neste trabalho.

1.2 Objetivos e metodologia

1.2.1 Objetivo Geral

Este trabalho propõe o desenvolvimento de uma nova versão para o *software* de recebimento de compras de produtos alimentícios através da Internet para a empresa *Delivery Much Online*.

1.2.2 Objetivos Específicos

Mais especificamente, o objetivo deste trabalho é:

- Construir o projeto para uma nova versão do *software* de recebimento de pedidos através da Internet do serviço *Delivery Much*;
- Identificar e definir uma tecnologia de comunicação bidirecional entre computadores para utilização no serviço através da Internet;
- Desenvolver a nova versão do *software* proposto.

1.3 Estrutura do Texto

A estrutura deste trabalho está organizada da seguinte maneira: no Capítulo 2 é apresentado o conceito de *e-commerce* voltado para o setor alimentício juntamente com projetos existentes. A apresentação geral do sistema, os conceitos das tecnologias e plataformas utilizadas, a proposta do projeto do *software* e a arquitetura do sistema serão tratadas no Capítulo 3. O Capítulo 4 apresenta o desenvolvimento do sistema de pedidos *online* juntamente com os módulos desenvolvidos. No capítulo 5 são mostradas as interfaces desenvolvidas em comparação à antiga versão do *software*. Por fim, no capítulo 6 são descritas as conclusões do desenvolvimento do *software* proposto, assim como possíveis trabalhos futuros que permitirão a continuidade do projeto.

2 TRABALHOS RELACIONADOS

Este capítulo enfoca trabalhos relacionados ao escopo deste trabalho e está dividido em três partes. Na primeira parte (item 2.1) será introduzido o conceito de comércio eletrônico voltado para o mercado brasileiro de *fast-food*. Os itens 2.2 e 2.3 descrevem sistemas de empresas que também oferecem serviços de compras de produtos alimentícios *online* no Brasil. O terceiro item finaliza este capítulo com conclusões parciais sobre esses serviços e o serviço prestado pela *Delivery Much Online*.

2.1 E-Commerce

O *e-commerce* (comércio eletrônico) é um tipo de transação comercial feita por meio eletrônico, como vender, comprar ou pagar contas pela Internet (O que é E-commerce?, 2013). O comércio eletrônico caracteriza-se por apresentar informações sobre os produtos comercializados de uma maneira digital e por quebrar barreiras geográficas na comercialização desses produtos e serviços. Este tipo de comércio traz conveniência e comodidade aos consumidores, pois além de simplificar a comparação de preços e vantagens entre produtos, os consumidores podem facilmente compartilhar informações entre si, o que influencia a decisão de compra dos demais usuários. (O que é E-Commerce?, 2013).

Segundo dados da 27ª edição da revista *Webshoppers* da e-bit, o setor de *e-commerce* finalizou o ano de 2012 com R\$ 22,5 bilhões de faturamento, significando um acréscimo nominal de 20% em relação ao ano de 2011, mantendo o crescimento dos últimos anos, como mostra a Figura 1. Para o ano de 2013 estima-se que o Brasil atingirá o faturamento de R\$ 28 bilhões com o comércio digital (GUASTI; ROTHER, 2013). Nesta mesma edição da revista *WebShoppers* constatou-se que no ano de 2012, 73% dos consumidores digitais efetuaram o pagamento de suas mercadorias através de cartão de crédito, contra 18% efetuados a partir de boleto bancário. Esse nicho de mercado está em constante expansão e acredita-se que será uma tendência para os próximos anos, a prova disso é o rápido crescimento das plataformas *e-commerce* e a confiança que os usuários estão adquirindo neste novo meio de realizar negócios.



Figura 1. Evolução do faturamento *e-commerce* (em bilhões) no mercado brasileiro, no período de 2008-2012 (GUASTI; ROTHER, 2013)

As facilidades das compras realizadas através da Internet estão mudando o comércio brasileiro. Os usuários digitais procuram cada vez mais produtos e serviços que estejam na *web*, tanto pela facilidade que a compra *online* proporciona, como pela comodidade da possibilidade de verificar e comparar os produtos que se deseja de uma maneira interativa e rápida.

2.2 *iFood*

A empresa *iFood* criada em 2011 chegou a marca de 50 mil pedidos realizados em apenas um mês. O serviço da *iFood* abrange mais de mil restaurantes clientes atuando em cinco capitais e cem cidades brasileiras (GUSMÃO, 2013). Na página inicial do serviço (Figura 2) o usuário tem a opção para filtrar os restaurantes por CEP, bairro ou por um tipo de cozinha específica.

A empresa *iFood* disponibiliza para seus clientes o *iConnect*, dispositivo que recebe as compras realizadas através da página *web* *iFood*. O *iConnect* utiliza sistema GPRS, dispensando linhas telefônicas ou conexões à Internet via cabo, o aparelho se mantém sempre *online* a espera do recebimento das compras realizadas. Esse sistema imprime um comprovante com detalhes do pedido, forma de pagamento e dados complementares necessários para realização da entrega da compra (IFOOD, 2013).

Além da convencional compra através da página *web*, os usuários desse serviço podem realizar compras pelos aplicativos desenvolvidos para as plataformas Android e IOS. Segundo matéria publicada no site Info (<http://exame.abril.com.br/pme/startups/noticias/mobile->

investe-r-5-5-milhoes-na-ifood) 21% das 50 mil compras recebidas pelo serviço são realizadas através de *smartphones*.

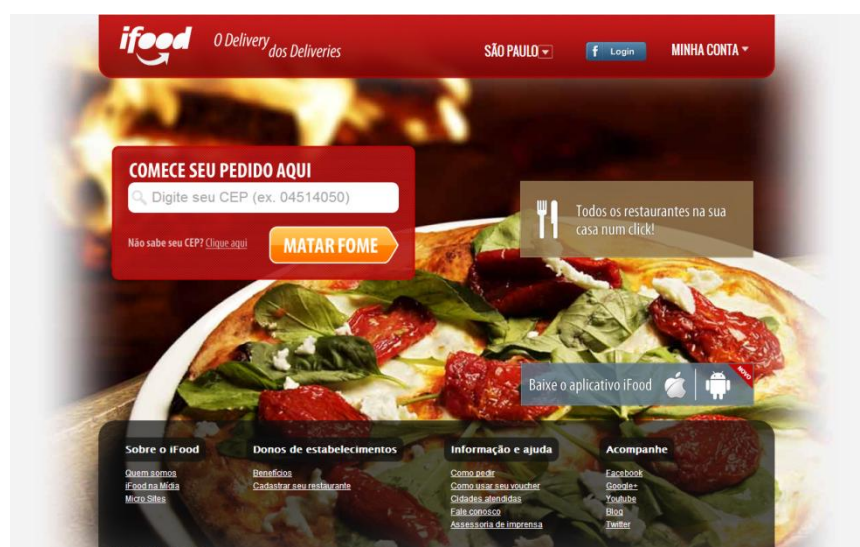


Figura 2. Página inicial *iFood* (<http://www.ifood.com.br>)

2.3 Peixe Urbano Delivery

“Criada no sul de Minas Gerais, com o nome de “O Entregador”, a empresa chamou a atenção do Peixe Urbano pelo talento da equipe e o potencial do negócio durante o *Startup Farm*, um programa de aceleração de *startups* com diversas edições em todo o Brasil. Alguns meses depois, no início de 2012, foi anunciada a aquisição do serviço, hoje chamado de Peixe Urbano Delivery” (PEIXE URBANO, 2013).

No site do sistema Peixe Urbano Delivery (Figura 3), a busca pelos restaurantes parceiros é feita através do CEP ou da pesquisa por bairros, o resultado da consulta pode ser classificado pelo tipo da cozinha, por estabelecimentos com promoções, estabelecimentos abertos ou por estabelecimentos que aceitam pagamento *online*. A compra pode ser realizada tanto pela página do serviço quanto pelos aplicativos disponibilizados para as plataformas IOS e Android.

O sistema Peixe Urbano Delivery contém um painel para as empresas clientes gerenciar e receber os pedidos realizados, esse painel foi desenvolvido para plataformas *web* e pode ser acessado através do endereço <http://restaurante.peixeurbano.com.br>.

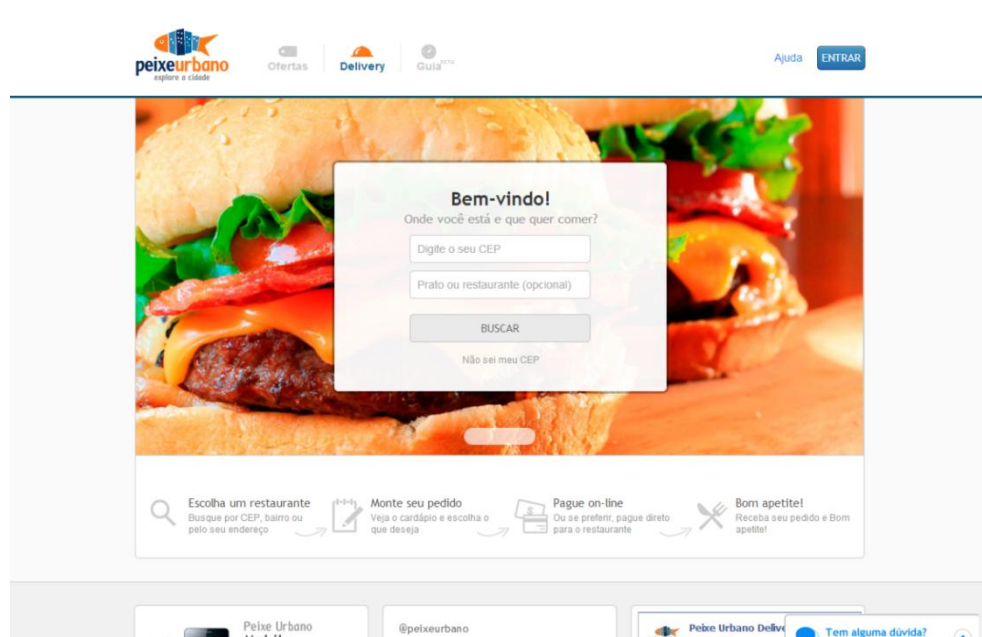


Figura 3. Página inicial Peixe Urbano Delivery (<http://delivery.peixeurbano.com.br>)

2.4 Conclusões parciais

O comércio de alimentos pela *web* tem ganhado força nas capitais. Os sistemas *iFood* e Peixe Urbano Delivery estão concentrados nas principais cidades do país, e nenhum desses serviços atua no interior do Rio Grande do Sul. A *Delivery Much Online* foi a primeira empresa desse setor a implantar esse tipo de serviço no interior do estado, e possui planos para expandir sua atuação a outras cidades do estado que não possuem esse tipo de serviço.

Os sistemas mencionados se diferenciam principalmente pela interface e pelas tecnologias empregadas. Enquanto a *iFood* utiliza o *iConnect* para receber os pedidos nas empresas clientes, o sistema Peixe Urbano Delivery utiliza uma aplicação *web* para realizar essa tarefa. No sistema *Delivery Much*, o *software Azis*, baseado em *software* livre e desenvolvido para plataforma *desktop*, é responsável por esse procedimento.

O novo *software Azis* foi projetado para manter-se atualizado sem a necessidade de intervenção humana, permitindo que mesmo que se façam alterações na arquitetura de todo o sistema, não seja necessário despender recursos humanos para realizar atualização manual nas máquinas onde o *Azis* é instalado. Essa tarefa de atualização pode ser dificultada quando requerida em dispositivos dedicados como é o caso do *iConnect*, onde modificações no sistema poderiam resultar em atualização ou troca do dispositivo.

Por se tratar de um *software desktop*, o Azis contém certas funcionalidades que não são encontradas em *softwares web*. Uma funcionalidade simples e muito útil é a opção para iniciar o Azis juntamente com o sistema operacional, servindo como lembrete para que a empresa cliente realize o *login* toda vez que o sistema operacional é iniciado.

Os alertas visuais também tiram proveito do sistema *desktop*. Quando um novo pedido é recebido, a janela de execução do Azis é levada a frente de todas as outras. Assim, o computador não precisa ser dedicado exclusivamente ao Azis e não precisa de constante verificação por parte do cliente, o próprio Azis se encarrega de alertar o recebimento do novo pedido.

3 CONCEITUALIZAÇÃO DO SISTEMA E ARQUITETURA

3.1 Delivery Much Online

Atuando no mercado santa-mariense desde outubro de 2011, a *Delivery Much Online* nasceu com a missão de mudar o paradigma de encomendas de alimentos na região, trazendo esse processo para o ambiente digital de modo a agilizar as atividades de escolha, pagamento e acompanhamento de pedidos.

O modelo de negócio da empresa visa agregar diversas empresas do setor alimentício que disponibilizam seus produtos por entrega. O objetivo inicial é trabalhar com um *website* centralizado de compras através da Internet com diversas empresas do ramo alimentício, centralizando as opções de alimentação de uma localidade em um só lugar.

O sistema *Delivery Much* pertencente a empresa *Delivery Much Online*, engloba todo o serviço de compra *online*. Um *screenshot* da página inicial é apresentado na Figura 4.

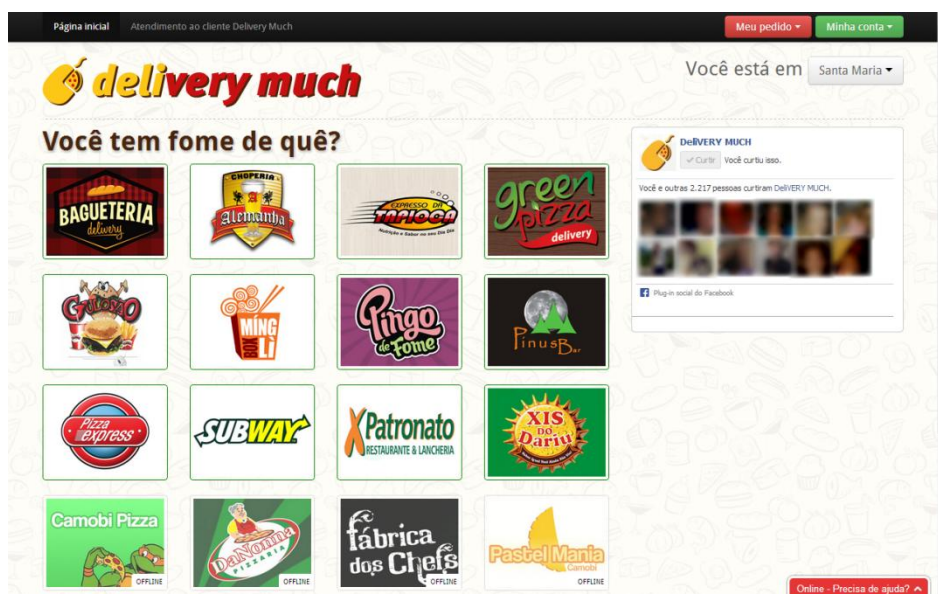


Figura 4. Screenshot da página inicial do website *Delivery Much* (<http://www.deliverymuch.com.br>)

Esse sistema possui três tipos de atores que podem ser divididos em:

- **Cientes:** os clientes da *Delivery Much Online* são as empresas que comercializam alimentos agregados ao serviço de entrega domiciliar, e que reconhecem que sua forma de oferecer este serviço pode ser melhorada a partir da informatização e otimização de seus processos. Esses clientes possuem uma

página personalizada dentro da página principal do *Delivery Much*, essa página contém as informações e o cardápio da empresa, necessários para realização das compras *online*.

- **Usuários:** o público que compra produtos pela Internet, ou seja, pessoas que utilizam o serviço de *e-commerce* prestado pela *Delivery Much Online*. Em suma, esses usuários também são clientes do serviço *Delivery Much*, de forma que, é de responsabilidade da *Delivery Much Online* manter o sistema de *e-commerce* atualizado e ergonomicamente confortável para que estes se sintam à vontade em seu uso. Para esses usuários, o serviço oferecido pela *Delivery Much Online* traz comodidade e praticidade. O objetivo é incentivar que esses usuários consumidores troquem a maneira convencional de pedir seus produtos por uma maneira que propicia rapidez, confiabilidade e funcionalidade.
- **Administradores:** a equipe de administradores do sistema *Delivery Much* é responsável pelo funcionamento e manutenção de todo o sistema. Funções deste grupo incluem: relacionamento com clientes e usuários, cadastro de novos clientes, acompanhamento de vendas e divulgação de promoções.

A *Delivery Much Online* criou em 2011 o Azis, nome dado ao *software* de recebimento de compras *online* para clientes. O projeto vigente visa expandir o mercado da empresa para outras cidades, objetivando essa expansão o sistema *Delivery Much* teve de ser reconstruído para portar maior escalabilidade. Assim, o *software* Azis também necessita de uma nova versão, que foi desenvolvida nesse trabalho, mas é necessário conhecer todo o cenário para compreender como o *software* requisitado interage com o restante do sistema. O sistema *Delivery Much* é dividido em:

- **Página web:** página onde são listadas as empresas clientes, cada cliente possui sua própria página com informações gerais e cardápio digital. A partir da página *web* os usuários podem escolher entre as empresas alimentícias listadas para consultar o cardápio, realizar a compra de produtos, efetuar pagamento *online* e acompanhar o *status* de entrega de suas compras, além dessas, outras funcionalidades como ajuda *online* e interação com redes sociais são disponibilizadas.

- **Painel de administração:** o painel de administração contém informações e funcionalidades necessárias para o gerenciamento e visualização de estatísticas aos administradores.
- **Software servidor:** o *software* servidor é responsável principalmente por detectar um novo pedido realizado e enviar esse ao *software* instalado nos clientes. Também é responsável por toda comunicação entre as funcionalidades do Azis e os dados requisitados pelo mesmo.
- **Software de recebimento de encomendas online (Azis):** este *software* é instalado nos computadores pertencentes aos clientes. A principal finalidade do Azis é realizar a comunicação com o *software* servidor, possibilitando assim o recebimento das compras realizadas através da página *web* pelos usuários. O Azis também permite aos clientes autonomia em relação a mudanças de valores dos produtos comercializados e de entrega, além de possibilitar a visualização de relatórios de vendas.

Quando um pedido é processado, ele recebe um *status*. O sistema *Delivery Much* trabalha com quatro *status* para controle de pedidos, os quais são visualizados pela equipe de administradores (através do painel de controle), pelos usuários (após a finalização do pedido os usuários são direcionados a uma página de acompanhamento) e gerenciados pelo *software* servidor ou pelos clientes através do Azis. Os *status* são classificados como segue:

- **Aguardando confirmação:** no momento em que a compra é finalizada, automaticamente o *status* do pedido é marcado como “aguardando confirmação” e o mesmo é enviado ao Azis. Esse *status* indica que o pedido foi recebido pelo Azis e será analisado por um funcionário da empresa cliente.
- **Preparando/entregando:** esse *status* indica que algum funcionário da empresa cliente visualizou o pedido e este já está em processo de preparo ou entrega.
- **Entregue:** em qualquer etapa do processo o usuário tem a opção de marcar seu pedido como entregue e logo após avaliar o serviço utilizado. Dessa maneira é possível obter um *feedback* sobre o serviço de

determinada empresa cliente, apontando eventuais melhorias ou reforçando boas práticas.

- **Erro ao transmitir para a empresa:** se durante o processo de compra ocorrer eventuais falhas no envio do pedido ao Azis, esse é sinalizado com *status* de erro e uma mensagem é enviada ao usuário e aos administradores do *Delivery Much* informando o ocorrido.

3.2 Ferramentas utilizadas

3.2.1 Java

“O Java é a base para praticamente todos os tipos de aplicações em rede e é o padrão global para o desenvolvimento e distribuição de aplicações móveis, jogos, conteúdo baseado na *Web* e *softwares* corporativos. Com mais de 9 milhões de desenvolvedores em todo o mundo, de forma eficiente, o Java permite que você desenvolva, implante e use aplicações e serviços estimulantes.” (Obtenha Informações sobre a Tecnologia Java, 2013).

Um dos grandes destaques do Java é forma como esta linguagem opera. Java foi desenvolvido para ser portátil, isto é, um mesmo *software* construído com esta linguagem pode executar em plataformas totalmente diferentes. Dessa forma, o esforço dos desenvolvedores é focado no projeto e na lógica do sistema e não em detalhes de baixo nível como a arquitetura da plataforma. A *Sun Microsystems*, criadora do Java, concebeu esta linguagem com a finalidade de convergir diferentes ambientes computacionais, o slogan “*Write once, run everywhere*” da *Sun* já trazia essa ideia de portabilidade.

Para executar um *software* em Java é necessário possuir a Máquina Virtual Java (*Java Virtual Machine, JVM*) instalada no sistema operacional. A Máquina Virtual Java não conhece nada sobre a linguagem de programação Java, apenas conhece um formato binário particular, os arquivos de classe. Um arquivo de classe contém instruções de Máquina Virtual Java, os chamados *bytecodes*, uma tabela de símbolos, assim como outras informações complementares (The Java Virtual Machine Specification Java SE 7 Edition, 2013).

Segundo (DEITEL; DEITEL, 2003), a linguagem de programação Java é totalmente orientada a objetos com forte suporte para técnicas de engenharia de *software*. Java conta com uma comunidade muito grande de programadores e colaboradores. Esses grupos mantêm uma

vasta lista de bibliotecas, o que facilita e diminui o retrabalho no desenvolvimento de *softwares* baseados nessa linguagem, pois muitos problemas comuns são facilmente resolvidos através da utilização dessas bibliotecas.

A linguagem Java foi escolhida para o desenvolvimento deste *software* principalmente por ser portátil. A portabilidade garante que não será necessário desenvolver versões de *software* para cada sistema operacional existente. Como os clientes deste serviço podem utilizar os mais variados tipos de sistemas operacionais o Java abstrai este problema, permitindo um foco maior no objetivo já proposto pela linguagem, na arquitetura e na lógica do *software*.

3.2.2 RMI

Java RMI (*Remote Method Invocation*) é um mecanismo que permite que um objeto de uma máquina virtual Java invoque métodos remotos de outro objeto em outra máquina virtual Java. Quando um objeto é invocado, seus dados são serializados e enviados para a máquina virtual remota, após o método ser executado nessa máquina remota, o resultado é serializado e enviado de volta à máquina virtual que iniciou a comunicação (Package java.rmi, 2013).

O RMI permite realizar de uma maneira fácil a comunicação entre servidor e clientes, permitindo ao desenvolver abstrair os detalhes de comunicação entre os dispositivos e focar no desenvolvimento do *software* como se este estivesse sendo executado localmente.

3.2.2.1 Dirmi

Este trabalho utiliza o *framework* Dirmi para suportar a comunicação entre clientes e servidor. O projeto Dirmi é um substituto para o Java RMI que facilita a comunicação bidirecional de objetos remotos, incluindo recursos como métodos assíncronos, *callbacks* e exceções remotas customizáveis (Dirmi, 2013). Uma das principais funcionalidades que o Dirmi oferece é a de não possuir restrições à tradução de endereços, o que permite que clientes e servidor possam exportar seus objetos remotos entre si sem configurações manuais adicionais.

A versão antiga (de 2010) do *software* Azis se comunicava unidirecionalmente com um *WebService* através de requisições HTTP (*Hypertext Transfer Protocol*). Para receber os pedidos em tempo hábil, o *software* precisava realizar requisições ao *WebService* em um espaço muito curto de tempo, o que despendia muita comunicação. Assim, uma das principais

características do novo *software*, proposto neste trabalho, consiste em realizar comunicação bidirecional, deixando que o servidor envie ao Azis os novos pedidos sem que esse precise requisitar ao servidor esses dados.

A escolha da utilização do Dirimi no desenvolvimento deste projeto foi motivada principalmente pelos seguintes fatores: (i) a facilidade que ele proporciona em comunicações bidirecionais em relação ao RMI e (ii) pelo desempenho eficiente em relação a outros *frameworks* de comunicação. Estes itens serão discutidos no texto que segue.

Baseado no RMI, o Dirimi auxilia a chamada de métodos remotos de clientes no lado servidor, dessa maneira o servidor consegue capturar objetos remotos dos clientes conectados e assim executar métodos remotos nesses clientes. Esta operação facilita a comunicação bidirecional entre os dispositivos, o que é vital para o projeto uma vez que o servidor deverá enviar dados periodicamente para os clientes conectados, assim, utilizando o recurso de *callback* do Dirimi e não exigindo *polling* por parte do *software* cliente como era feito pela primeira versão do *software*, economizando transferência de dados.

A técnica de *polling* consiste na realização de solicitações periódicas sobre o estado atual de uma determinada informação (Citap, 2013). Neste caso significa a insistente verificação pelo cliente de novas compras ao servidor. A técnica de *callback* permite que o cliente informe ao servidor qual método este deverá invocar quando novos pedidos estiverem disponíveis, não exigindo a verificação contínua desses dados.

O RMI também permite que a comunicação inversa (servidor para cliente) seja executada, porém necessita de configurações adicionais em cada nó para que esse serviço funcione de maneira adequada.

3.2.2.2 Comparação de desempenho de *frameworks*

Testes realizados pelos desenvolvedores do Dirimi mostram que este é até 25% mais rápido do que o próprio RMI. Estes testes consistem na medição de tempo desde a invocação de métodos remotos até a resposta dos mesmos.

Os resultados dos testes (resumidos na Tabela 1) tomam como base a comunicação via TCP (*Transmission Control Protocol*), que representa comunicação sem *overhead* adicional, assumindo assim o *score* máximo de 100. Os *frameworks* listados na Tabela 1 estão ordenados por desempenho, onde se pode observar que o Dirimi ocupa a segunda posição,

logo atrás do TCP. Resultados detalhados estão disponíveis na página do projeto (<http://sourceforge.net/apps/mediawiki/dirmi>), bem como o código utilizado para realização dos testes (Performance Dirmi, 2013).

Framework	Requisições por segundo	score
TCP	15065	100
Dirmi	12389	82
RMI	9197	61
SIMON 1.1	2298	15
JRemoting	2008	13

Tabela 1. Comparação entre *Frameworks* de comunicação (Performance Dirmi, 2010)

O desempenho do *framework* Dirmi mostra que este tem uma menor sobrecarga em relação a outros *frameworks* de comunicação, sendo superior ao próprio RMI, no qual o Dirmi é baseado.

3.3 Arquitetura do sistema *Delivery Much*

O diagrama de classes da antiga versão do sistema *Delivery Much* continha em sua totalidade 8 classes. Com a reestruturação do projeto esse número aumentou para 35 classes, tornando possível o escalonamento e o dinamismo do sistema.

A arquitetura do novo sistema *Delivery Much* é dividida em duas partes, *user-side* e *client-side*. A visão geral em nível de *hardware* e *software* dessa organização e divisão é ilustrada na Figura 5, e é assim definida:

- *User-side*: consiste na interface utilizada pelos usuários, é formada pelo *website*, aplicativos móveis para compras de produtos e pela parcela do servidor responsável por gerenciar esses serviços (concluir compras, permitir cadastros de usuários, etc) utiliza PHP (*Hypertext Preprocessor*) como principal linguagem de programação.
- *Client-side*: consiste na interface utilizada pelos clientes, é formada pelo *software* de recebimento de compras online (Azis), e pela parcela do servidor responsável por gerenciar esses serviços (interação com o Azis). Utiliza o Java como principal linguagem de programação.

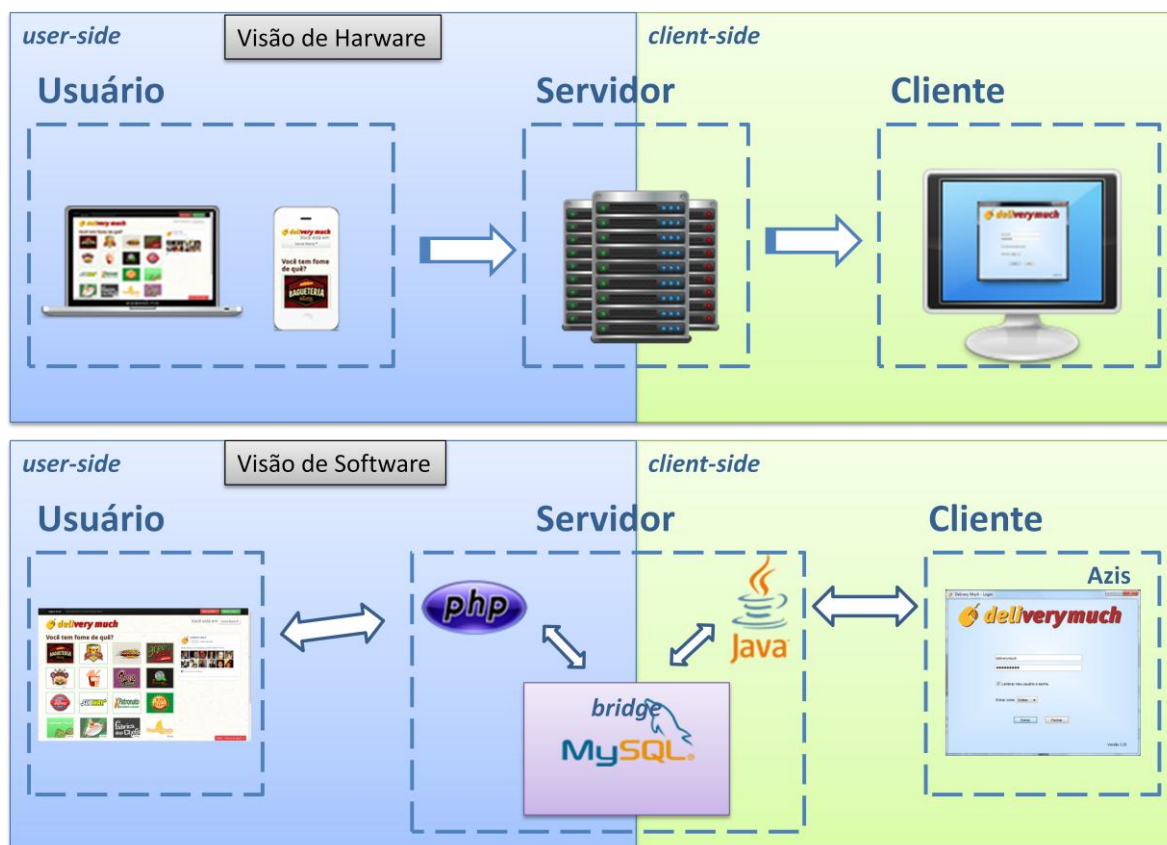


Figura 5. Visão de *hardware* e *software* do sistema *Delivery Much*, respectivamente

A integração entre *user-side* e *client-side* ocorre através do banco de dados MySQL. No momento em que um usuário realiza uma compra através do *website* ou de aplicativos móveis, a interface *user-side* registra os dados necessários na base de dados (implementada pelo MySQL). Então a interface do lado servidor *client-side* verifica a existência de novos pedidos na base de dados e envia os mesmos ao *software* Azis.

No lado *client-side* o servidor é responsável por classificar os dados e serializá-los como objetos para entregar ao Azis. Assim todo o trabalho de manipulação e busca das informações é realizado no servidor, facilitando e centralizando futuras alterações no banco de dados e na comunicação entre *user-side* e *client-side*. Essa característica também impede totalmente o acesso à base de dados pelo Azis, aumentando significativamente o nível de segurança de todo o sistema.

Para aumentar a confiabilidade do serviço, são utilizados servidores da *Amazon* com espelhamento, além de realizar processo de *backup* do banco de dados em intervalos de 30 minutos com destinos remotos para proteger o sistema de eventuais falhas.

3.4 *Software* de recebimento de encomendas (Azis)

O projeto do *software* proposto foi dividido em oito módulos com objetivos específicos:

- (i) **atualização:** responsável por manter o *software* atualizado.
- (ii) **comunicação:** responsável por realizar a comunicação com o servidor, invocando métodos remotos do servidor e permitindo invocação de métodos locais pelo servidor. Também é responsável pela detecção de eventuais falhas de conexão e o restabelecimento desta o quanto antes.
- (iii) **login:** responsável pela autenticação e obtenção de dados das empresas clientes. Deve realizar autenticação criptografada.
- (iv) **principal:** principal tela de trabalho, exibida após a realização do login. Este módulo gerencia os pedidos recebidos e contém links para os outros módulos do *software*.
- (v) **compras:** esse módulo é responsável pela exibição e disposição dos dados detalhados das compras.
- (vi) **relatórios:** módulo responsável pelo resgate e visualização de compras passadas.
- (vii) **entrega:** responsável pela exibição e gerenciamento de locais e valores de entrega.
- (viii) **produtos:** responsável pela exibição e gerenciamento dos produtos comercializados na página *web* da empresa cliente no *Delivery Much*.

Após a divisão desses módulos observou-se que havia duas finalidades gerais: **interação** e **funcionalidade**, e que essas poderiam dividir os módulos em grupos. Dessa maneira, seis desses módulos (login, principal, compras, relatórios, entrega e produtos) integram o grupo de interação, que visa usabilidade do cliente com o *software* e a apresentação dos dados de compras e da própria empresa cliente. Os dois módulos restantes (atualização e comunicação) foram inseridos no grupo de funcionalidade que fornecem suporte para que o *software* realize tarefas funcionais.

Os módulos descritos interagem entre si conforme ilustração da Figura 6. Essas interações serão descritas com maiores detalhes no capítulo 4.

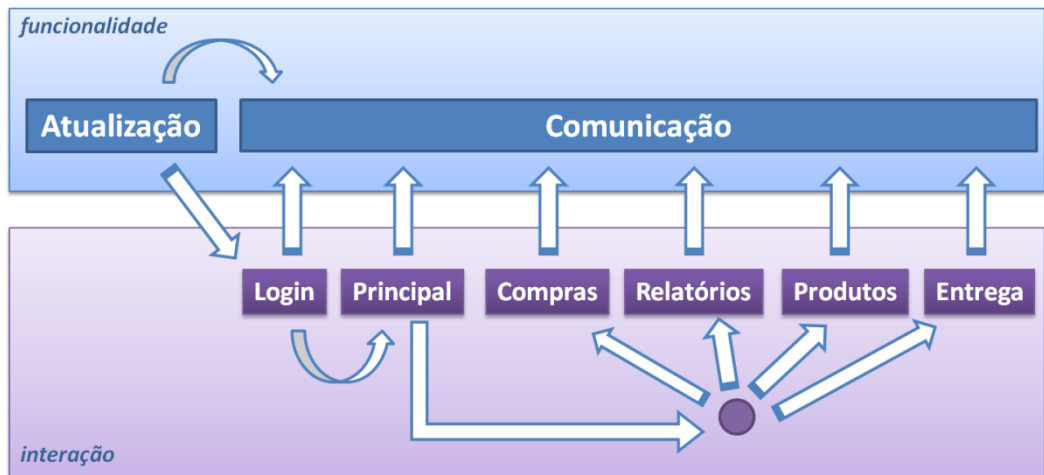


Figura 6. Interação entre módulos do *software* Azis

Por fim, uma interface de *Log* de erros foi construída para ser utilizada em todo o projeto. Dessa maneira, quando um evento anômalo ocorrer e esse gerar alguma exceção no *software*, essa interface guardará os dados obtidos desse evento em um arquivo, o qual será enviado à equipe *Delivery Much* para que este seja avaliado e corrigido em versões futuras.

4 DESENVOLVIMENTO

A linguagem de programação Java na sua versão 7, juntamente com o *framework* de comunicação Dirmi 1.1.2, foram utilizados para o desenvolvimento o *software* proposto neste trabalho. A IDE (*Integrated Development Environment*) Eclipse Indigo e o *plugin Window Builder* 3.7 (utilizado para criação de interfaces gráficas) auxiliaram esse processo de desenvolvimento.

A seguir os módulos apresentados no item 3.4 serão descritos com maiores detalhes. Estes módulos foram implementados na versão atual deste *software*.

4.1.1 Módulo de Atualização

O módulo de atualização do Azis é dividido em dois, a parte principal é separada do restante do projeto, e a parte secundária é integrada ao projeto principal. A divisão desse módulo ocorreu por dois motivos:

1. Um processo não pode modificar o arquivo que lhe deu origem, conseqüentemente não podendo realizar a auto-atualização, sendo necessário que outro processo o faça;
2. Uma pequena modificação, tanto no Azis quanto no módulo atualizador externo acarretaria na necessidade de atualização manual desses arquivos nos computadores clientes, essa atualização poderia gerar grandes transtornos para os clientes e para a equipe *Delivery Much*, acabando por contrariar a ideia de facilidade e mínima manutenibilidade do sistema.

Assim, o módulo de atualização principal é responsável por manter atualizados todos os arquivos do Azis, menos a si próprio, papel desempenhado pelo módulo de atualização interno, que é integrado ao processo principal.

Quando executado, o Azis inicia o módulo de atualização principal e encerra o seu próprio processo. Assim, o módulo principal de atualização realiza comunicação com o servidor que contém um módulo responsável pela verificação da versão do Azis e detecção da necessidade de atualização. Se uma nova versão for detectada, o módulo de atualização cria uma árvore listando os arquivos do diretório de instalação com sua respectiva *hash*. Essa árvore é enviada ao servidor de atualização que a partir dessas informações realiza a

verificação de quais arquivos não estão presentes no cliente e quais arquivos sofreram modificações. Com posse dessas informações o servidor otimiza o processo de atualização enviando apenas os arquivos necessários para a atualização do Azis, evitando transferências de arquivos redundantes, economizando tráfego e conseqüentemente tempo.

O módulo de atualização externo exibe uma tela onde é possível acompanhar o processo de atualização principal. O processo de atualização secundário é realizado em segundo plano durante a execução do *software* e não exibe informações desse processo. Caso o processo secundário falhe, este gerará um erro que é salvo pela interface de *Log* e enviado à equipe *Delivery Much* o quanto antes para providenciar a solução do caso específico.

4.2 Módulo de Comunicação

O módulo de comunicação é responsável por estabelecer e gerenciar conexões com o servidor remoto. Esse módulo possui uma interface bem definida compartilhada com o *software* servidor e é utilizado em todos os módulos do projeto. Sua principal função é enviar os dados requisitados pelos outros módulos a métodos específicos do *software* servidor e disponibilizar objetos para invocação do servidor.

Quando necessária a invocação de métodos no servidor, é possível resgatar a instância de comunicação através de uma classe *singleton* em qualquer parte do *software*, eliminando a necessidade de criação de vários objetos para comunicação ou disseminação de um único objeto por passagem de parâmetro em todos os módulos do projeto.

Como a comunicação entre *software* cliente e *software* servidor é realizada através de conexão à Internet, eventuais falhas de acesso à rede por parte dos clientes podem ocorrer. É essencial que falhas sejam detectadas e devidamente tratadas para resolver ou amenizar problemas resultantes. Nesse contexto foram desenvolvidas rotinas para detecção e tratamento de falhas de comunicação. A principal finalidade dessa função é detectar falhas de comunicação com o servidor e tentar restabelecer essa comunicação. Seu funcionamento consiste em uma *thread* que mantém constante comunicação com o servidor, e se em determinado intervalo de tempo não obtiver comunicação com este, irá invocar um método específico no servidor para teste de conectividade. Se essa *thread* não obtiver um número específico de respostas positivas consecutivas, esse módulo encerra a atual conexão e tenta nova conectividade com o servidor sem exibir alertas ou informações ao usuário do Azis sobre o atual estado. Caso após três tentativas (em média dois minutos) a falta de conexão

persistir, o módulo de comunicação informa a falta de conectividade ao módulo principal que finaliza sua tela de exibição e invoca o módulo de login informando a eventual falta de conectividade.

4.2.1 Módulo de Login

Durante a autenticação do cliente, o par <*login e senha*> são enviados ao servidor. Dados enviados ao servidor devem ser criptografados a fim de impedir que estes sejam interceptados durante a comunicação.

A criptografia dos dados é realizada pelo algoritmo MD5. Este algoritmo foi escolhido por gerar uma *hash* de 128 bits, o que torna sua quebra por força-bruta uma tarefa dispendiosa e computacionalmente cara (VALE, 2013) e por ser um algoritmo unidirecional, garantindo que a *hash* gerada não possa ser transformada no texto em que lhe deu origem.

Entretanto, o algoritmo MD5 apresenta uma pequena vulnerabilidade. Existem *WebServices* disponíveis na Internet onde é possível realizar buscas em grandes bases de dados por *hashes* conhecidas. Dessa maneira, se forem utilizados *logins* e/ou senhas consideradas conhecidas, e durante o processo de autenticação a comunicação for interceptada por terceiros, esses dados capturados provavelmente serão inseridos como entrada para esses *WebServices*, os quais possivelmente informarão uma sequência de caracteres que originará a *hash* gerada. Para contornar este problema é preciso evitar a entrada de *logins* e senhas consideradas comuns.

A maneira mais simples e eficaz de se chegar a esse objetivo é com a adição de prefixos pré-definidos aos campos de *login* e senha antes da realização de criptografia desses dados. Essa ação tem por finalidade alterar sequências de caracteres conhecidas, sendo que mesmo utilizando *login* e/ou senha conhecidos, a *hash* gerada será totalmente diferente da *hash* da cadeia de caracteres sem prefixos pré-definidos. Essa técnica dificulta a busca pelos dados que deram origem a ela, consequentemente dificultando o acesso ao serviço por pessoas não autorizadas. O processo de concatenação de prefixo aos dados é transparente ao cliente, permitindo que este utilize senhas fáceis de ser memorizadas e ao mesmo tempo difíceis de serem descobertas.

4.2.2 Módulo Principal

O módulo principal é assim chamado por ser o responsável por exibir e gerenciar os pedidos que chegam até o cliente e por fornecer acesso aos outros módulos do projeto. A tela do módulo principal é exibida logo após a realização de *login*.

No momento em que um usuário finaliza uma compra, o *software* Azis o receberá através do objeto local disponibilizado para invocação remota pelo módulo de comunicação, este módulo encaminhará a compra ao módulo principal. Por sua vez, o módulo principal adicionará o pedido na lista “aguardando confirmação”, emitirá um aviso sonoro, e se o sistema operacional utilizado for Windows, exibirá uma notificação na barra de tarefas, como ilustrado na Figura 7. Neste momento, a tela do *software* ficará em primeiro plano em relação a outros programas executados no computador, liberando o administrador do *software* da tarefa de verificar a tela principal repentinamente.

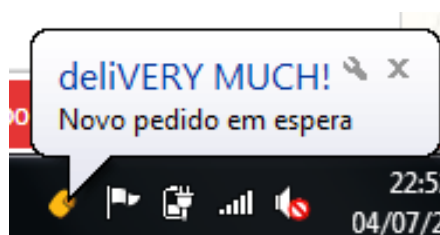


Figura 7. Notificação de novo pedido.

Na tela principal existem duas listas onde os pedidos são exibidos, conforme ilustrados na Figura 8. Essas listas são divididas em **(i) Aguardando confirmação**, e **(ii) Preparando/Entregando**, representando o *status* atual do pedido. Nestas listas consta o nome do usuário, o tempo corrido desde a realização do pedido, o valor total desse pedido e um botão que pode ser utilizado para avançar o *status* deste. Os itens destas listas possuem três cores (preto, amarelo, vermelho) que são modificadas de acordo com três variáveis: (i) *status* atual do pedido, (ii) tempo previsto para entrega e (iii) tempo passado desde a realização da compra. Essas cores facilitam a visualização de compras que necessitam de maior atenção em relação a outras.

Aguardando confirmação			
Fernando Giaretta	segundos atrás	R\$ 8,14	
Preparando/Entregando			
Lucas Zardo	segundos atrás	R\$ 1,00	

Figura 8. Pedidos nas listas “aguardando confirmação” e “preparando/entregando” no módulo principal

O acesso ao módulo de compras, onde os pedidos são detalhados, é realizado através de duplo clique sobre os dados do pedido listado.

4.2.3 Módulo de Compras

A tela de compras é dividida em três partes principais: (i) **pedido detalhado**, (ii) **informações gerais** e (iii) **informações de entrega**. Todos os produtos do pedido são listados como itens na área de **pedido detalhado**, nesses itens são exibidos o nome do produto, os adicionais desse produto e o seu respectivo valor.

Os dados referentes a valores (valor total do pedido, valor de entrega, troco, forma de pagamento), dados do usuário (nome, telefone) e tempo estimado para entrega são exibidos na área de **informações gerais**.

A área de **informações de entrega** possui todos os dados necessários para realizar a entrega do produto (forma de entrega, endereço, referência).

Outras informações e opções de interação estão dispostas nessa tela, pequenas informações são disponibilizadas para dar ênfase à informação útil, como a opção para alteração do *status* do pedido sem que seja necessário o retorno à tela principal, e a informação de primeira compra do usuário no estabelecimento, o qual sugere para o cliente da *Delivery Much Online* que este proporcione atenção ao seu cliente a fim de conquistar e fidelizar o novo usuário.

O contador de tempo exibido neste módulo possui a mesma finalidade do exibido no módulo principal. Nele são mostrados o horário em que o pedido foi realizado e o tempo que

já passou desde a realização desse pedido. O tempo desde a realização do pedido também é exibido em três cores (preto, amarelo e vermelho) que varia de acordo com o *status* do pedido e o tempo transcorrido, facilitando a rápida visualização de quais pedidos precisam de prioridade no momento.

Finalmente, nessa tela também é encontrada a opção para **impressão do pedido**, a qual conterá informações das três partes principais (pedido detalhado, informações gerais e informações de entrega) oferecendo mais uma opção para otimizar a organização interna do estabelecimento.

4.2.4 Módulo de Relatórios

O módulo de relatórios (captura de tela ilustrada na Figura 9) atualmente consiste na exibição dos pedidos já entregues, funcionando como histórico de vendas. Dois campos para inserção de datas possibilitam a escolha para filtro desse histórico em um determinado período. Na antiga versão essa função era acessada através de uma página *web*.

Esse módulo faz uso do módulo de compras, onde é possível que os pedidos listados sejam abertos em novas janelas exibindo os detalhes da compra, da mesma maneira que são exibidas as compras que se encontram no módulo principal. A única diferença entre essa tela de relatório detalhado e a tela do módulo de compras é a indisponibilidade de alteração de *status* do pedido na tela do relatório detalhado.

Também é disponibilizada a opção para impressão da lista geral desse histórico, facilitando a organização interna dos clientes.

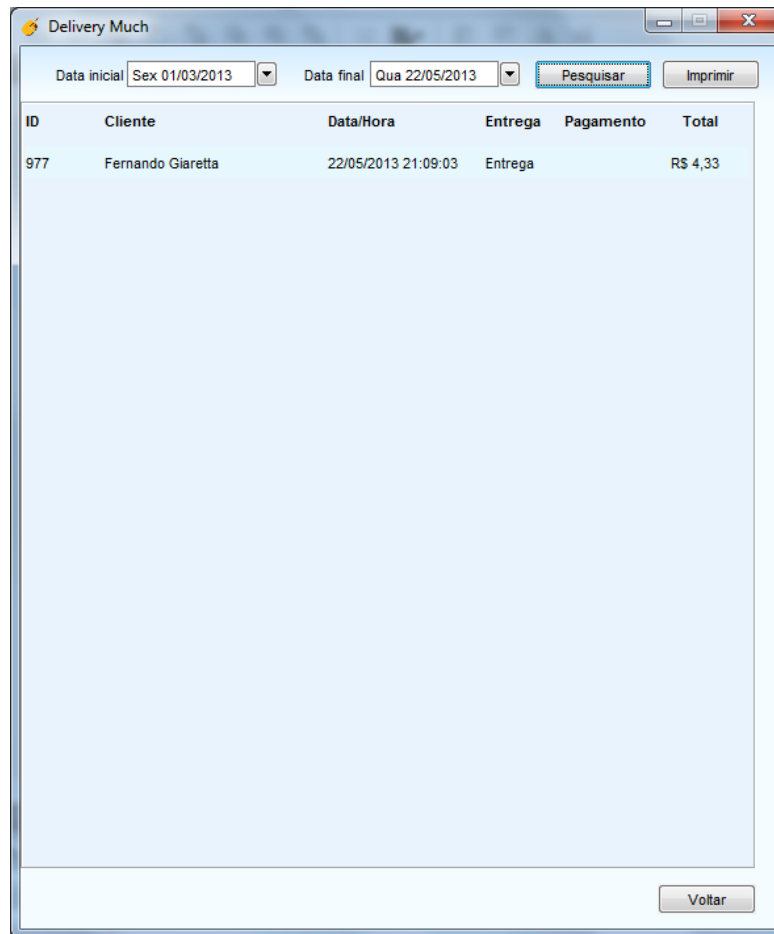


Figura 9. Módulo Relatórios com busca por período

4.2.5 Módulo de Entrega

O módulo de entrega agrega informações referentes a localidades onde a empresa cliente realiza esse serviço. Na tela desse módulo (Figura 10) são listados os bairros da cidade em que a empresa atua. Juntamente com cada bairro existem dois componentes que permitem a edição de informações referentes a disponibilidade de entrega e respectivo valor.

O componente valor consiste em um *spinner* que permite a alteração do valor cobrado para realização da entrega no bairro em questão, já o componente de entrega (exibido em forma de *checkbox*) informa se a empresa realiza ou não a entrega no bairro especificado.

Com a finalidade de tornar o sistema mais dinâmico foi criada a opção para inserção de bairros não listados. Esta opção é encontrada no canto inferior esquerdo, e quando acionada exibe um campo para entrada de texto onde é possível digitar o nome de algum bairro não listado dentre as opções. Quando um novo bairro é adicionado através desta opção, o mesmo é enviado para o painel de controle da equipe de administradores do *Delivery Much* onde

passará por aprovação prévia. Se aprovado, o novo bairro integrará a lista de bairros da cidade em questão e será visível a todas outras empresas que atuam nessa mesma cidade e aos usuários que efetuam compras pelo sistema.

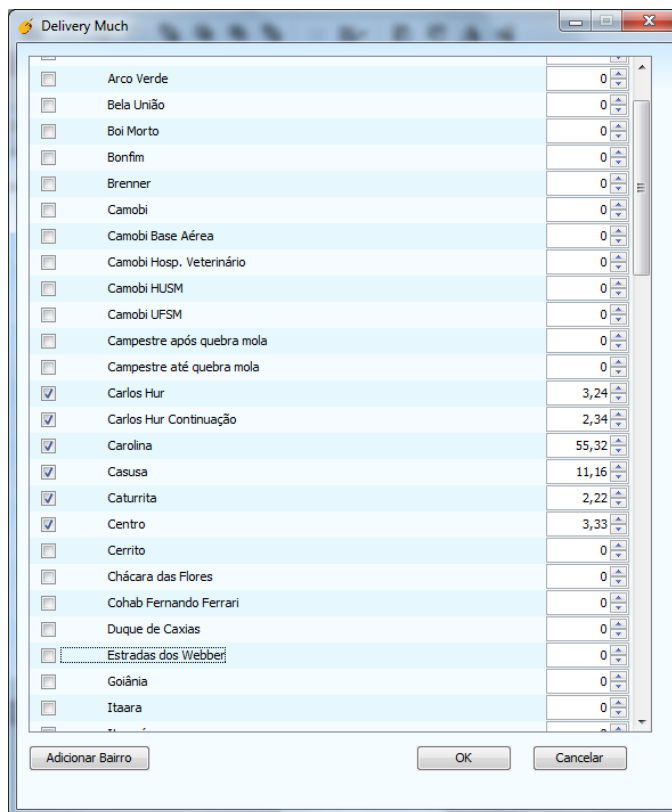


Figura 10. Módulo Entrega listando bairros de entrega com respectivo valor

4.2.6 Módulo de Produtos

Na tela de produtos (Figura 11) é possível visualizar todo o cardápio da empresa cliente. Cada item exibido no cardápio contém o nome do produto, sua descrição e a categoria a qual este está vinculado. Os itens listados nesse módulo são os mesmos exibidos na página *web* da empresa.

Os itens exibidos nessa tela podem ser habilitados ou desabilitados, conseqüentemente sua visualização será removida ou permitida na página da empresa. Essa opção possibilita que os próprios clientes controlem seu estoque. Dessa maneira, quando ocorrer falta de algum item é possível desabilitá-lo do cardápio em poucos passos sem necessidade de intervenção da equipe *Delivery Much*, como era feita na antiga versão.

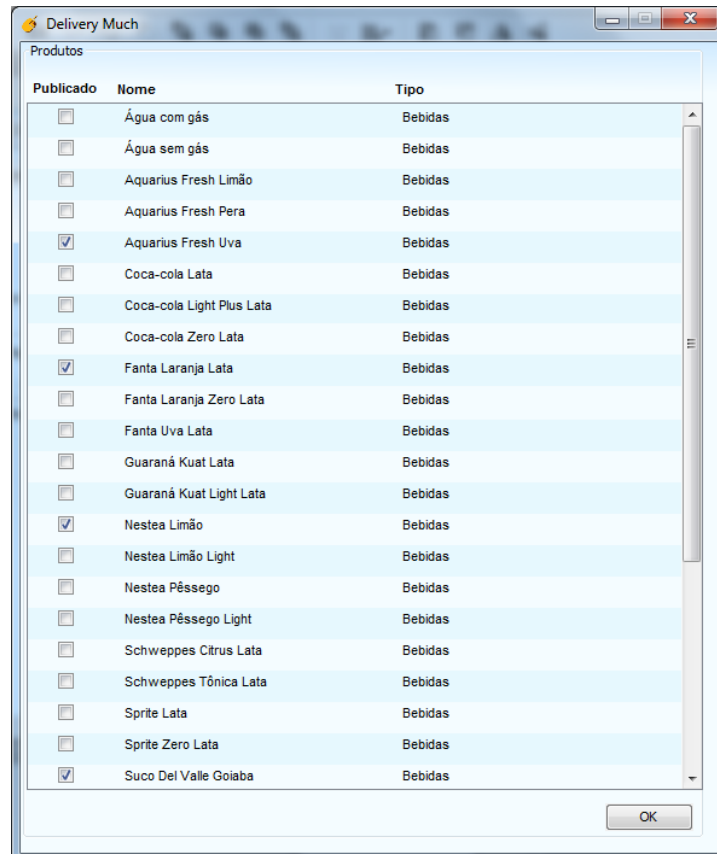


Figura 11. Módulo Produtos exibindo o cardápio da empresa

5 COMPARAÇÃO ENTRE A ANTIGA E A NOVA VERSÃO

Nesta seção são ilustradas e comparadas as principais interfaces gráficas de interação com o cliente da antiga e da nova versão do *software* Azis, assim como são detalhadas as funções que a nova versão desempenha com o restante do sistema.

5.1 Tela de atualização

A primeira tela exibida após a execução do *software* é a tela de atualização, ilustrada na Figura 12. Essa tela informa o andamento do processo de verificação de novas versões junto ao servidor e o processo de atualização em si, caso esta esteja disponível.

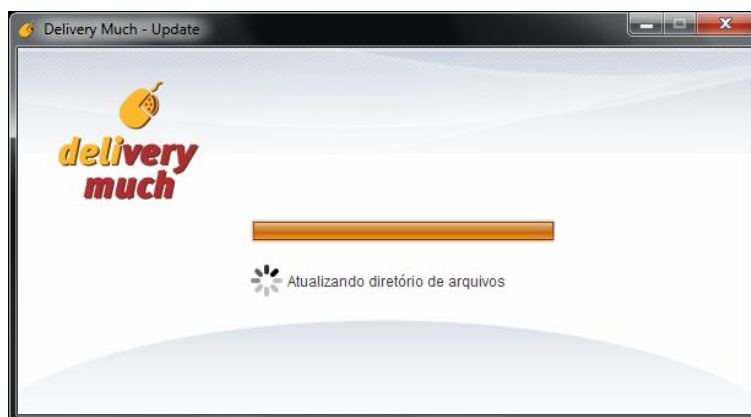


Figura 12. Tela de atualização

Caso o processo de atualização falhe por algum motivo, uma mensagem é exibida (Figura 13) informando o endereço *web* onde é possível realizar o *download* manual da última versão do *software*.



Figura 13. Falha durante processo de atualização

5.2 Tela de login

Após a verificação e possível atualização do Azis, é necessária a autenticação por parte do cliente. Dessa forma, a tela de login é exibida possibilitando a inserção do usuário e senha do cliente, além da escolha do *status online/offline*.

A escolha do *status* permite que após a validação dos dados a empresa apareça disponível ou indisponível para que os usuários realizem suas compras. A opção para realizar login em modo *offline* é disponibilizada para que funcionários da empresa cliente possam verificar o histórico de compras e realizar alterações nos dados sem aparecer disponível para comercialização de produtos. Ainda é disponibilizada a opção para salvamento de usuário e senha, não sendo necessária a inserção dos mesmos a cada novo *login*. Esses dados são salvos em arquivos binários, o que impossibilita a sua visualização por pessoas não autorizadas.

Após a inserção de usuário e senha nos devidos campos, e o acionamento do botão “Entrar”, a tela de *login* exibe o processo de autenticação e troca de informações com o servidor.

A nova interface gráfica (Figura 14), manteve a disposição dos itens de acordo com o padrão da primeira versão do Azis (Figura 15), mas seguiu novos padrões para apresentação de imagens e componentes.

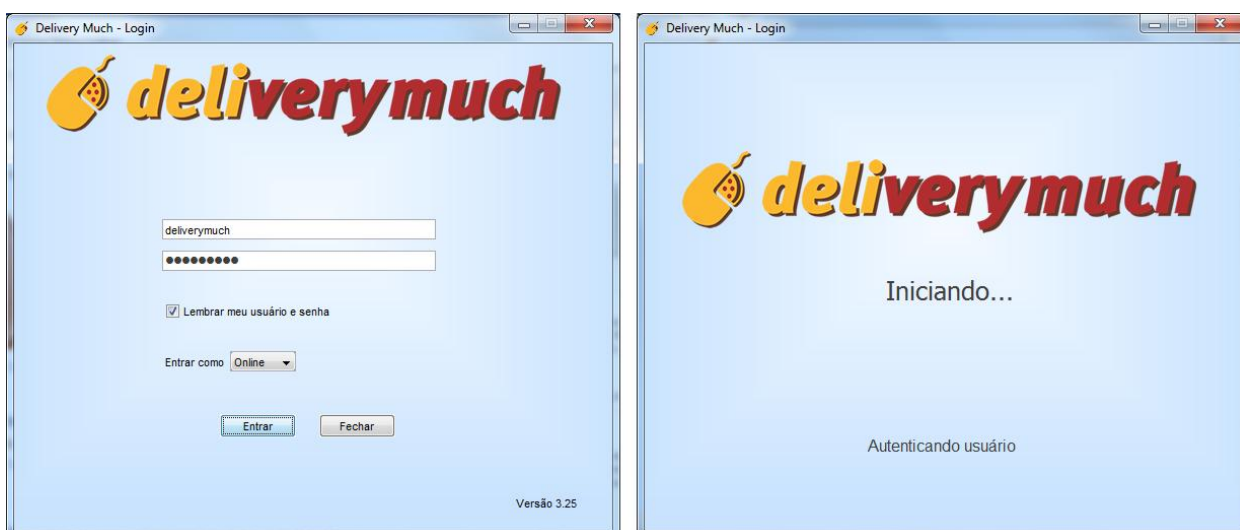


Figura 14. Tela de login antes e durante autenticação, respectivamente



Figura 15. Tela de login da antiga versão

5.3 Tela principal

Na antiga tela principal, ilustrada pela Figura 16, a interação com os dados disponibilizados no *website* eram nulos, a interação era possível somente com os pedidos dos usuários.

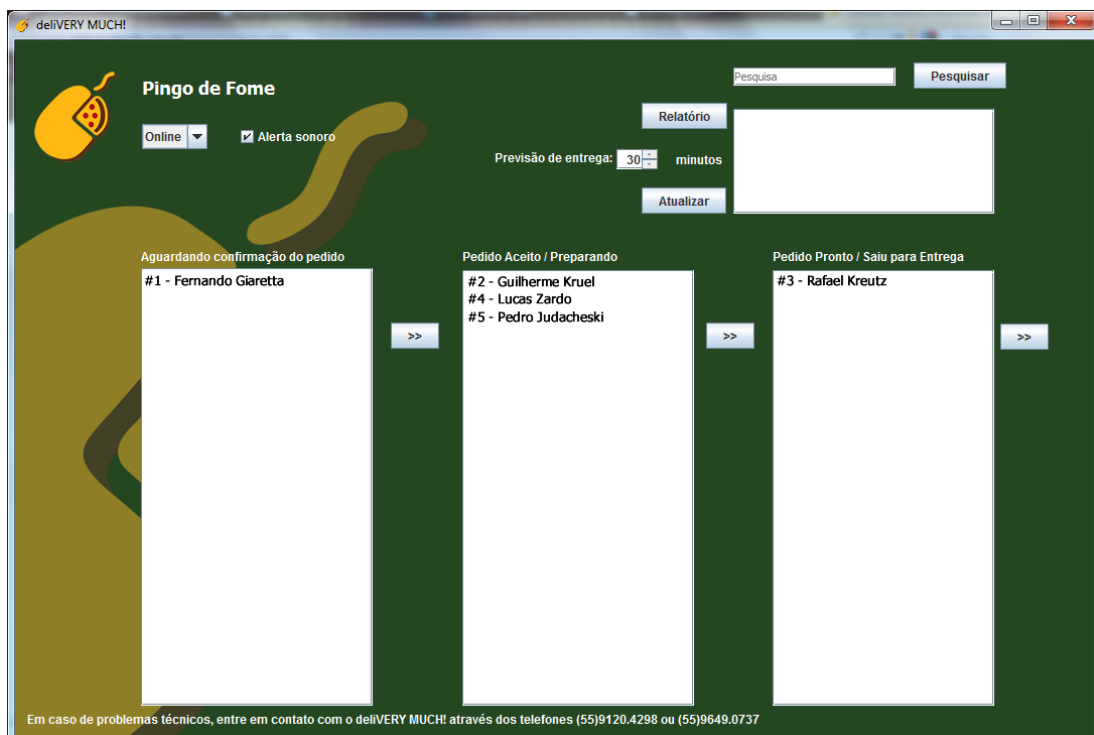


Figura 16. Antiga tela principal

Na tela principal da nova versão, ilustrada pela Figura 17, o cliente possui opção de modificação na descrição da empresa e previsão de entrega, esses dados são exibidos na página da empresa no sistema *Delivery Much*, permitindo autonomia em relação aos dados informados na página.

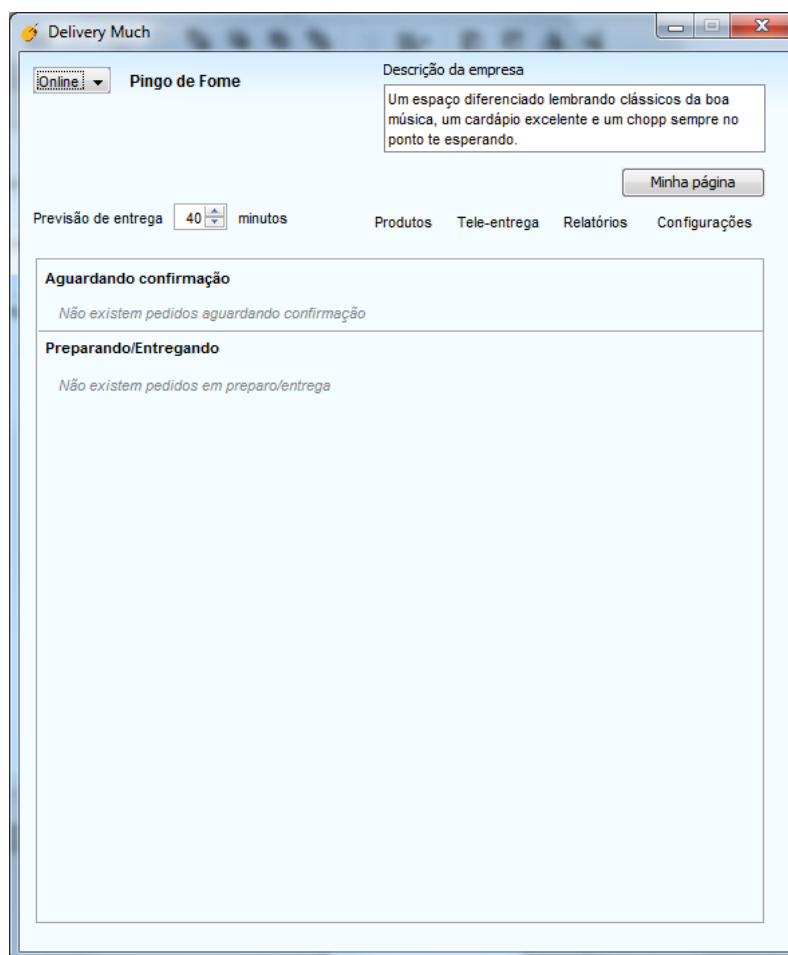


Figura 17. Tela do módulo principal na nova versão do Azis

A opção para modificação de *status* altera a disponibilidade de usuários efetuarem compras na empresa cliente (*status online/offline* na página *web*). Quando *offline* a mensagem ilustrada na Figura 18 é exibida na página da empresa para o usuário.

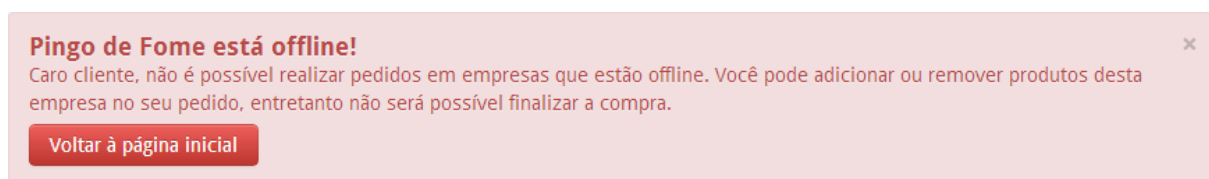


Figura 18. Aviso de empresa *offline* na página *Delivery Much*

O campo “Descrição da empresa” pode ser editado. Este campo permite a alteração de um pequeno texto exibido na página da empresa. Quando alterado, este texto é enviado ao painel de administração requisitando autorização dos administradores para ser exibido no *website*. O campo previsão de entrega altera o valor que informa o tempo médio para entrega do produto em um determinado momento, este valor pode ser modificado a qualquer momento e não necessita de autorização dos administradores. A ilustração dos dados exibidos no *website* desses dois campos pode ser visto na Figura 19.



Figura 19. Descrição e tempo médio de espera no *website Delivery Much*

Quando o *software* estiver com o *status online*, a qualquer momento uma nova compra pode ser recebida. Dessa maneira, ela será inserida na primeira lista, onde após visualização do mesmo pelo administrador do *software*, este deverá marcar a compra com o *status* “em preparo/entrega” através do ícone verde na tela principal ou do botão “Preparando/Entregando” contido na tela de compras. Essa ação informará ao usuário através da página de acompanhamento (Figura 20), que sua compra foi visualizada por um funcionário da empresa.



Figura 20. Status de acompanhamento de compra pelo usuário

A tela principal, também disponibiliza o acesso aos módulos de produtos, entrega, relatórios, compras e também à tela “Configurações” que contém ajustes básicos do *software*.

5.4 Tela de compras

Neste módulo a disposição dos dados teve como base a organização da antiga versão (Figura 21). As mudanças deste módulo consistem nas novas funcionalidades previamente listadas e o cuidado com a apresentação desses dados.

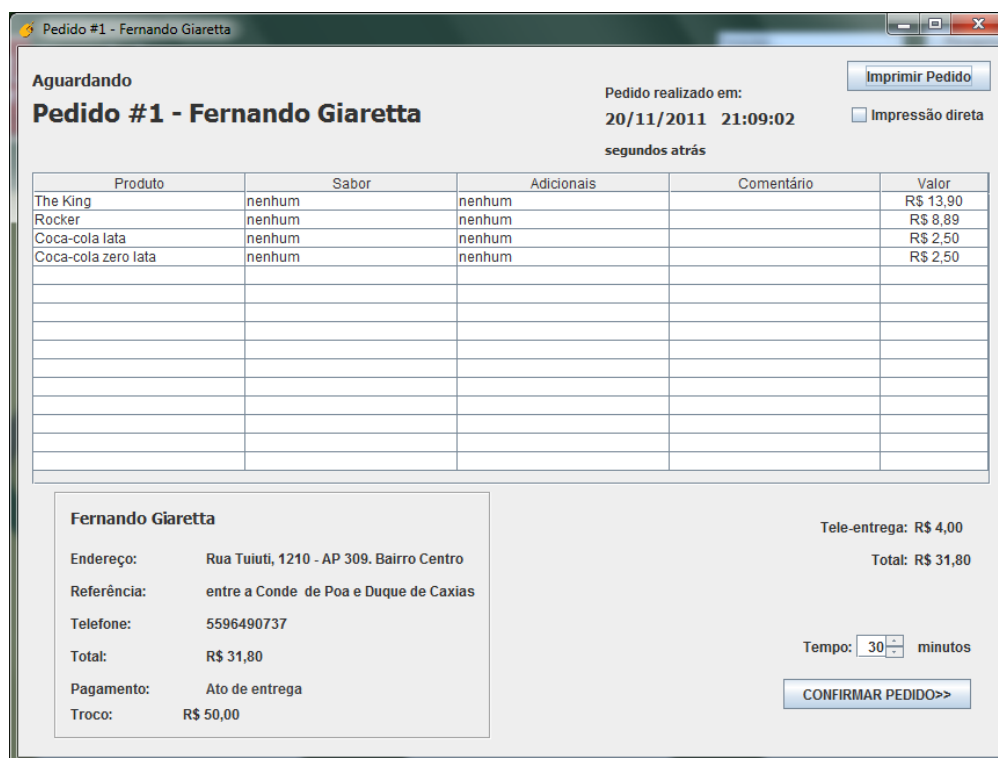


Figura 21. Tela de compras da antiga versão

A nova interface gráfica de compras (Figura 22) apresenta cores e ícones destacando as informações relevantes. Os dados das compras podem ser impressos tanto em impressoras A4 quando em impressora térmicas, utilizando o *checkbox* “Impressora térmica” próximo ao botão de impressão.

Assim como na antiga versão, o botão situado no canto inferior direito altera o *status* do pedido de acordo com o texto exibido neste.

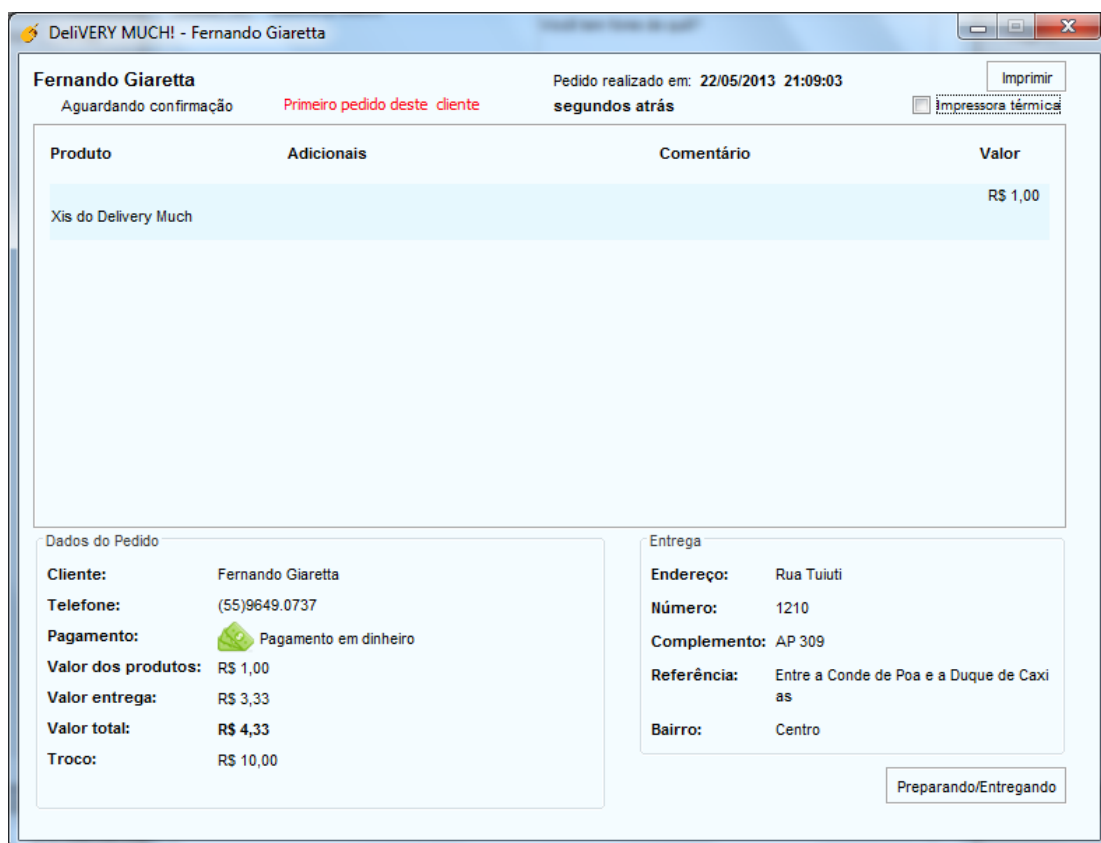


Figura 22. Tela de compras da nova versão

5.5 Avaliação

O *software* desenvolvido neste trabalho está em pleno funcionamento desde o início de junho de 2013. Respondendo bem as necessidades da empresa e sendo bem aceito pelos clientes da *Delivery Much Online*. De acordo com consulta informal aos clientes, o *software* em questão traz facilidade em relação ao antigo, dispondo suas opções, funcionalidades e informações de maneira simples e de fácil entendimento, além de apresentar uma interface gráfica mais discreta e organizada.

Com relação à conectividade e à comunicação com o servidor, a nova versão do *software* cumpre com suas tarefas, permitindo acesso à informação de compras e relatórios em tempo real. A funcionalidade do módulo de conexão responsável pela manutenção de conexão também exerce bem o seu papel, restabelecendo a conexão com o servidor o mais rápido possível quando esta conectividade por algum motivo falha.

É importante destacar o satisfatório funcionamento do módulo de comunicação, pois este permite que o servidor possa invocar métodos remotos no *software* cliente. Essa funcionalidade foi primeiramente requerida para que o servidor pudesse enviar os pedidos sem a necessidade de *polling* pelo Azis. Contudo este mecanismo permite a integração de novas funcionalidades ao projeto, como um serviço de *chat* interno, onde a equipe *Delivery Much* pode enviar mensagens diretamente à empresa cliente sem necessidade de utilizar outras formas de contato.

A *Delivery Much Online* aprovou a nova versão do Azis, destacando o bom funcionamento da conexão bidirecional e o módulo de atualização, que juntos permitem alto grau de controle de todos os nós do sistema, facilitando a expansão da empresa para outras cidades, diminuindo manutenções manuais e centralizando atualizações em apenas um lugar.

6 CONSIDERAÇÕES FINAIS

6.1 Conclusões

A construção do projeto e o desenvolvimento do *software* geraram uma nova versão do Azis compatível com a nova arquitetura e aprovada pela *Delivery Much Online*. Essa nova versão foi construída visando o suporte e a integração de novos módulos sem a necessidade de atualizações manuais, facilitando e otimizando o processo de desenvolvimento e distribuição de novas versões com o auxílio do módulo de atualização, o qual proporcionou o nível esperado para manutenção centralizada.

A divisão por módulos simplificou e facilitou o trabalho de desenvolvimento deste projeto assim como o desenvolvimento futuro, sendo possível modificar, adicionar ou remover módulos inteiros de maneira independente, conferindo com o grau de escalabilidade necessária para o projeto.

Como um dos principais objetivos do trabalho, a pesquisa e escolha de uma ferramenta que implementava comunicação bidirecional foi tomada como base do projeto. Em um primeiro momento o *framework* RMI foi designado para realizar essa tarefa. Após a construção do módulo de comunicação e devidos testes, este *framework* apresentou certa resistência em comunicações entre dois nós quando um destes estava por trás de uma rede que utilizava tradução de endereços. Dessa maneira, alternativas foram buscadas a fim de contornar o problema, assim o *framework* Dirmi foi apresentado como melhor substituto ao RMI. Por ser baseado no RMI, o Dirmi não necessitou de grandes alterações estruturais no projeto, solucionando o problema em questão.

A confiabilidade na transmissão de dados foi alcançada com a utilização do uso de dados concatenados a prefixos estáticos com o algoritmo de criptografia MD5. Já a disponibilidade do serviço é garantida pelos serviços de *backup* remoto e espelhamento de servidores proporcionado pela *Amazon*.

O *software* foi testado em ambientes Windows e Linux, funcionando em ambos os sistemas, diferindo apenas sobre as funcionalidades de notificação na barra de tarefas e atualização. Para exibir a notificação na barra de tarefas foi utilizada uma biblioteca do Windows, assim não sendo executada em outros ambientes. Para a atualização do *software*, pequenas dificuldades de permissão com o sistema operacional Windows foram encontradas,

atualmente esse processo só pode ser concluído com sucesso se o nível de controle de conta de usuário estiver configurado com nível baixo. Alternativas para requerimento de elevação de usuário estão sendo estudadas para que soluções possam ser agregadas a novas versões do *software* sem que seja necessária essa configuração adicional. Testes realizados no Ubuntu não apresentaram necessidade de configurações adicionais para que o processo de atualização fosse concluído com sucesso.

Atualmente o *software* encontra-se na versão 3.31, a numeração de versionamento passou de 1 para 3, não existindo a versão 2. Durante o seu primeiro mês de uso o Azis atendeu todas as funcionalidades necessárias para utilização do serviço *Delivery Much*. A partir deste ponto é possível a adição de novos módulos utilizando a facilidade da atualização automática e da divisão de módulos independentes.

6.2 Trabalhos futuros

No decorrer do planejamento e desenvolvimento do Azis novas ideias de funcionalidades surgiram. Dentre as diversas ideias podem-se destacar algumas que posteriormente serão avaliadas pela equipe e possivelmente integradas a futuras versões do *software*.

Um novo módulo que provavelmente irá integrar o Azis será o financeiro. O módulo financeiro exibirá informações do plano vigente com o serviço *Delivery Much*, valor total de vendas, lucratividade e valores a receber de pagamentos *online*. Atualmente esses dados são enviados mensalmente junto à fatura do serviço. A adição dessas informações ao *software* se faz necessária para acompanhamento parcial de vendas e valores a receber.

Como mencionado anteriormente, uma das novas funcionalidades previstas para o Azis é a adição de um *chat* interno para comunicação entre equipe *Delivery Much* e clientes. Esse *chat* permitirá que ambos iniciem uma conversação a fim de sanar dúvidas ou eventuais problemas, eliminando a necessidade de utilizar outras formas de comunicação (telefone, e-mail, etc).

A maior mudança ocorrerá no módulo de relatórios, o qual atualmente exhibe somente um histórico de compras. Novas versões do Azis irão agregar a este módulo gráficos que facilitarão a análise de vendas e lucratividade das empresas clientes. Gráficos sugeridos até o momento incluem números de pedidos em intervalos de tempo, itens mais vendidos,

quantidade de novos usuários conquistados e quantidade de pedidos realizados por um mesmo usuário. Informações para simulação de promoções com base nos valores dos produtos, lucratividade e itens mais vendidos devem ser estudadas em breve para apresentar esses dados aos clientes. Essas informações são fundamentais para o acompanhamento e fácil consulta dos clientes da *Delivery Much*, os quais poderão utilizar esses dados para tomar decisões sobre o seu negócio.

No módulo relatórios também serão adicionados gráficos com *feedback* dos usuários sobre os serviços prestados, bem como críticas escritas pelos próprios consumidores. Esses dados permitirão que os clientes da *Delivery Much Online* melhorem seus serviços em pontos fracos específicos e reforcem seus pontos fortes, conquistando e fidelizando clientes.

Tomando como premissa a atual popularização de dispositivos móveis, será estudada a viabilidade do desenvolvimento de uma versão *mobile* do Azis. Este possível projeto funcionará em paralelo ao *software desktop*, o qual poderá ser instalado em *smartphones* ou *tablets* facilitando a conquista de novos clientes e permitindo maior expansão de abrangência do serviço.

7 BIBLIOGRAFIA

ACESSO à Internet no Brasil chega a 84,3 milhões de pessoas. **Ibope** Disponível em: <<http://www.ibope.com.br/pt-br/relacionamento/imprensa/releases/Paginas/Acesso-%C3%A0-internet-no-Brasil-chega-a-83,4-milh%C3%B5es-de-pessoas.aspx>>. Acessado em: 13/Março/2013.

CITAP. **Citap** Disponível em: <<http://www.citap.com/documents/tcp-ip/tcpip019.htm>>. Acessado em: 14/Julho/2013.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems - Concept and Design**. 2ª Edição. [S.l.]: Addison-Wesley, 1994. 644 p.

DEITEL, H. M.; DEITEL, P. J. **Java: Como Programar**. 4ª Edição. [S.l.]: Bookman, 2003. 1386 p.

DIRMI. **Sourceforge** Disponível em: <<http://sourceforge.net/apps/mediawiki/dirmi/index.php?title=Dirmi&source=navbar>>. Acessado em: 19/Abril/2013.

GAZZONI, M. Varejo de alimentos busca lugar na Internet. **Estadão** Disponível em: <<http://economia.estadao.com.br/noticias/negocios,varejo-de-alimentos-busca-lugar-na-internet,91184,0.htm>>. Acessado em: 13/Março/2013.

GUASTI, P.; ROTHER, C. WebShoppers 27ª Edição. **WebShoppers**. p. 41, 2013.

GUSMÃO, G. Serviço iFood recebe mais de 50 mil pedidos por mês. **Info** Disponível em: <<http://info.abril.com.br/noticias/blogs/download-da-hora/iphone/servico-ifood-recebe-mais-de-50-mil-pedidos-por-mes/>>. Acessado em: 20/Maio/2013.

IFOOD. **iFood** Disponível em: <<http://www.ifood.com.br/delivery/iconnect/>>. Acessado em: 20/Maio/2013.

O que é e-Commerce. **Gestor de conteúdos** Disponível em: <<http://www.gestordeconteudos.com/tabid/3850/Default.aspx>>. Acessado em: 14/Março/2013.

O que é E-Commerce? **ArtMedia** Disponível em:
<<http://artmedia.net/br/produtosservicos/44-web-designing/191-o-que-e-e-commerce.html>>.
Acessado em: 19/Abril/2013.

O que é E-commerce? **Tudo sobre Marketing** Disponível em:
<<http://tudosobremarketing.wordpress.com/2009/05/14/o-que-e-e-commerce/>>. Acessado em:
19/Abril/2013.

OBTENHA Informações sobre a Tecnologia Java. **Oracle** Disponível em:
<http://www.java.com/pt_BR/about/>. Acessado em: 24/Abril/2013.

PACKAGE java.rmi. **Oracle** Disponível em:
<<http://docs.oracle.com/javase/6/docs/api/java/rmi/package-summary.html>>. Acessado em:
24/Abril/2013.

PEIXE URBANO. Quem Somos. **Peixe Urbano** Disponível em:
<<http://delivery.peixeurbano.com.br/quem-somos>>. Acessado em: 05/Junho/2013.

PERFORMANCE Dirimi. **SourceForge** Disponível em:
<<http://sourceforge.net/apps/mediawiki/dirimi/index.php?title=Performance>>. Acessado em:
11/Maio/2013.

THE Java Virtual Machine Specification Java SE 7 Edition. **Oracle** Disponível em:
<<http://docs.oracle.com/javase/specs/jvms/se7/jvms7.pdf>>. Acessado em: 24/Abril/2013.

VALE, C. R. Criptografia MD5. **DevMedia** Disponível em:
<<http://www.devmedia.com.br/criptografia-md5/2944>>. Acessado em: 29/Maio/2013.