

**FACULDADES INTEGRADAS DE TAQUARA  
CURSO DE SISTEMAS DE INFORMAÇÃO**

**UTILIZAÇÃO DA TECNOLOGIA J2ME PARA DESENVOLVIMENTO DE UMA  
APLICAÇÃO DE APOIO E TOMADA DE DECISÃO NO SETOR DE VENDAS DA  
ÁREA CALÇADISTA**

**LUIZ RODRIGO JARDIM DA SILVA**

**Taquara  
2008**

**LUIZ RODRIGO JARDIM DA SILVA**

**UTILIZAÇÃO DA TECNOLOGIA J2ME PARA DESENVOLVIMENTO DE UMA  
APLICAÇÃO DE APOIO E TOMADA DE DECISÃO NO SETOR DE VENDAS DA  
ÁREA CALÇADISTA**

Trabalho de Conclusão apresentado ao Curso de Sistemas de Informação das Faculdades Integradas de Taquara, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação, sob orientação do Prof. Mestre Francisco Assis Moreira do Nascimento.

**Taquara  
2008**

## RESUMO

Este trabalho apresenta um sistema direcionado para empresas da área calçadista, mais especificamente para representantes comerciais do setor de vendas. Sua principal finalidade é oferecer mobilidade aos seus usuários, e também fornecer informações para auxílio na tomada de decisão. Em um ambiente cada vez mais competitivo para as empresas brasileiras, principalmente do ramo calçadista, ferramentas que agreguem algum diferencial no suporte a clientes e no auxílio a vendedores são uma das alternativas existentes para o enfrentamento das diversidades existentes atualmente. A aplicação é um módulo independente integrado a um dispositivo móvel compatível com a tecnologia J2ME e interage com um sistema de gestão ERP.

**Palavras-chave:** Vendas. Tecnologias móveis. Área calçadista.

<b>1</b>	<b>INTRODUÇÃO</b> .....	6
<b>1.1</b>	<b>Contexto atual do mercado calçadista brasileiro</b> .....	7
1.1.1	Processo atual do setor de vendas calçadista brasileiro.....	9
1.1.2	Problemas no processo de vendas do setor calçadista.....	10
<b>1.2</b>	<b>Tecnologias móveis para minimizar problemas</b> .....	11
1.2.1	Mobilidade como vantagem competitiva.....	12
1.2.2	Tecnologias existentes.....	13
<b>1.3</b>	<b>Sistemas de apoio à decisão(SAD)</b> .....	15
1.3.1	Conceitos.....	16
1.3.2	Sistemas de apoio à decisão existentes.....	19
<b>1.4</b>	<b>Tecnologias para implementação de um sistema de apoio à decisão móvel</b> .....	19
1.4.1	J2ME.....	21
1.4.2	NetBeans.....	23
1.4.3	LWUIT ( <i>LightWeight User Interface Toolkit</i> )	24
1.4.4	Banco de dados Oracle.....	25
1.4.5	Servidor de aplicações Tomcat.....	26
<b>1.5</b>	<b>Metodologias para desenvolvimento de <i>software</i></b> .....	28
<b>2</b>	<b>SAVM – SISTEMA DE APOIO À VENDAS MÓVEL</b> .....	31
<b>3</b>	<b>DESENVOLVIMENTO DO SISTEMA SAVM</b> .....	35
<b>3.1</b>	<b>Metodologia</b> .....	35
3.1.1	Descrição dos requisitos ( <i>Product BackLog</i> ).....	36
3.1.2	Detalhamento dos casos de uso ( <i>Sprints</i> ).....	39
3.1.2.1	Autenticação.....	40
3.1.2.2	Mensagens .....	41
3.1.2.3	Consulta de pedidos.....	43
3.1.2.4	Análise de restrições.....	47
3.1.2.5	Empenho de estoque.....	48
3.1.2.6	Simulação de vendas.....	54
3.1.3	Implementação.....	56
<b>3.2</b>	<b>Instalação do sistema SAVM</b> .....	61
<b>4</b>	<b>USO DO SISTEMA SAVM</b> .....	62

<b>4.1</b>	<b>Efetuando autenticação.....</b>	<b>63</b>
<b>4.2</b>	<b>Verificando mensagens.....</b>	<b>64</b>
<b>4.3</b>	<b>Empenhando estoque.....</b>	<b>65</b>
<b>4.4</b>	<b>Consultando pedidos.....</b>	<b>68</b>
<b>4.5</b>	<b>Analisando restrições.....</b>	<b>70</b>
<b>4.6</b>	<b>Simulando vendas.....</b>	<b>71</b>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>76</b>
	<b>REFERÊNCIAS.....</b>	<b>77</b>

## 1 INTRODUÇÃO

Atualmente as empresas brasileiras do setor coureiro-calçadista não estão só competindo entre si, considerando o fator local, mas sim com empresas de outros estados e principalmente de outros países. Isto se deve principalmente a globalização econômica, onde as mesmas são obrigadas a expor suas economias locais e nacionais frente à competitividade internacional (ABICALÇADOS, 2007).

Com todo esse cenário, as empresas precisam buscar novas alternativas para que consigam atingir novos mercados e também se manterem dentro dele. Segundo Hamel e Prahalad (1996), a busca da competitividade das companhias se dá principalmente através da reengenharia de seus processos e sua melhora contínua, e também da reinvenção dos setores com regeneração das suas estratégias.

Um dos setores que precisam estar constantemente se adaptando com a realidade atual é o setor de vendas, pois é um dos principais canais de acesso dos consumidores. Para que uma empresa tenha sucesso no mercado em que está inserida, é necessário que um dos seus focos principais seja o atendimento ao cliente, procurando saber das suas necessidades, suas avaliações sobre o produto que adquiriu, saber que benefícios seus produtos ou serviços existentes oferecem, ou seja, é necessário prestar todo suporte necessário ao consumidor.

Tendo em vista a necessidade sempre constante das empresas de estarem agregando elementos que garantam sua competitividade no mercado, com este trabalho propõe-se desenvolver uma ferramenta que disponibilize de uma maneira fácil e ágil um suporte qualificado ao consumidor, permitindo assim que o representante comercial tenha em mãos informações que o auxiliem no suporte e na tomada das decisões. Para isto, o *software* deve ser executado em um dispositivo móvel (celulares, *Smartphones*, etc) e permitir ao vendedor o seu deslocamento até o cliente, garantindo assim sua mobilidade.

Aproveitando também a abrangência cada vez maior das redes de comunicações, de telefonia celular principalmente, a aplicação deve poder ser utilizada em qualquer lugar que exista cobertura dessas redes, ou seja, o representante poderá estar onde o cliente estiver.

O sistema proposto chamado SAVM (Sistema de Apoio a Vendas Móvel) consiste de um módulo independente, instalado em um dispositivo compatível com a tecnologia J2ME (Java 2 Micro Edition) e que interage com um sistema ERP já desenvolvido e em uso atualmente, chamado SisCon<sup>1</sup>.

Este trabalho está organizado da seguinte maneira: neste capítulo 1 é descrito o contexto atual do mercado calçadista brasileiro. Também são apresentadas as vantagens da utilização das tecnologias móveis atualmente. A utilização de sistemas de apoio à decisão também é mostrado, bem como seus conceitos. Já no capítulo 2 têm-se apresentada a descrição completa do *software*. No capítulo 3 será mostrado como foi desenvolvida a aplicação, sua instalação e requisitos básicos para seu funcionamento incluindo a metodologia utilizada. Com o capítulo 4 será possível saber como será a utilização do sistema e finalmente no capítulo 5 poderá ser vista as conclusões e considerações finais sobre o desenvolvimento desse trabalho.

## **1.1 Contexto atual do mercado calçadista brasileiro**

O setor calçadista no Brasil é um dos setores que mais está sofrendo com os efeitos da globalização, pois as empresas desse ramo estão cada vez mais buscando novos mercados. Um dos fatores importantes que contribuem para acirrar a competitividade do setor calçadista é o aquecimento da economia chinesa e de outros países asiáticos bem como do leste europeu. Esses países se beneficiaram muito com a globalização, pois não só conseguiram atingir novos mercados, como também tiveram oportunidade de importar mão-de-obra especializada de outros países, como do Brasil, por exemplo. Com isso, aliando-se ao domínio da tecnologia e com mão-de-obra de baixo custo e também de subsídios dos governos, essas empresas estão cada vez mais presentes no mercado calçadista (BRAZILIAN FOOTWEAR, 2007).

---

<sup>1</sup> *Software* de gestão empresarial de propriedade da empresa Dws Informática Com.Imp.Exp Ltda

O contexto econômico local também é um dos fatores importantes que impõe grandes desafios para as empresas do ramo calçadista, tanto para as que exportam seus produtos como para as que não exportam, pois diferenças cambiais, principalmente em comparação ao Dólar (moeda norte americana) e também ao Euro (moeda européia), diminuem o preço de venda dos produtos importados, facilitando assim sua entrada no país.

**Tabela 1: Relação dos cinco maiores produtores/exportadores/consumidores de calçados do ano de 2005**

Produtores		Exportadores		Consumidores	
País	Pares (milhões)	País	Pares (milhões)	País	Pares (milhões)
<b>China</b>	9.000,0	<b>China</b>	6.914,0	<b>EUA</b>	2.241,9
<b>Índia</b>	909,0	<b>HongKong</b>	741,0	<b>China</b>	2.096,5
<b>Brasil</b>	762,0	<b>Vietnã</b>	472,7	<b>Índia</b>	852,4
<b>Indonésia</b>	580,0	<b>Itália</b>	249,0	<b>Japão</b>	650,3
<b>Vietnã</b>	535,0	<b>Brasil</b>	217,0	<b>Brasil</b>	555,0

Fonte: ABICALÇADOS (2007).

**Tabela 2: Relação dos cinco maiores produtores/exportadores/consumidores do ano 2004**

Produtores		Exportadores		Consumidores	
País	Pares (milhões)	País	Pares (milhões)	País	Pares (milhões)
<b>China</b>	8.100,0	<b>China</b>	5.885,0	<b>China</b>	2.223,9
<b>Índia</b>	850,0	<b>HongKong</b>	744,6	<b>EUA</b>	2.130,7
<b>Brasil</b>	755,0	<b>Vietnã</b>	420,2	<b>Índia</b>	796,5
<b>Indonésia</b>	533,0	<b>Itália</b>	279,3	<b>Japão</b>	620,0
<b>Vietnã</b>	533,0	<b>Brasil</b>	212,0	<b>Brasil</b>	547,0

Fonte: ABICALÇADOS (2007).

**Tabela 3: Relação dos cinco maiores exportadores de calçados para o Brasil no ano de 2005**

País	Pares
<b>China</b>	13.795.636
<b>Vietnã</b>	1.140.135
<b>Indonésia</b>	566.348
<b>Tailândia</b>	267.436
<b>Itália</b>	60.311

Fonte: ABICALÇADOS (2007).

**Tabela 4: Relação dos cinco maiores exportadores de calçados para o Brasil no ano de 2004**

<b>País</b>	<b>Pares</b>
<b>China</b>	7.782.658
<b>Vietnã</b>	332.789
<b>Indonésia</b>	384.174
<b>Tailândia</b>	52.883
<b>Itália</b>	106.719

Fonte: ABICALÇADOS (2007).

Fazendo um comparativo entre as tabelas 1 e 2, é possível ter uma compreensão globalizada do mercado calçadista sobre os cinco maiores exportadores e consumidores nos anos de 2004 e 2005, e é possível afirmar que países como a China e Vietnã aumentaram consideravelmente suas exportações para mercados que antes eram explorados principalmente pelo Brasil e pela Itália. Ainda observando os dados das Tabelas 3 e 4, o Brasil está importando cada vez mais calçados. Segundo a Abicalçados (2007), as importações de calçados provenientes da China já somam 68,2% do total das importações desse produto, criando assim um cenário altamente competitivo para esse ramo.

Considerando o cenário atual e o previsto, a necessidade de se atingir novos mercados é essencial para que as empresas do ramo calçadista se mantenham ativas e competitivas. E uma das alternativas é a qualificação das forças de vendas, para que as mesmas tenham condições de explorar novos mercados e garantir um melhor atendimento ao cliente.

#### 1.1.1 Processo atual do setor de vendas calçadista brasileiro

Atualmente as empresas do ramo calçadista fazem suas vendas basicamente de duas maneiras: a primeira seria através de um representante comercial, que é a maneira mais comum; e a outra é através de suas lojas, que são as vendas feitas diretamente ao consumidor (pronta entrega). Nesse trabalho será abordado o processo mais comum que é através do representante comercial.

Inicialmente o representante comercial se desloca até o cliente para iniciar a venda. Junto com o cliente, o vendedor efetua o registro do pedido, com base no

catálogo de produtos disponibilizado pela empresa e, após, o pedido é enviado para fábrica. Então é feita uma análise de crédito do cliente, para verificar se o mesmo se enquadra financeiramente dentro dos critérios estipulados pela empresa.

Após a análise de crédito, o pedido entra em processo de produção. O processo de produção não indica que o pedido esteja sendo produzido, é necessário que se enquadre dentro de certos critérios que podem variar de fábrica para fábrica, sendo os principais: prazo de entrega estipulado pelo cliente ou empresa; a capacidade de produção; as matérias-primas disponíveis para que seja produzido etc.

Então, depois de produzido, o pedido entra em processo de faturamento. Nessa etapa é verificada mais uma vez a situação financeira do cliente junto à empresa e nos órgãos de proteção ao crédito. Se for liberado dessa etapa o pedido então finalmente é faturado e entra em processo de entrega.

#### 1.1.2 Problemas no processo de vendas do setor calçadista

Atualmente o processo de vendas das indústrias do setor calçadista possui vários problemas referentes à gestão de suas forças de vendas como um todo, entre as principais pode-se destacar:

- a) alocação de recursos humanos e físicos para digitação dos pedidos, que normalmente ocorre com a extensão dos seus turnos de trabalho;
- b) horários de picos para transmissão dos pedidos, ocorrendo assim sobrecarga nas suas redes de comunicações (internet), servidores de banco de dados, etc;
- c) alocação de recursos humanos e físicos para prestar suporte aos seus vendedores;
- d) escassez de informações para seus representantes, impossibilitando assim a resposta imediata ao cliente;
- e) pedidos emitidos para clientes que estão em situação de inadimplência no cadastro da empresa;
- f) pedidos emitidos que excedem o limite de crédito estipulado para o cliente;

- g) indisponibilidade de vendas de produtos em estoque online, não havendo possibilidade de reserva e empenho do estoque, sendo que para essa tarefa também é exigido alocação de recursos humanos da empresa para que seja possível realizar essa tarefa;
- h) grande dependência por parte dos vendedores do suporte da empresa ao qual representam, gerando assim um sistema totalmente burocrático de vendas;
- i) dificuldades nas negociações de preços de vendas com os clientes, gerando possíveis perdas de vendas;
- j) desgastes de relacionamento entre empresa e vendedores, aumentando assim a rotatividade de representantes e gerando possíveis perdas de consumidores;

## 1.2 Tecnologias móveis para minimizar problemas

A utilização de tecnologias móveis é uma das alternativas para que problemas corriqueiros do dia-a-dia das empresas sejam atenuados ou até mesmo resolvidos.

A expansão das tecnologias móveis nos últimos anos, devido a sua grande utilização, fez com que surgissem vários conceitos e termos, onde os principais são destacados abaixo (SACCOL e REINHERT, 2004):

- a) tecnologias de informação móveis (*Mobile*): são sistemas que se caracterizam pela sua relação entre a mobilidade e portabilidade, ou seja, é a capacidade de se utilizar em qualquer lugar um dispositivo de tecnologia de informação (KALAKOTA e ROBINSON, 2002 *apud* SACCOL e REINHERT, 2004);
- b) tecnologias de informação sem fio (*Wireless*): são tecnologias que utilizam dispositivos conectados a uma rede sem fio, através de um aparelho de comunicação sem fio, como por exemplo, infravermelho, *Bluetooth*, redes locais sem fio (*Wireless LAN*). Nesse tipo de tecnologia é importante destacar que o conceito de mobilidade não poderá ser utilizado na sua plenitude, pois geralmente o alcance é limitado (SACCOL e REINHERT, 2004);

- c) tecnologias de informação ubíquas: envolve a utilização de diversas tecnologias, diversos dispositivos heterogêneos conectados e se comunicando por meio de diferentes redes sem fio (SACCOL e REINHERT, 2004);
- d) comércio móvel (*m-commerce*): onde o processo de comercialização de produtos e serviços se faz por meio de dispositivos portáteis sem fios (celulares, PDAs). É considerada como a próxima geração do comércio eletrônico;

A utilização de ferramentas *m-commerce* é uma das alternativas frente as tecnologias existentes que podem ser utilizadas pelas empresas para automatizar suas forças de vendas, e prover soluções para auxiliar nas suas tomadas de decisões, principalmente entre seus representantes e seus clientes. O uso desse tipo de aplicação agrega também um diferencial de competitividade importantíssimo, sendo principalmente a mobilidade.

### 1.2.1 Mobilidade como vantagem competitiva

A mobilidade subentende que o acesso à internet viaja com o cliente, dando a este capacidade de agir imediatamente ao detectar alguma boa oportunidade de negócio. A mobilidade é um atributo que atrai tanto compradores quanto consumidores (TURBAN, 2004, p.165).

A mobilidade é uma das grandes vantagens competitivas na atualidade, pois significa estar conectado a qualquer hora, em qualquer lugar, com acesso às informações importantes para realizar qualquer atividade. Adotando uma estratégia de mobilidade, é possível ter acesso imediato aos dados para auxílio na tomada de decisão, através de dispositivos móveis como celulares, Notebooks, PDAs entre outros (INTEL, 2007).

Segundo Fator Brasil (2007) uma pesquisa realizada pela IDC América Latina com 301 empresas com mais de 500 funcionários do Brasil, Argentina e México, mostrou que no Brasil o mercado de mobilidade é muito pouco explorado, o que conseqüentemente, indica um maior potencial de mercado para empresas do setor.

Entre os principais motivos pelos quais as corporações devem disponibilizar ferramentas para automatizar os processos de maneira remota, pode se destacar uma maior eficiência da força de trabalho, mais flexibilidade, melhor apuração e diminuição de erros, uma maior conveniência em poder estar em outro ambiente de trabalho, aumento na possibilidade na atualização de dados empresariais em movimento, suprimento de novas necessidades de negócios, boa relação custo-benefício e um crescimento na velocidade no processamento das informações (DECISION REPORT, 2007).

Segundo a LVBA (2007) uma pesquisa realizada pela Nokia feita com 500 executivos, indica que a cultura de mobilidade empresarial está cada vez mais presente hoje em dia, sendo que 33% dos executivos informaram que 20% de seus funcionários podem ser considerados “funcionários móveis”, ou seja, passam pelo menos um dia da semana fora da empresa. Foram mencionados os seguintes benefícios competitivos obtidos com a adoção da mobilidade; redução do tempo de reposta ao cliente 35%; maior colaboração dos funcionários dentro da empresa 27% ; uma maior habilidade de trabalhar com múltiplos parceiros/fornecedores 12%.

Ainda segundo a pesquisa realizada pela Nokia, os entrevistados afirmaram que pretendem aumentar o uso de soluções móveis nos próximos dois anos, com uma maior possibilidade no uso: (i) acesso remoto a rede interna das empresas 41%; (ii) gestão de relacionamento com o cliente; e (iii) automação da equipe de vendas 17%.

### 1.2.2 Tecnologias existentes

Em 1992 a empresa americana Apple lançou um *Handheld*<sup>2</sup> chamado Newton. Sua tela era sensível ao toque, tinha 1 *Megabyte* de memória e tinha a capacidade para transmissão de dados de 38.5kbps. Esse modelo não teve muita aceitação na época, mas pode ser considerado o início da introdução de dispositivos móveis ao mercado (AMSYST, 2007).

---

<sup>2</sup> Computador pessoal, portátil, de dimensões reduzidas e com pouca capacidade de processamento.

Em 1996, outra empresa americana lançou no mercado o *Palm Pilot* 1000 e 5000, sendo que estes tiveram uma grande aceitação e chegaram a abranger 80% do mercado mundial, e existem até hoje. Também nesse período começaram a surgir outros dispositivos com o sistema operacional *Windows CE 1.0* da Microsoft, sendo que esses dispositivos também não tiveram seu uso massificado na área de computação móvel. Mas a partir de 2000 com o surgimento do sistema operacional *Pocket PC 2000*, embutidos em dispositivos da empresa HP (*Hewlett Packard*) e da Compaq, teve uma grande aceitação e com isso esse mercado começou a crescer (AMSYST, 2007).

A Symbian fundada em 1998 por uma das grandes fabricantes de celulares da época em conjunto com a empresa inglesa PSION, desenvolveu o sistema operacional *Symbian*, sendo que esses rodam na maioria dos *Smartphones* e *Handhelds* da Nokia, e atualmente está presente na maioria dos dispositivos móveis europeus (AMSYST, 2007).

Atualmente está ocorrendo a convergência das funcionalidades de dispositivos móveis distintos, como funções palmtops, câmeras fotográficas, GPS (*Global Position System*), celulares, notebooks, impressoras (AMSYST, 2007).

Com o avanço das tecnologias móveis, dispositivos, redes de comunicação e plataformas de desenvolvimento, foi possível integrar aplicações que antes só eram disponíveis nos computadores pessoais. Com isso, de acordo com Taurion (2004), os aplicativos existentes até então em *PDA*s, celulares, entre outros, deixaram de ser somente simples agendas de contato, aplicativos de entretenimento (jogos), para se tornarem ferramentas orientadas a negócios.

Entre as ferramentas disponíveis no mercado atualmente podem ser citadas as seguintes:

- a) *SFA*: é um sistema de automação de forças de vendas de propriedade da Abaccon Brasil, onde foi utilizada a linguagem de programação J2ME para seu desenvolvimento. É indicada para efetuar controle do processo comercial. Entre suas principais funcionalidades, de acordo com o fornecedor, pode-se destacar: cadastro de clientes, contatos e oportunidades; acesso a informações em qualquer lugar e qualquer momento; consulta e validação de crédito; customização de acordo com as necessidades do cliente (ABACOMM, 2008);

- b) *J-Sales*: ferramenta desenvolvida pela empresa brasileira Trevisan Tecnologia. É destinada a agilizar as informações no processo de vendas, reduzir as falhas na comercialização e dar suporte aos clientes. É possível integrá-lo ao sistema de gestão do cliente. Suas principais funcionalidades são: gestão e controle do dia-a-dia dos vendedores; acompanhamento do histórico dos clientes; registro de pedidos e visitas; gestão de estoque, etc (TREVISAN, 2008);

A evolução constante das tecnologias móveis está possibilitando que ferramentas comuns em PCs (computadores pessoais), sejam integradas a dispositivos portáteis sem fio. A expansão e o aumento da velocidade das redes sem fios, de celulares principalmente, também estão contribuindo significativamente para que isso ocorra, pois a capacidade de se transmitir dados e recebê-los se torna aceitável para o perfeito funcionamento desses aplicativos.

Aplicativos que auxiliam executivos, gestores, representantes comerciais na tomada de decisão são exemplos de ferramentas que podem ser utilizados em dispositivos móveis, pois estes atualmente já provêm de certa capacidade computacional para executá-los, possibilitando assim uma combinação muito importante de sistemas de apoio à decisão com mobilidade.

### **1.3 Sistemas de apoio à decisão (SAD)**

Atualmente, um dos principais insumos estratégicos para as organizações é a informação, o mundo organizacional está cada vez mais complexo, a diversidade de dados e opções à nossa volta é tanta que o processo de tomada de decisão necessita de um tratamento adequado. (CLERICUZI, 2006).

A necessidade de se obter informações cada vez mais precisas e eficazes estão fazendo com que as empresas busquem ferramentas computacionais que as ajudem no processo diário de apoio à tomada de decisões (CLERICUZI, 2006). Segundo Chiavenato (1993 *apud* CLERICUZI, 2006) grande parte do valor das

organizações estão se dando em cima das informações que as mesmas possuem e não mais em bens físicos e dinheiro, e isso se deve principalmente ao avanço das tecnologias de informação. Com isso pode-se observar que esse tipo de ferramenta está muito presente nas corporações atuais. E estão sendo integradas a diversos tipos de tecnologias existentes, como por exemplo, as móveis.

À medida que os dispositivos móveis se tornarem mais rápidos, mais seguros e mais acessíveis, a tendência natural é que esse tipo de tecnologia se torne cada vez mais presente no dia-a-dia das empresas. A possibilidade dos gestores, representantes comerciais e diretores utilizarem suas ferramentas tradicionais em qualquer lugar será muito maior, sendo que antes eram disponíveis somente em seus locais de trabalho e seus escritórios. Agregando assim um grande diferencial de competitividade e produtividade nas suas operações.

Pode-se afirmar com isso que os sistemas de apoio à decisão provavelmente serão umas das ferramentas que estarão integradas a dispositivos móveis sem fios, pois com o grande avanço das redes de comunicações sem fios que está ocorrendo atualmente, a sua utilização começa a se tornar viável e o conceito de mobilidade dentro das organizações se tornará uma realidade.

### 1.3.1 Conceitos

Um sistema de apoio de decisão (SAD) é um sistema de informação baseado em computador que combina modelos e dados, em uma tentativa de solucionar problemas semi-estruturados com grande envolvimento por parte do usuário (TURBAN, 2004, p.370).

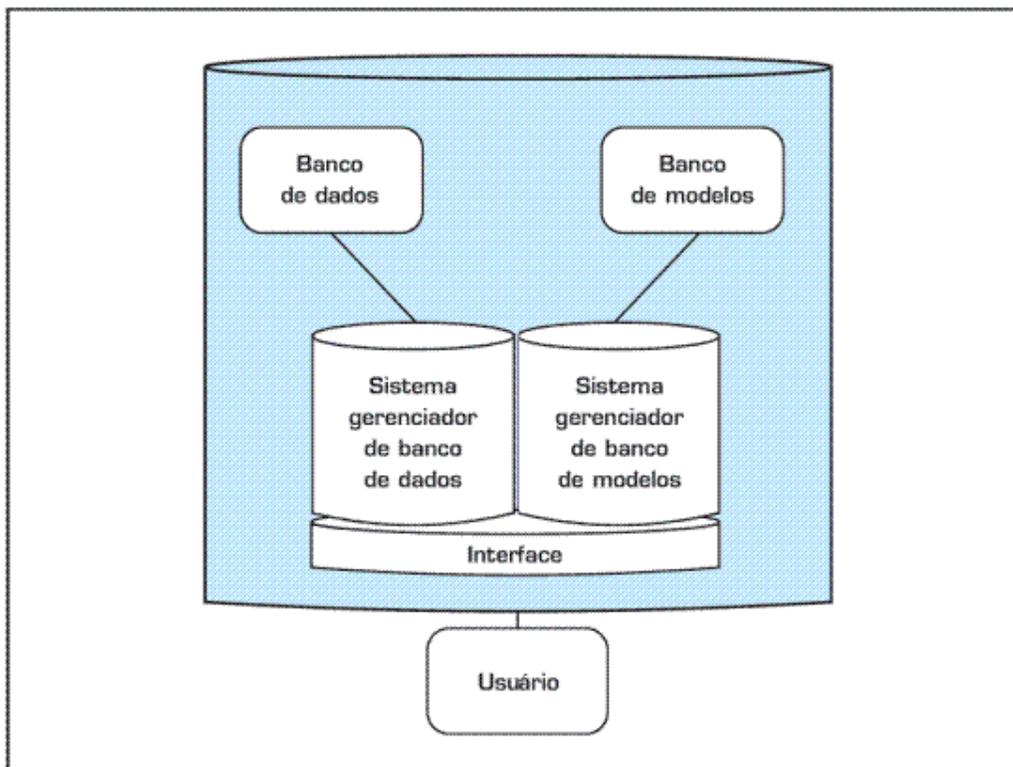
Segundo Turban (2004) muitas empresas atualmente estão usando sistemas de apoio à decisão para melhorar seus processos decisórios e isso se deve as seguintes necessidades citadas abaixo:

- a) necessidade de informações novas e cada vez mais precisas;
- b) necessidade de se obter informações mais rapidamente;

- c) as organizações estão enfrentando uma maior concorrência nos mercados internos e externos;
- d) as empresas estão operando em uma economia muito instável;
- e) volume muito grande de informações em seus bancos de dados;
- f) necessidade de se detectar diversos problemas de gestão;
- g) resposta imediata ao cliente;
- h) montagem de seus planejamentos estratégicos;
- i) mercado muito dinâmico;

Não há um consenso comum sobre o que exatamente constitui um sistema SAD, principalmente referente às suas características e também sobre suas capacidades. Conforme o autor, um sistema SAD é caracterizado por prestar apoio para tomadores de decisões em todos os níveis gerenciais, estratégico e operacional, tanto individuais como em grupos. É bem mais destacado para problemas não estruturados e semi-estruturado, onde é necessária a capacidade humana de julgamento.

Uma outra característica importante de um sistema de apoio à decisão é a possibilidade de *análises de sensibilidade*. É extremamente, pois o torna flexível e adaptável a inúmeras situações que existem em um processo decisório. Essa análise permite que o usuário final possa inserir seus próprios dados e podendo assim simular várias situações.



**Figura 1: Componentes de um sistema SAD**

Fonte: Sprague e Watson(1989) *apud* Clericuzi (2006, p.10).

Na figura 1 é apresentado o modelo conceitual de um sistema de apoio à decisão. De acordo com Sprague e Watson (1989) a arquitetura de um sistema SAD é composta de três subsistemas: um sistema gerenciador de bancos de dados (SGBD); um sistema gerenciador de bancos de modelos (SGM); e uma interface amigável para o usuário. Sendo que o banco de dados é associado a um SGBD para que este faça o gerenciamento de todos os dados armazenado, proporcione filtros, controle e segurança. Já o banco de modelos engloba todas as ferramentas necessárias para coletas de dados e para apoiar na decisão de acordo com o nível de estruturação. Enquanto a interface é responsável pela interação com o usuário.

Por fim, para se construir um sistema de apoio à decisão primeiramente é necessário que seja definida o nível de estruturação do problema e com isso seja possível distinguir as abordagens que deverá ser aplicado (CLERICUZZI, 2005).

### 1.3.2 Sistemas de apoio à decisão existentes

Entre as aplicações existentes e implementadas com sucesso pode-se destacar a ferramenta de apoio à decisão DSS Agent desenvolvida pela empresa Microstrategy. Este sistema foi posto em operação na empresa norte americana ShopKo, que possui uma rede de descontos e tem em torno de 130 lojas espalhadas por aquele país. A organização que atua em um ambiente extremamente competitivo e seu sucesso depende da capacidade de tomar decisões rápidas. Antes da implantação do *software*, o sistema de informação era totalmente rígido, sendo que suas informações eram fragmentadas e levavam-se horas ou até mesmo dias para se obter relatórios que auxiliassem os gestores na sua administração (TURBAN, 2004).

Os resultados obtidos com a implantação do SAD DSS Agent foram muito satisfatórios de acordo com Turban (2004) para a empresa ShopKo, pois deram condições da mesma de obter relatórios atualizados, informações sintetizadas de acordo com as suas necessidades, previsões precisas de giro de estoque, controle custos, e melhoraram consideravelmente a relação com seus clientes e fornecedores.

Entre as principais características da ferramenta, pode-se destacar:

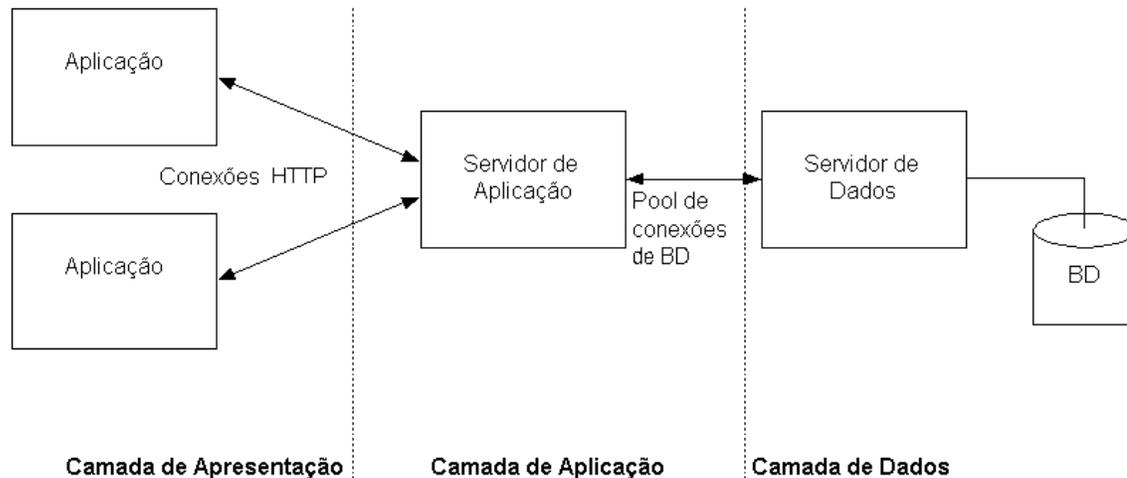
- a) estatísticas diárias de vendas por loja;
- b) resumo de dados de venda por departamento, região, etc;
- c) previsões de vendas e de estoques;
- d) diversas análises de compras de clientes, como o exemplo de clientes que compram somente produtos em liquidação;

## 1.4 Tecnologias para implementação de um sistema de apoio à decisão móvel

Para o desenvolvimento de um SAD, é necessária a utilização de diversas ferramentas e tecnologias. Nesse trabalho é utilizada a tecnologia J2ME usando

como IDE a ferramenta NetBeans. Foi utilizado o conceito de desenvolvimento em camadas, sendo: camada de apresentação (interface do usuário); camada de aplicação (lógica de negócio); camada de dados (servidores de dados);

Na figura 2 é representando um sistema típico em três camadas;



**Figura 2: Arquitetura de um sistema três camadas.**

Fonte: Jacques (2008).

A camada de apresentação é responsável pela interação com o usuário, exibe os resultados, fornece formulários para que essa interação seja feita. Já a camada de aplicação armazena toda a lógica de negócio, atende as requisições feita pelo cliente e as gerencia, é controlado por um servidor de aplicação. Enquanto que a camada de dados é responsável pelo armazenamento físico das informações, e atende aos pedidos efetuados pelo servidor.

A programação em camadas ganhou destaque no começo dos anos 90, onde o desenvolvimento de aplicações cliente/servidor era muito utilizado na época, e era considerado uma arquitetura em duas camadas. Esse conceito de programação era muito bem adaptada até a chegada e a popularização da internet, pois com isso os aplicativos começaram a serem acessados pelos navegadores, e nesse novo contexto não era viável escrever toda a lógica de negócio na camada cliente. Por outro lado não foi possível armazenar a lógica de negócios em servidores de banco de dados, pois estes possuíam limitações, devido às suas linguagens de programação (*storeds procedures, functions, triggers*) que não eram padronizadas, sendo exclusiva de cada servidor, esbarrando assim no problema da portabilidade (MSDN, 2008).

### 1.4.1 J2ME

Em 1990 a Sun MicroSystem criou a linguagem *Java*, que inicialmente se chamava *Oak*, mas devido a problemas de registro de marcas e patentes foi mudado para *Java*. A Sun tinha como objetivo desenvolver uma linguagem com a qual fosse possível se desenvolver programas que pudessem ser executados em pequenos dispositivos portáteis e distribuídos, independentes da plataforma proprietária.

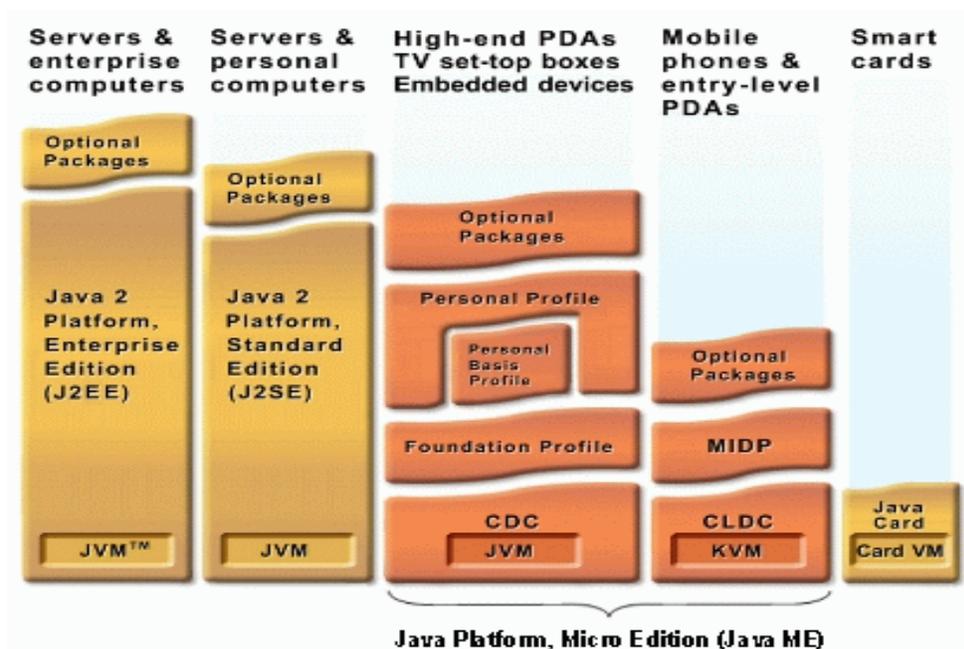
J2ME (Java 2 Micro Edition) é uma linguagem de programação para dispositivos móveis de propriedade da empresa Sun Microsystems. Segundo a Sun.Com (2007), é um ambiente robusto e flexível para que seja possível rodar aplicativos com pouca capacidade de processamento e memória (PDAs, TV set-top-boxes, celulares, impressoras, sistemas de navegação automotiva). Uma das características principais de *softwares* desenvolvidos nessa plataforma é a sua portabilidade, pois é possível executá-los em diferentes tipos de equipamentos, de fornecedores diferentes, bastando somente terem suporte para J2ME.

Atualmente a tecnologia J2ME é baseada em três elementos principais, conforme ilustrado na figura 3 (MUCHOW, 2005).

- a) configuração é a camada de *software* que contém bibliotecas básicas para que o aplicativo possa ser executado. Nela se encontram as máquinas virtuais KVM (*Kilobyte Virtual Machine*) e JVM (*Java Virtual Machine*), que são responsáveis por emular um sistema operacional próprio e com isso garantir a portabilidade da aplicação. Essa camada pode ser dividida entre CDC (*Connected Device Configuration*) e CDLC (*Connected Device Limited Configuration*), onde na camada CDC é possível executar programas com uma maior capacidade de processamento, como por exemplo, TV digital, PDAs, sistemas de navegação automotiva, enquanto que a camada CDLC que é indicada para equipamentos com limitada capacidade de processamento e memória, como por exemplo, celulares e palm hand-held;
- b) perfis são extensões de bibliotecas que possibilitam desenvolver aplicativos para um determinativo tipo de dispositivo e provem uma maior flexibilidade para o surgimento de novas tecnologias, bastando somente adicionar novos

perfis. Se destacam nessa arquitetura os perfis MIDP (*Midlevel Information Device Profile*) e *Foundation Profile*. A aplicação desenvolvida utilizando MIDP é chamada de *MIDlete*;

- c) pacotes adicionais são um conjunto de bibliotecas adicionais que são parecidos com os perfis, porém não define um ambiente completo de desenvolvimento e sim apenas uma extensão. Um exemplo de pacote adicional é *Wireless Messaging API* que é utilizado para enviar e receber mensagens SMS (*Short Messages Services*);



**Figura 3: Estrutura J2ME**  
Fonte: SUN.COM (2007).

De acordo com Muchow (2005), estima-se que apenas no Japão mais de 20 milhões de telefones móveis com suporte a tecnologia J2ME foram fabricados em 2001, e este número vem aumentando consideravelmente. Ainda segundo Muchow (2005), os interessados em utilizar essa plataforma para desenvolvimento poderão aumentar seus ganhos em produtividade e conseqüentemente em termos financeiros, pois oferece suporte a uma ampla variedade de dispositivos e portabilidade entre plataformas.

Para o desenvolvimento do sistema proposto decidiu-se pela utilização da tecnologia J2ME, pois esta apresenta as seguintes vantagens:

- a) baseada em java que é uma linguagem orientada a objetos, seguindo os mesmos padrões da plataforma Java;
- b) não há necessidade de se adquirir licenças para utilização, pois é distribuída tanto para fins comerciais ou não;
- c) é uma tecnologia que já está bem consolidada no mercado, tendo vários casos de sucesso já documentados e implementados;
- d) documentação extensa e variada disponibilizada pela Sun Microsystems;
- e) a portabilidade da linguagem java foi um dos principais fatores pela decisão de sua escolha, pois com isso é possível executar os aplicativos em uma gama maior de dispositivos;
- f) existem ferramentas de desenvolvimento (*IDE*) que podem ser integradas à tecnologia J2ME, facilitando assim a criação de aplicativos;

Durante o processo de escolha foram analisadas duas tecnologias existentes no mercado atualmente para desenvolvimento de aplicativos móveis, sendo o Microsoft Visual Studio 5.0 da Microsoft e o Android da Google. A plataforma de desenvolvimento da Microsoft só pode ser adquirida mediante a compra de licenças, e com isso acarretaria em um aumento dos custos financeiros do projeto. Aplicações criadas com o Visual Studio só podem ser executadas em dispositivos que possuem o sistema operacional *Windows Mobile*, e com isso diminuiria a portabilidade da ferramenta desenvolvida. Já o *Android* é uma tecnologia nova, existem poucos casos de sucesso implementados, sendo necessária uma alocação maior de tempo para o estudo e aprendizado, aumentando consideravelmente o risco na execução do projeto.

#### 1.4.2 NetBeans

Entre as principais IDEs Java disponíveis atualmente, foram analisadas para o desenvolvimento da aplicação a ferramenta Eclipse criada pela Eclipse Foundation, o JBuilder de propriedade da Borland Software Corporation e por fim o Netbeans.

NetBeans é uma IDE (*Integrated Development Environment*) para desenvolvimento gratuito de código aberto. Pode ser executado em várias plataformas, incluindo Windows, Linux, Solaris e MacOS. O NetBeans proporciona aos desenvolvedores todas as ferramentas de que necessita para criar aplicações profissionais multiplataforma, corporativas, web e aplicações móveis (NETBEANS, 2007).

NetBeans começou como um projeto estudantil na República Checa (originalmente chamado Xelfi), em 1996. O objetivo era escrever um ambiente de desenvolvimento parecido com a ferramenta RAD (*Rapid Application Development*) Delphi, que era um dos ambientes de desenvolvimento mais utilizados no mundo naquela época. Xelfi foi o primeiro Java IDE escrito em *Java*, com a sua primeira versão lançada em 1997.

Atualmente o NetBeans está na versão 6.1 e pode ser feito o *download* diretamente do seu site gratuitamente.

O NetBeans mostrou-se a ferramenta mais adequada para o desenvolvimento do projeto, e a justificativa da sua escolha é destacada abaixo:

- a) existir suporte total no desenvolvimento de aplicações J2ME;
- b) possuir integrado a sua *IDE* uma ferramenta de desenho de fluxos de navegação da aplicação, facilitando assim seu desenvolvimento;
- c) por ser uma ferramenta com características RAD, tem um ganho significativo em produtividade, diferentemente do Eclipse;
- d) disponível para uso de forma gratuita, diferente da ferramenta JBuilder que exige a compra de licenças para utilizá-la;

### 1.4.3 LWUIT

LWUIT (*LightWeight User Interface Toolkit*) é um conjunto de bibliotecas escritas em Java que auxiliam no desenvolvimento de *software* para dispositivos móveis, onde é possível criar aplicações com um visual mais atraente e consistente. Esta biblioteca é integrada com a plataforma J2ME e foi desenvolvida pela Sun, sendo disponibilizada para o uso gratuito sobre a licença GPLv2 (*General Public License, version 2*) (JAVA.NET, 2008).

O desenvolvimento de aplicativos para dispositivos móveis atualmente é um grande desafio, devido a grande variedade de dispositivos existente no mercado, com diferentes tipos de tecnologias. Uma das grandes dificuldades é a criação de componentes para interface com usuário (lista de menus, telas, cores, fontes, botões, campos para edição, etc), pois os mesmos podem ter comportamentos diferentes dependendo do tipo do dispositivo. Através da utilização da biblioteca LWUIT é possível desenvolver aplicativos para uma gama maior de dispositivos sem que percam suas características projetadas inicialmente (JAVA.NET, 2008).

Para que a aplicação proposta tivesse uma portabilidade maior e também uma aparência mais atraente, decidiu-se por utilizar a biblioteca LWUIT. Outro motivo considerado para sua escolha é devido a sua licença de utilização que é gratuita, diferentemente da biblioteca J2ME Polish (conjunto de bibliotecas semelhantes ao LWUIT) onde é necessário o pagamento de licença, caso se deseje distribuir a ferramenta comercialmente.

#### 1.4.4 Banco de dados Oracle

Banco de dados Oracle é uma sistema gerenciador de banco de dados de propriedade da empresa Oracle Corporation.

A empresa Oracle Corporation foi fundada em agosto de 1977, e se chamava Software Development Lab (SDL). Nessa época era apenas uma empresa de consultoria e tinha como presidente Bob Miner, e como engenheiros de *software* Ed Oates e Bruce Scott. Larry Ellison (atual presidente) trabalhava em um projeto da CIA (*Central Intelligence Agency*) chamado “oracle” e tinha como seu gerente Bob Miner (DEVMEDIA, 2007).

Larry Ellison estava acompanhando um trabalho desenvolvido pela IBM sobre banco de dados relacionais de 1970 que utilizava linguagem SQL, na época chamada *Sequel/2*. Então ele observou uma grande oportunidade que outras companhias do ramo não haviam percebido, quando encontrou uma descrição de um protótipo funcional de um banco de dados relacional e descobriu que nenhuma empresa tinha se empenhado em comercializar esse tipo de tecnologia. Elisson e

seus co-fundadores perceberam então que seria teriam um grande mercado a explorar (ORACLE, 2008).

Conforme Devmedia (2007), a primeira versão comercial do banco de dados Oracle foi vendida em 1979, para força aérea norte americana e foi totalmente escrita utilizando programação Assembly (programação de baixo nível). Sendo que nunca houve o lançamento de uma primeira versão, e sim a venda como segunda versão, pois os seus executivos na época temiam que fosse difícil comercializar um *software* de primeira distribuição. Ainda segundo Devmedia (2007), em 1983 o banco de dados foi reescrito utilizando a linguagem C, tornando-o mais portátil, e foi considerada a primeira versão 32 bits. Em 1997 é lançada a versão 8 já com suporte para aplicações *Web*. Atualmente a última distribuição de banco de dados comercializada pela Oracle é a versão 11g e introduz o conceito aperfeiçoado de *Grid Computin*<sup>3</sup> criada na versão anterior 10g.

Segundo Oracle (2008), a Oracle é a principal fornecedora de tecnologia de banco de dados atualmente e pode ser encontrada em diversas corporações pelo mundo, sendo utilizada em 98% dos escritórios de empresas citadas pela lista "Fortune 100".

Para o desenvolvimento do trabalho proposto foi escolhido o banco de dados oracle versão 8i, pois este já é utilizado pelo sistema ERP SisCon, e possui todas as informações necessárias para a interação com o aplicativo móvel.

#### 1.4.5 Servidor de aplicações Tomcat

O servidor de aplicações Tomcat foi desenvolvido pela Fundação Apache, é responsável pela compilação e execução de aplicativos web. Foi totalmente escrito em linguagem de programação Java e um dos principais requisitos para seu funcionamento é que seja instalado e configurado no mesmo servidor a versão Java J2SE (*Java 2 Standard Edition*).

---

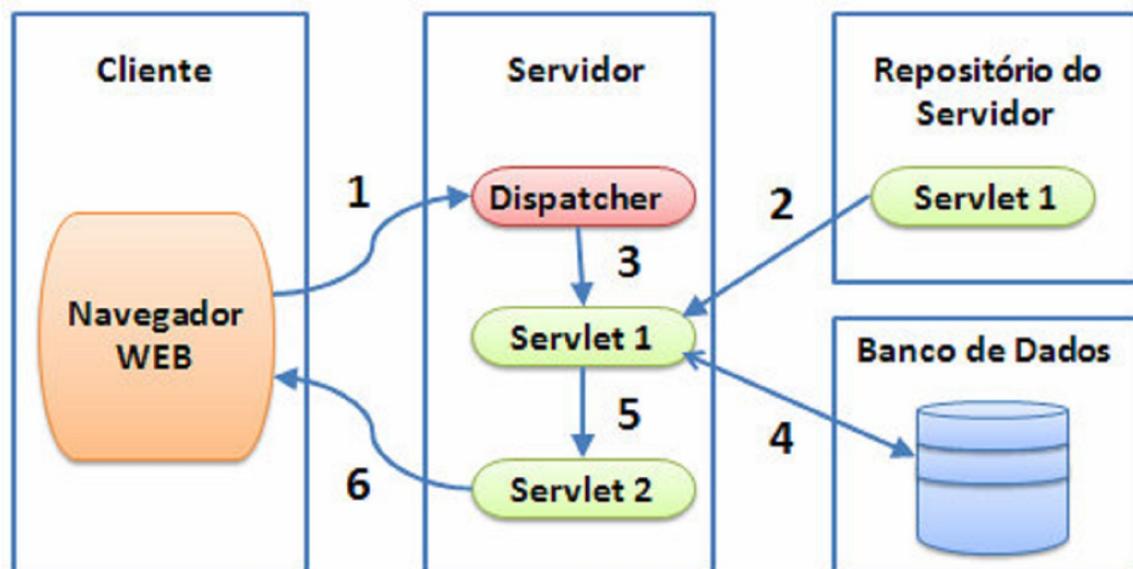
<sup>3</sup> Grid Computing é um novo conceito que explora as potencialidades das redes de computadores, com o objetivo específico de disponibilizar camadas virtuais de processamento de dados que permitem ao usuário ter acesso a aplicações altamente exigentes (CLUBE DO HARDWARE, 2008).

Tomcat pode ser usado livremente tanto para fins comerciais ou não. É considerado um dos principais concorrentes de tecnologias de servidores de aplicações, como o IIS (*Internet Information Services*) da Microsoft que executam páginas ASP (*Active Server Page*).

Segundo Apache (2008), uma das principais funcionalidades do servidor Tomcat é armazenar *Servlets* e páginas *JSP* para que no momento que houver uma solicitação de alguma requisição, o mesmo consiga compilá-las e executá-las, devolvendo como resposta documentos HTML.

Conforme SDN (2008), *Servlets* são classes escritas em Java, e têm a capacidade de receber requisições HTTP, capturando assim seus parâmetros, e com isso sendo capaz de processá-los em qualquer classe Java codificada, devolvendo como resposta páginas em HTML. Uma das suas características principais é a possibilidade de ter acesso a toda família de APIs (*Application Programming Interface*) nativas Java, entre elas pode-se destacar a JDBC API, que provêem acesso a diversos tipos de bancos de dados.

Na figura 4 é representado o funcionamento de um *Servlet* armazenado em um servidor Tomcat.



**Figura 4: Funcionamento de uma requisição HTTP para um *Servlet***  
 Fonte: SDN (2008).

Conforme a Figura 4, um cliente envia uma requisição via HTTP, e com isso o *Dispatcher* carrega o *Servlet* a partir de um repositório (caso o *Servlet1* ainda não

esteja carregado). O Servlet1 recebe a requisição do cliente e acessa o banco de dados, por meio de uma classe codificada em Java. Após os resultados são enviados do Servlet1 para o Servlet2, e este formata os resultados como uma página HTML e o devolve para o cliente.

De acordo com SDN (2008), entre as principais razões para a utilização de Servlets, pode-se destacar:

- a) eficiência: o código de inicialização é carregado somente uma vez pelo servidor de aplicações, aumentando assim sua performance;
- b) persistência: Servlets podem manter estados entre as requisições dos clientes, pois são mantidos em memória até sua finalização;
- c) portabilidade: são escritos em linguagem de programação Java, portanto não há necessidade de serem reescrito caso haja mudança de plataforma;
- d) robustez: tratam com eficiência as exceções;
- e) extensibilidade: por serem orientados a objetos, é possível utilizar conceitos de herança, polimorfismos, aumentando assim a reutilização de códigos;
- f) segurança: por serem executados em um servidor de aplicações, herdam todas as características de segurança já implementadas;

Para o desenvolvimento do trabalho proposto, optou-se por utilizar o servidor de aplicações Tomcat, por ser uma solução muito utilizada e difundida atualmente. Também não necessita aquisição de licenças, portanto não haverá custo financeiro para sua utilização, diferente do servidor IIS da Microsoft, do *WebSphere* da IBM e *Weblogic* da Oracle/BEA Systems. A portabilidade através da capacidade de executar códigos Java também é considerado um dos principais fatores para sua escolha.

## **1.5 Metodologias para desenvolvimento de *software***

Metodologia de desenvolvimento de *software* é um conjunto de regras e práticas recomendadas para a criação de programas de computador. Geralmente

essas regras passam por fases ou passos, que são subdivisões do processo para ordená-lo e melhor gerenciá-lo (SOMMERVILLE, 2000).

As metodologias de desenvolvimento podem ser divididas em metodologias tradicionais e metodologias ágeis. As metodologias tradicionais, também chamadas de pesadas, são divididas em fases bem definidas e geram grande quantidade de documentação e artefatos, geralmente são destinadas a grandes projetos de *software*. O principal *framework* utilizado atualmente para sua implementação é o RUP (*Rational Unified Process*).

Já as metodologias ágeis têm como grande característica a desburocratização no processo de desenvolvimento. São definidos por (Beck, 2001b) doze princípios básicos para que uma metodologia seja considerada ágil, sendo:

- a) a prioridade é satisfazer o cliente através de entregas contínuas e freqüentes;
- b) receber bem as mudanças de requisitos, mesmo em uma fase avançada do projeto;
- c) entregas de softwares funcionando, a partir de algumas semanas ou alguns meses, sempre no menor tempo possível;
- d) as equipes de negócio e de desenvolvimento devem trabalhar juntas diariamente;
- e) manter equipe motivada, fornecendo ambiente, apoio e confiança necessário para realizarem suas tarefas;
- f) a maneira mais eficiente da informação circular através de uma conversa face-a-face;
- g) ter o sistema funcionando é a melhor medida de progresso;
- h) processos ágeis promovem o desenvolvimento sustentável;
- i) atenção contínua a excelência técnica e a um bom projeto aumentam a agilidade;
- j) simplicidade é essencial;
- k) as melhores arquiteturas, requisitos e projetos provêm de equipes organizadas;
- l) em intervalos regulares, a equipe deve refletir sobre como se tornar mais eficaz;

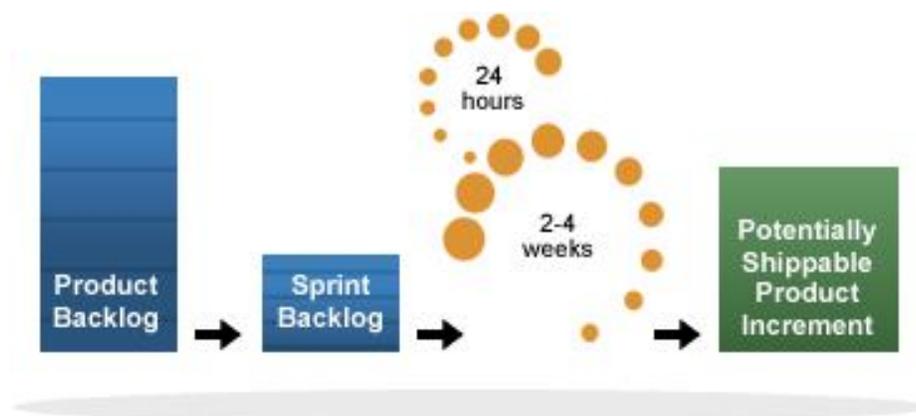
Atualmente as principais metodologias ágeis utilizadas são:

- a) XP (*Extreme Programming*) (BECK, 1999a) ;
- b) FDD (*Feature Driven Development*) (DE LUCA, 2002);
- c) ASD (*Adaptive Software Development*) (HIGHSMITH, 2000) ;
- d) Scrum (SCHWABER e BEEDLE, 2002);

A metodologia escolhida para o desenvolvimento do projeto é a Scrum, agregando alguns artefatos característicos do RUP. Sua escolha se deve principalmente pela necessidade de se gerar muitas atualizações em um curto espaço de tempo. Devido a necessidade de se realizar testes em diferentes tipos de dispositivos, como verificação do *layouts* de telas e funcionamento como um todo da aplicação.

Scrum é uma metodologia ágil focada na entrega do produto, em um menor tempo possível. Ela se enquadra nos doze princípios básicos propostos por (BECK et. al., 2001b).

Na figura 5 é exemplificado o funcionamento da metodologia Scrum.



**Figura 5: Funcionamento da metodologia SCRUM.**  
 Fonte: SCRUM ALLIANCE (2008).

No *Product Backlog* são definidas as listas de requisitos, geralmente feito pelo analista de negócios. Esses requisitos são agrupados de acordo com suas prioridades, apresentando as seguintes informações: descrição do requisito, tempo estimado para sua conclusão e responsável pelo desenvolvimento. Estando definidos os requisitos, os mesmos serão então alocados em *Sprints Backlog*, onde vários destes formam um único *Sprint* ou iteração, tendo uma duração média de duas a quatro semanas. Diariamente acontecem reuniões com toda a equipe de desenvolvimento para que todos sejam informados do processo como um todo e também que sejam expostas possíveis dificuldades encontradas. No final de cada

*Sprint* é gerada uma nova versão do produto e disponibilizado para o cliente, sendo ainda definida uma equipe para acompanhar sua implementação e sua gestão de qualidade (SCHWABER e BEEDLE, 2002).

## **2 SAVM – SISTEMA DE APOIO À VENDAS MÓVEL**

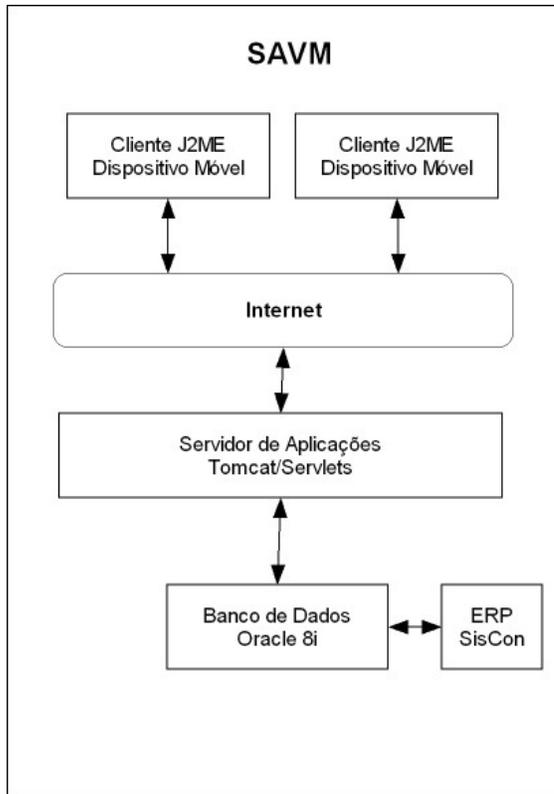
O sistema proposto nesse trabalho é uma ferramenta que será integrada em qualquer dispositivo móvel que tenha suporte a linguagem de programação J2ME. É uma aplicação direcionada ao setor de vendas para empresas do ramo calçadista, sendo possível com ela realizar consultas, empenhar produtos disponíveis em estoque, simular preços de vendas, receber informações de promoções e queimas de estoque, analisar restrições cadastrais. Tendo como características principais, a fácil utilização por parte do usuário, realização de operações para auxílio na tomada de decisões, e principalmente agilidade na resposta ao cliente.

A aplicação desenvolvida é composta de uma interface que é instalada e executada em um dispositivo móvel, e é utilizado para que o usuário informe os dados, solicite o processamento e então receba os resultados. O tráfego de requisições e respostas é realizado por meio de redes sem fio (*Wireless*, redes de comunicação de telefonia celular), ou qualquer outra que seja possível trafegar informações utilizando protocolo HTTP.

As requisições solicitadas pela interface cliente são recebidas pelo servidor de aplicações utilizando o protocolo HTTP. No servidor são processadas essas requisições por meio de classes escritas em código Java (Servlets). A resposta é de responsabilidade dos Servlets, pois estes formatam os dados e as devolvem para o cliente que solicitou.

O sistema comercial SisCon é o sistema ERP utilizado para integração com a aplicação desenvolvida. Essa aplicação fornece todos os dados necessários, para que o cliente J2ME consiga utilizá-los. O ambiente comum de acesso entre o cliente e o ERP SisCon é o banco de dados;

Na figura 6 é apresentada a arquitetura do sistema SAVM.



**Figura 6. SAVM - Arquitetura**

O sistema ERP SisCon é um sistema de gestão empresarial de propriedade da empresa Dws Informática Com.Imp.Exp Ltda. É utilizado principalmente por empresas do ramo coureiro-calçadista.

O sistema SisCon é responsável pelo gerenciamento e o fornecimento das informações para o SAVM. Foram necessárias adaptações no sistema ERP, sendo solicitadas e implementadas pela empresa proprietária. Abaixo constam as modificações realizadas:

- a) criação de um módulo para gerenciar os usuários que irão se autenticar e ter acesso as informações;
- b) criação de um módulo para registrar as mensagens para os vendedores;
- c) criação de um módulo para gerenciamento dos produtos empenhados;
- d) criação das *views* (tabelas virtuais ou visões de banco de dados):
  - v\_estoquedisponivel – estoque disponível online;
  - v\_cores – cadastro de cores;

- v\_cadastroproduto – cadastro dos produtos;
- v\_tabelapreco – tabela de preço dos produtos;
- v\_consultapedido – dados dos pedidos;
- v\_clientes – cadastro de clientes;
- v\_pendenciasfinanceiras – pendências financeiras de clientes;

Abaixo são apresentadas as telas referentes a cada módulo adaptado e operacionalizado pelos usuários do sistema SisCon.



The image shows a screenshot of a software window titled "SisCon - Gerenciamento Acesso Vendedores". The window contains a login form with the following elements:

- Usuário:** A text input field containing the value "LJOA025".
- Senha:** A text input field containing masked characters "XXXXXXXXXX".
- Vendedor:** A dropdown menu with the selected value "JOÃO CARLOS DA SILVA".
- Checkboxes:** Two checkboxes are checked: "Acesso Mobile" and "Acesso Web".
- Buttons:** Two buttons are located at the bottom: "Confirmar" and "Sair".

**Figura 7: SisCon - Gerenciamento de acesso dos vendedores**

Através do gerenciamento de acesso dos vendedores, é possível criar novos usuários (vendedores) com seus respectivos login e senha. É possível ainda habilitar/desabilitar seu acesso aos dados da empresa.

SisCon - Mensagens Vendedores - Mobile

Vendedor: <<TODOS OS VENDEDORES>>  
<<TODOS OS VENDEDORES>>  
JOÃO CARLOS DA SILVA  
PAULO RIBEIRO SILVEIRA  
SANTOS REPRESENTAÇÕES

Data:

Mensagem

PROMOÇÃO CALÇADOS DE SEGURANÇA REF: MASC-4056 E FEM-7882. PROMOÇÃO VÁLIDA ATÉ DIA 20/11/2008.

Enviar Mensagem Sair

**Figura 8: SisCon – Registro de mensagens para vendedores**

Com o módulo registro de mensagens é possível que o operador do sistema SisCon cadastre qualquer tipo de mensagem para um vendedor específico ou para vários vendedores, para que recebam através do SAVM.

Vendedor: JOÃO CARLOS DA SILVA

Referência	Variação	Cor	Número	Empenhado(Pares)	Preço(Par)	Comissão
MASC-4467	500	PRETO	40	10	34,58	4,00%
FEM-4478	501	BRANCO	38	45	32,45	3,50%
BV52	500	PRETO	40	15	45,00	4,00%
LUVA-VAO	0	NATURAL	PP	45	15,56	0,00%

Excluir Empenho Sair

Figura 9: SisCon – Produtos empenhados

Através do módulo produtos empenhados, é possível que os operadores do sistema SisCon consigam gerenciar todos os produtos reservados pelos vendedores. Pode ser feito a exclusão de um empenho, alteração de um preço de venda ou alteração do percentual de comissão de um vendedor.

### 3 DESENVOLVIMENTO DO SISTEMA SAVM

#### 3.1 Metodologia

Para o desenvolvimento do sistema proposto foi utilizada a metodologia ágil Scrum, agregando artefatos característicos da metodologia RUP. Os artefatos combinados com a metodologia escolhida foram: diagrama de casos de uso, diagrama de atividade, diagrama MER (Modelo Entidade – Relacionamento).

Todos os requisitos gerais do sistema foram definidos no *Product Backlog*. A partir dessa definição foram criados seis *Sprints* principais, sendo que após a sua conclusão já foi possível disponibilizar uma versão da aplicação para execução de testes. Ainda no *Product Backlog* foi definido o diagrama de casos de uso da aplicação e o diagrama MER. Cada *Sprint* possui um diagrama de casos de usos detalhado e um diagrama de atividades pertinente a ele. Por fim, cada *Sprint* possui um *Sprint Backlog*, onde contém informações para o seu desenvolvimento. Cada *Sprint* teve uma duração média de trinta dias;

### 3.1.1 Descrição dos requisitos (*Product Backlog*)

A seguir serão descritos os requisitos necessários para a construção da aplicação.

- a) solicitação junto à empresa proprietária do sistema SisCon, do usuário e senha para conexão no banco de dados Oracle 8i ;
- b) o sistema deve fornecer uma interface para que o operador informe seu nome de usuário e senha para autenticação (autenticação);
- c) após ter feito a autenticação com sucesso, o sistema deve apresentar ao usuário uma lista de menus para que possa acessar o modulo desejado;
- d) a aplicação irá disponibilizar uma tela de mensagens, para que o operador consiga verificá-las (mensagens);
- e) o sistema deve mostrar uma tela para que o usuário possa efetuar o empenho do estoque disponível (empenho);
- f) o sistema deve mostrar uma tela para que o operador possa efetuar consultas de pedidos de seus clientes (consulta de pedidos);
- g) a aplicação deve disponibilizar uma tela para que seja possível o operador verificar possíveis pendências financeiras junto ao cadastro da empresa (análise de restrições);
- h) o sistema deve mostrar uma tela para que o usuário consiga simular preços de vendas (simular vendas);

Na figura 10 é apresentado o diagrama de casos de uso do sistema SAVM

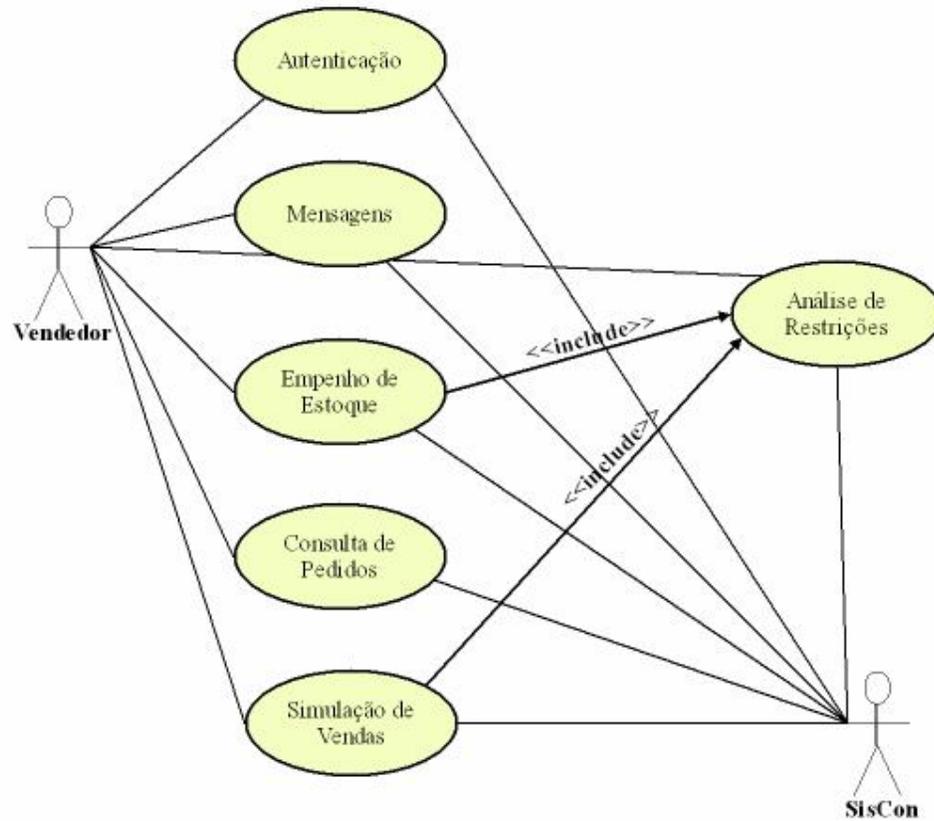


Figura 10: SAVM - Diagrama de casos de uso

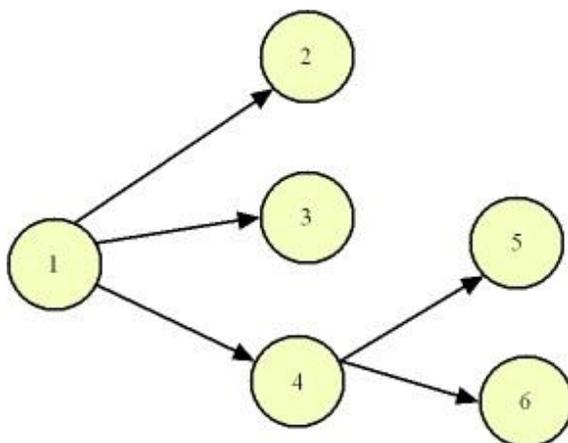


Figura 11: SAVM - Priorização do caso de uso

- 1 – Autenticação
- 2 – Mensagens
- 3 – Consulta de pedidos
- 4 – Análise de restrições
- 5 – Empenho de estoque
- 6 – Simulação de vendas

Na figura 12 é apresentado o diagrama MER (Modelo Entidade-Relacionamento).

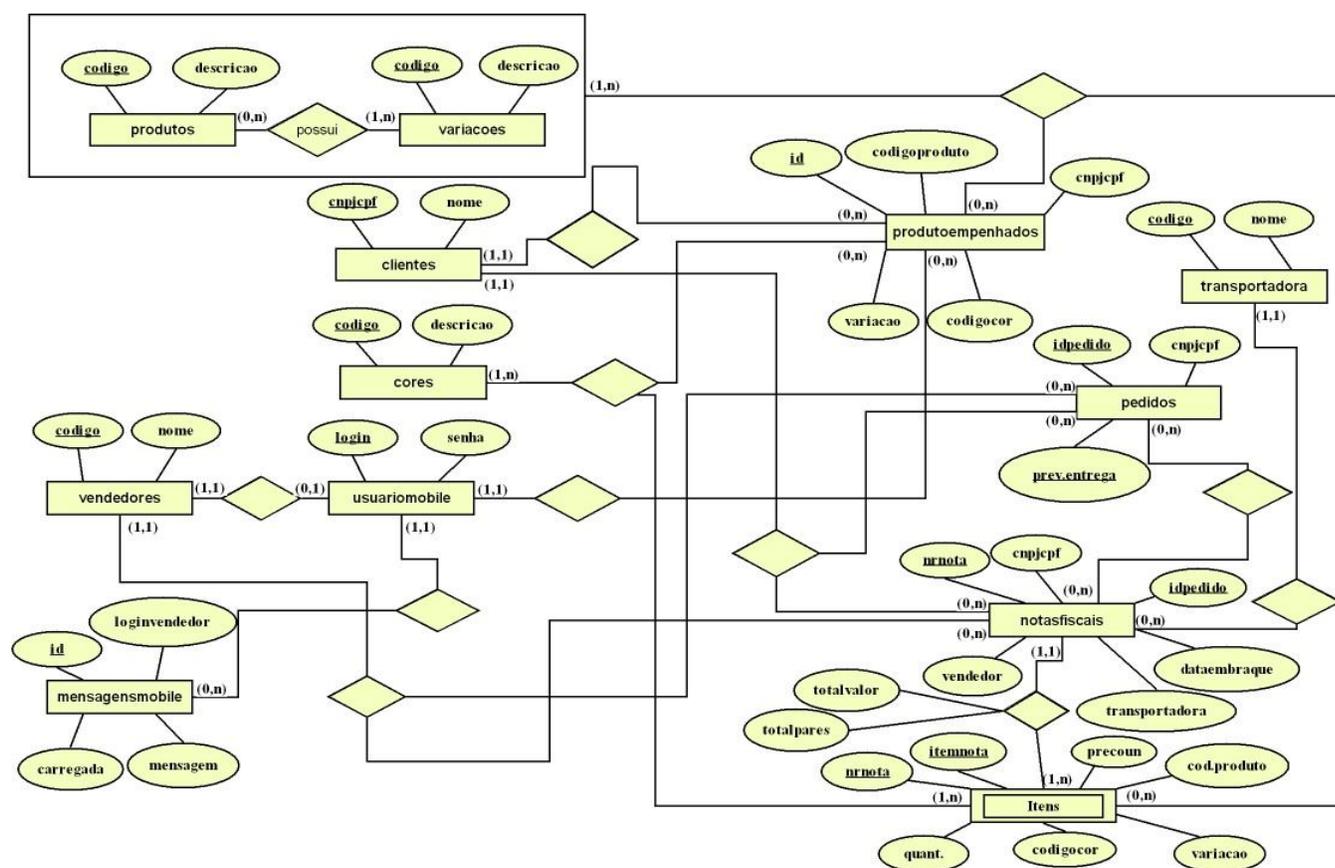


Figura 12: SAVM – MER (Modelo Entidade-Relacionamento)

### 3.1.2 Detalhamento dos casos de uso (*Sprints*)

Cada *Sprint* corresponde a um módulo do SAVM, sendo que após sua conclusão será disponibilizado uma versão para testes.

Cada módulo do sistema proposto irá se comunicar com um Servlet armazenado no servidor de aplicações Tomcat.

Na figura 13 é ilustrado um diagrama de blocos que representa a comunicação entre os módulos do dispositivo móvel (SAVM) e os servlets (ServidorSAVM).

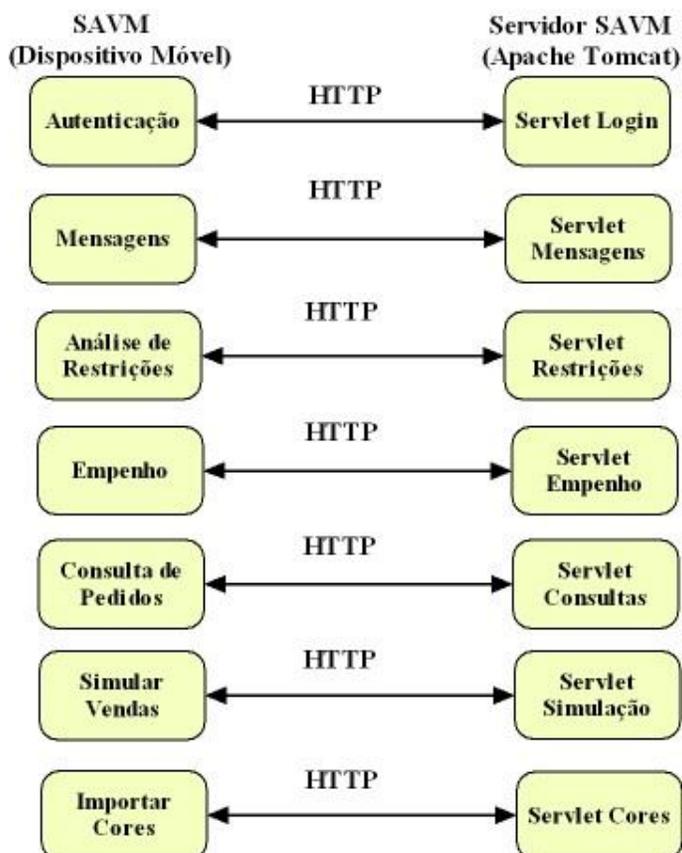


Figura 13: Diagrama de blocos – Servlets

Abaixo será detalhada cada tarefa (*Sprint Backlog*), sendo desenvolvida na seguinte ordem: autenticação, mensagens, consulta de pedidos, análise de restrições, empenho de estoque, simulação de vendas.

### 3.1.2.1 Autenticação

O módulo Autenticação deve apresentar as seguintes características:

- no momento que for iniciada a aplicação no dispositivo móvel, o sistema deve mostrar uma tela com dois campos de texto editáveis para que o usuário possa informar o nome de usuário e a senha;
- na tela login o sistema deve apresentar dois botões, sendo o botão login para que o usuário possa efetuar a autenticação e o botão sair para que possa encerrar a aplicação;
- após pressionar o botão login, o sistema deve iniciar uma conexão HTTP com o ServletLogin para efetuar a autenticação;

A seguir na figura 14 é apresentado um diagrama detalhando o caso de uso autenticação.

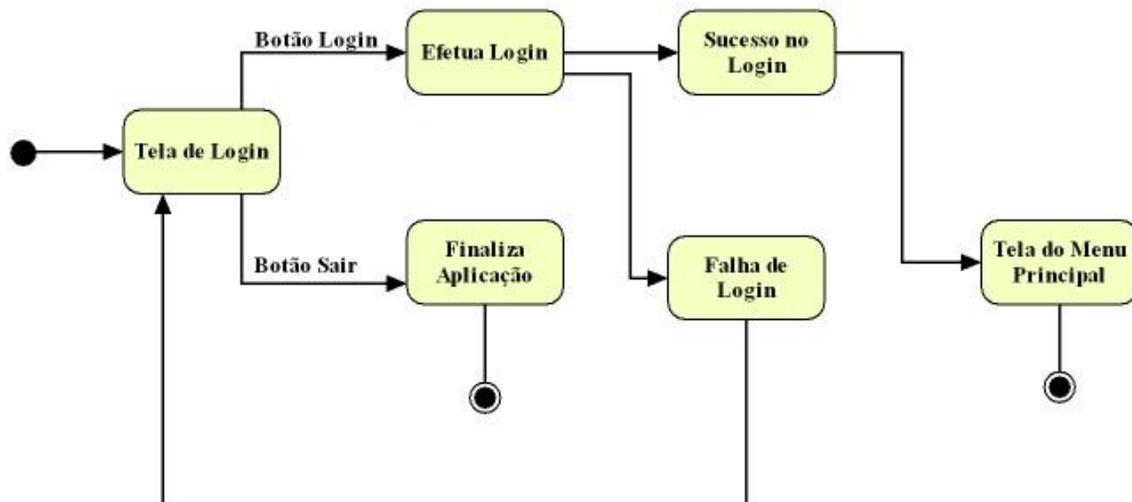


Figura 14: Autenticação – Caso de uso detalhado

Na figura 15 é apresentado o diagrama de atividades do módulo autenticação.

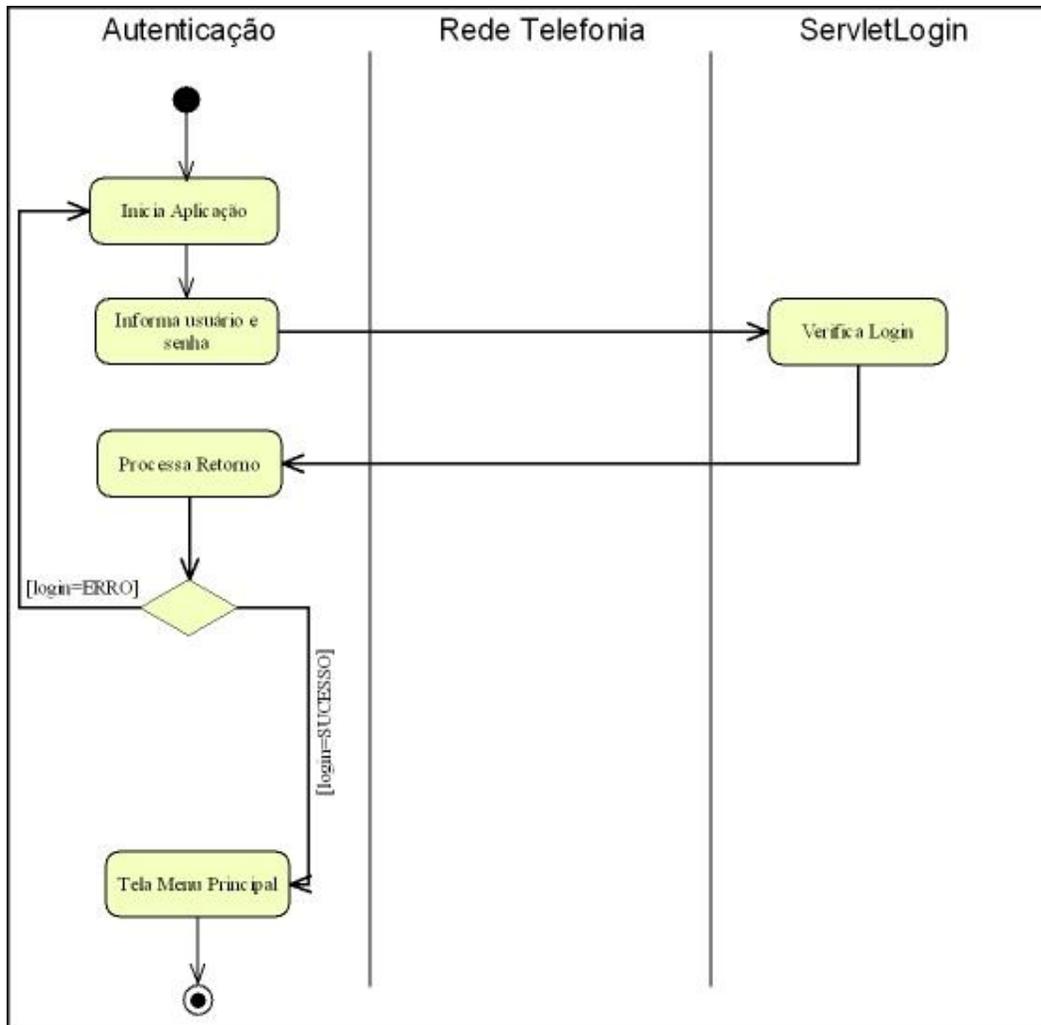


Figura 15: Autenticação - Diagrama de atividades

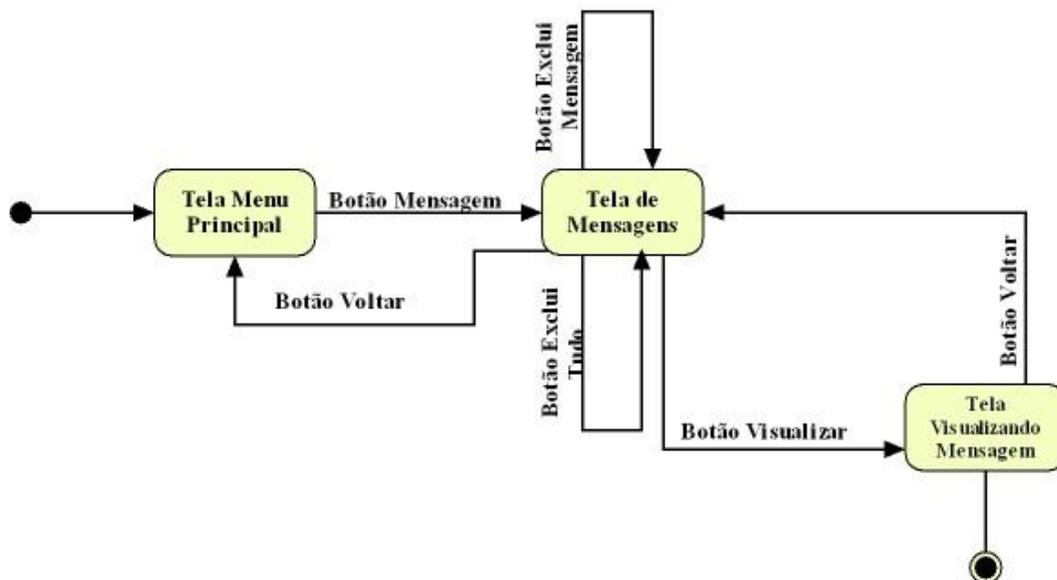
### 3.1.2.2 Mensagens

O módulo Mensagens deve apresentar as seguintes características:

- a) no momento que é pressionado o botão mensagens no menu principal, a aplicação deve abrir uma conexão HTTP com o ServletMensagens e então recuperar as mensagens novas que sejam pertencentes ao usuário logado;
- b) a aplicação somente deverá recuperar as mensagens novas e então gravar no sistema de arquivos do dispositivo móvel;

- c) o sistema deve apresentar a tela mensagens, listando as novas mensagens juntamente com as antigas existentes no dispositivo;
- d) o sistema deve apresentar um botão para visualizar uma mensagem, um botão para excluir uma única mensagem e um botão para excluir todas as mensagens do dispositivo móvel;
- e) pressionando o botão visualizar, o sistema deve apresentar a tela visualizando mensagens para que apresente a mensagem por completo. Nessa mesma tela a aplicação deve apresentar um botão voltar, para que seja possível voltar à tela que mostra todas as mensagens;
- f) o sistema deve apresentar um botão voltar, para que volte ao menu principal;

Na figura 16 é apresentado o caso de uso detalhado do módulo mensagens.



**Figura 16: Mensagens – Caso de uso detalhado**

Na figura 17 é apresentado o diagrama de atividades que representa o recebimento das mensagens.

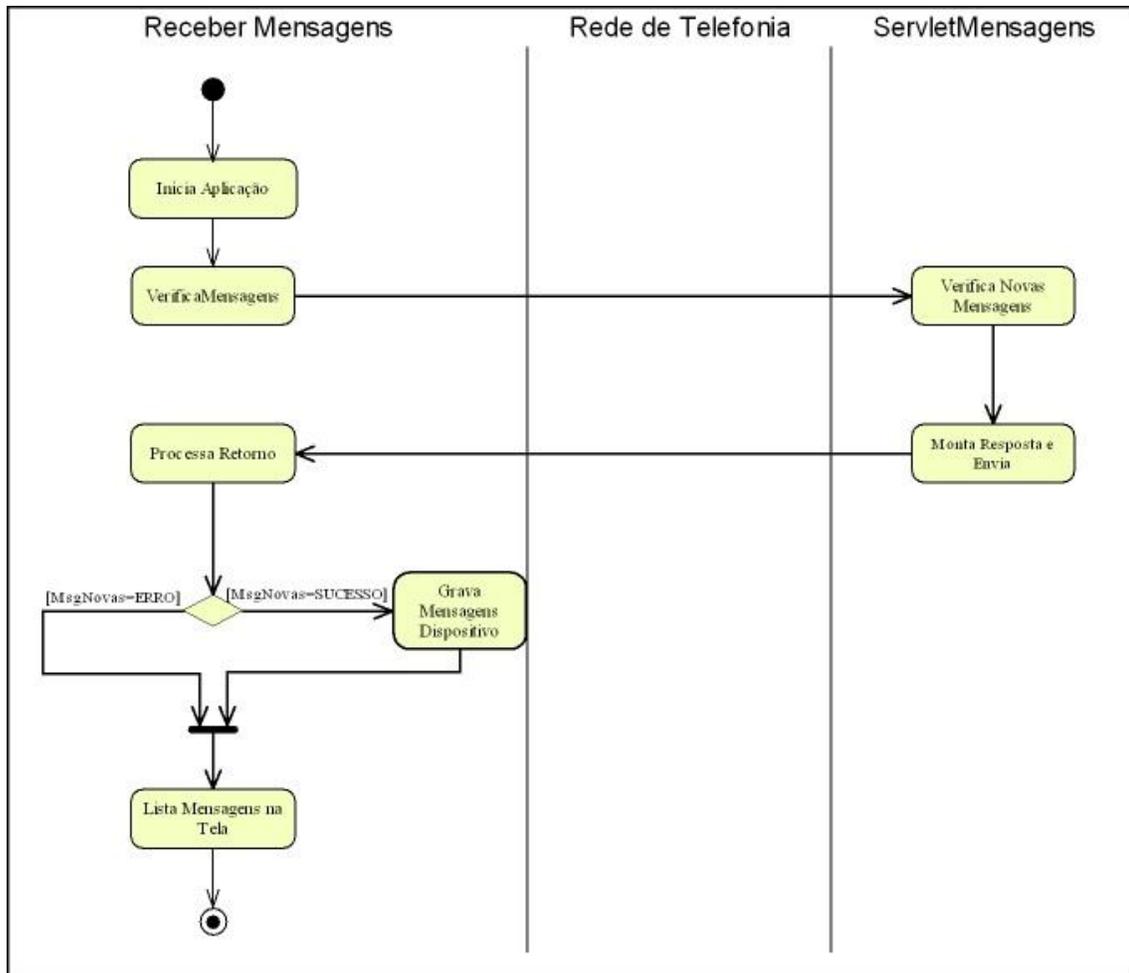


Figura 17: Receber Mensagens - Diagrama de atividades

### 3.1.2.3 Consulta de pedidos

O módulo Consulta de Pedidos deve apresentar as seguintes características:

- no momento que for pressionado o botão consulta pedidos no menu principal, o sistema deve apresentar a tela consultas pedidos;
- na tela consulta pedidos deve constar os seguintes campos textos editáveis juntamente com o(s) botão(e)s:
  - campo nota fiscal;
  - campo pedido;
  - campo Cpf/Cnpj;

- campo quantidade registros;
  - botão pesquisar;
- c) caso o usuário informe a nota fiscal ou o número do pedido, o cpf ou cnpj, o operador poderá pressionar o botão pesquisar, sendo que após esta ação o sistema deve abrir uma conexão HTTP com o ServletConsultas para que seja efetuada a consulta;
- d) após realizada a consulta por cpf ou cnpj o sistema deve apresentar a tela visualiza pedidos com uma lista que contenha as notas fiscais/pedidos ordenadas por data de faturamento decrescente, sendo que o número de registros listados seja limitado de acordo com o valor informado no campo quantidade registros. A aplicação deve mostrar o botão seleccione para que o operador consulte uma nota/pedido específico;
- e) após o operador consultar por pedido/nota ou selecionar uma nota específica o sistema deve apresentar as informações na tela visualiza dados pedido;
- f) as seguintes informações devem ser apresentadas: cpf/cnpj do cliente, nome do cliente, numero do pedido, nota fiscal, data previsão de entrega, data do embarque, transportadora, quantidade total de pares, valor total do pedido;
- g) o sistema deve apresentar um botão voltar em cada tela do módulo consulta de pedidos;

Na Figura 18 é apresentada o caso de uso detalhado do módulo consulta de pedidos.

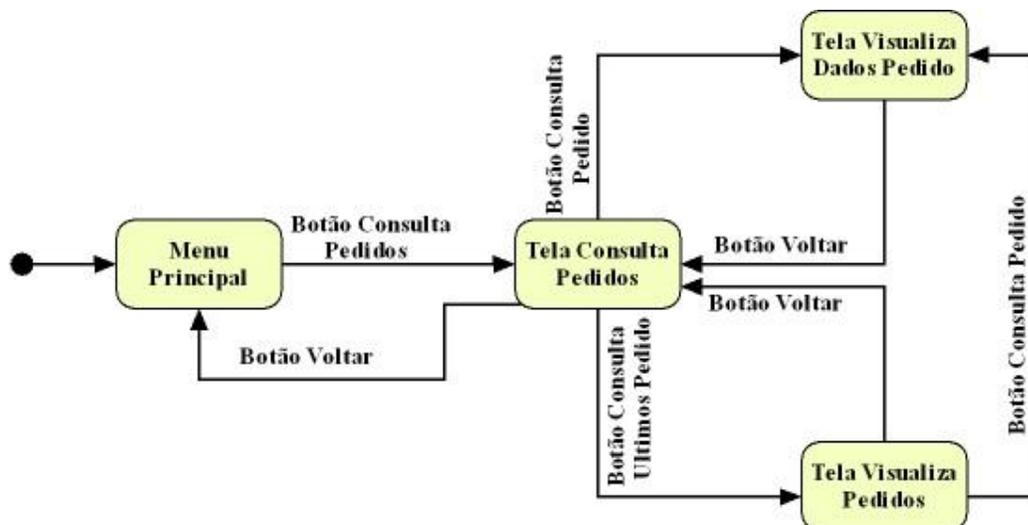


Figura 18: Consulta de Pedidos – Caso de uso detalhado

Na figura 19 é apresentado o diagrama de atividades do módulo consulta pedidos.

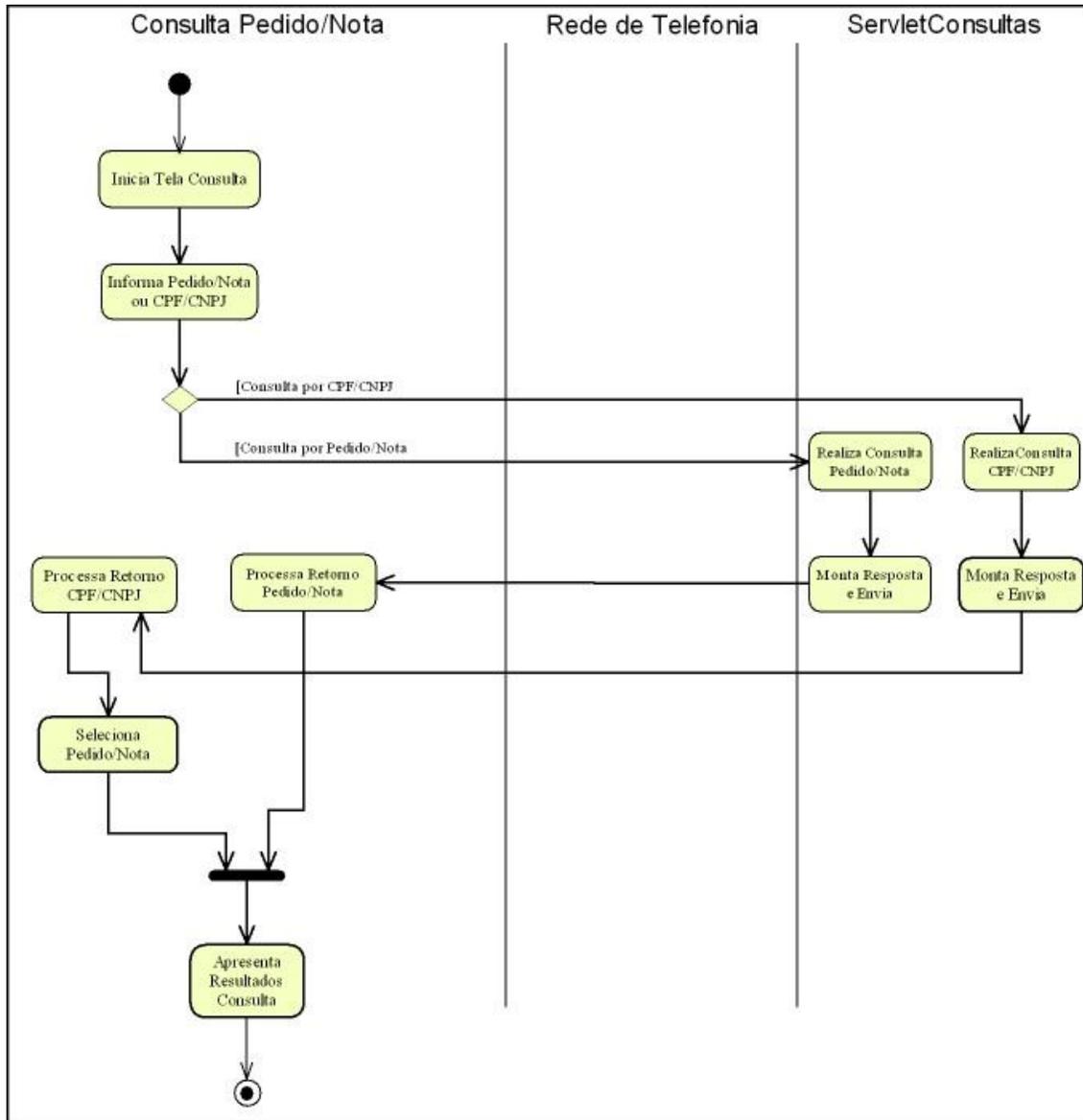


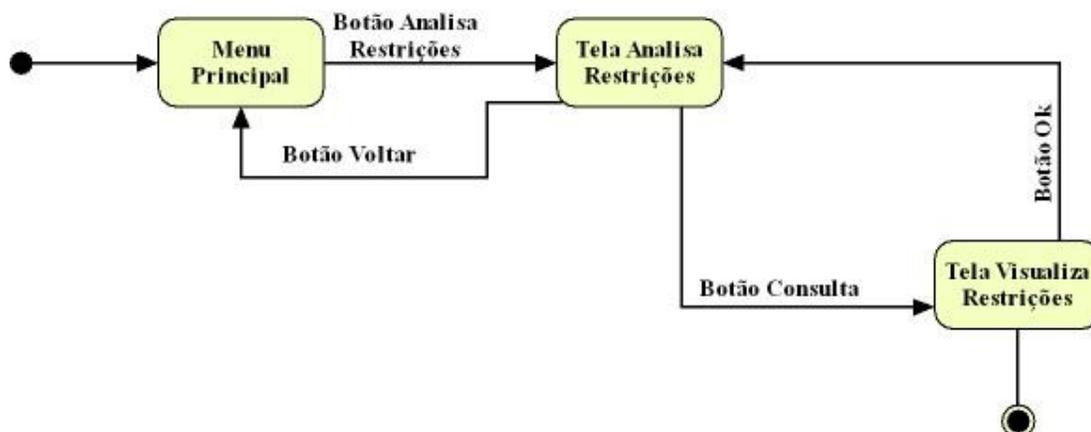
Figura 19: Consulta pedidos – Diagrama de atividades

### 3.1.2.4 Análise de restrições

O módulo Análise de Restrições deve apresentar as seguintes características:

- a) no momento que for pressionado o botão análise de restrições no menu principal, o sistema deve apresentar a tela analisa restrições, onde deve constar um campo texto para informação do cpf ou do cnpj do cliente;
- b) deve mostrar uma botão consulta;
- c) no momento que for pressionado o botão consulta, o sistema deve abrir uma conexão HTTP com o ServletRestrições para que seja efetuada a consulta;
- d) após efetuada a consulta, o sistema deve apresentar uma tela com os resultados;
- e) deve constar um botão voltar em cada tela módulo análise de restrições;

Na figura 20 é apresentado o caso de uso detalhado do módulo análise restrições.



**Figura 20: Analisa Restrições – Caso de uso detalhado**

Na figura 21 é apresentado o diagrama de atividades do módulo análise de restrições.

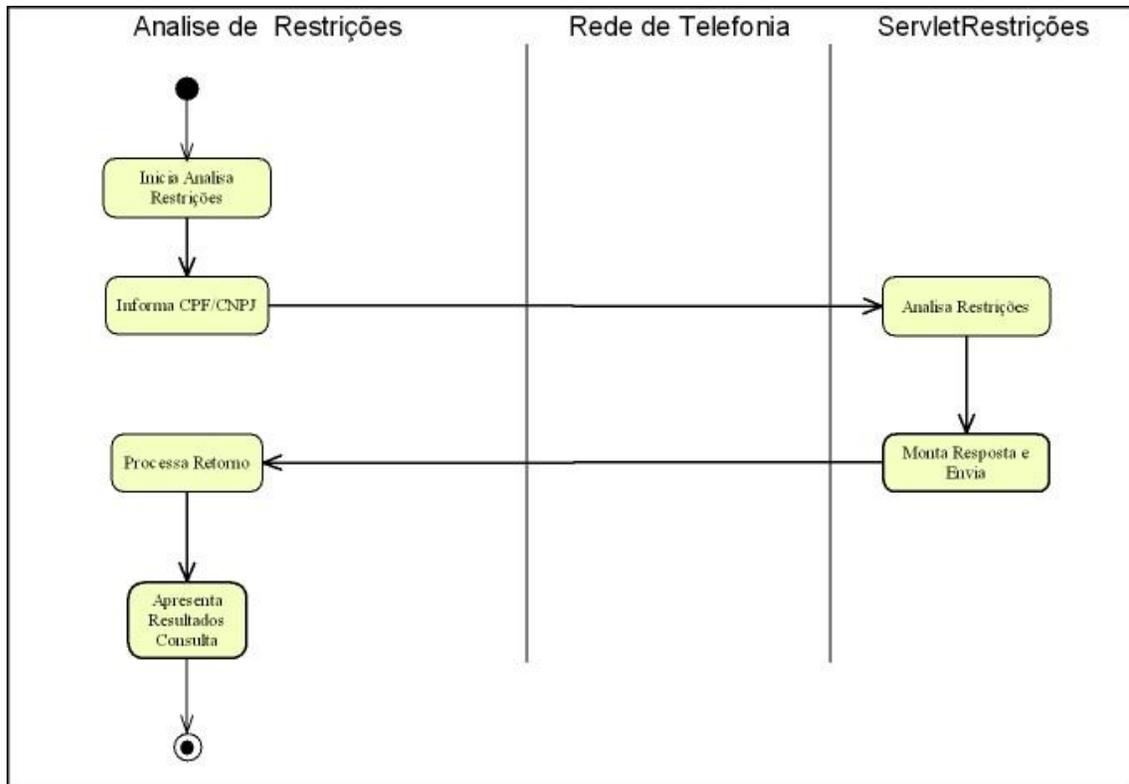


Figura 21: Análise de Restrições – Diagrama de atividades

### 3.1.2.5 Empenho de estoque

O módulo Empenho de Estoque deve apresentar as seguintes características:

- no momento que for pressionado o botão empenho no menu principal, o sistema deve apresentar a tela empenho estoque;
- na tela empenho estoque deve constar os seguintes campos editáveis e os seguintes botões:
  - campo referência – código do produto para venda;
  - campo variação – código da variação do produto;
  - campo cor – descrição da cor do produto;
  - campo número – numeração do produto exemplo(33, 34,35...);
  - botão verifica estoque – consulta estoque disponível;
  - botão importar cores – importa as cores do cadastro;
  - botão voltar - volta ao menu principal;

- c) pressionando o botão verifica estoque o sistema deve abrir uma conexão HTTP com o ServletEmpenho e então efetuar uma consulta para verificar a disponibilidade de estoque e então apresentar a tela seleção de produtos com os dados retornados;
- d) na tela seleção produtos a aplicação deve mostrar as seguintes informações:
- referência consultada, preço unitário;
  - uma tabela com os dados do produto em estoque: número, quantidade em estoque, total empenhado;
- e) na tela Seleção Produtos deve constar os seguintes botões:
- botão empenhar – seleciona um determinado produto da lista e então empenha;
  - botão nova consulta – retorna para a tela empenho estoque para uma nova consulta
  - botão concluir – abre a tela concluir empenho;
- f) na tela concluir empenho devem constar as seguintes informações e os seguintes botões:
- um resumo dos produtos empenhados: referência, cor, preço por par;
  - botões: grade, encerrar, excluir produto, nova consulta;
  - botão excluir produto – exclui produto selecionado;
- g) na tela concluir empenho quando o botão grade for pressionado o sistema deve abrir a tela grade empenhada;
- h) na tela concluir empenho quando o botão encerrar for pressionado o sistema deve abrir a tela finalizar empenho;
- i) na tela finalizar empenho deverão constar os campos, cpf/cnpj e os botões finalizar, cancelar;
- j) na tela finalizar empenho quando for pressionado o botão finalizar, o sistema deve abrir uma conexão HTTP com o ServletEmpenho;
- k) no ServletEmpenho deverá haver uma consulta para verificar possíveis pendências financeiras do cliente;

- l) no ServletEmpenho, no momento do registro do empenho, o sistema deve fazer uma nova consulta de estoque disponível, para verificar se o produto escolhido ainda possui estoque;
- m) caso ocorra sucesso na finalização do empenho o sistema deve mostrar uma mensagem de sucesso;
- n) para cada tela do módulo deve constar um botão voltar;

Na figura 22 é apresentado o caso de uso detalhado do módulo empenho;

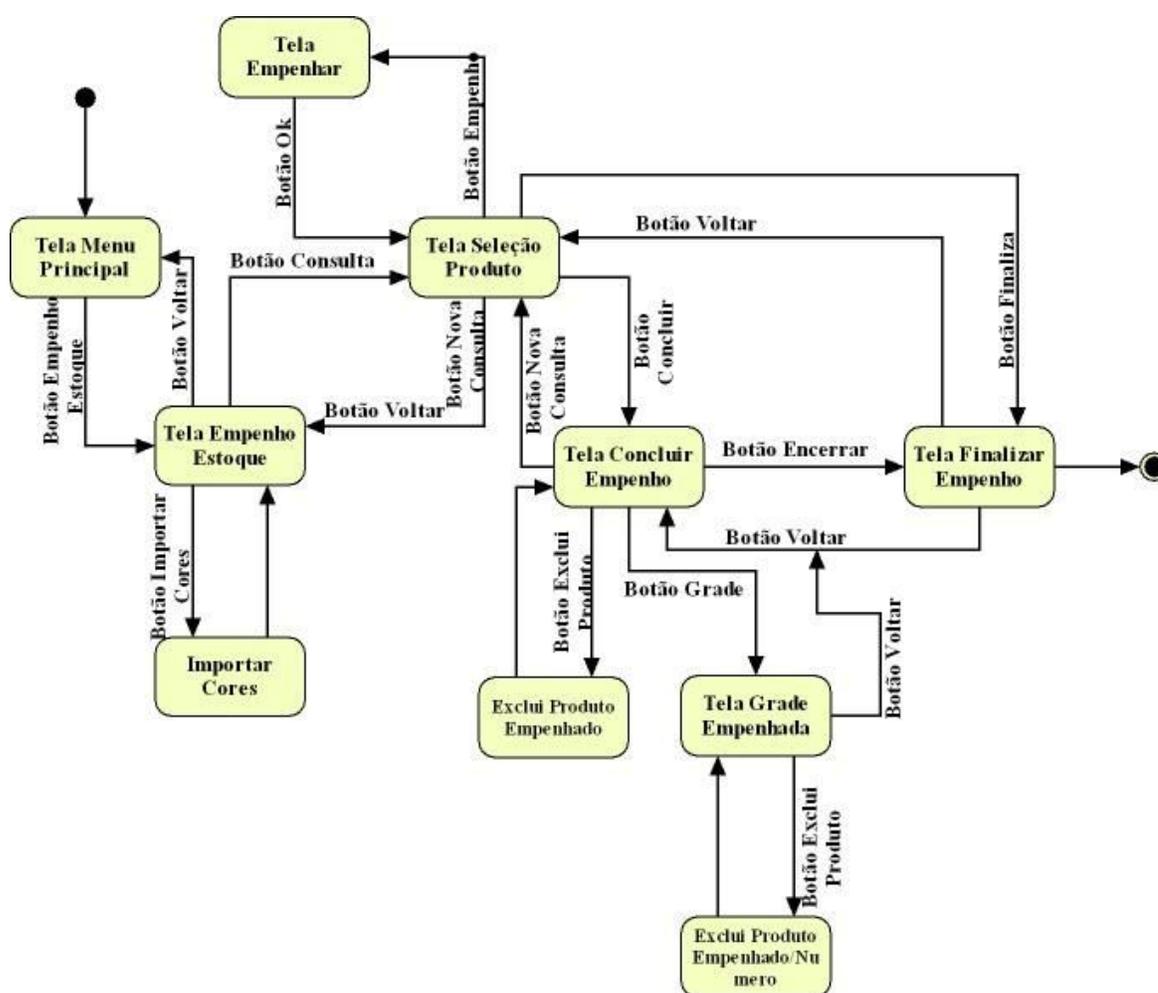


Figura 22: Empenho Estoque – Caso de uso detalhado

Na figura 23 é apresentado o diagrama de atividades, ilustrando a atividade Verifica Estoque.

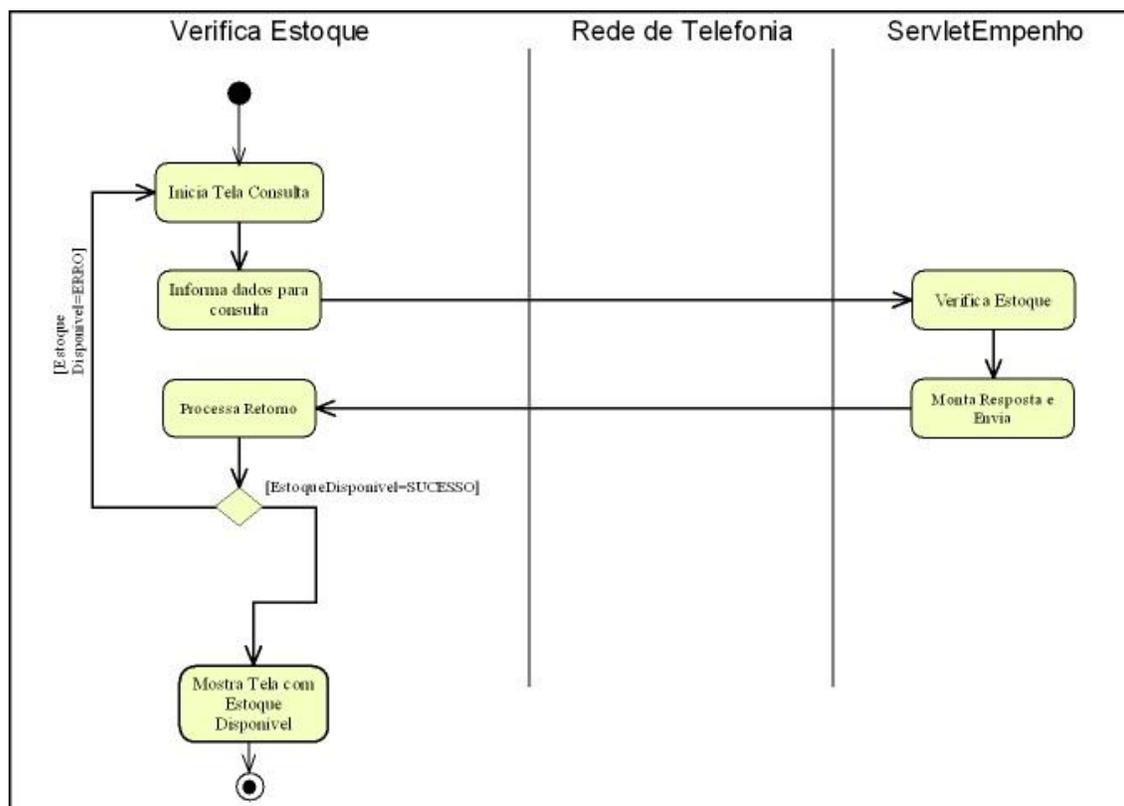
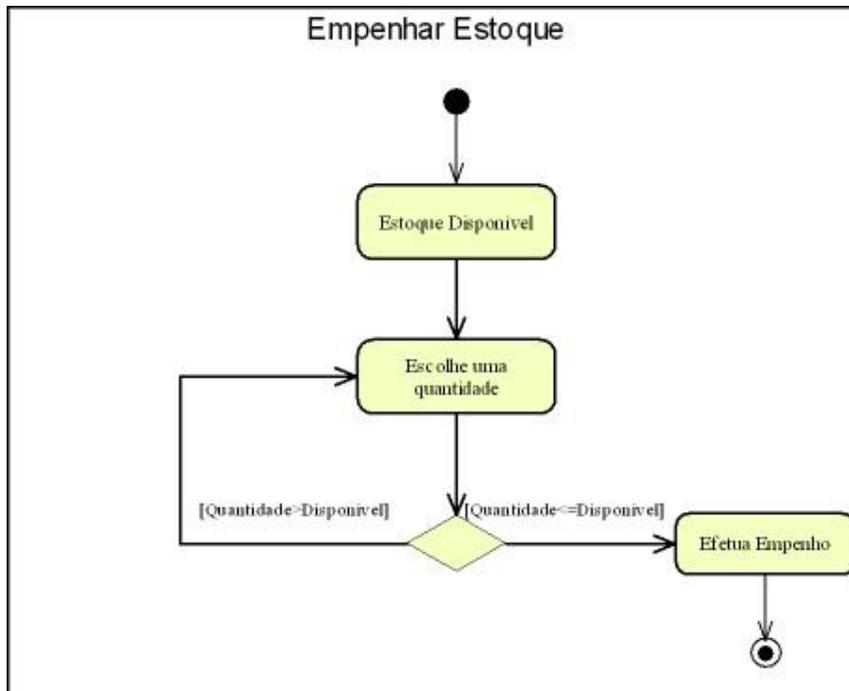


Figura 23: Verifica Estoque – Diagrama de atividades

Na figura 24 representa o diagrama de atividades do módulo empenho ilustrando a atividade Empenhar Estoque.



**Figura 24: Empenhar Estoque – Diagrama de atividades**

A figura 25 representa o diagrama de atividades do módulo empenho ilustrando a atividade Finaliza Empenho.

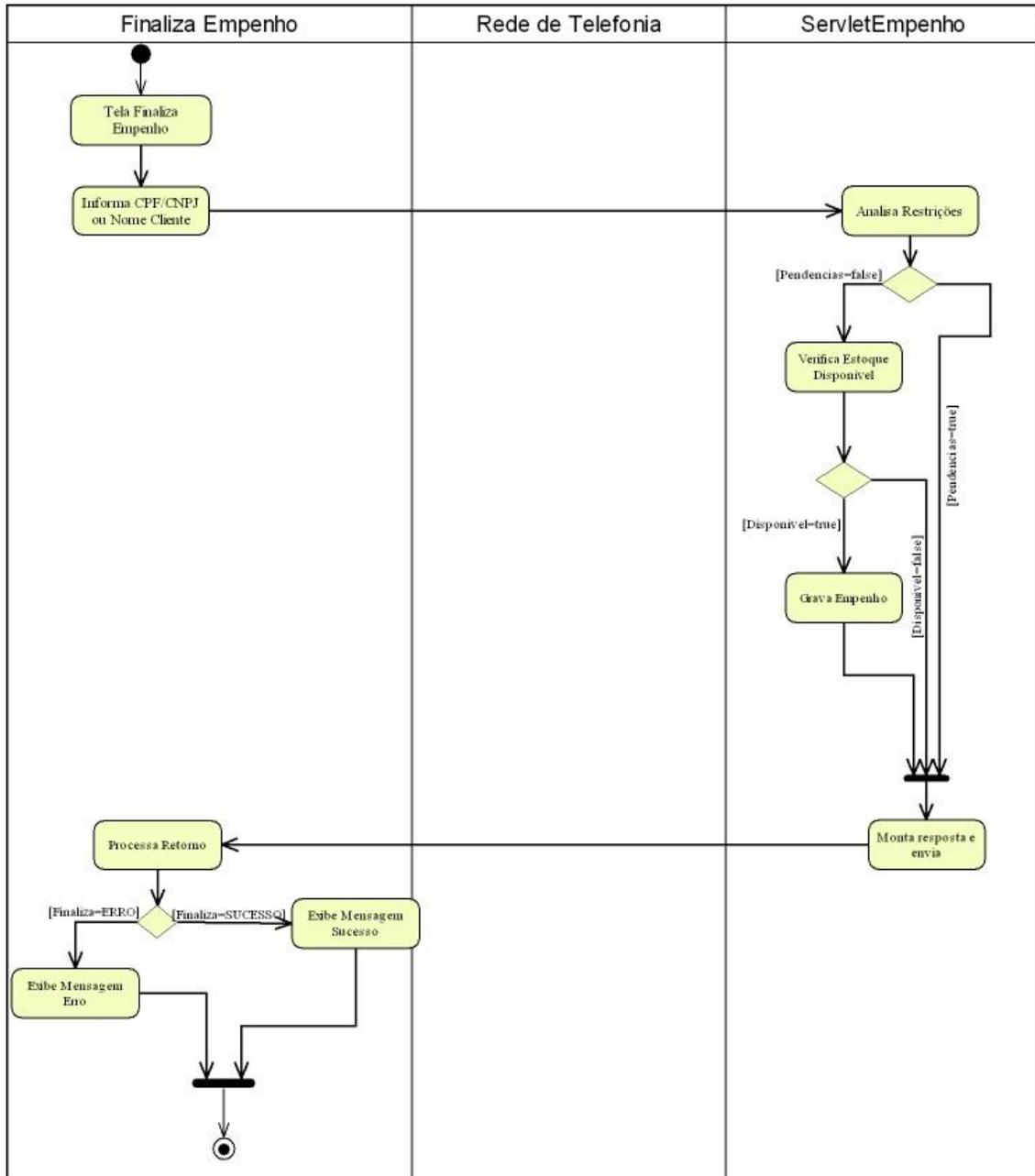


Figura 25: Finaliza Empenho – Diagrama de atividades

### 3.1.2.6 Simulação de vendas

O módulo Simulação de Vendas deve apresentar as seguintes características:

- a) no momento que for pressionado o botão simulação no menu principal do sistema, a aplicação deve apresentar a tela simulação venda;
- b) na tela simulação venda deve constar os seguintes campos editáveis e os seguintes botões:
  - campo referência – referência do produto;
  - campo variação – variação do produto;
  - botão consultar – consulta a tabela de preço do produto;
  - botão simular – simula um preço de venda do produto;
- c) se o operador pressionar o botão consultar, o sistema deve iniciar uma conexão HTTP com o ServletSimulação e efetuar a consulta, após deve mostrar a tela resultado consulta com uma tabela contendo as seguintes informações: preço por par, percentual de comissão e quantidade base;
- d) se o usuário pressionar o botão simular o sistema deve apresentar a tela realiza simulação;
- e) na tela realiza simulação a aplicação deve mostrar os campos: quantidade compra; desconto na comissão; botão simular;
- f) para cada tela do módulo simulação deverá constar um botão voltar;

Na figura 26 é apresentado o caso de uso detalhado do módulo simulação;

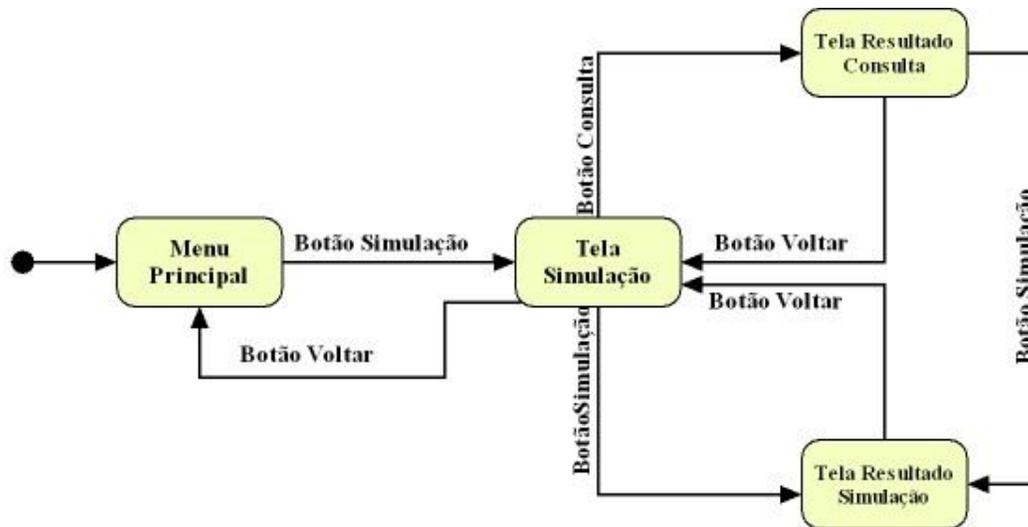


Figura 26: Simulação de vendas – Caso de uso detalhado

Na figura 27 é apresentado o diagrama de atividades do módulo simulação vendas.

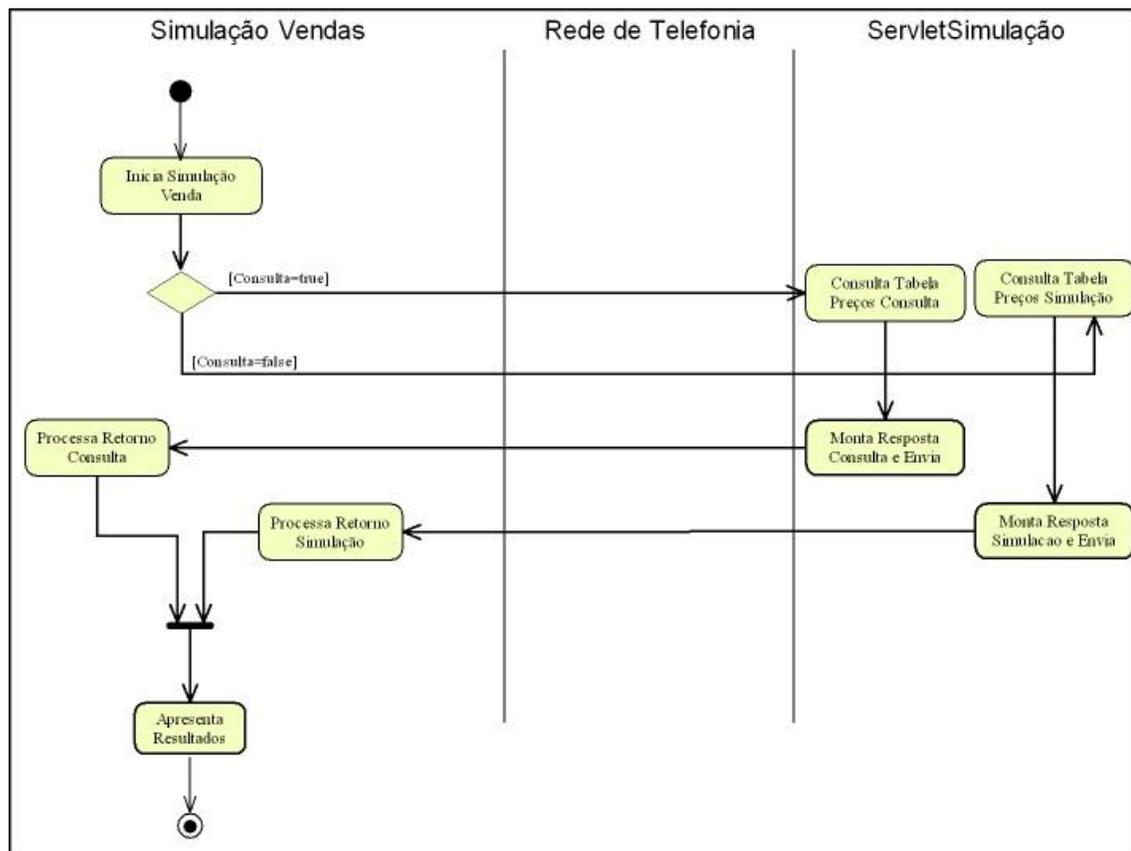


Figura 27: Simulação de vendas-Diagrama de atividades

### 3.1.3 Implementação

Pode-se dividir a codificação do sistema desenvolvido em duas etapas principais: codificação no dispositivo móvel e no servidor.

No dispositivo móvel foi utilizado quase na sua totalidade a biblioteca LWUIT para codificação. Através dessa biblioteca foi possível criar telas, campos textos, botões, listas, menus etc.

Abaixo é apresentado um método que cria uma tela padrão no dispositivo móvel.

<b>CÓDIGO PARA CRIAÇÃO DAS TELAS</b>
<pre>//método para criar uma tela private void montaFormularioSimulacao() {     //verifica se já não está na memoria     if (frmSimulacao == null){         //cria comando ok         cmd_Ok = new Command("Consultar");         // cria comando simular         cmd_Simular = new Command("Simular");         //cria comando voltar         cmd_Voltar = new Command("Voltar");         //cria tela         frmSimulacao = new Form(cg.NOME_PROJETO+                                " - Simulação");          //seta o layout         frmSimulacao.setLayout(new BoxLayout(                                BoxLayout.Y_AXIS));          //adiciona os comandos a tela         frmSimulacao.addCommand(cmd_Voltar);         frmSimulacao.addCommand(cmd_Simular);         frmSimulacao.addCommand(cmd_Ok);         //define um listener         frmSimulacao.setCommandListener(this);     }     //mostra a tela no dispositivo     frmSimulacao.show(); }</pre>

**Quadro 1: Dispositivo móvel - codificação padrão para criação de telas utilizando LWUIT**

Para que o usuário consiga executar alguma operação no dispositivo é necessário o uso de botões e menus. Como mostrado no quadro 1, foram criados os comandos cmd\_OK, cmd\_Simular, cmd\_Voltar. Também na mesma tela é

definido um *listener* (ouvinte), e este será responsável por receber as ações, tratá-las e então processá-las. No quadro 2 é mostrado o tratamento das ações acionadas pelo usuário.

<b>TRATAMENTO DAS AÇÕES</b>
<pre>//confere qual comando foi pressionado public void actionPerformed(ActionEvent comando) {     //confere o comando     if (comando.getCommand() == cmd_Ok) {         realizaConsulta(edtReferencia.getText(),                         edtVariacao.getText(), usuarioLogado);     } else if (comando.getCommand() == cmd_Voltar) {         projetoPrincipal.frmMenuPrincipal.show();     } //confere o comando     } else if (comando.getCommand() == cmd_Simular) {         montaFormularioRealizaSimulacao();     } }</pre>

**Quadro 2: Dispositivo móvel -tratamento de ações utilizando LWUIT**

Para efetuar a comunicação entre o dispositivo móvel e o servidor de aplicações foi utilizado uma classe padrão chamada *RequisicaoHttp*. Ela encapsula a classe *Connector* do J2ME (DEVMOBILE, 2008). Para efetuar uma conexão é necessário informar o endereço do servidor de aplicações (URL), juntamente com a porta, o servidor que contém os servlets, e por fim o servlet responsável pelo processamento. Abaixo no quadro 3 é mostrado um trecho de um código realizando uma conexão com o servidor.

<b>EFETUANDO UMA CONEXÃO HTTP</b>
<pre>//Método iniciaRequisição RequisicaoHttp requisicao = RequisicaoHttp.iniciaRequisicao( // endereço: http://localhost:8080/ServidorSAVM cg.URL + //ServletSimulacao     nomeServlet , //parâmetros .valores que serão tratados no servlet "opcao=" + opcao + "&amp;" + "usuario=" + usuario + "&amp;" + "referencia=" + referencia + "&amp;" + "numeroVariacao=" + numeroVariacao, //tipo de passagens dos parâmetros (GET ou POST) RequisicaoHttp.POST, //listener para tratamento da resposta do servlet this,</pre>

```
//id que representa a thread
1,
//tempo para execução da requisição
cg.TIME_OUT
);
```

**Quadro 3: Dispositivo móvel- realizando uma conexão com a classe RequisicaoHTTP**

A classe `RequisicaoHTTP` possui um *listener* para tratamentos das respostas enviadas pelo servlet requisitado. No momento que é concluída uma requisição é verificada se houve ou não sucesso na conexão. Podem ser executadas várias requisições simultaneamente, sendo que seu controle é efetuado de acordo com o `Id` enviado na requisição. Abaixo no quadro 4 é mostrado o tratamento de uma resposta enviada por um servlet.

<b>TRATAMENTO DE RESPOSTAS DE UM SERVLET</b>
<pre>public void requisicaoConcluida(boolean sucesso,                                String mensagemErro,                                byte[] dadosRecebidos,                                int id) {      if (sucesso) {         //verifica o ID da requisicao         if (id == 1) {             //processa as informações recebidas do servlet         }     }else{         //trata o erro da conexão     } }</pre>

**Quadro 4: Dispositivo móvel - tratamento das respostas envidas pelo servlet utilizando classe `RequisicaoHTTP`**

As informações são trafegadas no formato de *array* de *bytes*. Foi definido que a organização dos dados seria feita da seguinte maneira:

- ponto e vírgula(;) – é o caractere que irá separar os registros (linhas);
- vírgula(,) – é o caractere que irá separar os campos (colunas);

As informações recebidas do servidor são então organizadas em um *array* de *bytes* e então convertidas para uma única string, conforme apresentado no quadro 5.

<b>ORGANIZAÇÃO DOS DADOS RECEBIDOS</b>
<pre> public void requisicaoConcluida(boolean sucesso,                                String mensagemErro,                                byte[] dadosRecebidos,                                int id) {     //transforma o array de bytes em string     String dados = new String(dadosRecebidos).trim();     //transforma a String em array de strings     //representando os registros (linhas)     String registros[] = cg.split(dados, ';');     For (int i=0; registros.length&lt;0; i++)         //transforma o registro em um array de strings         //representando os campos(colunas)         String campos[] = cg.split(registros[0], ',');     } } </pre>

**Quadro 5: Dispositivo móvel - processamento das respostas enviadas pelo servidor**

Tendo recebido os dados, é necessário em certos casos armazená-los no dispositivo. Foi utilizado no desenvolvimento da aplicação o framework Floggy para realizar toda a parte de persistência. No quadro 6 é apresentado um trecho do código fonte mostrando sua utilização.

<b>FRAMEWORK FLOGGY</b>
<pre> //gravando no dispositivo móvel Mensagens mensagens = new Mensagens(); //popula o objeto mensagens.setCodigo(i); //popula o objeto mensagens.setLida(0); ); //popula o objeto mensagens.setMensagem(campos[1]); ); //grava dados PersistableManager.getInstance().save(mensagens); //recuperando dados no dispositivo móvel ObjectSet todasMensagens = PersistableManager.getInstance().find(Mensagens.class,                                      null, null); //excluindo dados no dispositivo móvel PersistableManager.getInstance().delete(mensagem); </pre>

**Quadro 6: Dispositivo móvel – gravando, recuperando e excluindo dados (FLOGGY, 2008)**

No momento que o servidor recebe uma requisição é iniciado então o processamento e após envia uma resposta para quem solicitou. As informações

serão recebidas em forma de parâmetros enviados pelo método `iniciaRequisicao()` mostrado no quadro 3. Cada parâmetro é interpretado no servlet e direcionado ao seu método correspondente, sendo então processado. No quadro 7 é mostrado um trecho do código fonte de um servlet executando uma requisição e então enviando a resposta para o dispositivo móvel.

<b>SERVLET RECEBENDO UMA REQUISIÇÃO</b>
<pre>//utilizando passagens de parâmetros pelo método POST protected void doPost(HttpServletRequest request,                       HttpServletResponse response)     throws ServletException,            IOException {processRequest(request,  response);  //parametro1 String String opcao = request.getParameter("opcao"); //parametro2 String referencia=request.getParameter("referencia"); //parametro3 numeroVariacao=request.getParameter("numeroVariacao"); String retorno=""; //verifica qual opção escolhida if (opcao.equals("realizaConsulta")){     //método realizaConsulta()     retorno=realizaConsulta(referencia, numeroVariacao);     //envia a resposta para o dispositivo móvel     response.getWriter().write(retorno);     // envia a resposta para o dispositivo móvel     response.getWriter().flush(); } }</pre>

**Quadro 7: Servidor - codificação de um servlet**

Foram criados arquivos de configurações tanto para o servidor como para a aplicação que será executada no dispositivo móvel. Esse arquivo é uma classe escrita em Java e contém toda a parametrização mínima para que o sistema como um todo funcione. Esse arquivo é identificado com o nome de `ConfiguracoesGerais.java`. Abaixo é apresentado um trecho do seu código fonte.

<b>ARQUIVOS DE CONFIGURAÇÕES DO DISPOSITIVO MÓVEL</b>
<pre> public class ConfiguracoesGerais {     public final PersistableManager pm =         PersistableManager.getInstance();      public final String NOME_PROJETO = "SAVM";     public final String URL =         "http://localhost:8081/ServidorSAVM/";     public final String URL2 =         "http://www.dws.com.br:8080/ServidorSAVM/";      public final String ERRO = "ERRO";     public final String SUCESSO = "SUCESSO";     public final String DIR_IMAGENS = "/img/";     public final String AVISO = "AVISO";     public final int TIME_OUT = 200000;     public final int BLACK = 0x000000;     public final int WHITE = 0xFFFFFFFF; } </pre>

**Quadro 8: Dispositivo móvel – Arquivo ConfiguracoesGerais.java**

<b>ARQUIVOS DE CONFIGURAÇÕES DO SERVIDOR</b>
<pre> public class ConfiguracoesGerais {     private final String SGBD="Mysql";     private final String BD_URL_MYSQL =         "jdbc:mysql://local:3306/savm";     private final String BD_URL_ORACLE         ="jdbc:oracle:thin:@localhost:1521:savm";     private final String BD_USER = "usuario";     private final String BD_PASS = "senha";     private final String SUCESSO ="SUCESSO";     private final String ERRO ="ERRO";     private final String AVISO="AVISO";     private String BD_CLASS_ORACLE="oracle.jdbc.OracleDriver";     private String BD_CLASS_MYSQL="com.mysql.jdbc.Driver"; } </pre>

**Quadro 9: Servidor – Arquivo ConfiguracoesGerais.java**

### 3.2 Instalação do sistema SAVM

O sistema pode ser instalado em qualquer dispositivo móvel que suporte a plataforma J2ME MIDP 2.0 e CLDC 1.1. Com a adição da biblioteca LWUIT na aplicação é recomendável que o dispositivo possua uma resolução de tela de 320 x

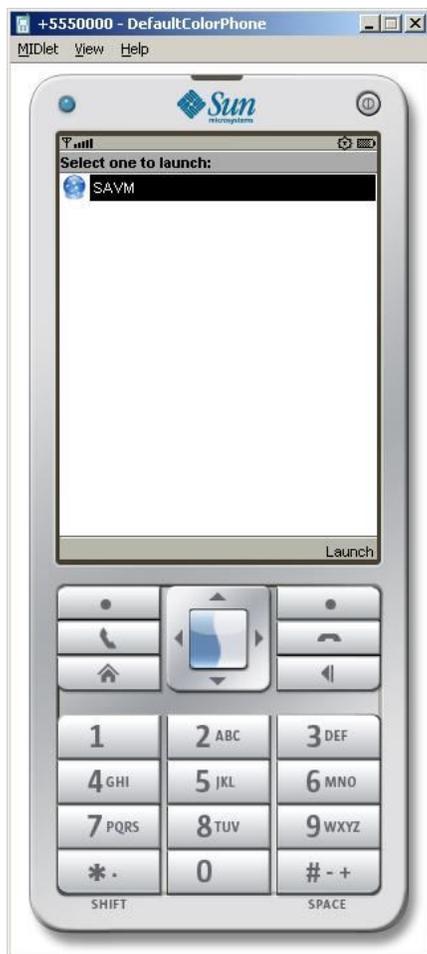
240 pixels. Essa configuração é característica da maioria dos *SmartPhones* existentes no mercado atualmente.

Para realizar a instalação do sistema no dispositivo são necessários os arquivos SAVM.jad e SAVM.jar. Esses arquivos podem ser carregados via navegador *Web* do próprio aparelho, via cabo, *Bluetooth* ou qualquer outro meio que permita comunicação e transferência de arquivos entre computador e dispositivo móvel. Tendo carregados os arquivos no aparelho o processo de instalação se iniciará automaticamente. Caso não inicie a instalação, será necessário executar o arquivo SAVM.jar no próprio dispositivo e então instalá-lo.

#### **4 USO DO SISTEMA SAVM**

A demonstração do uso da ferramenta SAVM foi realizado no emulador que acompanha o IDE NetBeans.

Na figura 28 é apresentado o emulador, com a aplicação selecionada para que seja iniciada.



**Figura 28: Emulador – Inicializando**

#### **4.1 Efetuando autenticação**

No momento que é iniciado o aplicativo SAVM no dispositivo móvel, a primeira tela que aparece para o usuário é a de login (mostrada figura 29).

Nesse momento é preciso que o operador forneça ao aplicativo o seu usuário e sua senha, disponibilizado pela empresa que utiliza o sistema SisCon. Sem a autenticação do usuário o sistema fica indisponível para sua utilização.



Figura 29: Autenticação



Figura 30: Menu Principal

Após efetuada a autenticação com sucesso a aplicação apresenta o menu principal do sistema para acesso aos módulos desejados (mostrado na figura 30).

#### 4.2 Verificando mensagens

No momento que o operador escolhe a opção do menu principal Mensagens, a aplicação irá efetuar uma consulta para verificar se existem novas mensagens, e após mostrará as mensagens para que possa visualizar, ou excluí-la.



Figura 31: Mensagens

Figura 32: Visualizando

Na figura 31 é demonstrada a seleção de uma mensagem para visualização, que é exibida na figura 32.

### 4.3 Empenhando estoque

Selecionando a opção empenho, do menu principal, o sistema abrirá para o operador a tela empenho estoque. O usuário deverá informar obrigatoriamente a referência, a variação e a cor do produto que deseja consultar, conforme figura 33.

Pressionando o botão verificar estoque, o sistema mostrará uma tabela com todos os produtos disponíveis, como mostrados na figura 34. Ainda, na mesma tela (figura 33), é possível fazer uma atualização no cadastro de cores pressionando o botão importar cores.



Figura 33: Verificando Estoque

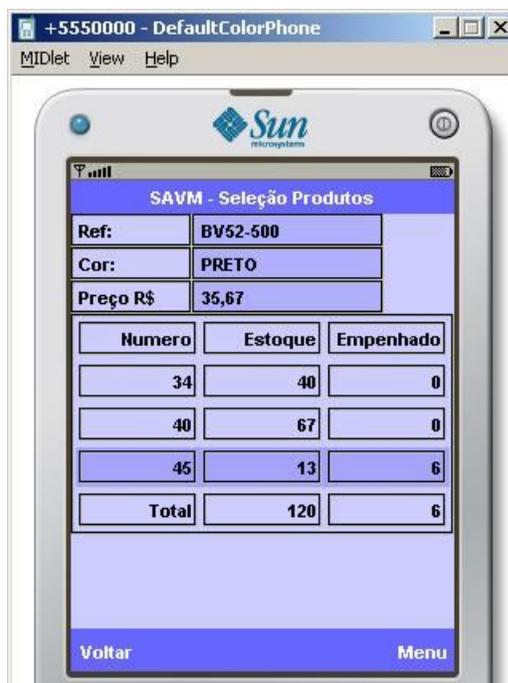


Figura 34: Seleção Produtos



Figura 35: Empenhando



Figura 36: Concluir Empenho

Depois de efetuada a consulta de estoque, caso possua algo disponível, o operador poderá escolher um determinado número e então pressionar o menu e

escolher o botão empenhar. Nesse momento abrirá uma tela, conforme figura 35, mostrando um campo para que o usuário digite a quantidade que deseja empenhar.

Tendo feito a escolha, o usuário poderá escolher a opção concluir. Feito isso a aplicação mostrará a tela concluir empenho, mostrado na figura 36. É apresentado um resumo dos produtos escolhidos. Ainda nessa mesma tela o usuário poderá visualizar a grade escolhida, poderá excluir um determinado produto, realizar uma nova consulta, ou encerrar.



Figura 37: Finaliza Empenho

Figura 38: Mensagem Sucesso

Tendo escolhido a opção encerrar, o sistema abrirá a tela finaliza empenho, conforme figura 37, com os campos cpf/cnpj e cliente. O cliente pode ser do banco de dados do sistema SisCon ou pode ser um cliente novo. Caso o cliente não seja cadastrado, o operador poderá informar somente o nome. E por fim se o cliente for cadastrado, o usuário deverá informar seu cpf ou cnpj e então pressionar o botão finalizar. Nesse momento será efetuada a gravação do empenho, sendo apresentando uma tela de confirmação de sucesso.

Concluído com sucesso o processo de empenho, o sistema indisponibiliza a quantidade reservada do estoque disponível. Sendo necessário no final do dia o

vendedor acessar o sistema SisCon para completar os dados cadastrais dos clientes ( caso sejam novos), bem como outros dados referente ao pedido.

#### 4.4 Consultando pedidos

Selecionado a opção Consulta Pedido do menu principal, o sistema abrirá para o operador uma tela, conforme figura 39, para que informe a nota fiscal ou o pedido ou o cnpj/cpf e escolha a quantidade de registros que deseja mostrar, caso consulte por cnpj/cpf.

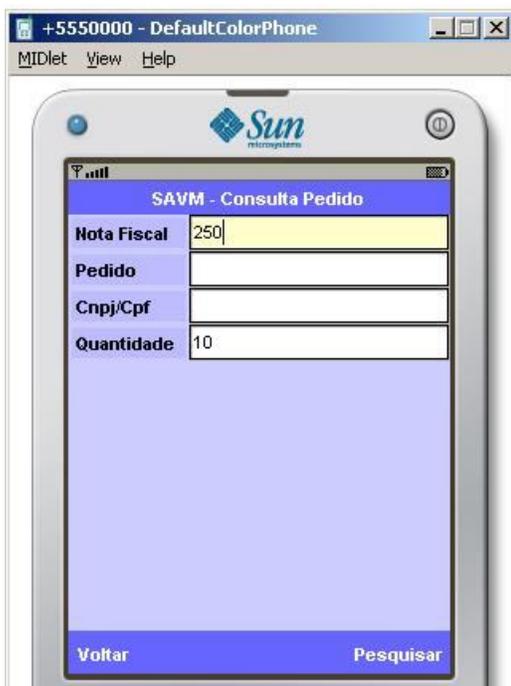


Figura 39: Consulta Pedido



Figura 40: Resultado Consulta

Caso o usuário não informe o cpf/cnpj, e então informar a nota ou o pedido, no momento que for pressionado o botão pesquisar a aplicação mostrará os dados da nota fiscal ou pedido solicitado, como mostrado na figura 40.



Figura 41: Consulta por CPF/CNPJ

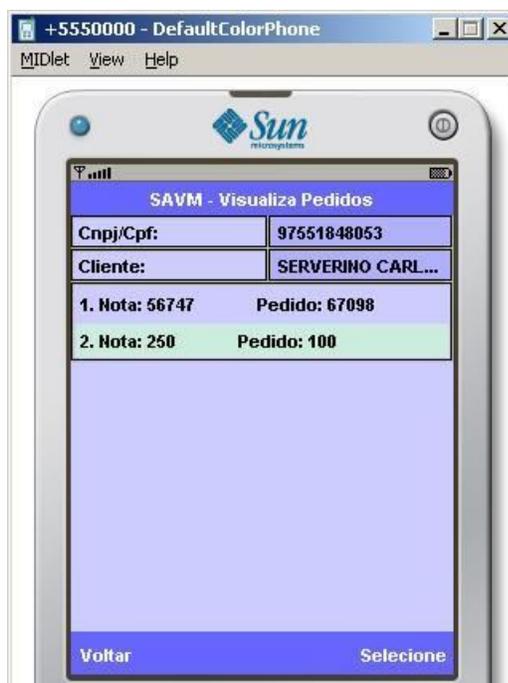


Figura 42: Visualiza Pedidos



Figura 43: Visualiza Dados Pedido

Se o operador informar o cpf/cnpj, e então acionar o botão pesquisar, o sistema abrirá a tela visualiza pedidos, conforme figura 42, para que selecione uma nota específica e mostre os dados do pedido e nota fiscal, mostrado na figura 43.

#### 4.5 Analisando restrições

Selecionando no menu principal a opção Análise Restrições, abrirá para o operador uma tela para que informe o cpf ou cnpj do cliente, conforme mostrado na figura 44. No momento que o botão consultar for pressionado o sistema irá verificar se o cliente possui alguma pendência financeira ou qualquer tipo de bloqueio que impossibilite uma venda. Na tela visualiza restrições (figura 45), é mostrado o resultado de uma consulta, apresentando pendências junto ao cadastro da empresa.

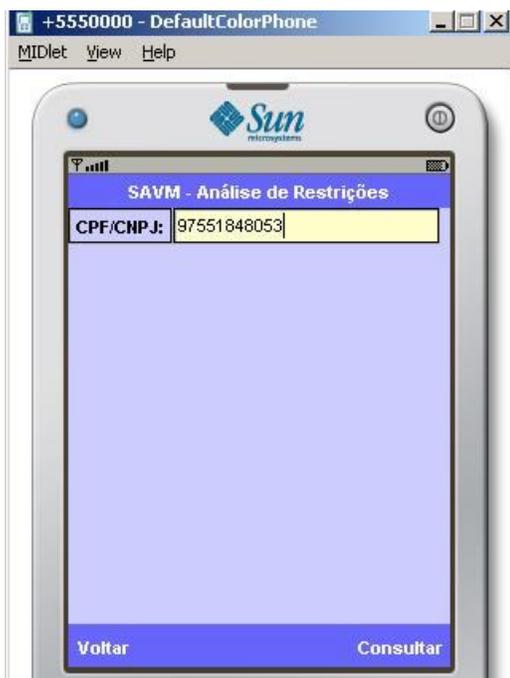


Figura 44: Análise Restrições



Figura 45: Visualiza Restrições

## 4.6 Simulando Vendas

Selecionando a opção Simulação do menu principal, abrirá para o operador a tela simulação pedindo para que seja informada a referência do produto juntamente com a variação. Na mesma tela o usuário poderá realizar uma consulta ou simular um preço de venda. No momento que for pressionado o botão consultar será mostrado o preço de venda do produto, a comissão paga para o vendedor e a quantidade base. Nas figuras 46 e 47 é apresentada a tela inicial e a tela com o resultado da consulta respectivamente.

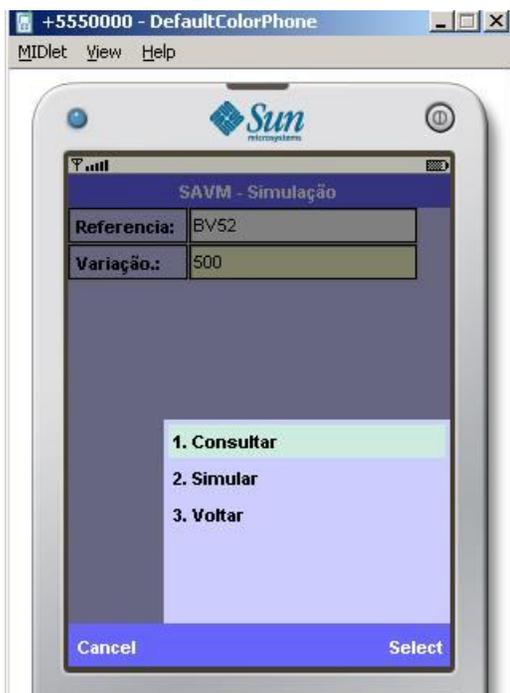
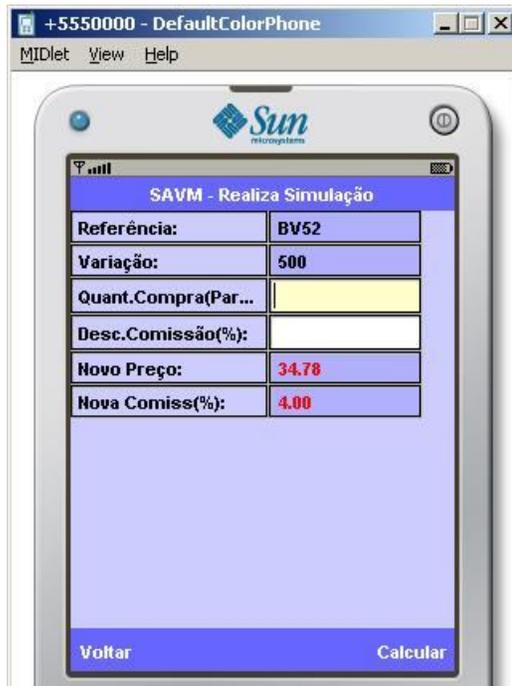


Figura 46: Simulação



Figura 47: Simulação- resultado consulta

Após pressionar o botão simular o sistema abrirá a tela realiza simulação (mostrada na figura 48). Será apresentada a referência e a variação informada inicialmente. Também abrirá dois campos textos editáveis para a informação da quantidade de compra e o desconto da comissão. Por fim será mostrado o novo preço de venda e a nova comissão, que serão os resultados da simulação.



**Figura 48: Realiza Simulação**

Inicialmente o novo preço e a nova comissão serão os valores padrões que constam na tabela de preço.

Caso o usuário informe a quantidade que o cliente deseja comprar e após pressionar o botão calcular, o sistema irá fazer um cálculo, e então com base nas informações da tabela de preço tentará conceder um desconto no preço de venda. O cálculo é feito basicamente levando em consideração a quantidade base mínima de compra para a concessão de desconto.

Por exemplo, se o cliente deseja comprar 100 pares de um determinado produto, e a quantidade base for de 100 pares, tendo como percentual de desconto 5%, e como preço de venda de tabela R\$ 34,78 por par, o sistema realizará cálculo mostrado abaixo:

$$\text{Novo Preço Venda} = 34,78 - (34,78 * 5\%)$$

$$\text{Novo Preço Venda} = \text{R\$ } 33,04$$

Se a quantidade comprada ultrapassar a quantidade base o sistema aplicará novamente o mesmo desconto sobre o preço já calculado. Por exemplo, caso o cliente deseja comprar 200 pares o sistema realizará o seguinte calculo:

$$\text{Novo Preço Venda} = 34,78 - (34,78 * 5\%)$$

$$\text{Novo Preço Venda} = 33,04 - (33,04 * 5\%)$$

$$\text{Novo Preço Venda} = \text{R\$ } 31,39$$

É possível ainda que o sistema sugira ao vendedor que o mesmo ofereça ao cliente a compra de uma quantidade maior, caso a quantidade informada seja próxima à quantidade base da tabela de preço, tendo como benefício o ganho de um desconto. Por exemplo, caso o vendedor informe somente 87 pares, o sistema entenderá que é uma quantidade de compra aproximada da quantidade base e então abrirá uma tela sugerindo a compra de 100 pares.

Abaixo, na figura 49 e 50 é apresentada a tela que realiza o cálculo do desconto pela quantidade de compra, bem como o sistema sugerindo uma nova quantidade de compra para que o vendedor ofereça ao cliente.

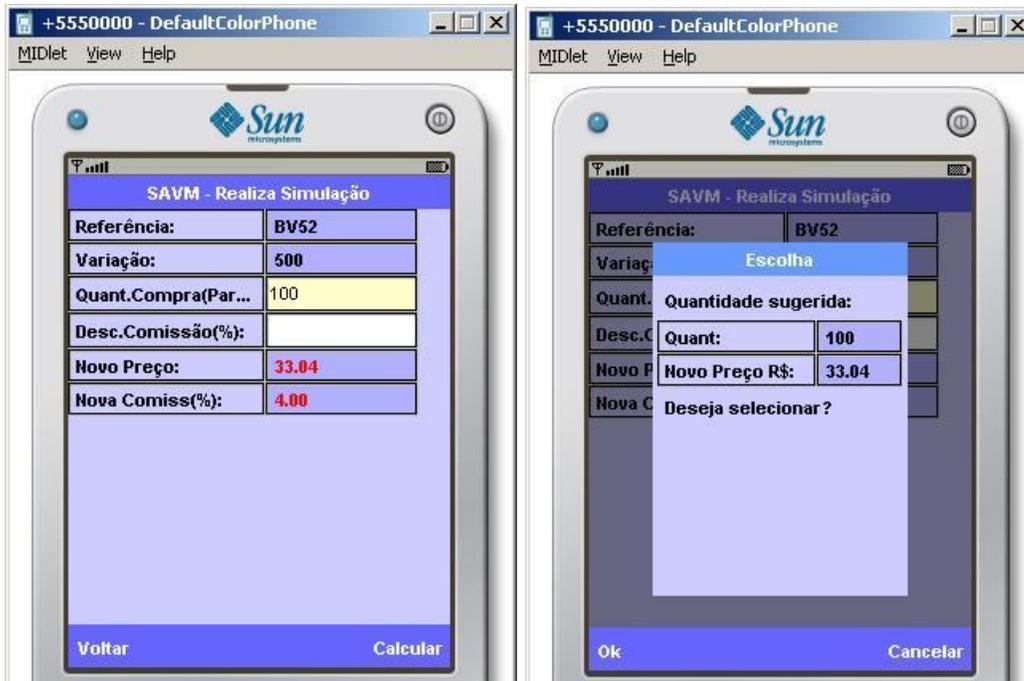


Figura 49: Desconto por quantidade

Figura 50: Sistema sugerindo quantidade

Ainda na tela “realiza simulação” o vendedor poderá conceder o desconto da sua própria comissão em cima do preço de venda de um determinado produto para o cliente. Para isso basta informar o percentual de quanto ele deseja descontar da sua comissão. Por exemplo, caso o vendedor deseja retirar o 35% da sua comissão e conceder ao cliente, tendo como comissão inicial 4% e preço de venda R\$ 34,78 por par de produtos, o sistema irá realizar o seguinte calculo:

$$\text{Nova Comissão} = 4 - (4 \cdot 35\%)$$

$$\text{Nova Comissão} = 2,6\%$$

$$\text{Novo Preço Venda} = 34,78 - (4 \cdot 1,4\%)$$

$$\text{Novo Preço Venda} = \text{R\$ } 34,29$$

O cálculo do desconto por comissão é apresentando abaixo na figura 51.

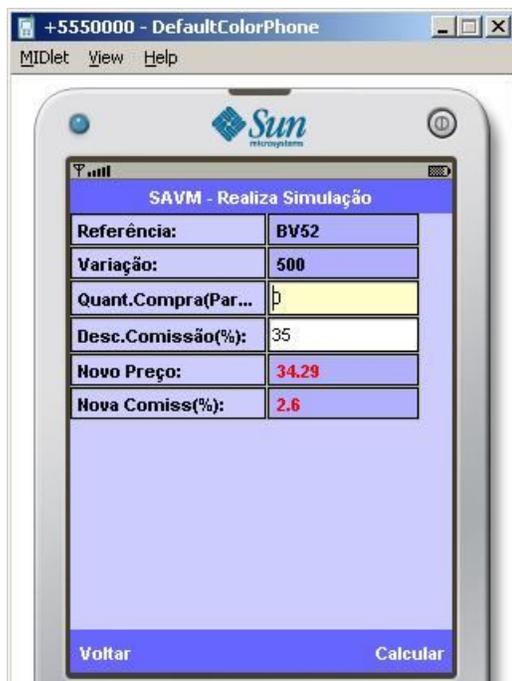


Figura 51: Desconto por comissão

Para o cálculo dos descontos, tanto por quantidade quanto por comissão, o preço de venda não será menor que o preço mínimo definido na tabela de preço.

## 5 CONCLUSÃO

Este trabalho apresentou o desenvolvimento de uma ferramenta móvel que auxilia representantes comerciais de empresas do ramo calçadista na tomada de decisões junto aos seus clientes. Foram apresentadas as dificuldades enfrentadas atualmente por essas empresas e a necessidade sempre constante de estar agregando nas suas operações diferenciais para que sejam competitivas perante seus concorrentes. Também foram descritas as tecnologias utilizadas no seu desenvolvimento e a metodologia usada.

Espera-se que o sistema SAVM seja efetivamente utilizado pela empresa que o adquirir, pois durante a sua criação foi dada uma atenção diferenciada para que a ferramenta fosse de simples utilização, tivesse uma interface amigável e que fosse possível extrair resultados práticos e objetivos por parte do usuário, ajudando assim na prestação de um suporte diferenciado ao cliente.

Para trabalhos futuros, deseja-se agregar mais funcionalidades no módulo simulação de vendas, como por exemplo, definir mais critérios de negociações de preços de vendas, como análise no histórico do cliente, pontualidade em pagamento, definindo assim algum tipo de vantagem para o mesmo, caso ele se enquadre em determinados critérios.

## REFERÊNCIAS

ABACOMM. **Abacomm do Brasil**. Disponível em:  
<<http://www.abacomm.com.br/v1>> Acesso em 16 jun.2008.

ABICALÇADOS. **Resenha Estatística**. Disponível em:  
<[http://www.abicalcados.com.br/documentos/resenha\\_estatistica/Resenha%20Estatistica%202007%20ATUALIZADA.doc](http://www.abicalcados.com.br/documentos/resenha_estatistica/Resenha%20Estatistica%202007%20ATUALIZADA.doc) > Acesso em: 28 out. 2007.

\_\_\_\_\_. **Globalização**. Disponível em: <  
[http://www.abicalcados.com.br/documentos/literatura\\_tecnica/GLOBALIZA%C7%C3O.doc](http://www.abicalcados.com.br/documentos/literatura_tecnica/GLOBALIZA%C7%C3O.doc)> Acesso em: 25 nov. 2007.

APACHE. **Apache Tomcat**. Disponível: < <http://tomcat.apache.org/>> Acesso em: 15 jun. 2008.

AMSYST. **Mobilidade em análise**. Disponível em:  
<[http://www.amsyst.com.br/artigos\\_20061106.shtml](http://www.amsyst.com.br/artigos_20061106.shtml)> Acesso em: 15 nov. 2007.

BRAZILIAN FOOTWEAR. **Resumos de palestras**. Disponível em: <  
[http://www.brazilianfootwear.com.br/documentos/downloads/arq\\_Texto%20pos%20seminario.doc](http://www.brazilianfootwear.com.br/documentos/downloads/arq_Texto%20pos%20seminario.doc)> Acesso em: 25 nov. 2007.

BECK, K. **Extreme Programming Explained: Embrace Change**, 1st Edition, Addison-Wesley, 1999a.

BECK, K. et al. **Agile Manifesto**. 2001b. Disponível em:  
<<http://www.agilemanifesto.org>>. Acesso em: 24 jun. 2008.

CLERICUZI, Adriana Zenaide; ALMEIDA, Adiel Teixeira de; COSTA, Ana Paula Cabral Seixas. **Aspectos relevantes dos SAD nas organizações: um estudo exploratório**. Prod. , São Paulo, v. 16, n. 1, Apr. 2006.

CLUBE DO HARDWARE. **Grid Computing**. Disponível em: <  
<http://www.clubedohardware.com.br/artigos/124/2>> Acesso em 21 jun. 2008.

DECISION REPORT . **Mobilidade deve chegar as PMEs**. Disponível em:  
<[http://www.decisionreport.com.br/publique/cgi/cgilua.exe/sys/start.htm?from\\_info\\_in dex=9&infoid=1129&sid=16&tpl=printerview](http://www.decisionreport.com.br/publique/cgi/cgilua.exe/sys/start.htm?from_info_in dex=9&infoid=1129&sid=16&tpl=printerview)> Acesso em 14 nov. 2007.

DE LUCA, J.; COAD, Peter; LEFEBVRE, E. **Java Modeling In Color With UML: Enterprise Components and Process (Textbook Binding)**. Prentice Hall, 2002.

DEVMEDIA. **História da Oracle**. Disponível em: <[http://www.devmedia.com.br/articles/viewcomp.asp?comp=4685&hl=\\*Banco\\*%20and%20\\*de\\*%20and%20\\*Dados\\*](http://www.devmedia.com.br/articles/viewcomp.asp?comp=4685&hl=*Banco*%20and%20*de*%20and%20*Dados*)> Acesso em 21 jun. 2008.

DEVMOBILE. **Componente reutilizável HttpURLConnection**. Disponível em: <<http://devmobile.blog.br/2008/04/07/comunicacao-paralela-com-servidor-em-aplicacoes-j2me-com-httpconnection-usando-componente-reutilizavel>>. Acesso em 20 mai. 2008.

FATOR BRASIL. **Tendências de mobilidade e convergência na América Latina e no mundo**. Disponível em: <[http://www.revistafatorbrasil.com.br/ver\\_noticia.php?not=9498](http://www.revistafatorbrasil.com.br/ver_noticia.php?not=9498)> Acesso em: 14 nov. 2007.

FLOGGY. **J2ME persistence framework**. Disponível em: <<http://floggy.sourceforge.net/>> Acesso em 10 jun. 2008.

HAMEL, Gary; PRAHALAD, C. K. **Competindo pelo Futuro**. Rio de Janeiro: Campus, 1996.

HIGHSMITH, James A. **Adaptive Software Development: A Collaborative Approach to Managing Complex Systems**. Dorset House Publishing, 2000.

INTEL. **“O que é mobilidade”**. Disponível em: <<http://www.sitesintel.com.br/liberdade/mobilidade.htm>> Acesso em: 15 nov. 2007.

JACQUES. **Introdução e motivação: Arquiteturas em n camadas**. Disponível em: <<http://www.dsc.ufcg.edu.br/~jacques/cursos/j2ee/html/intro/intro.htm>> Acesso em 21 jun. 2008.

JAVA.NET. **Lwuit Mobile and Embedded Project**. Disponível em: <<https://lwuit.dev.java.net/>> Acesso em: 20 ago. 2008.

LVBA. **LVBA Comunicação e Propaganda**. Disponível em: <[http://www.lvba.com.br/portugues/imprensa/release/nokia\\_0744.php](http://www.lvba.com.br/portugues/imprensa/release/nokia_0744.php)> Acesso em: 5 nov. 2007.

MSDN. **Desenvolvimento em camadas**. Disponível em: <[http://www.microsoft.com/brasil/msdn/tecnologias/arquitetura/Layers\\_Developing.mspx](http://www.microsoft.com/brasil/msdn/tecnologias/arquitetura/Layers_Developing.mspx)> Acesso em 21 jun. 2008.

NETBEANS. **NetBeans IDE**. Disponível em <<http://www.netbeans.org/products/ide/>> Acesso em: 28 out. 2007.

MUCHOW, John W. **Core J2ME, Tecnologia e MIDP**. 1.ed. São Paulo:Pearson, 2005.

ORACLE. **História da oracle: inovação, liderança, resultados** . Disponível em: <<http://www.oracle.com/corporate/story.html>>. Acesso em: 15 abr. 2008.

SACCOL, Zanela A.; REINHARD, Nicolau. Tecnologias de Informação Móveis, Sem Fio e Ubíquas: Definições, Estado-da-Arte e Oportunidades de Pesquisa. **Revista de Administração Contemporânea**. n.4, vol.11. Dezembro, 2007.

SCHWABER, K; BEEDLE, M. **Agile Software Development with SCRUM**. New Jersey:Prentice Hall, 2002.

SCRUM ALLIANCE. **ScrumAlliance**. Disponível em: <[http://www.scrumalliance.org/pages/what\\_is\\_scrum](http://www.scrumalliance.org/pages/what_is_scrum)> Acesso em: 30 jul. 2008.

SDN J2EE. **Sun Developer Network-Servlets**. Disponível em: <<http://java.sun.com/products/servlet/overview.html>> Acesso em 21 out. 2008.

SDN IMAGENS. **Sun Developer Network**. Disponível em: <<http://java.sun.com/products/sjwtoolkit/images>> Acesso em 15 out. 2007.

SOMMERVILLE, I. **Software Engineering**. 6.ed. Addison Wesley, 2000.

SUN.COM. **Sun Microsystems**. Disponível em: <http://java.sun.com/javame/index.jsp> Acesso em: 28 out. 2007.

SPRAGUE, Jr.; WATSON, H. **Decision support systems: putting theory into practice**. USA: Prentice-Hall, 1989.

TAURION, Cezar. **Software livre: potencialidades e modelos de negócio**. Rio de Janeiro: Brasport, 2004.

TREVISAN. **Trevisan Tecnologia**. Disponível em: <<http://www.trevisantecnologia.com.br>> Acesso em 18 jun.2008.

TURBAN, E. et al. **Tecnologia da Informação para Gestão**. 3.ed. Porto Alegre:Bookman, 2004.