

DESENVOLVIMENTO DE SOFTWARE EMBARCADO PARA CONTROLE DE PEDIDOS DE PIZZARIA COM J2ME

Herbert Oliveira Rocha

Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia de Roraima (IFRR).
herberthb12@gmail.com

Jefferson Thomas de Oliveira

Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia de Roraima (IFRR).
profetahb@gmail.com

Marcelo de Farias Matos

Tecnólogo em Análise e Desenvolvimento de Sistemas pelo Instituto Federal de Educação, Ciência e Tecnologia de Roraima (IFRR).
marcelofmatos@gmail.com

Pierre da Costa Viana Júnior

Mestre em Engenharia Elétrica pela Universidade Federal do Pará – UFPA, Bacharel em Ciência da Computação pela Universidade da Amazônia (UNAM) e Professor do Instituto Federal de Educação, Ciência e Tecnologia de Roraima (IFRR).
pierreviana@gmail.com

RESUMO

Este trabalho apresenta a proposta do uso de uma alternativa tecnológica voltado para o setor do comércio alimentício, especificamente na área de atendimento ao cliente, com dispositivos moveis por meio da linguagem Java para dispositivos móveis – Java 2 Micro Edition (J2ME). Descrevemos as etapas de elaboração do projeto para implantação do protótipo. O protótipo é composto por dois módulos nos dispositivos móveis – Atendimento dos Pedidos e Controle dos Pedidos – e um módulo no servidor que terá como principal função o gerenciamento do estabelecimento, podendo também ser acessado via navegador de Internet.

PALAVRAS - CHAVE

Desenvolvimento de Sistemas. Dispositivo Movel. Linguagem Java, Wi-fi.

ABSTRACT

This paper presents the proposal of using an alternative technology turned to the food sector trade, specifically in the area of customer service, with mobile devices through the Java language for mobile devices - Java 2 Micro Edition (J2ME). We describe the stages of drawing up the project for the prototype implementation. The prototype is composed of two modules in mobile devices - Customer Orders and the Control of Orders - and a module on the server will have the primary role of managing the establishment, can also be accessed by the internet browser.

KEYWORDS

System Development, Mobile Devices, Java Language, Wi-fi.

RESUMEN

En este trabajo se presenta la propuesta de utilizar una alternativa tecnológica volcada al comercio de alimentos, específicamente en el área de atención al cliente, con los dispositivos móviles a través del lenguaje Java para dispositivos móviles - Java 2 Micro Edition (J2ME). Se describen las etapas de elaboración del proyecto para la implementación del prototipo. El prototipo está compuesto de dos módulos en los dispositivos móviles – atención de las solicitudes - y un módulo en el servidor tendrá la función primordial de la gestión del establecimiento, también se puede acceder a través del navegador de Internet.

PALABRAS - CLAVE

Sistema de Desarrollo. Mobile. Java, Wi-Fi.

INTRODUÇÃO

O principal objetivo da tecnologia sem fio é proporcionar mobilidade, ou seja, permitir que os usuários obtenham qualquer tipo de informação, em movimento (LEAL, 2004).

No cenário atual onde os colaboradores das empresas precisam enfrentar viagens, engarrafamentos e deslocamentos constantes, a possibilidade de acessar relatórios de vendas, agenda de compromissos, entre outras informações relevantes ao desempenho de suas funções, a qualquer momento, o acesso a essa tecnologia torna-se vital para o crescimento das organizações.

Leal (2004) destaca vantagens importantes, tais como: não ser preciso necessariamente criar aplicações novas, podendo-se simplesmente adicionar módu-

los de mobilidade às soluções convencionais existentes.

O desenvolvedor pode projetar soluções móveis completamente integradas aos recursos já em operação na organização do cliente. Esta prática permite agregar valor às ferramentas pré-existentes evitando o descarte de tecnologia e os custos desta operação, ganhando flexibilidade ao negociar novos projetos contornando a resistência natural dos clientes a mudanças drásticas.

Até o surgimento da plataforma J2ME as aplicações tinham que ser escritas na linguagem nativa de cada dispositivo usando bibliotecas proprietárias, o que tornava estas aplicações incompatíveis com dispositivos diferentes.

Segundo Almeida (2004), a tecnologia J2ME trouxe flexibilidade e liberdade aos desenvolvedores quanto aos modelos de aparelhos e operadoras de telefonia móvel, e a tendência é que em breve torne-se padrão na indústria de dispositivos móveis.

Nas seções seguintes serão abordados a análise dos dados da pesquisa, o histórico e arquitetura J2ME, bem como o processo de desenvolvimento de aplicações usando essa tecnologia.

ANÁLISE DOS DADOS DA PESQUISA

Utilizou-se para o levantamento e análise dos dados o método qualitativo por meio de entrevistas com proprietários de estabelecimentos, tendo o cuidado ao considerar os depoimentos e informações dos entrevistados para validação dos resultados. A pesquisa foi realizada em seis pizzarias da cidade de Boa Vista-RR, que possuem um grande número de clientes assíduos, na tentativa de conhecer as percepções, a satisfação, as expectativas e as opiniões dos usuários-proprietários, através de um roteiro estruturado de perguntas.

Segundo a pesquisa sobre o uso de sistema informatizado das pizzarias, apenas 40% utilizam-no e 60% não usam nenhum tipo de gerenciamento informatizado nos serviços (Figura 1). Isto pode representar alguns inconvenientes ao atendimento aos clientes, como atrasos e troca de pedidos.

A análise de pedidos semanais feitos pela empresa, em que 16,67% atendem até 100 pedidos; 33,33% atendem aproximadamente 200 pedidos e 50% atendem no máximo a 300 solicitações, representa a demanda de necessidade na agilidade dos atendimentos, conforme Figura 2.

Outro dado, a ser considerado, foi que nenhum proprietário-usuário conhecia ou utilizava dispositivo móvel.

A maioria das pizzarias pesquisadas usa sistemas manuais, com a utilização de bloco de anotações para efetuar o pedido do cliente. Apenas um pequeno

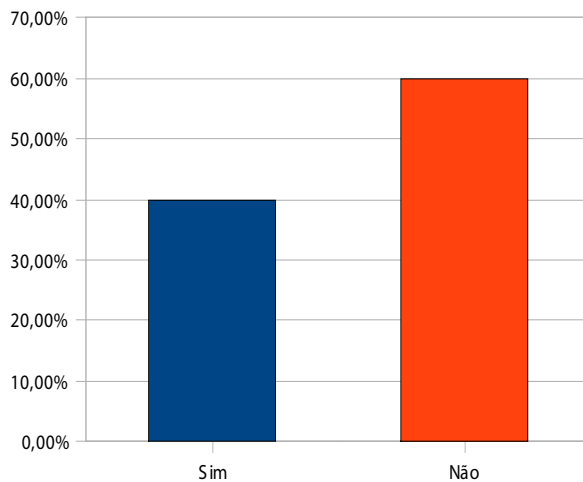


Figura 1
Uso de sistema informatizado

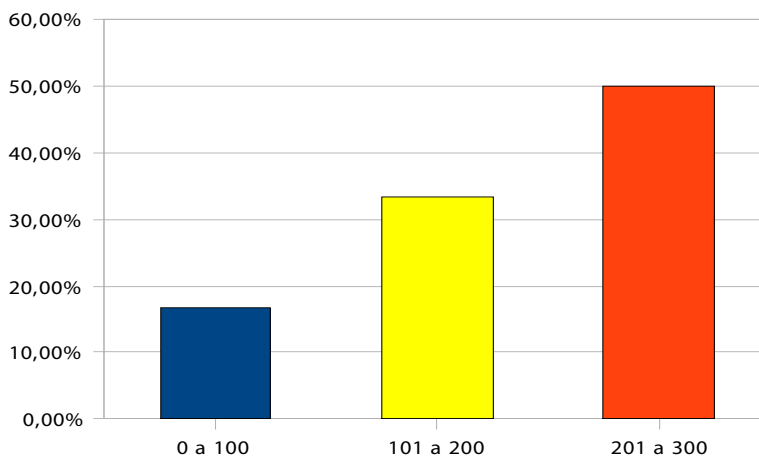


Figura 2
Quantidade de pedidos realizados semanalmente

percentual faz uso de um sistema para controle dos pedidos, mas não possuem o recurso de acesso de dispositivos móveis.

O sistema a ser adotado não necessita de um treinamento mais complexo para uso pelos colaboradores, bastando o conhecimento de informática básica, o que pode diminuir custos com formação mais específica para a operacionalização.

Para Amorim (2005), o “advento da telefonia móvel aliado à expansão da Internet criou o ambiente necessário para atender a demanda crescente por informação e comunicação em tempo real e sempre disponível.”

Nesse contexto os empresários precisam compreender que o investimento em informática agrega benefícios aos produtos e serviços, proporcionando um maior retorno econômico e, principalmente, oferecendo um melhor atendimento aos seus clientes.

De acordo com Pressman (2002), o software tornou-se a força motora de uma empresa, sendo responsável pelo gerenciamento do fluxo de informações e, por isso, é um diferencial entre as organizações.

Diante do exposto, apresenta-se como alternativa um protótipo de gerenciamento de informações e serviços, dividido em dois módulos.

ESPECIFICAÇÕES PARA O MÓDULO I

O primeiro módulo consiste em um sistema web desenvolvido na plataforma J2EE (*Java 2 Enterprise Edition*), onde são aplicadas algumas de suas funções, tendo em relevância que para a utilização deste primeiro módulo o usuário só necessitará de um navegador web (browser).

O sistema funciona sem a necessidade de fazer instalação de softwares ou configurações complexas, atendendo às carências relacionadas às atividades do processo de funcionamento da pizzaria como o acompanhamento administrativo do local, possuindo uma base de dados, desenvolvido no SGDB *PostgreSQL* em que estes dados serão armazenados e estarão acessíveis tanto na *Intranet* como na Internet.

Este módulo apresenta várias funcionalidades tais como: autenticação de usuário, reserva de mesas, registro de atendimentos interno e externo, gerência de pedidos, relatório de vendas. A Figura 3 apresenta uma tela do Módulo I com suas principais funções.



Seja Bem Vido (a): Herbert



Desenvolvido by Adober - 2008 - GPL

Figura 3
Tela principal do Módulo I

Todas essas funcionalidades são acessíveis somente por usuários cadastrados no sistema e devidamente autenticados através da tela apresentada no Figura 4.

A conexão entre os computadores se realiza através de rede *wireless*. Alguns relatórios ficam disponíveis via Internet, com acesso restrito à equipe administrativa. A Figura 5 apresenta a arquitetura do Módulo I, que também pode ser chamado de Módulo Servidor.

ESPECIFICAÇÕES DO MÓDULO II

O segundo modulo é constituído por *software* embarcado, desenvolvido na especificação J2ME (*Java 2 Micro Edition*) que pode ser utilizado em alguns celulares, *smart phones* e *palms*, trazendo mobilidade ao sistema. A Figura 6 apresenta o sistema funcionando em um simulador.



Figura 4
Tela de autenticação do Módulo I

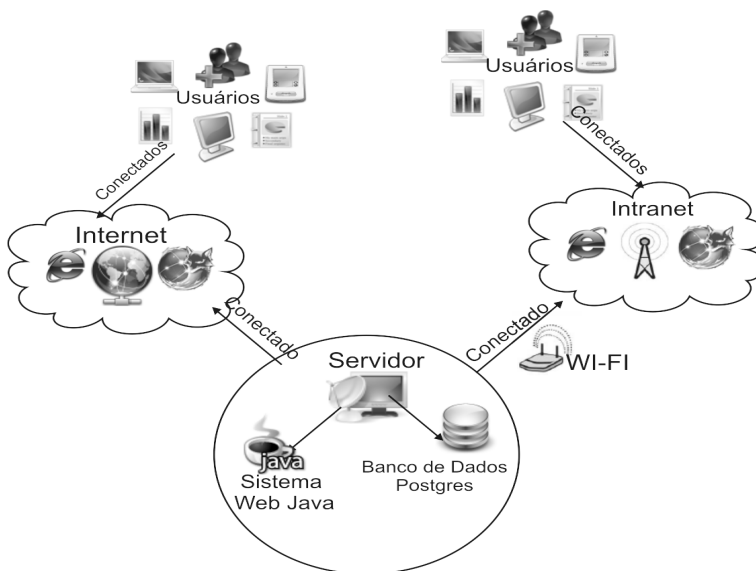


Figura 5
Arquitetura do Modulo I (módulo server do sistema)

A plataforma Java Micro Edition ou J2ME como é mais conhecida é o segmento do Java destinado aos dispositivos de pequeno porte, tais como telefones celulares, PDAs, *smartphones*, televisores digitais de nova geração, sistemas de navegação automotiva, comutadores e roteadores de rede, componentes para automação residencial.

Com a tecnologia J2ME é possível criar aplicações especiais para tirar proveito das vantagens e particularidades dos aparelhos móveis, aplicações integradas ao serviço das operadoras, que tenham a interatividade esperada para um pequeno aparelho e que respondam de acordo com a situação e a atenção do usuário.

A sua principal função vincula-se no atendimento ao público, facilitando o serviço efetuado pelos garçons, que precisam ser exatos na listagem dos pedidos solicitados pelos clientes. A Figura 7 apresenta a lista de pedidos de um cliente.

No Módulo II o funcionamento organiza-se da seguinte forma: cada garçom utiliza um *palm*, que por meio deste se comunicará com o Módulo I do sistema, transmitindo via *wireless* os pedidos feitos pelos clientes. Depois de fazer os pedidos, o garçom pode atender outros clientes, não tendo mais que retornar ao balcão para passar os pedidos em papel. Além disso, o mesmo pedido fica disponível para consulta entre outros setores da empresa. Todo esse processo é permitido somente pelos garçons cadastrados no banco de dados, conforme representação no módulo I do sistema.

O primeiro setor localiza-se na cozinha contendo a lista dos pedidos desejados e suas respectivas observações. Os cozinheiros recebem estes dados que serão apresentados em uma tela de computador personalizada, através do sistema web.

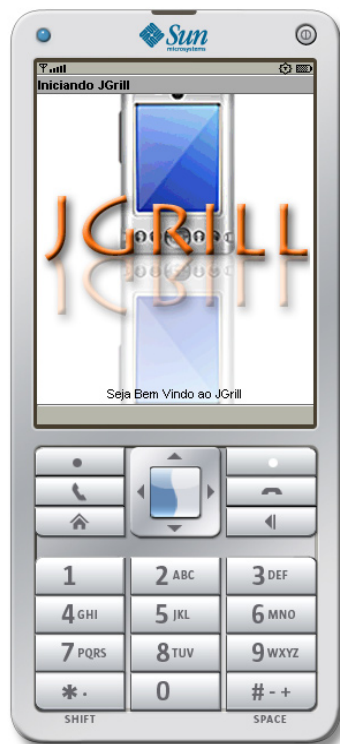


Figura 6
Tela inicial do Módulo II
(Módulo Cliente)

O segundo setor localiza-se no balcão de atendimento com o caixa, que receberá as informações sobre o pedido, calculando o valor total.

O cozinheiro finaliza o pedido e o sistema informará ao dispositivo móvel do garçom que efetuou o pedido. O cliente pode solicitar a sua conta para qualquer garçom, pois todos os colaboradores poderão consultar o valor correspondente ao cliente que o solicitou. Todo esse processo visa à clareza nas informações do pedido e à agilidade no atendimento ao cliente. O cliente efetuará o pagamento no caixa, concluindo o processo de atendimento. A Figura 8 explica como todos estes processos funcionam.

No desenvolvimento do protótipo, adotou-se o padrão MVC (*Model-View-Controller*). Um dos principais objetivos do padrão MVC é a organização do código de uma aplicação em camadas, realizando assim a separação física dos componentes do software. Desta forma, a organização em camadas é a chave para a independência entre os componentes, visando agrupar componentes por responsabilidades em comum (FRAGMENTAL, 2008).

As camadas de separação física e lógica proposta pelo modelo são: a camada *View*, em que são apresentadas todas as interfaces que interagem com o usuário permitindo a realização de requisições ou entrada de dados.

Na camada *Controller* é desenvolvida toda a parte que controla o comportamento da aplicação, servindo como uma intermediária entre a *View* e o *Model*. E por último, a camada *Model* é a responsável por conter o código da ação que foi requisitada como, por exemplo, se comunicar com outra camada que busque dados em um banco (BUSCHMANN, 2007).

A IDE (*Integrated Development Environment*) escolhida para o desenvol-

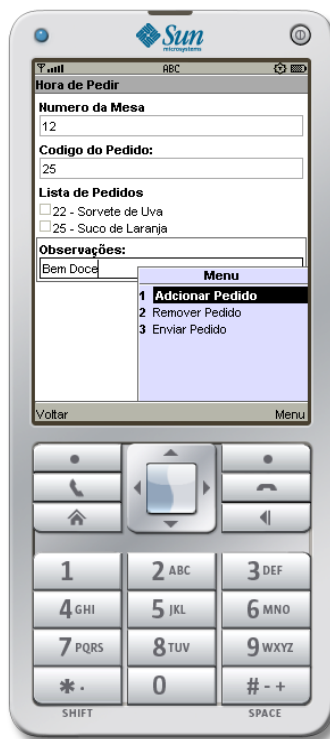


Figura 7
Tela de pedido

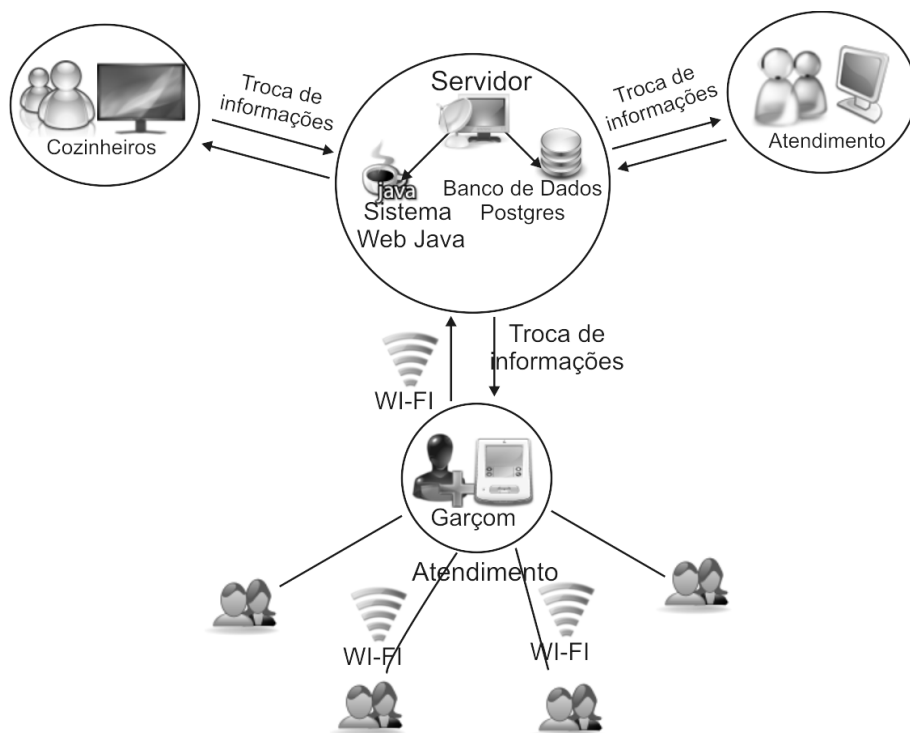


Figura 8
Arquitetura do Módulo II (módulo mobile) do sistema.

vimento e codificação do sistema, tornou-se um elemento importante neste projeto, uma vez que alguns argumentos como praticidade, confiabilidade e outras condições objetivas, tais como tempo, recursos materiais e tecnológicos, devem ser considerados para a sua realização.

O desenvolvimento dos módulos pertinentes ao sistema foram desenvolvidos na IDE *NetBeans* por ser modular e basear-se em padrões da linguagem de programação Java.

O projeto *NetBeans* consiste em uma IDE de código-fonte aberto e em uma plataforma de aplicativo que utiliza uma estrutura genérica para construir qualquer tipo de aplicativo. A IDE pode ser executada em muitas plataformas, incluindo *Windows*, *Linux*, *Mac OS X* e *Solaris*.

Gonçalves (2008) afirma que a IDE é de fácil instalação e de uso imediato, permitindo assim uma melhor integração com os membros da equipe de desenvolvimento.

A versão utilizada da IDE *NetBeans* foi a 6.1, que já contava com inúmeros e recursos compatíveis com as necessidades da equipe, sendo que algumas destas características solidificaram a escolha:

- Realce de semântica;
- Auto-completar código e análise de tipo;
- Correções rápidas e verificações semânticas;
- Refatoração;
- Assistentes para arquivos de configuração XML e controladores *Spring Web MVC*;
- Gerenciamento de banco de dados;
- Especificação de dependências de biblioteca usando caminhos de biblioteca relacionada (para padrão Java, *Web*, e todos os tipos de projeto Java EE);
- Emulador *Mpowerplayer MIDP* para aplicativos MIDP no *MacOS X* (disponível a partir da Central de *plug-ins*);
- *SVG Composer* para componentes personalizados SVG;
- Documentação e estabilidade avançada;
- Modelagem UML Visual.

A utilização desta ferramenta proporcionou um ambiente agradável de desenvolvimento. Além disso, o tempo de adaptação tornou-se favorável diante de um vasto campo de opções e funções que esta IDE proporciona para os desenvolvedores.

O *Hibernate* foi a solução utilizada como forma mais completa de integrar a orientação a objetos com o banco de dados do tipo relacional. O uso deste *framework* para mapeamento objeto-relacional contribuiu para o ágil desenvolvimento, diminuindo a complexidade entre os programas Java e o *PostgreSQL*.

Para documentação do projeto, adotou-se o padrão para *Javadoc*, que é fortemente utilizado em códigos Java. Para Bashan (2005), toda a descrição de pacotes, classes, atributos, métodos e parâmetros encontram-se junto com o código-fonte, sendo possível gerar arquivos a partir destas descrições com o *Javadoc*.

O *Struts* é um *framework* baseado para *web* e tem se tornado, de fato, padrão na comunidade J2EE. O *Struts* provê uma implantação do modelo MVC-2 para construções de aplicações *web* (DUDNEY, 2003), que foi adicionado ao

projeto como ferramenta para codificação. O uso do *Struts* no projeto visa à utilização de recursos de organização das ações do sistema e da validação de dados que o *framework* proporciona.

Os desenvolvedores que trabalham em aplicações eletrônicas convencionais estão acostumados a kits de ferramentas e a ambientes de desenvolvimento os quais aproveitam a igualdade entre as aplicações (HUSTED, 2004). Os *frameworks* da aplicação fornecem uma base comum reutilizável podendo servir como fundação para um produto específico.

Para o servidor de aplicação *web* foi escolhido o *Glassfish*. Desenvolvido pela empresa *Sun Microsystems*, é um dos mais completos com suporte a especificações Java EE, além de comercialmente, poder ser utilizado para depurar outros servidores de aplicação.

A utilização de determinadas especificações Java padronizadas pela Sun foram fundamentais para possibilitar o desenvolvimento do sistema. Utilizou-se a tecnologia, Utilizou-se a API que padroniza do lado servidor alguns dos componentes da tecnologia J2EE (*Java Enterprise Edition*), com regras de desenvolvimento definidas como implantação de interfaces responsáveis por tarefas definidas ou regras de herança. Ela tem como vantagens o tratamento automático de segurança, persistência, transação dentre outros serviços (SAMPAIO, 2007).

Na codificação deste projeto foi utilizada também, algumas estruturas da especificação Java J2EE, tal como o *Java Servlet*, que estende as funcionalidades de um *webserver* e pode fazer uso de qualquer outra API. As regras de negócio foram codificadas com a utilização de *servlet*, onde as mesmas possuíam códigos com a linguagem Java.

Na listagem de dados, *layout* da pagina e outras informações pertinentes à exibição foi utilizado *JavaServer Pages* (JSP), o qual faz as funções de um *script* em uma página *web* e é compilado como *servlet* em seu container.

Para o tratamento do layout na JSP utilizou-se CSS (*Cascading Style Sheets*), HTML (*Hypertext Markup Language*) e quando relacionado à transferência de informações entre uma JSP e *Servlet* incorporou-se o EL (*Expression Language*). Primeiramente introduzido no JSTL (*JavaServer Pages Tag Library*) o EL é uma linguagem de programação que permite operações necessárias a uma linguagem de scripts: controlar fluxos, atribuir e receber valores de variáveis, apresentar um valor na tela a partir de expressões, acessar propriedades de um objeto em Java, chamar os métodos de um *Java Beans* e outras diversas funções.

No desenvolvimento do módulo móvel, há uma aplicação *Java 2 Micro Edition*, e para a codificação deste módulo foram utilizadas estruturas como *MiddletVisual*, *Visual Mobile Designer* (VMD), possuindo a estrutura para codificação

de acordo com esta especificação e outros itens como: o fluxograma de um determinado pacote, um palco visual para a criação das telas pertinentes ao dispositivo móvel, podendo fazer a utilização de clicar e arrastar para o desenvolvimento das telas e itens de análise.

Para um tratamento visual mais específico foi utilizada a estrutura já compatível com a *MidletVisual*, *SVG (Scalable Vector Graphic)* e juntamente com API específica *JSR-266 (Java Specification Request)*. Esses possuem suporte e são incluídos no Netbeans Mobility Pack para *CLDC (Limited Device Configuration) / MIDP (Mobile Information Device Profile)*.

Podem-se adicionar arquivos gráficos baseados em vetores, pois embora compactados, podem ter uma alta qualidade. O *SVG* também tem suporte a *scripts* de animação, permitindo personalizar e atribuir uma melhor interatividade para o usuário. A Figura 9 apresenta uma animação criada em *SVG* para o sistema de Pizzaria.

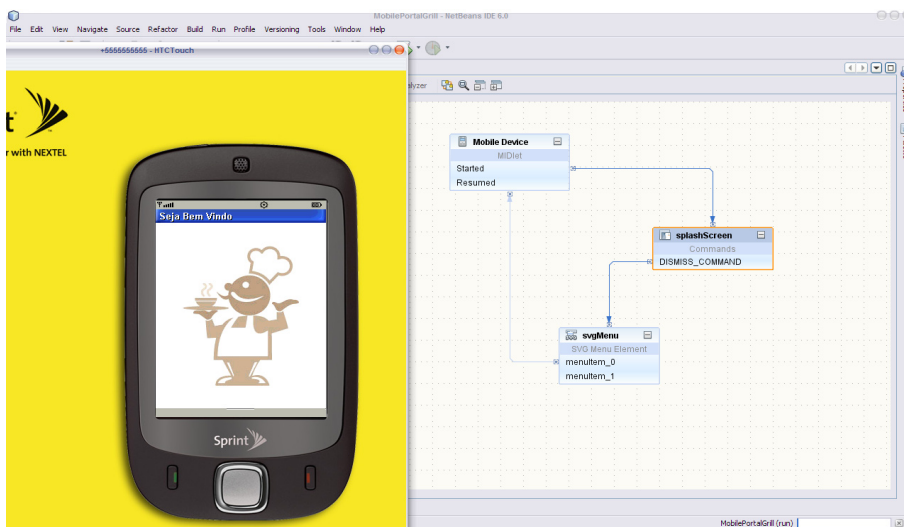


Figura 9
Visual Mobile Designer (VMD) e uma animação para o sistema de Pizzaria.

CONCLUSÃO

A utilização do *Linux Ubuntu* como sistema operacional do servidor da aplicação, nos ofereceu uma série de facilidades, tais como: um instalador perso-

nalizado, um conjunto de ferramentas de software disponíveis a partir de servidores para permitir a instalação de novos softwares, métodos padrões para configuração e notificações de problemas nos softwares, e grandes repositórios de softwares em servidores acessíveis através da internet (HILL,2006).

Fez-se a instalação padrão utilizando-se dos seguintes softwares: *jdk*, *glassfish*, *postgresql*, *rsync* (para automação de *backups* do sistema). O protótipo está em fase de teste, apresentando até o momento 60% dos requisitos plenamente atendidos. Este teste foi feito por simulação em laboratório por softwares específicos, como o *Wireless ToolKit*, da IDE *Netbeans*.

Atualmente não existem pizzarias utilizando o sistema, porém há empresas locais interessadas. Diante das análises feitas, notou-se também a necessidade do desenvolvimento de mais funções no sistema para agregar valor, como por exemplo, o controle de estoque, controle de comissão para os garçons e atendimento de pedidos via Internet.

REFERÊNCIAS BIBLIOGRÁFICAS

ALMEIDA, Leandro Batista de. **Introdução à J2ME e Programação MIDP**. 5 ed. Mundo Java, Curitiba, maio 2004. Reportagem de Capa, p. 20-27.

AMORIM, Andrea Rodrigues de. **Desenvolvimento de Aplicações Móveis com J2ME**. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação). Universidade Luterana do Brasil, 2005. Disponível em:< <http://www.ulbra.tche.br/~tcc-canoas/concluidos2005-2.html>>. Data de acesso: 25 de maio de 2008.

BASHAN, Brian, SIERRA, Kathy, BATES, Bert. **Use a Cabeça!: JSP & Servlets**. Ed. Alta Books 2005.

GONÇALVES, Edson. **Desenvolvendo Aplicações Web com NetBeans IDE 6**. Ed. Ciência Moderna, 2008

BUSCHMANN, F. et. al. **Pattern-Oriented Software Architecture: “A System of Patterns”**. Chichester, UK: John Wiley & Sons, 2007.

PRESSMAN, Roger S. **Engenharia de Software**. 5. ed. Rio de Janeiro: McGraw-Hill, 2002.

FRAGMENTAL **Tecnologia - Artigo MVC e Camadas**, <http://fragmental.com.br/wiki/index.php?title=MVC_e_Camadas> Acessado em 11 de junho de 2008.

LEAL, Maurício. **Começando com Java Wireless**. 18 ed. Java Magazine, Rio de Janeiro, 2004. Primeiros Passos, p. 16-19.

SAMPAIO, Cleuton. **Guia do Java: Enterprise Edition 5**. Ed. Brasport, Rio de Janeiro, 2007.

HUSTED, Ted, DUMOULIN, Cedric, FRANCISCUS, George, WNTERFELDT, David **Struts in Action** . Greenwich: Manning Publications, 2004.

DUDNEY, Bill, LEHR, Jonathan. **Jakarta Pitfalls - Time-saving Solutions for Struts**, ANT, Junit and Cactus . Indianapolis: Willey Publishing. 2003.

HILL, Benjamin Mako, BACON, Jono, BURGER, Corey, JESSE, Jonathan, KRSTIC, Ivan. **The Official Ubuntu Book**. ed. Prentice Hall, 2007.