

ALBERTO YUKIO ABE
EDUARDO ROCHA YANAI
FÁBIO JUN ONISHI

**PORTAL DE COMÉRCIO ELETRÔNICO BASEADO EM DISPOSITIVOS
MÓVEIS E NA PERSONALIZAÇÃO DA IHC: SISTEMA TANAMÃO**

Projeto de Formatura apresentado à
disciplina PCS 2050 – Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo.

São Paulo
2004

ALBERTO YUKIO ABE
EDUARDO ROCHA YANAI
FÁBIO JUN ONISHI

**PORTAL DE COMÉRCIO ELETRÔNICO BASEADO EM DISPOSITIVOS
MÓVEIS E NA PERSONALIZAÇÃO DA IHC: SISTEMA TANAMÃO**

Projeto de Formatura apresentado à
disciplina PCS 2050 – Laboratório de
Projeto de Formatura II, da Escola
Politécnica da Universidade de São Paulo.

Área de Concentração:
Engenharia de Computação

Orientador:
Prof. Dr.
Jorge Luis Risco Becerra

São Paulo
2004

Nossas dúvidas são traidoras

E nos fazem perder

O bem que poderíamos conquistar,

Se não fosse o medo de tentar.

William Shakespeare

AGRADECIMENTOS

Ao nosso orientador Prof. Dr. Jorge Luis Risco Becerra, pelo constante apoio e aconselhamento durante o projeto.

Ao nosso amigo Rubens Jun Hirakawa, que iniciou este trabalho conosco e foi de suma importância para a sua concepção.

Aos nossos familiares, pelo incentivo e compreensão nos momentos mais difíceis.

Aos nossos colegas de turma e professores por todos esses anos de amizade e aprendizado na faculdade.

A todas as pessoas que contribuíram de alguma maneira com o desenvolvimento deste trabalho.

RESUMO

O presente trabalho tem como objetivos o estudo, o desenvolvimento e a implementação de um sistema de cotação de preços e venda de produtos, através de dispositivos móveis conectados à Internet. Este sistema consiste de um portal *web*, com interface homem-computador personalizada ao perfil de acessos do cliente. As tecnologias de interesse englobam: o modelo de comunicação *Web Services*, as arquiteturas de aplicações corporativas baseadas no padrão *J2EE*, o sistema de informação *CRM*, e o protocolo para apresentação de informações em dispositivos móveis conhecido como *WAP*. O texto apresenta os testes e resultados experimentais obtidos com a aplicação de tais tecnologias na arquitetura do sistema. Além disso, mostra algumas considerações acerca do projeto, a fim de transmitir a experiência obtida pelos autores a futuros projetistas de sistemas similares.

ABSTRACT

The objectives of the following work are the study, the development and the implementation of a system that performs prices quotation and products selling through mobile devices connected to the Internet. This system consists of a web portal with a human-computer interface personalized to the customer's access profile. The related technologies are: the Web Services communication model, the architectures for enterprise applications based on the J2EE standard, the CRM information system, and the protocol for presenting information in mobile devices known as WAP. The text presents the tests and experimental results obtained with the application of such technologies in the architecture of the system. Also, it shows some conclusions concerning the project, in order to transmit to future architects of similar systems the experience acquired by the authors.

SUMÁRIO

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

1	INTRODUÇÃO	15
1.1	Contexto Geral	15
1.1.1	A popularização dos dispositivos móveis	15
1.1.2	A difusão do conceito de relacionamento com o cliente (CRM)	16
1.1.3	O surgimento de um novo padrão	16
1.1.4	A busca por conforto e comodidade	17
1.2	Objetivo	18
1.3	Escopo da Solução	18
1.4	Organização	18
2	ASPECTOS TECNOLÓGICOS E CONCEITUAIS	20
2.1	WAP – Wireless Application Protocol	20
2.1.1	Introdução	20
2.1.2	Arquitetura	22
2.1.3	Benefícios	25
2.2	Web Services	26
2.2.1	Introdução	26
2.2.2	SOAP e Web Services	27
2.2.3	SOAP e RPC	29
2.2.4	Documento WSDL	30
2.3	Customer Relationship Management	31
2.3.1	Introdução	31
2.3.2	Aspectos Importantes	33
2.3.3	Arquitetura	34

2.4	Plataforma J2EE.....	37
2.4.1	Introdução	37
2.4.2	Arquitetura J2EE	38
2.4.3	Tipos de Enterprise Java Beans	40
2.5	Processo Unificado	41
2.6	Processo em Cascata.....	43
3	ESPECIFICAÇÃO DO PROJETO DE FORMATURA.....	45
3.1	Requisitos Funcionais.....	45
3.2	Requisitos Não-Funcionais	46
3.3	Características dos Usuários	47
3.4	Restrições	48
3.5	Arquitetura do Sistema.....	49
3.6	Modelo de Casos de Uso.....	52
3.7	Modelo de Classes	56
3.8	Modelo da Interface Homem-Computador	61
3.9	Metodologia	64
4	PROJETO E IMPLEMENTAÇÃO	67
4.1	Descrição do Produto Final.....	67
4.2	Testes e Resultados.....	76
4.2.1	Comparação de preços	76
4.2.2	Compra de produtos	78
4.2.3	Verificação do status do pedido.....	80
4.2.4	Personalização	80
4.2.5	Interface HTML.....	82
5	CONSIDERAÇÕES FINAIS	83
5.1	Conclusões	83
5.2	Trabalhos Futuros	84
ANEXO A – ANÁLISE ORIENTADA A OBJETOS DO SISTEMA		
ANEXO B – ESTUDO DA IHC DO SISTEMA TANAMÃO		
ANEXO C – CRONOGRAMA DA FASE DE IMPLEMENTAÇÃO		

LISTA DE REFERÊNCIAS

APÊNDICE – MANUAL DO USUÁRIO

LISTA DE TABELAS

Tabela 1 – Planejamento de atividades dos recursos.	66
Tabela 2 – Simulação de cotação de preços.	77
Tabela 3 – Banco de dados de transações de débito em banco.	79
Tabela 4 – Banco de dados de transações de cartão de crédito.	79
Tabela 5 – Banco de dados do Web Service do Fornecedor Multi-Coisas.....	79
Tabela 6 – Banco de dados do Web Service do Fornecedor Uau Vênus.....	79
Tabela 7 – Banco de dados do Sistema Tanamão.	80
Tabela 8 – Exemplos de resultados obtidos para um consumidor.	80
Tabela 9 – Registros de Consumidores.....	82
Tabela 10 – Registros de Fornecedores.	82
Tabela 11 – Tabela das distâncias convertidas da interface WAP/WML.....	98

LISTA DE FIGURAS

Figura 1 – Modelo WAP [4].	22
Figura 2 – Arquitetura WAP [4].	24
Figura 3 – Estrutura de uma mensagem SOAP [5].	28
Figura 4 – Dimensões das estratégias de marketing[6].	32
Figura 5 – Arquitetura de informação do CRM básico [6].	34
Figura 6 – Arquitetura tecnológica do CRM básico [6].	35
Figura 7 – As duas dimensões do Processo Unificado [15].	42
Figura 8 – Fases do Processo em Cascata [16].	44
Figura 9 – Arquitetura macro do Sistema Tanamão.	49
Figura 10 – Arquitetura J2EE em 3 camadas para o servidor de aplicação.	50
Figura 11 – Arquitetura completa do Sistema Tanamão.	51
Figura 12 – Diagrama de Casos de Uso preliminar.	52
Figura 13 – Diagrama de Casos de Uso do Sistema Tanamão.	54
Figura 14 – Diagrama de Classes preliminar do Sistema Tanamão.	56
Figura 15 – Diagrama de Classes final do Sistema Tanamão.	58
Figura 16 – Diagrama de Estados de Navegação da interface WAP/WML.	62
Figura 17 – Diagrama de Estados de Navegação da interface HTTP/HTML.	63
Figura 18 – Sistema Tanamão: Tela inicial.	68
Figura 19 – Sistema Tanamão: Tela de Login.	68
Figura 20 – Sistema Tanamão: Menu de opções.	69
Figura 21 – Sistema Tanamão: Promoções identificadas para o consumidor.	69
Figura 22 – Sistema Tanamão: Categorias de produtos.	70
Figura 23 – Sistema Tanamão: Tipos de produtos.	70
Figura 24 – Sistema Tanamão: Listagem de produtos.	71
Figura 25 – Sistema Tanamão: Seleção do endereço para entrega.	71
Figura 26 – Sistema Tanamão: Confirmação de transação de compra.	72
Figura 27 – Tela para cadastro de novos consumidores.	73
Figura 28 – Tela para cadastro de novos fornecedores.	74

Figura 29 – Tela para criação de uma nova promoção.	75
Figura 30 – Diagrama de Interação do método efetuarPedido da classe Portal.	95
Figura 31 – Diagrama de Interação do método gerarProdutos da classe Portal.	96
Figura 32 – Diagrama de Interação do método verificarPedidos da classe Portal.	96
Figura 33 – Estrutura de navegação marcada para análise de métricas de hipertexto. ..	97

LISTA DE ABREVIATURAS E SIGLAS

API	– Application Programming Interface
CGI	– Common Gateway Interface
CNPJ	– Cadastro Nacional de Pessoa Jurídica
CORBA	– Common Object Request Broker Architecture
CPF	– Cadastro de Pessoas Físicas
CRM	– Customer Relationship Management
DW	– Data Warehouse
EJB	– Enterprise Java Beans
GPRS	– General Packet Radio Service
GUI	– Graphical User Interface
HTTP	– HyperText Transfer Protocol
IDE	– Integrated Development Environment
IHC	– Interface Homem-Computador
J2EE	– Java 2 Enterprise Edition
JAXM	– Java API for XML Messaging
JAXP	– Java API for XML Parsing
JDBC	– Java DataBase Connectivity
JMS	– Java Message System
JNDI	– Java Naming and Directory Interface
JNI	– Java Native Interface
JSP	– Java Server Pages
IP	– Internet Protocol
RPC	– Remote Procedure Call
SOAP	– Simple Object Access Protocol
TCP	– Transport Control Protocol
TI	– Tecnologia da Informação
URI	– Uniform Resource Identifier
URL	– Unique Resource Locator

XML	– eXtensible Markup Language
W3C	– World Wide Web Consortium
WAP	– Wireless Application Protocol
WDP	– Wireless Datagram Protocol
WML	– Wireless Markup Language
WSDL	– Web Services Description Language
WSP	– Wireless Session Protocol
WTLS	– Wireless Transport Layer Security
WTP	– Wireless Transaction Protocol

1 INTRODUÇÃO

Apresentamos a seguir os principais aspectos que motivaram a execução deste trabalho, assim como a definição dos objetivos e do escopo do projeto realizado.

1.1 Contexto Geral

O sistema foi idealizado inicialmente imaginando-se uma aplicação que, ao mesmo tempo, representasse o estado da arte em termos de tecnologia, e atendesse às necessidades atuais do consumidor moderno. Assim sendo, listamos a seguir os principais fatos motivadores deste projeto.

1.1.1 A popularização dos dispositivos móveis

Após a privatização do setor de telecomunicações no Brasil, o número de aparelhos de telefonia celular cresceu a tal ponto que chegou a ultrapassar, no ano de 2003, o número de aparelhos de telefonia fixa [1]. Esse crescimento criou um nicho de mercado ainda em maturação: o de serviços oferecidos para dispositivos móveis.

Além do celular, outro tipo de dispositivo móvel que está se tornando cada vez mais acessível é o computador de mão ou *Handheld* (i.e. *Palm* e *Pocket PC*). O barateamento do *hardware*, a expansão das redes sem-fio e o aumento da capacidade de processamento destes aparelhos fazem com que a relação custo/benefício diminua a cada dia.

Com a popularização das tecnologias de acesso à Internet via dispositivo móvel (WAP e GPRS, por exemplo), pode-se prever uma sensível queda nos custos de infra-estrutura nos próximos anos. Como consequência, surgirá uma demanda considerável por serviços ao consumidor final que sejam disponibilizados para acesso via dispositivo móvel.

1.1.2 A difusão do conceito de relacionamento com o cliente (CRM)

Nos últimos anos observou-se a propagação do conceito de CRM em todos os setores da indústria de serviços. A idéia de “conhecer o cliente” é antiga, existe desde os primórdios do comércio na civilização humana. Mas com o desenvolvimento tecnológico presenciado na última década, o CRM passou a ser utilizado em larga escala, graças ao suporte da Tecnologia da Informação.

Atualmente, o CRM é base da estratégia de *marketing* e vendas da maioria das empresas, pois fornece informações para tomada de decisões, fortalece o relacionamento com o cliente e permite um melhor planejamento estratégico, diminuindo custos e gerando dividendos.

1.1.3 O surgimento de um novo padrão

Desde o lançamento dos primeiros programas aplicativos para computador, um grave problema tem sido enfrentado pelos usuários e, principalmente, pelos engenheiros de *software*: a incompatibilidade na comunicação entre programas de diferentes fabricantes, de diferentes plataformas ou de diferentes gerações. Ao longo dos anos, foram disponibilizadas várias soluções para este problema, algumas obtiveram até um certo sucesso, porém nenhuma se tornou consenso entre os grandes fabricantes da indústria de TI.

Sensível a este fato, no início desta década o W3C – *World Wide Web Consortium* [2] reuniu os gigantes do mercado (IBM, Sun, Microsoft e outros) para estabelecer um padrão de comunicação entre aplicativos pela Internet. Surgia então a arquitetura *Web Services*, um modelo de comunicação entre aplicativos, via Internet, utilizando os padrões SOAP e WSDL, que são baseados em outro padrão, o XML.

Apesar de ser uma inovação recente e ainda com problemas inerentes à tecnologia, nota-se atualmente uma grande difusão do *Web Services* dentro da indústria de *software*, apresentando este padrão como a nova aposta da indústria de TI para os próximos anos.

1.1.4 A busca por conforto e comodidade

O desenvolvimento tecnológico sempre esteve intimamente ligado à busca da melhoria da qualidade de vida das pessoas. Isso fica mais evidente ainda nos dias de hoje com o surgimento da Internet, que trouxe consigo o comércio eletrônico, o *e-mail*, as ferramentas de pesquisa e tantas outras inovações que já fazem parte do nosso dia-a-dia. Mas a vida moderna também tem seus inconvenientes, e o mais crítico de todos é o gerenciamento do tempo.

Devido a esses fatores, as pessoas estão cada vez mais à procura de serviços rápidos e que estejam disponíveis a qualquer hora, em qualquer lugar. Por isso, o grupo motivou-se a desenvolver um sistema que tenha a mobilidade e a praticidade como seus principais benefícios, e que possibilite a prestação de serviços essenciais ao nosso dia-a-dia. Enfim, uma aplicação que melhore a qualidade de vida das pessoas.

É dentro deste contexto geral, com a finalidade de atender às necessidades do crescente mercado de aplicações móveis, apresentando soluções personalizadas ao consumidor e aproveitando uma interface padrão de oferecimento de serviços, que idealizamos o Sistema Tanamão, tema do nosso projeto.

1.2 Objetivo

Os objetivos deste projeto são o estudo, o desenvolvimento e a implementação de um sistema de cotação de preços e venda de produtos, utilizando dispositivos móveis conectados à Internet (principalmente aparelhos celulares) como interface com os clientes. Este sistema será um tipo de portal (*site* da Internet onde serviços diversos são oferecidos) personalizado para cada consumidor. A personalização será feita baseada em conceitos de gerenciamento do relacionamento com o cliente (CRM – *Customer Relationship Management*). O codinome adotado para o projeto é “Sistema Tanamão”.

1.3 Escopo da Solução

Será desenvolvido um protótipo capaz de realizar pesquisas, comparação de preços e transações de compra de alguns produtos e serviços fictícios. O funcionamento do sistema será testado e demonstrado utilizando um software de emulação de acesso WAP em aparelhos celulares. Para efeito de simulação, serão implementados alguns *Web Services* (que não são parte do sistema) para fornecer dados dos produtos. Transações financeiras serão simplificadas, e implementadas como *Web Services* de bancos ou operadoras de crédito.

1.4 Organização

Este documento está organizado em seis capítulos, 3 anexos e apêndice. Esta estrutura visa seguir as diretrizes estabelecidas pelo Serviço de Bibliotecas da Escola Politécnica da USP para a elaboração de monografias, teses e dissertações.

O primeiro capítulo, a Introdução, faz a apresentação do projeto através da definição dos objetivos, do escopo de trabalho e da motivação para a pesquisa sobre o assunto, fornecendo esclarecimentos sobre a abordagem do tema e mostrando uma primeira imagem do projeto.

Já o segundo capítulo, Aspectos Tecnológicos e Conceituais, visa apresentar as tecnologias e conceitos empregados no desenvolvimento do Projeto de Formatura, permitindo um entendimento mais profundo sobre os aspectos técnicos descritos no documento.

Em Especificação do Projeto de Formatura, terceiro capítulo, é feito o detalhamento da estrutura do projeto, através da especificação dos requisitos, da listagem das características funcionais e da apresentação de modelos e diagramas que possibilitam a repetição do projeto por outros pesquisadores.

Ainda no terceiro capítulo são discutidas as metodologias empregadas no projeto, estabelecendo-se os resultados esperados em todas as etapas do desenvolvimento do sistema. No capítulo seguinte, Projeto e Implementação, descreve-se o produto final, apresenta-se os testes realizados, os resultados obtidos e aceitação do projeto.

Finalmente, o quinto e último capítulo traz as Considerações Finais sobre o projeto, apresentando a discussão de idéias e opiniões do grupo, o aprendizado adquirido e as conclusões.

2 ASPECTOS TECNOLÓGICOS E CONCEITUAIS

O objetivo deste capítulo é apresentar e discutir os aspectos conceituais mais importantes para o entendimento do projeto de formatura. Assim sendo, prioriza-se o ponto de arquitetura das tecnologias e suas possíveis aplicações.

2.1 WAP – Wireless Application Protocol

Conhecer o protocolo WAP é fundamental para entender como e por quê utilizamos esta tecnologia como interface do sistema com os consumidores do Sistema Tanamão. Assim sendo, apresentamos a seguir uma breve introdução, uma visão geral da arquitetura e os benefícios da implementação de sistemas utilizando o WAP.

2.1.1 Introdução

O *Wireless Application Protocol* (Protocolo para Aplicações Sem-Fio, em português), ou simplesmente WAP, é um conjunto de protocolos que está sendo definido de forma a prover serviços de Internet e de telefonia para aparelhos móveis sem-fio. É um padrão aberto desenvolvido e controlado pelo *WAP Forum* [3], entidade que congrega as principais empresas internacionais de telecomunicações e informática.

O WAP está posicionado na convergência de duas tecnologias de rede que estão se desenvolvendo rapidamente: a comunicação sem-fio e a Internet. Os mercados para estas tecnologias vêm crescendo continuamente, alcançando novos consumidores, mas a maioria das tecnologias desenvolvidas para a Internet foi projetada para computadores pessoais ou maiores, com uma banda relativamente larga, em redes de dados geralmente confiáveis.

Porém, os dispositivos portáteis sem-fio apresentam, em geral, um ambiente de computação mais restrito se comparado ao de computadores pessoais. As limitações fundamentais são [4]:

- CPU's menos poderosas;
- Menos memória (ROM e RAM);
- Consumo de potência restrito;
- Visores menores;
- Dispositivos de entradas diferentes (ex. teclas de um telefone).

Igualmente, redes móveis de dados apresentam um ambiente mais hostil de comunicação se comparadas às redes convencionais. Devido às limitações de potência, espectro disponível e mobilidade, as redes de dados sem-fio tendem a possuir [4]:

- Menos banda passante;
- Maior latência;
- Menos estabilidade de conexão;
- Menos previsibilidade de disponibilidade.

Para navegar na Internet é preciso que os aparelhos celulares possuam um “micro-navegador” instalado. No entanto, eles não podem acessar os *websites* normalmente vistos no computador, devido às limitações citadas anteriormente. Por isso, foi criado o WAP, uma nova tecnologia adequada ao aparelho celular, que permite o acesso a *mini-sites* [4].

Os navegadores que conhecemos, como *Internet Explorer* ou *Netscape*, interpretam páginas HTML da Internet em computadores pessoais. Um micro-navegador WAP é um *software* similar, muito mais simples, criado para funcionar em celulares e outros aparelhos sem-fio, interpretando páginas WML.

Os *sites* WAP, criados na linguagem WML, são baseados quase que totalmente em texto, com pouquíssimas imagens monocromáticas. WML é a sigla de *Wireless Markup Language*. O WML está para a WAP assim como o HTML está para a WWW (*World Wide Web*). É a linguagem utilizada para construir os *mini-sites* para os celulares (chamados "*decks*"), divididos em telas ou páginas (conhecidas como "*cards*").

2.1.2 Arquitetura

O modelo WAP é muito similar ao modelo WWW. Isso fornece vários benefícios para os desenvolvedores de aplicações, incluindo um modelo familiar de programação, uma arquitetura estabelecida e a habilidade de alavancar ferramentas existentes (ex. servidores *web*, ferramentas XML, etc.). Otimizações e extensões foram criadas de maneira a explorar as características do ambiente sem-fio. Sempre que possível padrões existentes foram adotados ou foram utilizados como ponto de partida para a tecnologia WAP [4].

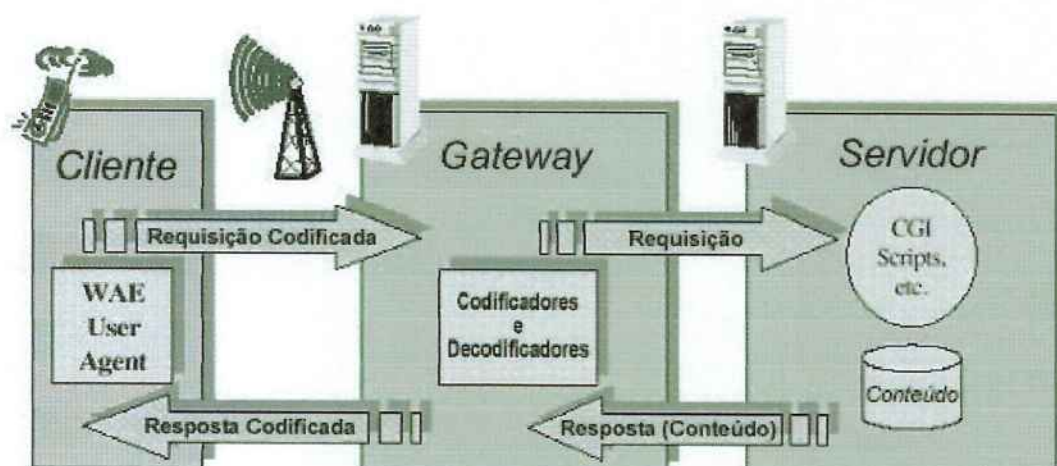


Figura 1 – Modelo WAP [4].

Aplicações e conteúdos WAP são especificados em um conjunto de formatos bem conhecidos baseados nos formatos WWW. Conteúdos são transportados utilizando-se um conjunto de protocolos de comunicação também baseados nos protocolos WWW [4].

O WAP define um conjunto padrão de componentes que permitem a comunicação entre dois terminais móveis e os servidores da rede, incluindo [4]:

- Modelo padrão de nomes: é utilizado URL, padrão WWW, para identificar os conteúdos WAP nos servidores de origem. URIs são utilizados para identificar recursos locais em um dispositivo, ex. funções de controle de chamada;
- Tipos diferentes de conteúdo: a todos os conteúdos WAP são fornecidos tipos consistentes com os tipos do WWW. Isso permite ao agentes WAP dos usuários processar corretamente o conteúdo baseado no seu tipo;
- Formatos padrão de conteúdo: formatos de conteúdo WAP são baseados na tecnologia WWW e incluem marcadores de *display*, informação de calendário, objetos de cartão eletrônico de negócios, imagens e linguagens *script*.
- Padrão de protocolos de comunicação: os protocolos de comunicação WAP permitem a comunicação de pedidos feitos pelo *browser* de um terminal móvel e o servidor *web* da rede.

Os tipos de conteúdos e protocolos WAP foram otimizados para dispositivos portáteis sem-fio. O WAP utiliza a tecnologia de *proxy* para conectar o domínio sem-fio e o domínio da WWW. O *proxy* WAP compreende tipicamente as seguintes funcionalidades [4]:

- Gateway de Protocolo: O *gateway* de protocolo traduz pedidos de uma pilha de protocolo WAP (WSP, WTP, WTLS, e WDP) para a pilha de protocolos WWW (HTTP e TCP/IP);
- Codificadores e Decodificadores de Conteúdo: Os codificadores de conteúdo traduzem o conteúdo WAP para formatos compactos para reduzir o tamanho dos dados na rede.

Essa infraestrutura garante que usuários de um terminal móvel possam navegar em uma grande variedade de conteúdos e aplicações WAP, e que os desenvolvedores de aplicações possam construir serviços de conteúdo e aplicações que rodem numa larga base de terminais móveis. O *proxy* WAP permite que aplicações e conteúdo possam ser hospedados em um servidor WWW padrão e possam ser desenvolvidos utilizando-se tecnologias WWW já estabelecidas tais como CGI *scripting* [4].

A arquitetura WAP fornece um ambiente extenso e escalável para o desenvolvimento de aplicações para dispositivos de comunicação móveis. Isso é alcançado através do projeto de uma pilha inteira de protocolos (ver Figura 6). Cada camada da arquitetura é acessível através da camada acima dela, tais como através de outros serviços e aplicações [4].



Figura 2 – Arquitetura WAP [4].

2.1.3 Benefícios

As relações comerciais nunca mais serão as mesmas. A tecnologia WAP está revolucionando o comércio eletrônico, criando um novo espaço, muito mais imediato e eficiente para os consumidores finais e todas as empresas realizarem suas transações comerciais em um tempo mínimo e com toda a segurança.

Tem-se acesso a toda a rede de serviços que a Internet oferece como enviar e receber *e-mails*, trocar mensagens instantâneas, participar de salas de bate-papo (*chat*), consultar informações financeiras, ler as principais notícias, verificar as condições de trânsito, previsão do tempo, buscar dicas de bares, restaurantes e cinema e acessar as mais diversas informações, com toda a conveniência e mobilidade.

Imagine se uma pessoa acabasse de chegar a uma cidade e pudesse buscar através de seu celular o restaurante mais próximo, o cardápio do dia e o valor da refeição. Pode-se entrar no *site* do restaurante e fazer a reserva. Além disso, pode-se entrar em outro *site* que mostra qual o melhor caminho para chegar ao restaurante. Não perderia tempo procurando por ruas desconhecidas ou pagando uma fortuna para o taxista. Tudo isso feito com alguns simples apertar de botões enquanto espera para pegar sua bagagem no aeroporto.

Inúmeras situações podem ser citadas para se aproveitar da mobilidade de um serviço de informações. A empresa poderá integrar clientes, funcionários e fornecedores com apenas alguns “cliques”, oferecendo serviços com maior eficiência, e mantendo contato com todos os funcionários da empresa.

2.2 Web Services

A arquitetura do Sistema Tanamão baseia-se no consumo de serviços distribuídos, disponibilizados pelos fornecedores dos produtos. E a melhor forma de implementar isto é utilizando o padrão *Web Services*, que representa o estado da arte em termos de comunicação entre aplicações. Portanto, apresentamos a seguir uma introdução teórica sobre o assunto a fim de possibilitar a compreensão da arquitetura utilizada no projeto.

2.2.1 Introdução

Quando a Internet começou a se popularizar, por volta do meio dos anos 90, as tecnologias presentes permitiam ao usuário conectar-se a um *site* e baixar o conteúdo deste. O HTML (*Hyper Text Markup Language*) era a linguagem "*de-facto*" que permitia a apresentação da informação presente na rede. Nos últimos anos, porém, novas tecnologias e *frameworks* de desenvolvimento estão surgindo, permitindo uma maior integração entre os diversos aplicativos e serviços disponíveis na internet. Este novo modelo em crescimento deve tratar tarefas complexas (como o gerenciamento de transações, por exemplo) através da disponibilização de serviços distribuídos que utilizem interfaces de acesso simples e bem definidas. Esses serviços ou aplicativos distribuídos são conhecidos como *Web Services* [5].

Para ilustrar a utilização de *Web Services* em uma situação real, imaginemos um *site* de vendas pela Internet que necessite validar o crédito do comprador antes de proceder com a venda. O sistema então acessa um serviço (*Web Service*) que cuida de todos os passos necessários à verificação de crédito: checa o histórico das compras efetuadas pelo consumidor na empresa, checa a situação de crédito do consumidor no sistema público, etc. O *Web Service* obtém estes dados e retorna a situação de crédito deste consumidor para o site. Este é apenas um exemplo, entre tantos, de utilização de *Web Services*.

2.2.2 SOAP e Web Services

Web Services são identificados por uma URI (*Unique Resource Identifier*), e são descritos e definidos usando XML. Um dos motivos que tornam *Web Services* atrativos é o fato deste modelo ser baseado em tecnologias padrões, em particular XML e HTTP. *Web Services* são usados para disponibilizar serviços interativos na *web*, podendo ser acessados por outras aplicações. SOAP (*Simple Object Access Protocol*) está se tornando padrão para a troca de mensagens entre aplicações e *Web Services*, já que é uma tecnologia construída com base em XML e HTTP [5].

SOAP é um protocolo projetado para invocar aplicações remotas através de RPC (*Remote Procedure Calls* – Chamadas Remotas de Procedimento) ou trocas de mensagens, em um ambiente independente de plataforma e linguagem de programação. Portanto, o SOAP é um padrão normalmente aceito para utilizar-se com *Web Services*. Desta forma, pretende-se garantir a interoperabilidade e intercomunicação entre diferentes sistemas, através da utilização de uma linguagem (XML) e mecanismo de transporte (HTTP) padrões [5].

As principais características do SOAP são [5]:

- Definido pelo consórcio W3C.
- Protocolo baseado em XML para a troca de informações em um ambiente distribuído;
- Padrão de utilização com *Web Services*;
- Normalmente utiliza HTTP como protocolo de transporte.

Uma mensagem SOAP consiste basicamente dos seguintes elementos [5]:

- **Envelope:** Toda mensagem SOAP deve contê-lo. É o elemento raiz do documento XML. O Envelope pode conter declarações de *namespaces* e também atributos adicionais como o que define o estilo de codificação (*encoding style*). Um "*encoding style*" define como os dados são representados no documento XML;
- **Header:** É um cabeçalho opcional. Ele carrega informações adicionais, como por exemplo, se a mensagem deve ser processada por um determinado nó intermediário (é importante lembrar que, ao trafegar pela rede, a mensagem normalmente passa por diversos pontos intermediários, até alcançar o destino final). Quando utilizado, o *Header* deve ser o primeiro elemento do Envelope.
- **Body:** Este elemento é obrigatório e contém o *payload*, ou a informação a ser transportada para o seu destino final. O elemento *Body* pode conter um elemento opcional *Fault*, usado para carregar mensagens de status e erros retornadas pelos "nós" ao processarem a mensagem.

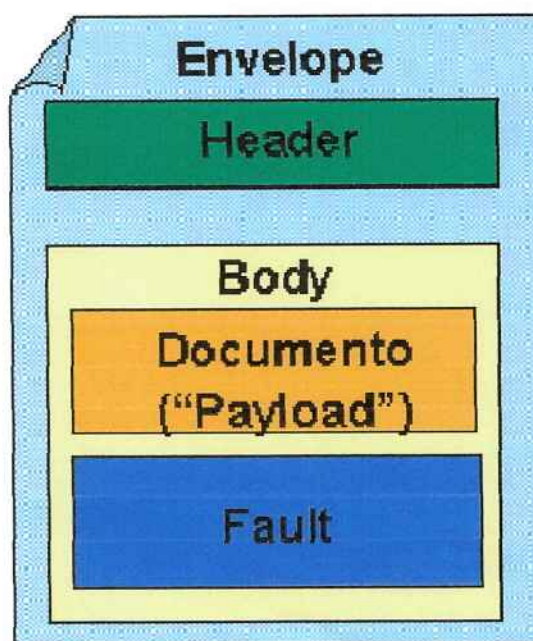


Figura 3 – Estrutura de uma mensagem SOAP [5].

2.2.3 SOAP e RPC

Entre outras implementações, o SOAP foi desenhado para empacotar e transportar chamadas de RPC, e para isto utiliza-se dos recursos e flexibilidade do XML, sob HTTP.

RPCs ou chamadas remotas de procedimento são chamadas locais a métodos de objetos (ou serviços) remotos. Portanto, podemos acessar os serviços de um objeto localizado em um outro ponto da rede através de uma chamada local a este objeto, sendo que cada chamada ou requisição exige uma resposta.

Processo de uma chamada RPC: Antes de serem enviadas pela rede, as chamadas de RPC (emitidas pela aplicação cliente) são encapsuladas (ou serializadas) segundo o padrão SOAP. O serviço remoto, ao receber a mensagem faz o processo contrário, desencapsulando-a e extraíndo as chamadas de método. A aplicação servidora então processa esta chamada, e envia uma resposta ao cliente. O processo então se repete: a resposta é também serializada e enviada pela rede. Na máquina cliente, esta resposta é desencapsulada e é repassada para a aplicação cliente [5].

A especificação SOAP (definida pela W3C) define as seguintes informações, como necessárias em toda chamada de RPC [5]:

- A URI do objeto alvo;
- O nome do método;
- Os parâmetros do método (requisição ou resposta);
- Uma assinatura do método opcional;
- Um cabeçalho (*header*) opcional.

2.2.4 Documento WSDL

De que forma um cliente de um *Web Service* sabe qual o formato dos métodos a serem chamados e quais os parâmetros a serem passados? Como cliente e serviço sabem como processar uma requisição?

Para solucionar estes tipos de problemas foi criado um documento que utiliza uma linguagem chamada WSDL. WSDL ou *Web Service Description Language* é uma linguagem baseada em XML, utilizada para descrever um *Web Service*. Um *Web Service* deve, portanto, definir todas as suas interfaces, operações, esquemas de codificação, entre outros neste documento [5].

Um documento WSDL define um XML *Schema* para descrever um *Web Service*. Tão logo o cliente tenha acesso à descrição do serviço a ser utilizado, a implementação do *Web Service* pode ser feita em qualquer linguagem de programação. Normalmente são utilizadas linguagens construídas para interação com a *web*, como, por exemplo, *Java Servlets* ou ASP, que, em seguida, chamam um outro programa ou objeto [5].

Basicamente, quando o cliente deseja enviar uma mensagem para um determinado *Web Service*, ele obtém a descrição do serviço (através da localização do respectivo documento WSDL), e em seguida constrói a mensagem, passando os tipos de dados corretos (parâmetros, etc) de acordo com a definição encontrada no documento. Em seguida, a mensagem é enviada para o endereço onde o serviço está localizado, a fim de que possa ser processada [5]. O *Web Service*, quando recebe esta mensagem valida-a conforme as informações contidas no documento WSDL. A partir de então, o serviço remoto sabe como tratar a mensagem, como processá-la (possivelmente enviando-a para outro programa) e como montar a resposta ao cliente.

2.3 Customer Relationship Management

A personalização da interface-homem-computador do Sistema Tanamão com os consumidores será baseada em conceitos de CRM, logo a compreensão dos principais conceitos deste tipo de sistema de informação é vital para entender o funcionamento do sistema. Por isso, apresentamos a seguir um breve histórico do CRM, alguns aspectos relativos à sua arquitetura, e outras informações relevantes.

2.3.1 Introdução

De modo simplificado, o *Customer Relationship Management* (Gerenciamento do Relacionamento com o Cliente, em português) ou CRM pode ser entendido como uma estratégia que permite à empresa como um todo ter uma visão única de seu cliente, estando mais atenta a oportunidades de negócio com o mesmo. Esta estratégia surgiu como evolução de correntes anteriores, causada pela crescente competição entre as empresas e pela constante insatisfação dos clientes [6].

A primeira estratégia de *marketing*, o Marketing em Massa, era baseada em *market share*, no oferecimento de produtos e na propaganda de marca. Teve pouco impacto na lealdade dos consumidores, pois se omitia com relação aos serviços ao consumidor.

Surgiu então uma segunda corrente, o Marketing Direcionado, que se baseou no foco em clientes-alvo que poderiam dar um retorno direto à empresa. Este modelo apresentou um ganho com relação ao anterior, porém ainda era ineficiente e tornou-se irritante aos consumidores [6].

Percebeu-se então a necessidade de uma interação “1 para 1” entre empresa e consumidor, a fim de tornar esta relação mais íntima e mais produtiva. Isto possibilitou a evolução das estratégias de *marketing* e o aparecimento do conceito de CRM, como é descrito na Figura 4.

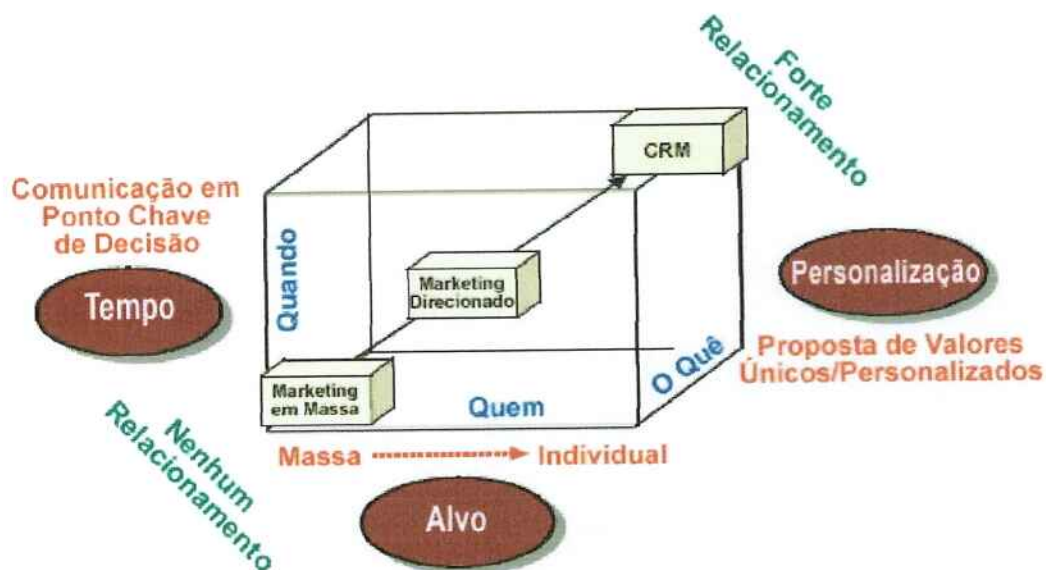


Figura 4 – Dimensões das estratégias de marketing[6].

Estudos indicam que é 6 vezes mais caro conquistar um novo cliente do que reter um já existente. Isso tem motivado a indústria a obter um maior conhecimento sobre seus clientes, através da busca dos seguintes objetivos [6]:

- Aumentar a fidelidade do cliente lucrativo;
- Aumentar a eficiência nos custos e a eficácia das campanhas de *marketing*;
- Buscar oportunidades de venda cruzada;
- Reduzir atritos com o cliente;
- Personalizar produtos, ofertas e preços;
- Planejar um único contato com o cliente por campanha.

2.3.2 Aspectos Importantes

Quando se adota uma estratégia de CRM, o foco se desloca do produto para o cliente [7]. Todo contato deve ser registrado, e cada novo diálogo deve se basear na conversa anterior.

É necessário aproveitar todas as interações que a corporação tem com o cliente no sentido de captar dados e transformá-los em informações que possam ser disseminadas pela organização [8].

A eliminação de redundâncias, também chamada de limpeza e filtragem dos dados, também é fundamental e pode ser feita através do emprego de um *Data Warehouse*, classificado por alguns analistas de mercado como o “coração” do CRM [9].

Sob a ótica da tecnologia, o CRM é um conjunto de aplicativos de *software* que visam, de forma integrada, apoiar as empresas em seus processos de relacionamento com o cliente e na sua gestão. Para obter os benefícios da implantação de um sistema de CRM, é preciso fazer uso intensivo da tecnologia da informação, mas não apenas isso.

Também é necessário mudar a cultura da organização, através de treinamento e comprometimento de funcionários. Implementar tecnologias de CRM sem redesenhar os processos internos da empresa e sem criar um modelo de relacionamento e de atendimento ao cliente, poderá ser apenas um projeto de informatização do *call center* (centro de contato telefônico com o cliente) ou da área de vendas, não conduzindo aos resultados esperados.

Não é a tecnologia, por melhor que seja, que constrói o relacionamento com os clientes, mas sim as pessoas envolvidas no processo [10].

2.3.3 Arquitetura

A arquitetura do CRM básico, do ponto de vista de fluxo de informação, pode ser visto na figura a seguir. Destaca-se a extração de dados de transações (i.e. compras), dados de conta (i.e. preferências, dados pessoais, dados profissionais) e dados externos (i.e. dados demográficos, dados de crédito), e os vários passos necessários para construir o sistema de informações sobre o cliente [6].

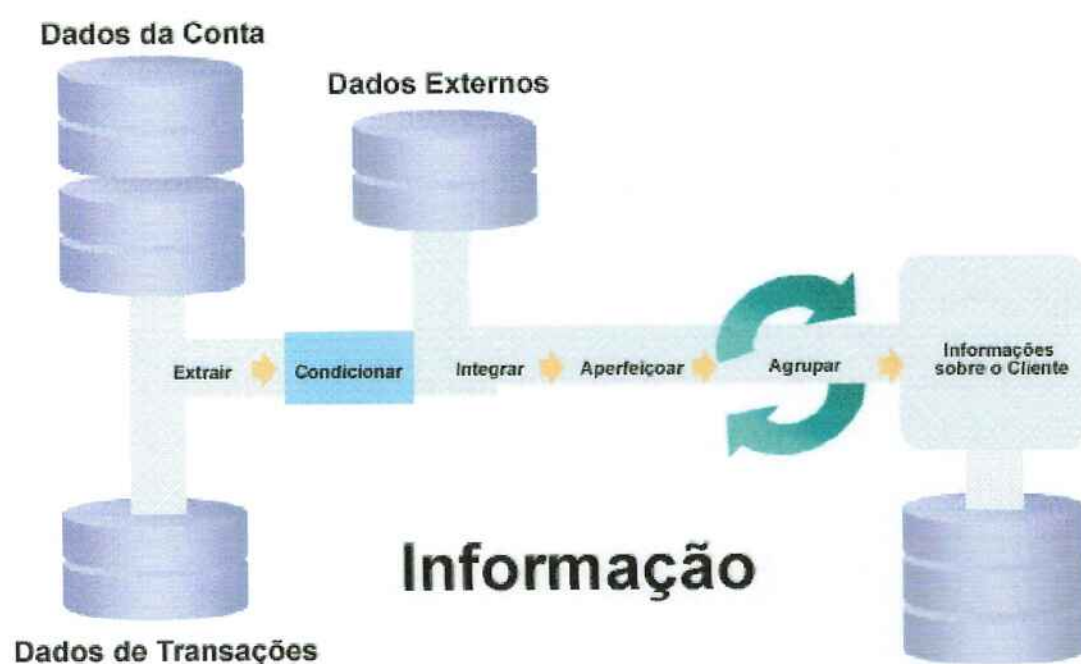


Figura 5 – Arquitetura de informação do CRM básico [6].

Várias tecnologias são utilizadas para implementar as funcionalidades do CRM (Figura 6): *Data Mining*, Processos Analíticos, Consultas & Relatórios, Estatísticas e Acesso à Informação, e todos envolvem a principal base de conteúdo: o *Data Warehouse* [6].

Na avaliação de muitos consultores de mercado, o DW (*Data Warehouse*) é classificado como o coração da arquitetura de CRM na medida em que permite trabalhar com as informações colhidas de vários pontos da empresa de forma mais completa e consistente.

A grosso modo, o DW é um banco de dados físico que fica separado dos demais bancos de dados da empresa, sendo alimentado pelos sistemas transacionais.



Figura 6 – Arquitetura tecnológica do CRM básico [6].

Uma alternativa menos dispendiosa e mais simples para as empresas que desejam construir uma base de dados única e consistente é a utilização dos *Data Marts*, que seriam na verdade "*mini-Data-Warehouses*" montados por departamentos. Os *Data Marts* podem surgir de duas maneiras: *top-down* e *bottom-up*. O *top-down* é quando uma empresa cria um DW e depois parte para a segmentação, ou seja, divide o DW em áreas menores, gerando pequenos bancos orientados por assuntos departamentalizados. A situação inversa caracteriza o *bottom-up*.

Por desconhecer a tecnologia do DW ou por dispor de poucos recursos, a empresa pode optar por, inicialmente, criar um banco de dados para uma área específica e, gradativamente, a empresa pode implementar outros *Data Marts*, até resultar num *Data Warehouse* [9].

2.4 Plataforma J2EE

Aplicações empresariais modernas geralmente mantêm no mínimo três camadas: um cliente, um servidor *middleware* e um banco de dados [12].

Projetos em múltiplas camadas resolvem vários problemas: separar interface do usuário, lógica de negócio e dados da aplicação torna o projeto como um todo mais fácil de ser compreendido e permite que diferentes equipes trabalhem em paralelo em cima de diferentes componentes. Porém, existe um ponto negativo: o aumento da sobrecarga de comunicação entre as camadas [12]. É justamente nesse ponto em que surgem as vantagens da utilização da plataforma J2EE (*Java 2 Enterprise Edition*) da *Sun Microsystems*.

2.4.1 Introdução

A Plataforma J2EE define especificações para a construção de sistemas em múltiplas camadas, garantindo segurança dos dados, troca de mensagens, padrões de componentes, alta performance e robustez, pois é voltada ao desenvolvimento de soluções corporativas [11].

A plataforma J2EE é basicamente um modelo de aplicação distribuído em múltiplas camadas no qual a lógica da aplicação é dividida em componentes de acordo com sua função. Isto permite que vários componentes da aplicação que compõem uma aplicação J2EE sejam instalados em máquinas diferentes [11].

No mundo do J2EE, existem geralmente quatro camadas: um cliente, uma aplicação Web construída ou com *servlets* Java ou com *Java Server Pages* (JSP), uma camada de lógica de negócio construída em *Enterprise Java Beans* (EJB) e uma camada de persistência que acessa um banco de dados relacional [12].

A plataforma define integração com outros sistemas através de CORBA, adquire APIs para o padrão XML, componentes distribuídos – *Web Service*. Além de que, a Camada de Apresentação Cliente, Camada de Apresentação Servidor, Camada de Lógica de Negócio e a Integração de Sistemas são definidas com tecnologias Java que definem o modelo de aplicação [11].

A plataforma J2EE cresce a cada momento, ganhando novas APIs, tecnologias, *frameworks* específicos para camadas de aplicação e *design patterns*. Tudo isto permite que sistemas *web*, *desktops*, dispositivos portáteis e de integração tornem-se mais robustos, estáveis, seguros e com alta performance [11].

2.4.2 Arquitetura J2EE

A arquitetura J2EE é composta por combinações sobre as seguintes camadas [13]:

- Aplicação Stand-Alone ou Applet: é uma aplicação GUI típica que utiliza os serviços de um ou mais EJBs armazenados no servidor de aplicações;
- Servlet ou página JSP: trata-se de uma aplicação *web* típica que também se utiliza dos serviços de um ou mais EJBs armazenados no servidor de aplicações;
- Servidor de Aplicações: gerencia o ciclo de vida de vários EJBs e é responsável pela comunicação entre estes e seus clientes. O Servidor de Aplicações também pode fazer acesso ao banco de dados, substituindo os EJBs;
- Servidor de Dados: em geral é um banco de dados relacional, mas também pode ser um servidor de diretórios ou uma aplicação legada em um *mainframe*. Na verdade não é parte da arquitetura J2EE, pois não executa classes Java (embora alguns bancos como o Oracle e o DB2 sejam capazes de rodar procedimentos armazenados escritos em Java).

O coração do J2EE são os *Enterprise Java Beans* ou EJBs. Um *Java Bean* é um componente Java reutilizável, preparado para uso dentro de IDEs visuais. Um EJB é um *Java Bean* preparado para ser hospedado de forma transparente por um Servidor de Aplicações [13].

A idéia é que a lógica das aplicações seja implementada como um conjunto de EJBs, já que estes poderão ser hospedados em máquinas adequadas em termos de performance e segurança, ou mesmo em várias máquinas separadas para distribuição de carga e tolerância a falhas [13].

A interface homem-computador é fornecida por aplicações GUI ou páginas HTML contendo *Servlets* ou páginas JSP que delegam trabalho para componentes EJB. Como um EJB também é um *Java Bean*, pode ser facilmente manipulado por um ambiente de desenvolvimento visual. Ou então, a camada de apresentação pode ser uma aplicação em linguagens como PHP, *Perl* ou *Python*, que são capazes de criar instâncias e trocar mensagens com componentes EJB. Com um pouco mais de trabalho, esta camada também pode ser formada por aplicativos C ou Pascal, utilizando a tecnologia JNI (*Java Native Interface*) [13].

O Servidor de Aplicações é responsável pela distribuição, comunicação e tolerância a falhas da aplicação. Isto possibilita ao desenvolvedor do componente se preocupar apenas com o código da aplicação. O Servidor de Aplicações também é responsável por fornecer acesso ao EJB através de uma interface CORBA, e ainda, permite a criação de *Servlets* para fornecer acesso via SOAP [13].

Outra vantagem da utilização de componentes EJB é a existência de uma série de APIs que fornecem capacidades adicionais às já citadas. Seguem alguns exemplos:

- JDBC: fornece acesso a bancos de dados relacionais;
- JavaMail: permite envio e recebimento de *e-mails*;
- Java Transaction API: fornece coordenação de transações distribuídas;
- JMS: permite troca de mensagens e eventos assíncronos;
- JAXP e JAXM: permite manipulação e troca de documentos XML;
- JNDI: fornece acesso a serviços de diretórios, como NIS e LDAP.

2.4.3 Tipos de Enterprise Java Beans

Existem 3 tipos de componentes EJB [13]:

- Session Bean: é um EJB cujo tempo de vida se limita à duração da sessão com um cliente, e serve apenas àquele cliente;
- Entity Bean: é um EJB cujo tempo de vida vai além de uma sessão e que pode ser compartilhado por vários clientes;
- Message Bean (ou *Message-Driven Bean*) é um EJB cujo tempo de vida vai além de uma sessão de um cliente, mas que serve apenas àquele cliente. É utilizado em operações assíncronas através do *Java Message System*.

Normalmente, os componentes *Session Beans* e *Message Beans* atuam como clientes de um ou mais componentes *Entity Beans*, a fim de desempenhar as suas funções. Os EJBs de uma aplicação freqüentemente requisitam os serviços de EJBs escritos para outras aplicações, e este é um dos benefícios em se utilizar uma arquitetura de objetos distribuídos [13].

2.5 Processo Unificado

Este modelo prevê o desenvolvimento de sistemas por iterações que geram produtos intermediários bem definidos, sendo que as iterações devem ser definidas para cada tipo de projeto. Isso reflete um modelo mais real de desenvolvimento. Além disso, o Processo Unificado permite a implantação efetiva de práticas comercialmente comprovadas de desenvolvimento de *software*, chamadas “boas práticas” [14]. São elas:

- Desenvolver *software* iterativamente;
- Gerenciar requisitos;
- Usar arquiteturas baseadas em componentes;
- Modelar *software* visualmente;
- Verificar a qualidade do *software*;
- Controlar mudanças no *software*.

O Processo Unificado pode ser descrito em duas dimensões: uma dimensão representa o aspecto dinâmico do processo, e é expressa em termos de ciclos, fases, iterações e *milestones*; a outra dimensão apresenta o aspecto estático do processo, ou seja, como ele é descrito em termos de atividades, artefatos, recursos e fluxos de trabalho [14].

Da perspectiva dinâmica, o Processo Unificado é dividido em 4 fases, sendo elas [15]:

- Concepção: estabelecimento da viabilidade do sistema através dos requisitos mais relevantes;
- Elaboração: definição de uma arquitetura estável e estimativa do custo da fase de construção, com base no refinamento de requisitos;
- Construção: produção de um produto de *software* pronto como versão inicial operacional;
- Transição: transferência do sistema para o ambiente do usuário.

Da perspectiva estática, o Processo Unificado é dividido em atividades que podem ser agrupadas nos seguintes fluxos de trabalho [15]:

- Atividades Técnicas: compreende as atividades normalmente consideradas no desenvolvimento de *software*: requisitos, análise, projeto, implementação e testes;
- Gerência: compreende as atividades de planejamento e controle do projeto;
- Ambiente: compreende as atividades de automação do processo e desenvolvimento do ambiente de manutenção;
- Avaliação: compreende as atividades de avaliação das tendências do processo e da qualidade do produto;
- Implantação: compreende as atividades para transferir o produto final para o usuário.

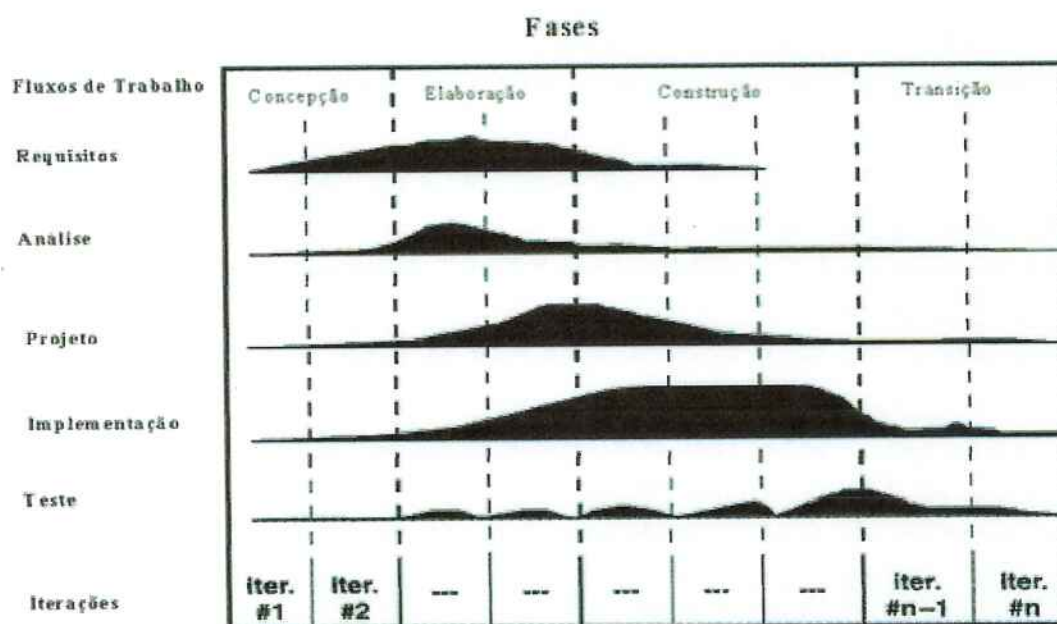


Figura 7 – As duas dimensões do Processo Unificado [15].

2.6 Processo em Cascata

O Processo em Cascata é o mais antigo sistema estruturado de desenvolvimento. Embora nos últimos anos tenha sido alvo de ataque por ser rígido e fora da realidade, ainda é largamente utilizado, pois atende de maneira rápida as necessidades do cliente [16]. É considerado a base de outros modelos de processo, pois é praticamente um modelo “genérico” para desenvolvimento de software.

O Processo em Cascata consiste das seguintes fases [16, 17]:

- Engenharia do Sistema: refere-se ao estudo de todos os aspectos dos processos de negócio desejados, com o objetivo de determinar como tais aspectos se relacionam entre si, e quais aspectos serão incorporados ao sistema [16];
- Análise de Software: esta fase consiste do levantamento dos requisitos do software, com o objetivo de determinar como eles serão implementados. É essencial que haja intensa comunicação entre cliente e desenvolvedor [16];
- Projeto: uma vez que os requisitos foram coletados e analisados, é preciso identificar detalhadamente como o sistema será construído para realizar as tarefas necessárias. Mais precisamente, a fase de Projeto é focada nos requisitos de dados (quais informações serão processados pelo sistema?), na construção do *software* (como a aplicação será construída?) e na construção da interface (quais padrões serão seguidos?) [16];
- Implementação: também conhecido como programação ou codificação, essa fase envolve a criação do sistema de *software* em si. Requisitos e especificações do sistema são transformados em código de máquina através de técnicas de programação [16];

- Testes: como o *software* é criado e adicionado ao sistema em desenvolvimento, testes são necessários para assegurar que este funciona corretamente e eficientemente. Os testes são focados geralmente em 2 áreas: eficiência interna e eficácia externa. O objetivo dos testes de eficácia externa é verificar se o *software* funciona de acordo com o projetado, e se implementa todas as funcionalidades necessárias. A meta dos testes de eficiência interna é garantir que o código é eficiente, padronizado, e bem documentado [16];
- Manutenção: é praticamente impossível entregar ao cliente um sistema que não possua defeitos, e por isso este processo é de suma importância para tratar as falhas quando se manifestam. A atividade de manutenção do *software* pode exigir novas iterações em qualquer uma das fases de desenvolvimento, devido a correção de defeitos, mudanças de requisitos, atualização do sistema, etc. Resultados: sistema modificado (programas e documentos) [15].

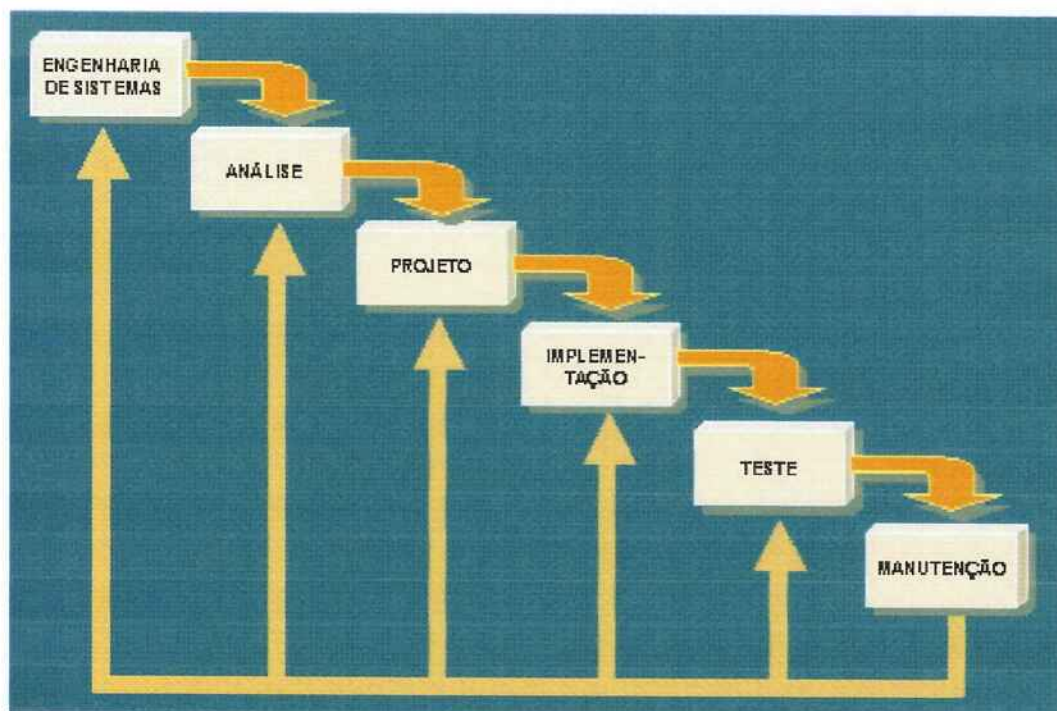


Figura 8 – Fases do Processo em Cascata [16].

3 ESPECIFICAÇÃO DO PROJETO DE FORMATURA

Este capítulo contém as informações relativas ao projeto e implementação do projeto de formatura. São apresentadas a especificação de requisitos, a arquitetura e a modelagem do Sistema Tanamão.

3.1 Requisitos Funcionais

Os requisitos funcionais do Sistema Tanamão foram elaborados a partir do objetivo e do escopo do Projeto de Formatura. São eles:

- O sistema deve ser acessível pelo consumidor através do protocolo WAP;
- O acesso do consumidor deve ser controlado através de usuário e senha;
- Efetuar pesquisa de produtos ou serviços de diferentes fornecedores/prestadores, através da comunicação com sistemas de diferentes arquiteturas;
- Possibilitar a realização de transações financeiras seguras para viabilizar a compra de produtos pelo portal;
- Gerenciar o relacionamento com o cliente, ou seja, implementar uma lógica de CRM que possibilite armazenar, organizar e tratar as informações que forem geradas em cada contato com o cliente;
- Utilizar tais informações para personalizar a interface com o usuário, e ainda, auxiliar estrategicamente as campanhas de marketing do portal.

3.2 Requisitos Não-Funcionais

Os requisitos não-funcionais do sistema são os seguintes (não foram alterados após a revisão):

- Usabilidade: os usuários do sistema serão principalmente consumidores preocupados com a simplicidade e a facilidade na busca de informações. Logo a interface deve ser baseada em objetos de utilização intuitiva (hipertexto), possibilitando sua utilização por uma pessoa de formação média usuária da Internet;
- Confiabilidade: tratando-se de um sistema que realiza transações comerciais e financeiras, torna-se necessário a existência de mecanismos que assegurem seu correto funcionamento, com uma confiabilidade estimada de 95%.
- Disponibilidade: apesar dos prejuízos que podem ser ocasionados por períodos de indisponibilidade, não se trata de um sistema crítico. Logo sua disponibilidade não precisa ser elevada, estima-se um valor em torno de 90%;
- Segurança de acesso (*security*): como dados confidenciais de clientes estarão sendo armazenados e processados, é vital que a segurança de acesso seja uma preocupação na implementação do sistema. Assim, o tráfego de dados deve ser utilizar-se de criptografia, e ainda, deve existir uma política de controle de acesso aos dados e arquivos do *software* (observação: o escopo do projeto não inclui segurança dos dados, sendo recomendada para uma futura implementação comercial);
- Manutenibilidade: deve ser garantida através da implementação de arquitetura em camadas e da utilização de protocolos e tecnologias baseados em padrões;
- Portabilidade: deve ser garantida através da utilização da tecnologia Java.

3.3 Características dos Usuários

É importante prever as características dos usuários para que seja feito um projeto eficiente das interfaces homem-computador. Assim sendo, foram identificados três tipos de usuários para o Sistema Tanamão, a saber:

- Consumidores: são os principais usuários, público-alvo do sistema. Estima-se que possuam formação de nível médio, tenham experiência básica no uso de computadores, para acesso à Internet, e na utilização de aparelhos celulares, para acesso ao portal via WAP. Em geral não conhecem os processos de negócio;
- Fornecedores: são usuários esporádicos que acessarão o sistema apenas via Internet. Estima-se que possuam formação de nível médio e conheçam alguns aspectos dos processos de negócio para que possam utilizar as funções do sistema;
- Administradores: são usuários freqüentes que utilizarão funções de administração. Estima-se que conheçam bem os processos de negócio e possuam conhecimento avançado sobre a arquitetura do sistema.

De acordo com as informações apresentadas acima, pode-se definir que as interfaces devem seguir os modelos normalmente encontrados na Internet e no WAP, ou seja, páginas compostas por formulários, botões, *hyperlinks* e outros objetos de hipertexto que tornam a navegação bastante intuitiva.

Além disso, é possível projetar quais tipos de informação devem ser exibidos, e quais podem ser omitidos ou serem incorporados de maneira implícita. Os resultados desta análise apresentam-se no projeto de interface apresentado no tópico 3.8 e no Anexo B – Estudo da IHC do Sistema Tanamão.

3.4 Restrições

As principais restrições ao projeto são:

- Limitação de hardware: o computador a ser utilizado como servidor do sistema não terá uma configuração ideal, por uma questão de custo;
- Dependência de outros sistemas: a integração com sistemas de fornecedores e bancos será simulada através de um protótipo a ser desenvolvido paralelamente ao Sistema Tanamão, para manter a independência de fatores externos ou interesses de terceiros;
- Limitações de prazo: como não se trata de uma aplicação crítica, será dada mais atenção à confiabilidade do que à disponibilidade. Mas ambos os requisitos serão atendidos de acordo com a disponibilidade de tempo. A segurança (*security*) do sistema deve ser questão de alta prioridade numa implementação comercial, para garantir a privacidade dos clientes. Porém, como este não é o foco do projeto, não será implementado nenhum tipo de criptografia nas transações financeiras do protótipo.

3.5 Arquitetura do Sistema

Partindo da definição do problema – criar um sistema de busca, comparação de preços e venda de produtos, acessível por dispositivos móveis e com personalização da interface com o consumidor – analisamos o cenário em que se aplicaria o Sistema Tanamão. E, com base em sua interface com os consumidores e com os fornecedores, estabelecemos a arquitetura macro apresentada na Figura 9.

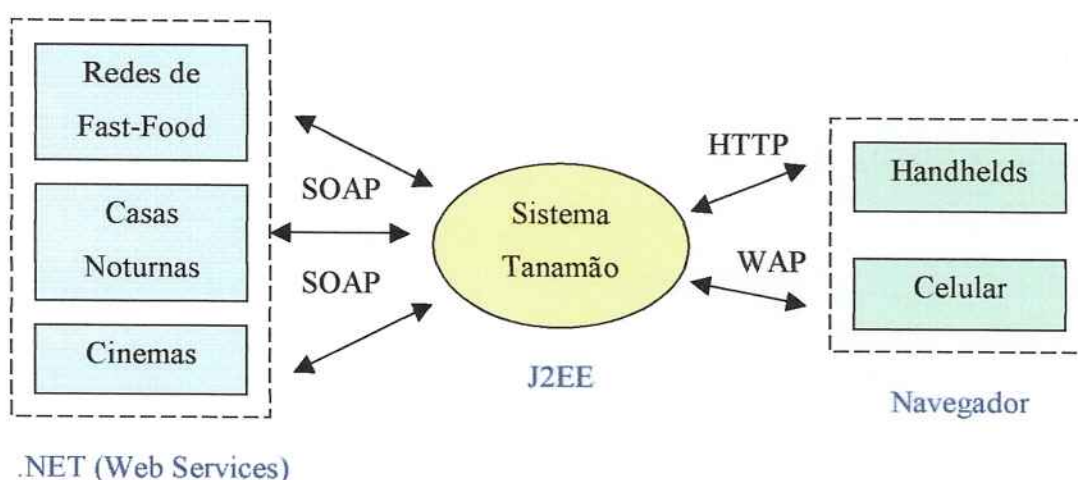


Figura 9 – Arquitetura macro do Sistema Tanamão.

Os fornecedores de produtos irão disponibilizar *Web Services* que permitirão ao Sistema Tanamão consultar dados dos produtos, efetuar pedidos e pagamentos. O padrão *Web Services* foi escolhido por proporcionar interoperabilidade entre diferentes sistemas (como foi discutido em 1.1.3). A interface será então baseada na troca de mensagens XML via protocolo SOAP.

Já a interface com o consumidor será baseada nos protocolos HTTP e WAP, pois já são bastante difundidos e não exigem instalação adicional de nenhum *software* nos dispositivos móveis.

Este desenho do sistema é baseado na arquitetura básica de *Web Services*, onde existem, geralmente:

- Um Servidor de Serviços, que fornece os *Web Services*;
- Um Servidor de Aplicação, que consome os serviços *Web Services*;
- Um Servidor de Publicação, que publica os serviços *Web Services*;

Embora a implementação deste último seja simples, não será realizada, pois foge do escopo do nosso problema. Partiremos do princípio de que os serviços já são conhecidos, ou seja, não existe a necessidade de sua publicação.

Como já foi discutido anteriormente, o Servidor de Serviços não é parte integrante do Sistema Tanamão. No modelo de negócio estabelecido, os *Web Services* são oferecidos pelos fornecedores, de acordo com suas estratégias e objetivos. Porém, para efeito de simulação e testes, construiremos alguns protótipos de Servidores de Serviços, utilizando a plataforma.NET para reforçar o aspecto da interoperabilidade.

O Servidor de Aplicação é o próprio Sistema Tanamão, responsável por toda a lógica de negócio (a ser implementada com *Enterprise Java Beans*), interface homem-máquina (que utilizará *Java Server Pages*) e persistência de dados (com banco de dados MySQL). Será implementado baseado na arquitetura J2EE, utilizando o servidor de aplicação JBoss 3.2.5.



Figura 10 – Arquitetura J2EE em 3 camadas para o servidor de aplicação.

Com a junção das duas arquiteturas apresentadas, é possível obter a arquitetura completa do sistema, baseada em camadas e em componentes (Figura 11). As vantagens desta arquitetura são:

- Permite clientes e servidores de diferentes plataformas;
- Não demanda a instalação de aplicativos no cliente (apenas um *software* de navegação é requerido);
- *Containers* disponibilizam o *Middleware*, um conjunto de serviços padrões;
- Possui maior reusabilidade devido ao uso de componentes;
- Reduz o tempo de desenvolvimento e custo;
- Aumenta a qualidade e a interoperabilidade.

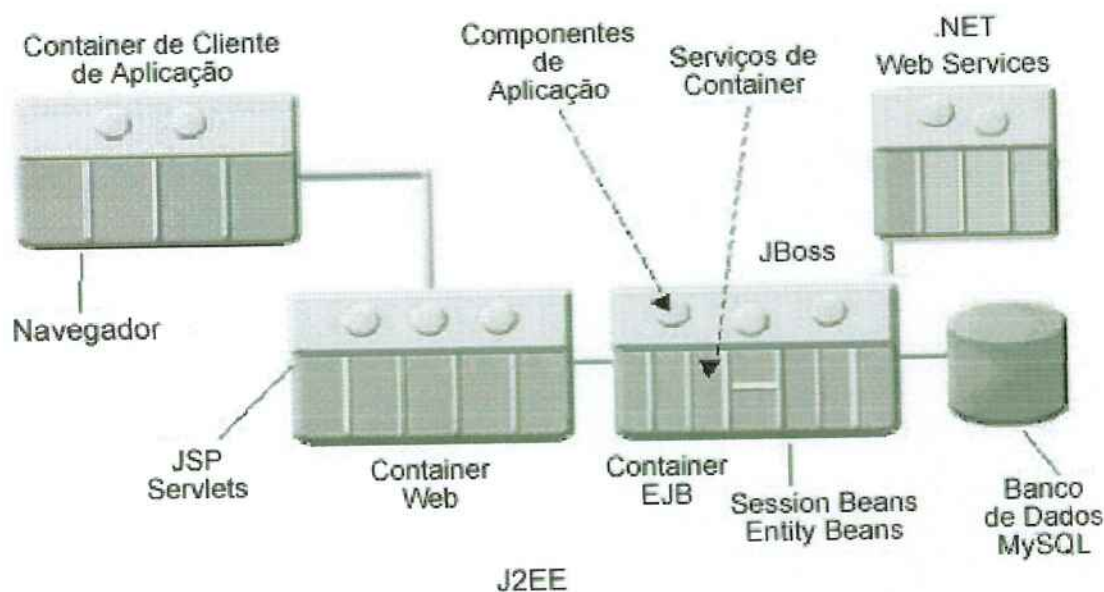


Figura 11 – Arquitetura completa do Sistema Tanamão.

3.6 Modelo de Casos de Uso

O Diagrama de Casos de Uso preliminar, desenvolvido na fase de especificação do sistema, é apresentado na Figura 12. Foi modelado a partir das funções do software identificadas inicialmente, e oferece uma visão do sistema do ponto de vista do usuário consumidor. Possui os seguintes casos de uso: Cadastrar, Login, Pesquisar Preços e Comprar Produto/Serviço. O único ator presente, Consumidor, representa as interações do usuário consumidor com o Sistema Tanamão.

Durante a fase de revisão da especificação do projeto, foram identificados novos requisitos para o sistema, conforme discutido no capítulo anterior. Em virtude disto, novas funcionalidades foram incorporadas ao software, provocando a necessidade de uma revisão no Modelo de Casos de Uso. Além disso, notou-se que o modelo fornecia uma visão superficial do sistema e não abrangia todos os pontos especificados nos requisitos.

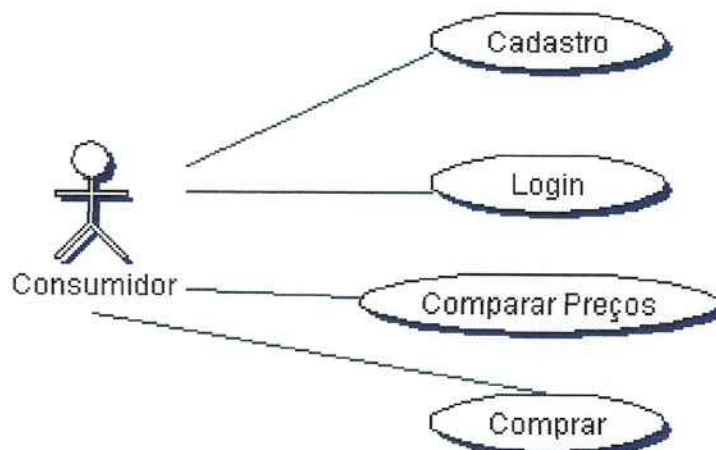


Figura 12 – Diagrama de Casos de Uso preliminar.

Conseqüentemente, havia a necessidade em acrescentar ao modelo os casos de uso relacionados à lógica de negócio, tais como comunicação com os *Web Services* e a lógica de CRM. Isto permitiria uma visão mais transparente e detalhada do funcionamento do sistema e da sua interação com os usuários.

A fim de atender a tais necessidades, foi feita a redefinição do Modelo de Casos de Uso, conforme ilustrado na Figura 13. Efetuou-se a divisão lógica do sistema em 4 subsistemas, conforme critério de semelhança funcional dos casos de uso:

- Interface Consumidor: agrupa os casos de uso relacionados à interface com o usuário consumidor, representando a camada de apresentação das funcionalidades e de ativação das principais funções do *software*. Estes casos de uso já estavam previstos no modelo preliminar. Seu principal ator é o Consumidor. Mais detalhes no capítulo 1.2 do Anexo A;
- Interface Fornecedor: contém os casos de uso que implementam a interface do sistema com o usuário fornecedor de produtos ou serviços. Não estava prevista no modelo preliminar, sendo adicionada para atender à especificação revisada dos requisitos. Seu principal ator é o Fornecedor. Mais detalhes no capítulo 1.3 do Anexo A;
- Interface Administrador: apresenta a interface do sistema com o usuário administrador, seus casos de uso foram elaborados para atender aos requisitos revisados. Seu principal ator é o Administrador. Mais detalhes no capítulo 1.4 do Anexo A;
- Lógica de Negócio: engloba os casos de uso que implementam a lógica de negócio, representando funções de CRM, de comunicação com *Web Services* e de persistência de dados. Estes casos de uso são apresentados para facilitar o entendimento sobre o funcionamento do sistema. Não possuem atores, pois são na realidade inclusões de outros casos de uso. Mais detalhes no capítulo 1.5 do Anexo A.

O Modelo de Casos de Uso revisado mostrou-se adequado e o grupo concluiu que este ilustrava satisfatoriamente o funcionamento do sistema, possibilitando o prosseguimento dos trabalhos. O grupo refletiu então, por algum tempo, sobre as razões pelas quais o modelo preliminar tinha se apresentado tão insipiente e pouco representativo, e listou-as:

- Falta de planejamento sobre o tempo gasto na elaboração do modelo;
- Deficiência no levantamento e definição dos requisitos do sistema;
- Simplificação das funções do *software*, causando uma visão superficial do sistema;
- Desatenção à grande representatividade visual que o modelo oferece.

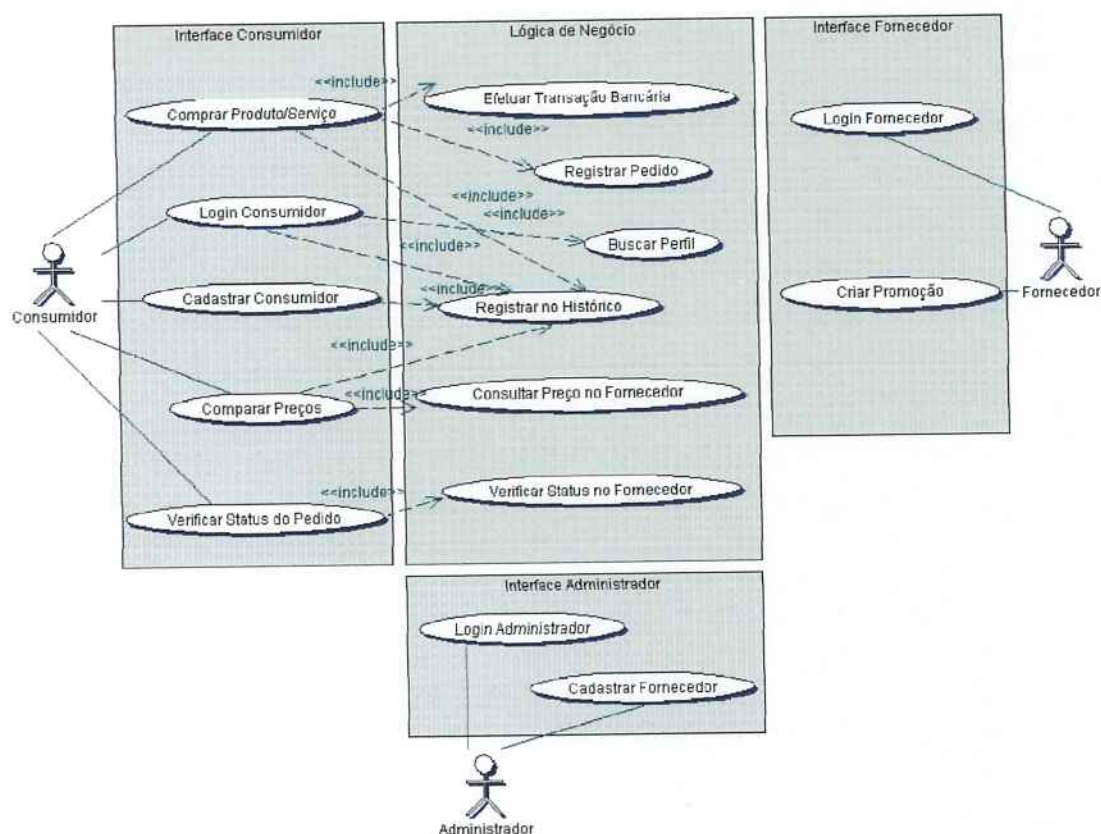


Figura 13 – Diagrama de Casos de Uso do Sistema Tanamão.

Concluiu-se então que não existem modelos certos ou errados para um dado sistema, mas sim modelos adequados e modelos pouco representativos, e cabe ao analista, baseado em sua experiência e bom senso, avaliar qual a dimensão correta para a modelagem dos casos de uso.

3.7 Modelo de Classes

O Modelo de Classes inicial (Figura 14) foi elaborado a partir dos casos de uso identificados na primeira fase do projeto de formatura. Como foi citado no tópico anterior, foram feitas mudanças drásticas na especificação, resultando em uma significativa alteração das classes do sistema. Vale lembrar que, como a implementação foi baseada na arquitetura J2EE, a maior parte das classes representa na realidade o papel lógico que os *Enterprise Java Beans* desempenham no sistema.

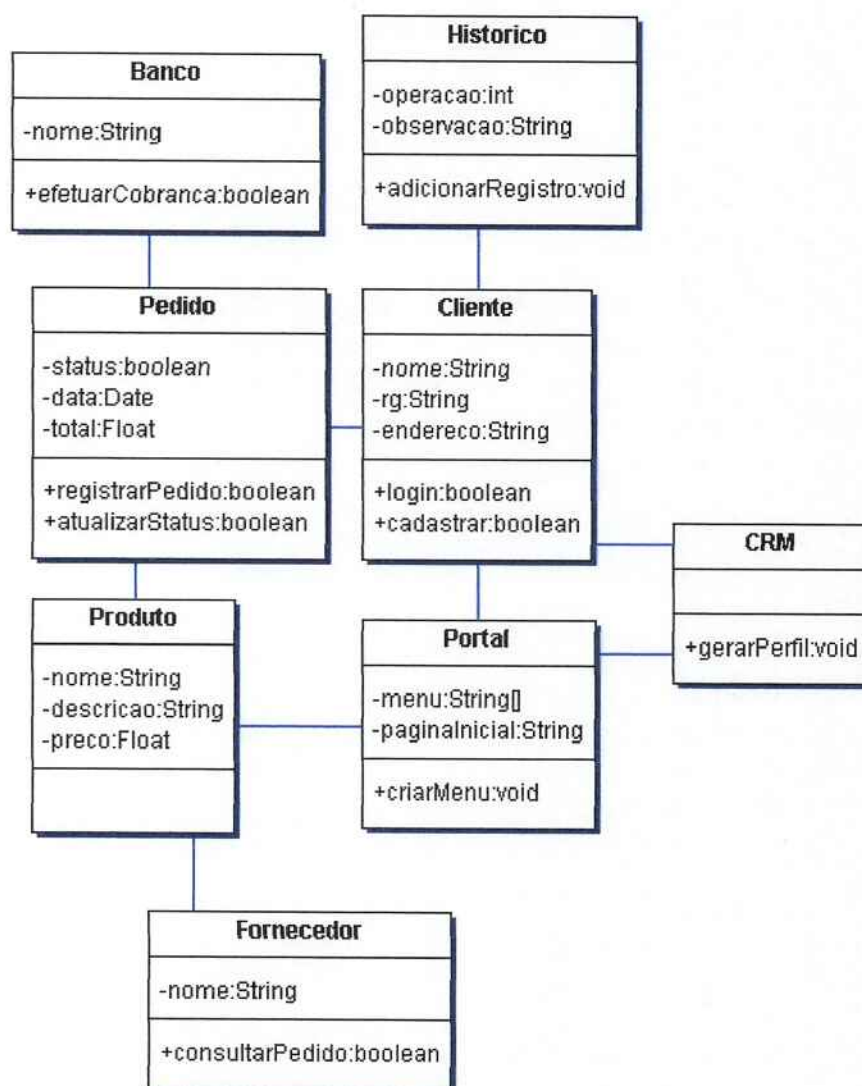


Figura 14 – Diagrama de Classes preliminar do Sistema Tanamão.

Segue uma descrição sucinta das classes identificadas inicialmente no sistema:

- Classe Cliente: é a representação do cliente no sistema. Armazena os dados pessoais, profissionais e preferências do cliente, além de implementar as funções que refletem a atuação do cliente, tais como login e cadastro;
- Classe Produto: armazena as informações referentes ao produto ou serviço, tais como nome, descrição, preço unitário, etc;
- Classe Pedido: armazena informações (data, total, status) relativas a compras efetuadas pelo cliente;
- Classe Fornecedor: é responsável pela comunicação (interface) com os sistemas dos fornecedores;
- Classe Banco: é responsável pela comunicação (interface) com os sistemas dos bancos e outras instituições financeiras;
- Classe Histórico: armazena informações de quaisquer contatos realizados com o cliente, seja via campanhas de *marketing*, seja via eventos de compra de produtos;
- Classe Portal: é responsável pela geração da interface personalizada com o cliente, através da utilização de menus, páginas personalizadas, ferramentas de busca, etc;
- Classe CRM: implementa a lógica de CRM, ou seja, traça perfis conforme histórico e dados do cliente e determina a atuação do sistema em estratégias de *marketing*.

Durante a fase de revisão da especificação, como já foi dito, algumas mudanças foram feitas neste modelo. Além disso, no decorrer da implementação do sistema, foram detectadas algumas necessidades e restrições não identificadas anteriormente. Por exemplo, como a arquitetura J2EE estabelece uma divisão das classes em classes de persistência de dados e classes de processamento, foi preciso dividir alguns métodos e agrupar outros. E ainda, era necessário adicionar classes que tratassem a comunicação com os *Web Services* dos fornecedores e dos bancos.

Assim sendo, elaboramos o diagrama de classes final do Sistema Tanamão, que é apresentado a seguir (Figura 15):

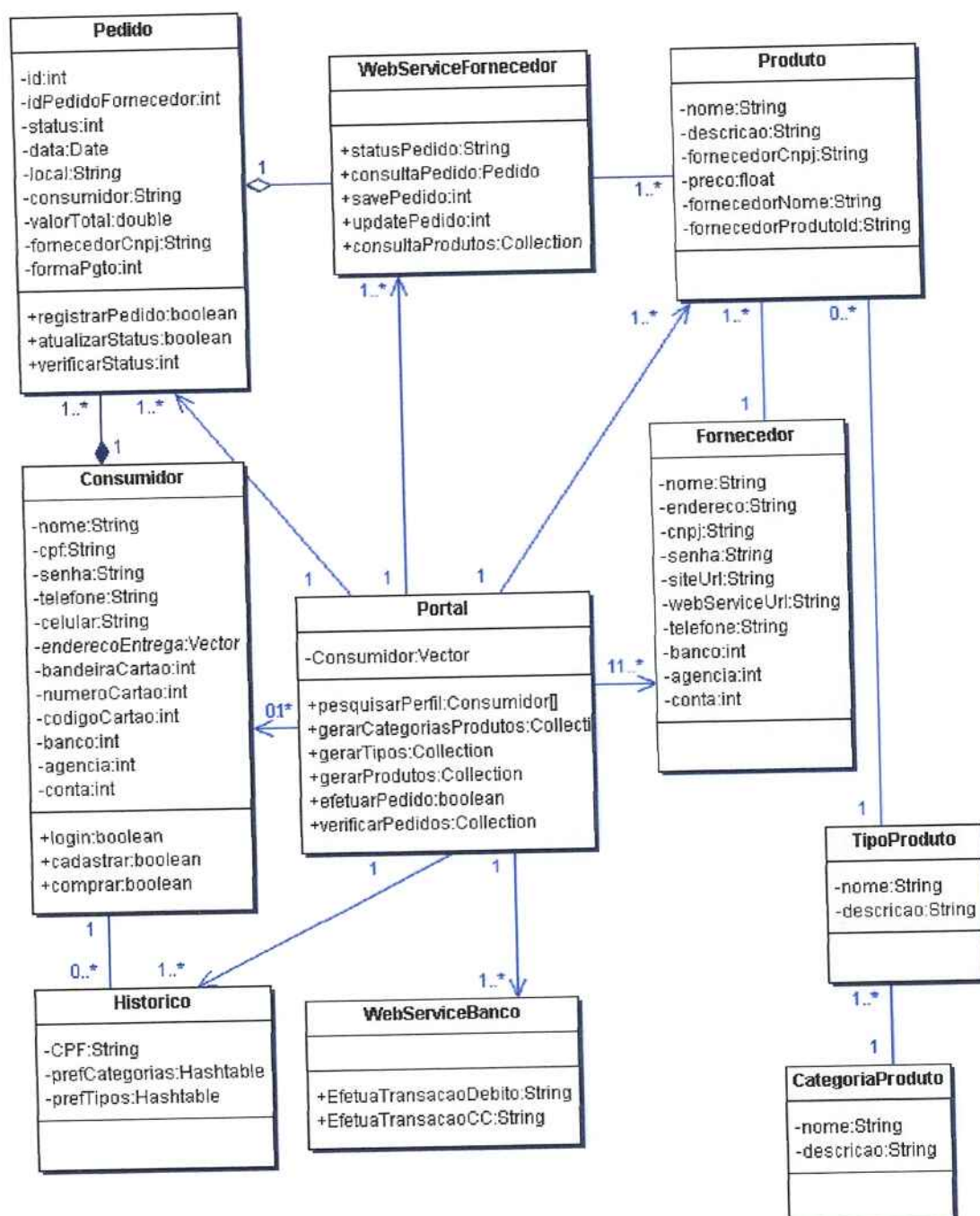


Figura 15 – Diagrama de Classes final do Sistema Tanamão.

Segue a descrição resumida das classes implementadas no Sistema Tanamão:

- Consumidor: informações gerais sobre o consumidor, o usuário do nosso sistema. Contém, além do seu cadastro, informações sobre endereços possíveis de entrega, além de dados bancários (cartão de crédito/conta). É implementada por um *Entity Bean*;
- Fornecedor: informações cadastrais do fornecedor (nome, endereço, telefone, etc), além da URL utilizada para contato com o seu *Web Service* e dados bancários (para efetuar o pagamento dos produtos solicitados a ele). É implementada por um *Entity Bean*;
- Histórico: armazena informações sobre a atividade do usuário no sistema, para utilização no CRM. Com isto, será possível personalizar a interface para cada usuário. É implementada por um *Entity Bean*;
- Pedido: é o registro do pedido no banco de dados local do Sistema Tanamão, para qualquer produto de qualquer fornecedor. Independe do banco de dados do fornecedor, que terá um registro próprio dos seus pedidos (este será feito através da interação por *Web Services*). É implementada por um *Entity Bean*;
- Produto: é a descrição de um produto, acrescida de informações sobre o fornecedor correspondente. É implementada por uma classe Java comum;
- Portal: é a classe principal do Sistema Tanamão. Provê a interface para todos os serviços disponibilizados. As páginas JSP geradas para o usuário utilizarão somente métodos desta classe, sendo esta uma fachada para os métodos de negócio (implementados pelos *Session Beans*). Desta forma, a camada de interface homem-máquina não necessita saber detalhes de implementação, apenas se comunicar com o Portal. Dentre as funcionalidades providas, podemos citar: a obtenção da lista de produtos, a execução de um pedido e o registro de atividades do usuário. É implementada por um *Session Bean*;
- TipoProduto/CategoriaProduto: cada produto está classificado num tipo de produto e uma categoria de produto. Isto é utilizado para agrupar os itens, de forma a facilitar a navegação (melhorando a usabilidade). É implementada por um *Entity Bean*;

- WebServiceBanco: através desta classe podemos realizar transações bancárias, efetuando pagamentos do cliente diretamente para o fornecedor. Isto pode ser feito através de cartões de crédito ou de débito bancário. É implementada por um *Session Bean*;
- WebServiceFornecedor: essa classe tem por finalidade encapsular as chamadas para os webservices de qualquer fornecedor, fornecendo uma interface padronizada para consultar produtos, realizar pedidos e consultar status de pedidos. Esses *Web Services* podem estar em qualquer servidor. Basta que tenhamos a url cadastrada no sistema para permitir oferecer seus produtos e serviços. Desta forma, é possível a integração direta com vários fornecedores. É implementada por um *Session Bean*.

Como pode ser notado, foram incluídas as classes *WebServiceFornecedor* e *WebServiceBanco* para tratar a comunicação com os *Web Services*. Além disso, foram adicionadas as classes *TipoProduto* e *CategoriaProduto* para realizar a classificação dos produtos no portal. Anteriormente, tal classificação estava projetada para ser feita apenas no banco de dados, porém a estrutura de *Entity Beans* exige que as classes reflitam as entidades do banco de dados. Houve ainda a incorporação das funções da classe CRM pela classe Portal, pois notamos que a primeira tinha era uma classe “artificial”, sem representatividade no desenho do sistema.

Apesar de haver alterações significativas no modelo de casos de uso durante a revisão, e na estrutura de implementação das classes durante a fase de implementação, pode-se notar que o modelo de classes final está bastante próximo do modelo preliminar. Isto mostra que este último, ao contrário do que ocorreu no caso do modelo de casos de uso (tópico 3.6), já representava adequadamente o funcionamento lógico interno do Sistema Tanamão. Conclui-se então que o grupo já possuía conhecimento suficiente sobre o sistema ao elaborar o modelo de casos de uso, porém, não soube como modelá-lo formalmente.

3.8 Modelo da Interface Homem-Computador

A interface homem-computador do Sistema Tanamão é dividida em dois subsistemas: o portal WML, que implementa o sistema de busca e venda, e o portal HTML, que disponibiliza o formulário para cadastro de consumidores, o sistema de administração e a interface com os fornecedores de produtos e serviços.

Baseando-se nos requisitos do sistema, nas características dos usuários (neste caso, os consumidores do portal), e nas limitações do WAP/WML, desenvolvemos a estrutura de navegação do sistema para a interface com o consumidor (Figura 16).

Esta estrutura foi desenvolvida com foco em três metas: prover alta usabilidade aos usuários, diminuir a sobrecarga cognitiva e equilibrar a dinamicidade da interface. Tais aspectos são discutidos detalhadamente no Anexo B – Estudo da IHC do Sistema Tanamão.

Já o segundo subsistema, a interface HTML, possui importância secundária no sistema, e por isso não foi o alvo principal de nossos esforços. A maior parte das páginas consiste de formulários de cadastro sem necessidade de lógica, e por isso são de simples implementação. O Diagrama de Estados de Navegação desta interface é apresentado na Figura 17.

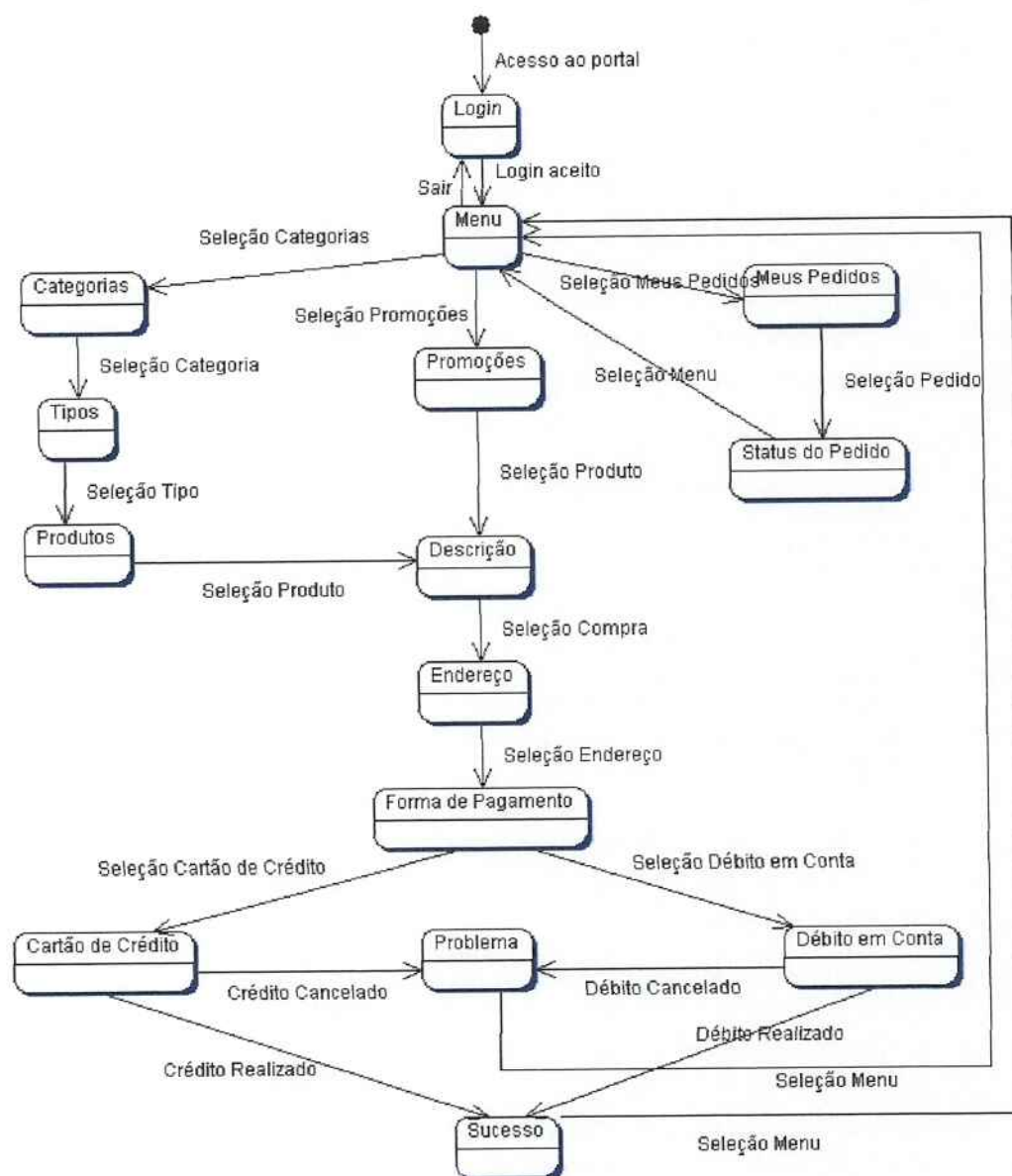


Figura 16 – Diagrama de Estados de Navegação da interface WAP/WML.

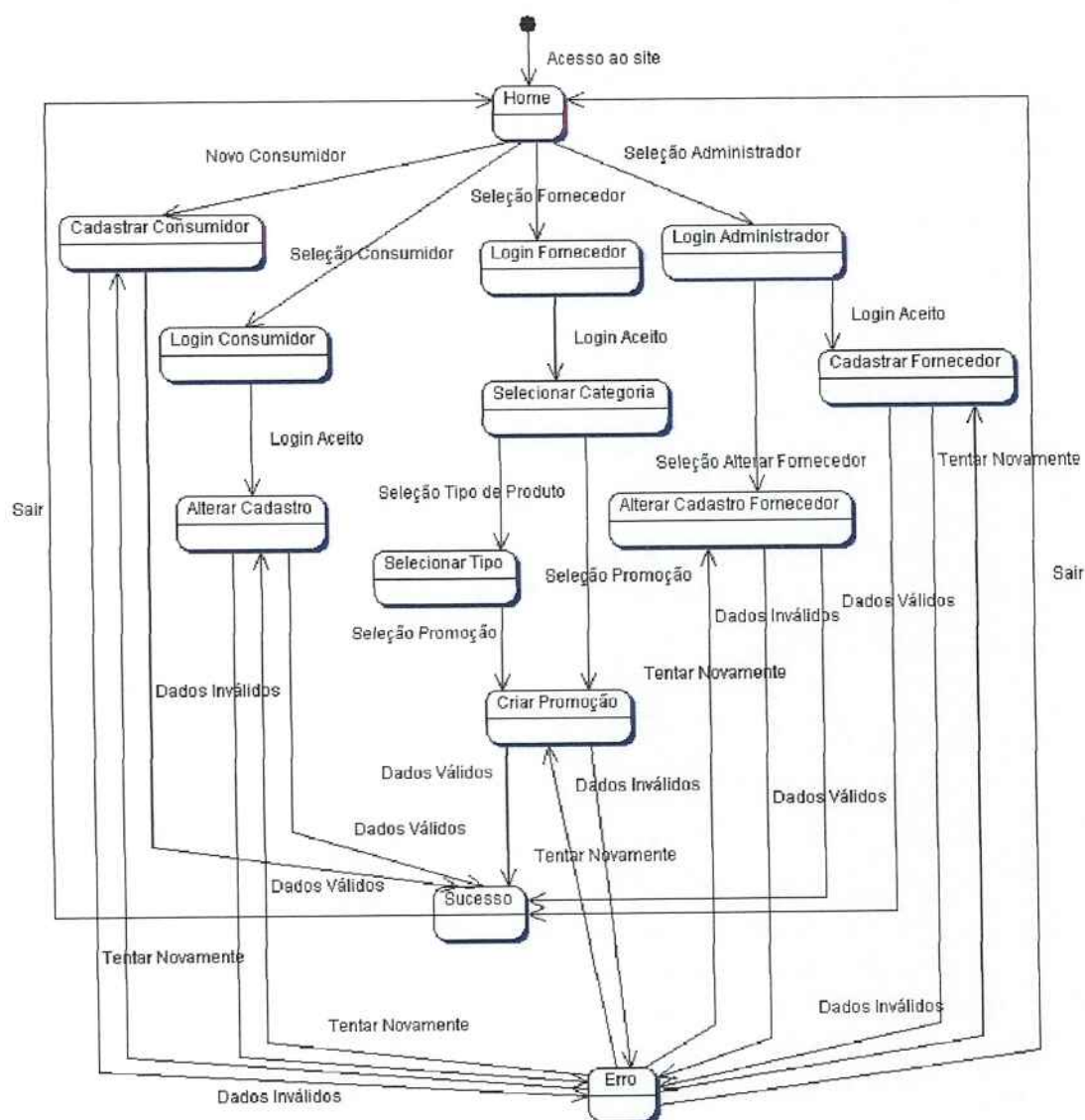


Figura 17 – Diagrama de Estados de Navegação da interface HTTP/HTML.

3.9 Metodologia

O Processo Unificado, que utiliza técnicas aplicadas a grandes equipes, foi escolhido inicialmente como metodologia de desenvolvimento de *software*, devido ao interesse do grupo no aprendizado deste modelo.

Com o início das atividades do projeto, os integrantes do grupo começaram a sentir dificuldades em seguir as diretrizes do Processo Unificado. Todos já conheciam e tinham trabalhado com o Processo em Cascata, e traziam consigo “vícios” que se transformaram em obstáculos durante o andamento do projeto. Além disso, o Processo Unificado é reconhecidamente mais adequado a grandes equipes de desenvolvimento, onde as atividades se desenrolam paralelamente, onde a rastreabilidade de requisitos é essencial, onde a flexibilidade do processo tem grande influência sobre o sucesso do projeto.

Por isso, como a equipe do projeto é reduzida e os requisitos do sistema são claramente conhecidos desde o início do projeto, decidimos então mudar a metodologia, passando a adotar o Processo em Cascata.

O Processo em Cascata apresenta fases distintas e separadas de especificação e desenvolvimento, sem execução concorrente de atividades. Devido a isso, dificulta-se a acomodação de mudanças depois que o processo está em execução, e torna-se inviável responder a mudanças de requisitos. Por isso, é mais apropriado quando os requisitos são bem compreendidos desde o início do projeto, como é o caso do Sistema Tanamão.

Utilizando a metodologia do Processo em Cascata, definimos então o seguinte plano para nosso projeto:

Fase	Atividades	Produtos	Período
Engenharia do Sistema	<ul style="list-style-type: none"> • Definição do escopo do sistema; • Avaliação de sistemas atualmente em uso; • Definição de metas e objetivos do sistema; • Estudo da viabilidade de automatizar; • Estimativa de tempo e recursos; • Escolha da metodologia de projeto; • Identificação de recursos de <i>hardware</i> e <i>software</i>. 	<ul style="list-style-type: none"> • Viabilidade do sistema; • Especificação dos requisitos do sistema. 	Janeiro a Fevereiro
Análise	<ul style="list-style-type: none"> • Identificação dos requisitos de <i>software</i>; • Definição do balanceamento entre operação humana e automação; • Geração de modelo independente de tecnologia; • Especificação dos critérios de aceitação. 	<ul style="list-style-type: none"> • Especificação dos requisitos de <i>software</i>. 	Fevereiro a Março

Fase	Atividades	Produtos	Período
Implementação	<ul style="list-style-type: none"> • Estabelecimento de padrões de codificação; • Seleção de pacotes e bibliotecas; • Transformação em código executável. 	<ul style="list-style-type: none"> • Código. 	Julho a Outubro
Testes	<ul style="list-style-type: none"> • Comparação do sistema com a especificação; • Verificação do funcionamento interno e externo dos programas; • Aceitação; • Instalação. 	<ul style="list-style-type: none"> • Sistema instalado. 	Novembro
Manutenção	<ul style="list-style-type: none"> • Alterações no sistema devido a erros, a mudanças de plataforma de <i>software</i> ou a mudanças de <i>hardware</i>. 	<ul style="list-style-type: none"> • Sistema modificado (programas e documentos) 	Dezembro

Tabela 1 – Planejamento de atividades dos recursos.

Optou-se por utilizar o paradigma de Orientação a Objetos nas fases de análise e implementação, pois este se indicou mais adequado à arquitetura do sistema (baseada no padrão J2EE). O cronograma detalhado das atividades da fase de implementação do projeto está no Anexo C, e os resultados obtidos com a aplicação da metodologia de Processo em Cascata são discutidos no capítulo 5.1.

4 PROJETO E IMPLEMENTAÇÃO

O objetivo deste capítulo é apresentar o Sistema Tanamão como produto final, ilustrar suas principais funcionalidades, listar os testes efetuados e os resultados obtidos, e ainda, realizar uma avaliação crítica sobre a conclusão do projeto.

4.1 Descrição do Produto Final

O Sistema Tanamão consiste de um portal WAP complementado por um *site* HTML que permite aos consumidores realizar cotação de preços e compra de produtos através de dispositivos móveis, preferencialmente celulares. Estas interfaces são geradas através de páginas JSPs compiladas em um servidor de aplicações da plataforma J2EE. Neste mesmo servidor são implementados componentes EJBs que realizam toda a lógica de negócio, ou seja, a personalização da interface, a comunicação com os *Web Services* e a persistência de dados.

Como produto final, apresentamos a interface homem-computador do portal WAP, visualizada através do emulador M3Gate [17]. Parte do conteúdo é personalizado de acordo com o perfil de interação do consumidor com o sistema. É importante frisar que o produto final é não só a interface do sistema, mas principalmente o conteúdo gerado, que é fundamentado nos princípios de CRM e na arquitetura de *Web Services*.

Todas as telas foram projetadas de acordo com a especificação da interface realizada no tópico 3.8. Além disso, foram observadas algumas convenções na construção do hipertexto, a fim de adequar o sistema ao estudo efetuado no Anexo B.

A Figura 18 apresenta a tela de inicial da interface WAP do Sistema Tanamão, ponto onde pode ser selecionada a opção “Login” para entrada no sistema. Se nada for selecionado em 5 segundos, a tela de Login carrega automaticamente.



Figura 18 – Sistema Tanamão: Tela inicial.

Na tela de Login (Figura 19), o usuário deve digitar o seu CPF e senha, que devem ter sido cadastrados previamente pela interface HTML.



Figura 19 – Sistema Tanamão: Tela de Login.

Na tela seguinte são apresentadas 3 opções para o consumidor (Figura 20): verificar promoções de interesse, acessar o menu de categorias ou verificar o status dos pedidos efetuados.



Figura 20 – Sistema Tanamão: Menu de opções.

Caso seja escolhida a opção de promoções, o sistema mostra produtos que estejam configurados como promoção para o perfil daquele consumidor (Figura 21). Por exemplo, se o usuário comprar muitos sanduíches pelo sistema, ofertas relacionadas a este produto irão aparecer. Cada consumidor receberá somente promoções de interesse.



Figura 21 – Sistema Tanamão: Promoções identificadas para o consumidor.

Caso a opção de verificar pedidos seja selecionada, é apresentada uma lista dos últimos pedidos registrados no sistema. Com a seleção de um dos pedidos, é exibido seu status.

Selecionando a opção de acessar categorias, é possível navegar pelo menu de produtos, organizados por categoria e tipo de produto. São mostrados, no último nível, os produtos de diversos fornecedores, com seu respectivo preço. Desta forma, pode ser feita a comparação entre produtos para uma melhor escolha (Figuras 22, 23 e 24).



Figura 22 – Sistema Tanamão: Categorias de produtos.



Figura 23 – Sistema Tanamão: Tipos de produtos.

Para a compra que a compra seja efetuada, alguns dados devem ser confirmados. O primeiro deles é o endereço de entrega. São exibidos para escolha os endereços já cadastrados no sistema. Adicionalmente, pode ser cadastrado um novo endereço de entrega através do portal (Figura 25).



Figura 24 – Sistema Tanamão: Listagem de produtos.



Figura 25 – Sistema Tanamão: Seleção do endereço para entrega.

No passo seguinte são solicitados os dados para o pagamento. Duas formas de pagamento são aceitas: cartão de crédito e débito em conta. Os dados da forma de pagamento escolhida são solicitados. Para cartões de crédito, é necessário informar ao sistema a bandeira, o número e o código do cartão. E para débito em conta, é preciso informar o banco, a agência, o número da conta e a senha. Estes dados (com exceção à senha) podem ser previamente cadastrados na interface HTML do Sistema Tanamão. Se tudo ocorrer como previsto, uma tela final de confirmação é exibida, apresentando os dados do pedido (Figura 26).



Figura 26 – Sistema Tanamão: Confirmação de transação de compra.

Omitimos algumas telas de menor importância, que em geral apresentam apenas mensagens de erro ou de sucesso em operações no portal.

O outro módulo do Sistema Tanamão, a interface HTML, é ilustrada a seguir. Como pode ser visto no diagrama de estados de navegação (Figura 17), trata-se de um sistema simples que complementa as funcionalidades do portal WAP.

Uma das principais funcionalidades implementadas na interface HTML é o cadastro de novos consumidores. Para melhorar a usabilidade do sistema, não é necessário que o consumidor se cadastre no portal Tanamão via WAP, já que isso pode se tornar cansativo devido à interface pouco amigável.

Desta forma, para ter acesso aos serviços do portal Tanamão o usuário pode se cadastrar previamente no sistema por uma interface HTML, mais adequada para este tipo de operação. A tela de cadastro para novos consumidores é exibida na Figura 27.

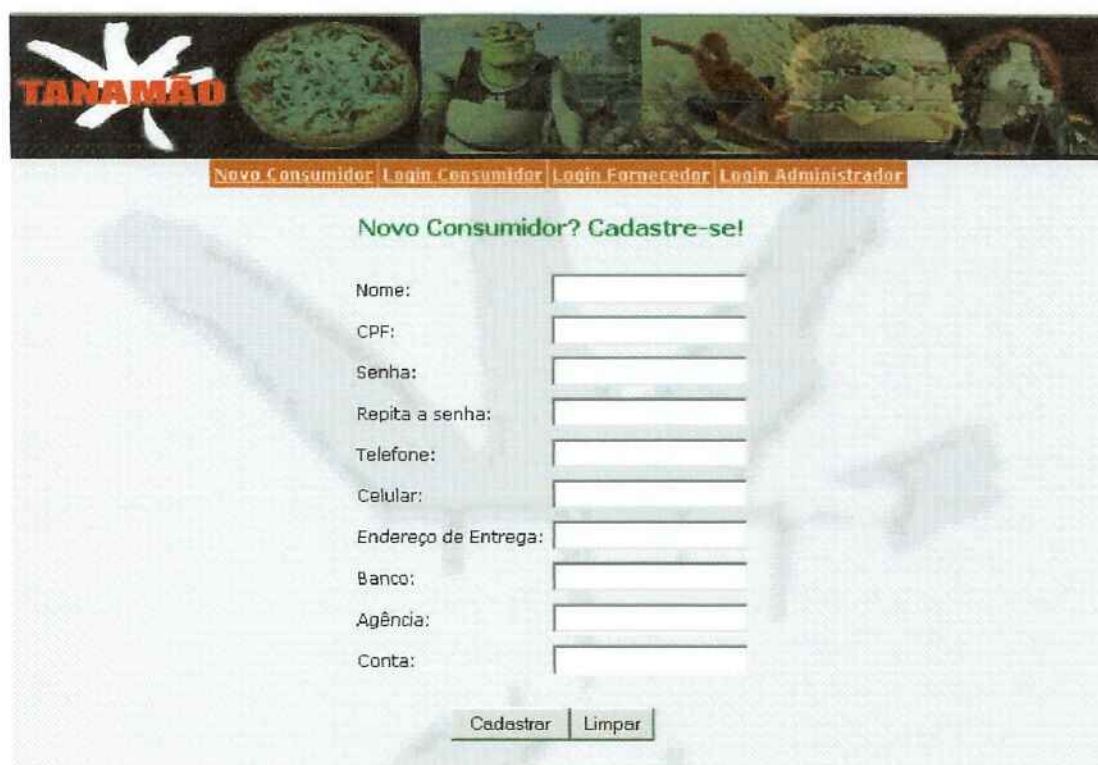
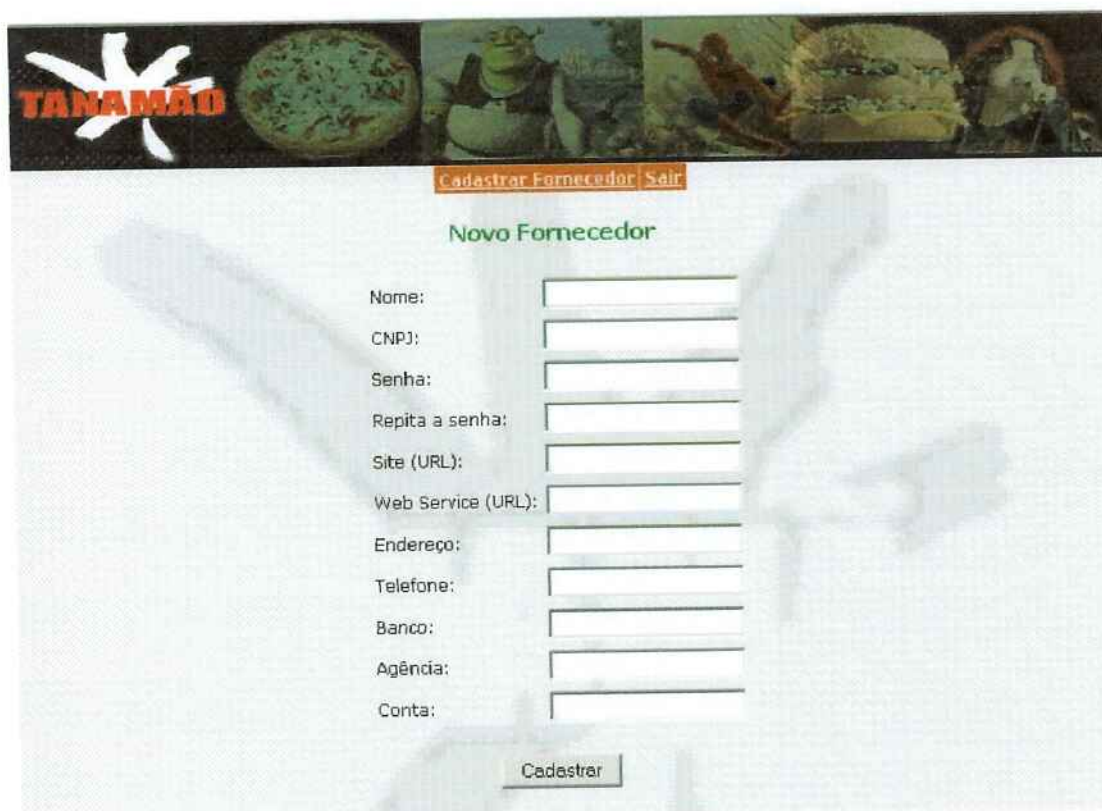


Figura 27 – Tela para cadastro de novos consumidores.

Nesta tela, o consumidor cadastra seu CPF que será usado como Login, sua senha, alguns dados para contato, um endereço de entrega e ainda dados de uma conta que pode ficar pré-cadastrada no sistema para futuras compras.

Outra tela de suma importância é o cadastro de fornecedor, que é realizado por usuários administradores do sistema. Através de um formulário são adicionadas informações necessárias para a inclusão de um novo fornecedor no mecanismo de cotação de preços do portal (Figura 28).



The screenshot shows a web interface for the 'Tanamão' system. At the top, there is a banner with the 'TANAMÃO' logo and several images of food items. Below the banner, there are two buttons: 'Cadastrar Fornecedor' and 'Sair'. The main heading is 'Novo Fornecedor'. The form contains the following fields with labels to their left: 'Nome:', 'CNPJ:', 'Senha:', 'Repita a senha:', 'Site (URL):', 'Web Service (URL):', 'Endereço:', 'Telefone:', 'Banco:', 'Agência:', and 'Conta:'. Each label is followed by a text input field. At the bottom of the form, there is a 'Cadastrar' button.

Figura 28 – Tela para cadastro de novos fornecedores.

O administrador deve cadastrar entre outros dados, o CNPJ do fornecedor, que será utilizado como Login no sistema, e a URI do *Web Service*. A partir disto, o Sistema Tanamão passa a “enxergar” um novo fornecedor e passa a cotar os preços de seus produtos no portal.

Por fim, a última tela de grande importância nesta interface é a tela de criação de novas promoções (Figura 29). Se um fornecedor já está cadastrado no sistema, ele pode adicionar produtos promocionais ao portal, escolhendo qual o perfil de consumo dos usuários que pretende atingir.



Figura 29 – Tela para criação de uma nova promoção.

4.2 Testes e Resultados

A seguir são apresentados os principais testes e resultados obtidos com o Sistema Tanamão. Procuramos enfatizar os aspectos funcionais mais relevantes para a análise do sistema.

4.2.1 Comparação de preços

Uma das funcionalidades previstas no Sistema Tanamão é a comparação de preços dos produtos. Os produtos devem ser classificados segundo uma categoria e um tipo, sendo assim possível comparar o preço de qualquer produto de um mesmo tipo e mesma categoria.

Após escolher uma Categoria (p.ex. Fast-food, Cinema ou Show) o usuário pode escolher o Tipo de produto que é uma subclassificação de uma categoria (p.ex. Pizza, Sanduíche, Ação, Suspense, Comédia, Rock, Pop, etc).

Os produtos serão armazenados apenas nos fornecedores, que possuirão *Web Services* com alguns *Web Methods* padronizados para listar todos os produtos que atendem a estes classificadores. Os fornecedores estarão previamente cadastrados no Sistema Tanamão, com o endereço do *site* (URL) e do *Web Service* (URL do *Web Service*).

Todos os fornecedores cadastrados serão pesquisados. Caso não possuam os produtos requisitados, apenas não participarão da comparação de preços, não podendo vender seus produtos. Para os fornecedores com os produtos disponíveis, aparecerão os produtos listados com o preço e o nome do fornecedor.

Através do Sistema Tanamão, realizaram-se testes com várias Categorias e vários Tipos, em dois fornecedores, que não fazem parte do escopo da nossa implementação, mas que foram implementados com o intuito de possibilitar a realização de testes.

Os resultados foram compilados na tabela a seguir:

Categoria	Tipo	Nome	Preço no Fornecedor	
			Multi-Coisas	Uau Vênus
Cinema	Acao	Kill Bill 2	10	15
Cinema	Acao	Star Wars 3	10	15
Cinema	Acao	Terminator	10	15
Cinema	Suspense	A Vila	8	7
Cinema	Suspense	O efeito Borboleta	8	7
Cinema	Suspense	Pânico	8	7
Cinema	Comedia	As Branquelas	10	11
Cinema	Comedia	Quem vai ficar com Mary?	10	9
Cinema	Comedia	O terminal	10	8
Fast-food	Pizza	Mussarela	9	10
Fast-food	Pizza	Calabresa	9	10
Fast-food	Pizza	Portuguesa	12	11
Fast-food	Pizza	Baiana	10	9
Fast-food	Sanduiche	Hamburguer	2	2,5
Fast-food	Sanduiche	X-Salada	3	3,5
Fast-food	Sanduiche	X-Burguer	2,5	3
Fast-food	Sanduiche	X-Tudo	5	4
Show	Rock	Linkin Park	80	82
Show	Rock	Velhas Virgens	30	25
Show	Rock	Dishwalla	50	55
Show	Pop	Capital Inicial	20	15
Show	Pop	Evanescence	50	55
Show	Pop	Lifhouse	50	55

Tabela 2 – Simulação de cotação de preços.

Apesar de existirem apenas dois fornecedores implementados, a inclusão de novos fornecedores está prevista e são facilmente incorporados, bastando que atendam a especificação e padronização dos *Web Services* e um simples cadastro no sistema, através de um *site* com *login* e senha, pela interface HTML.

4.2.2 Compra de produtos

Uma vez comparado o preço de um produto, o usuário pode escolher o fornecedor que melhor agrada, segundo o preço ou a confiança que o usuário deposita em um fornecedor específico.

O próximo passo é confirmar o endereço de entrega previamente cadastrado pela interface web ou inserir um novo através do teclado do celular. Em uma versão futura, pode optar-se em colocar um CEP e o sistema consultará um *Web Service* dos correios que devolverá o endereço, bastando digitar o número da casa e complemento.

Há duas possibilidades para pagamento: através de um cartão de crédito ou através de débito bancário. Escolhendo a opção de cartão de crédito, o sistema aceita os cartões Visa, MasterCard e Amex, mas isto pode ser alterado, conforme o cadastro de novas operadoras de cartão de crédito. O usuário precisa digitar o número do cartão de crédito e o seu código, já que por questões de segurança, os mesmos não serão armazenados no sistema.

Com a opção de débito em conta, o usuário precisa digitar o número do banco, agência, conta e senha, mas isto pode ser previamente cadastrado pela interface HTML (exceto a senha), evitando-se a dificuldade de digitação no celular.

Em ambas opções, o sistema irá acessar um *Web Service* de um banco para efetuar a transação bancária. A implementação do mesmo também não está no escopo do sistema, que se limita a utilizá-lo. No entanto, para efeito de testes, um *Web Service* simplificado de um banco foi criado para permitir efetuar transações bancárias. Desta forma, o pedido pode ser efetuado.

Realizando-se algumas compras pelo sistema, obtivemos os seguintes resultados armazenados no banco de dados (Tabelas 3 a 7):

Id	BancoOrigem	AgenciaOrigem	ContaOrigem	BancoDestino	AgenciaDestino	ContaDestino	Valor
1	222	0282	974661	456	0789	987654	40
2	123	0456	789789	987	0654	789321	100
3	124	0456	123789	987	0654	123321	30

Tabela 3 – Banco de dados de transações de débito em banco.

Id	Bandeira	NumCartao	BancoDestino	AgenciaDestino	ContaDestino	valor
1	Visa	1234567	123	789	987654	50
2	CredCard	9876543	987	123	465872	18
3	Amex	1357924	456	123	165488	80

Tabela 4 – Banco de dados de transações de cartão de crédito.

IdPedido	Status	Data
1	Pago	19/10/2004
2	Pago	19/10/2004
3	Pago	19/10/2004

Tabela 5 – Banco de dados do Web Service do Fornecedor Multi-Coisas.

IdPedido	Status	Data
1	Pago	19/10/2004
2	Pago	20/10/2004
3	Pago	20/10/2004

Tabela 6 – Banco de dados do Web Service do Fornecedor Uau Vênus.

idPedido Fornecedor	consumidor	data	local Entrega	Valor Total	status	produtos	forma Pgto	fornecedor
1	2	19/10/2004	1	40	Entregue	(BLOB)	2	1
2	1	20/10/2004	1	100	Aguardando Entrega	(BLOB)	2	1
1	1	20/10/2004	2	30	Aguardando Entrega	(BLOB)	2	2
2	1	19/10/2004	1	50	Entregue	(BLOB)	0	2
3	1	20/10/2004	1	18	Aguardando Entrega	(BLOB)	0	2
3	2	20/10/2004	1	80	Aguardando Entrega	(BLOB)	0	1

Tabela 7 – Banco de dados do Sistema Tanamão.

4.2.3 Verificação do status do pedido

Cada usuário pode verificar o status dos seus pedidos, bastando para isso estar logado no sistema. Através do usuário pode-se obter todos os pedidos realizados pelo mesmo e, assim, obter o status de cada pedido. O status pode adquirir as seguintes formas: Pago, aguardando pagamento, pedido entregue, número de cartão inválido, senha inválida, conta inválida, aguardando entrega.

Os resultados obtidos para o consumidor 2, nos dados acima foram:

IdPedido	Status	Data
1	Pago	19/10/2004
3	Pago	19/10/2004

Tabela 8 – Exemplos de resultados obtidos para um consumidor.

4.2.4 Personalização

A utilização do sistema por cada usuário é guardada no banco de dados, permitindo verificar qual é a sua preferência por Tipos de produtos e Categorias de produtos. Toda vez que o usuário entra em uma Categoria ou Tipo de produto, o sistema irá incrementar o contador de visitas.

Automaticamente o sistema irá verificar quais são os interesses do usuário e irá gerar a página com as Categorias e Tipos de maior interesse, facilitando a navegação do usuário.

Na página de promoções aparecerão os produtos em promoção, gerado a partir da interface web, com a utilização de um fornecedor, disponibilizando poucos produtos com fácil acesso e pouca navegação.

Um exemplo de personalização das páginas, quando um usuário utiliza-se frequentemente de pesquisas para a Categoria Fast-food e cinema:

Ordenação inicial do menu de seleção de categorias:

1. Cinema
2. Fast-food
3. Show

Ordenação após alguma utilização do sistema, conforme citado:

1. Fast-food
2. Cinema
3. Show

O fornecedor pode utilizar o serviço de criar promoções, como exemplo pizza de mussarela por um preço de R\$8,00, listando seu produto na página de promoções. O número de produtos em promoção tem um limite configurável e, após atingir este limite, o sistema irá substituir um produto cuja validade tenha se expirado (por ordem de data, apagando o mais antigo) ou impedir a criação de uma campanha até que se expire uma promoção.

4.2.5 Interface HTML

Através da interface HTML, o usuário pode cadastrar-se no sistema e, desta forma, evitar o trabalho de digitação via teclado de celular, melhorando a usabilidade.

O usuário cadastra o nome, CPF, senha, celular, telefone, endereços de entrega (o consumidor pode cadastrar o endereço de entrega pelo celular também), número do banco, agência, conta, bandeira e número do cartão de crédito. A senha do banco ou do cartão na é armazenada por questões de segurança.

CPF	Senha	nome	celular	telefone
123456789	cRDtpNCeBiqI5KOQsKVyrA0sAiA=	Alberto	9861-1234	6332-4567
789123456	2jmj7l5rSw0yVb/vJWAYkK/YBwk=	Eduardo	9985-9989	4567-7893
987654321	I5KOQsKVyrAWAYkK/YBwk	Fabio	8188-4888	1234-7894

Endereço	banco	agência	noConta	bandeira	noCartao
Rua do Campeao, 1	123	123	46576	1	
Rua do Gargalo, 99	456	123	46732	2	56489
Rua da Perdição, 69	789	456	21324	3	

Tabela 9 – Registros de Consumidores.

O fornecedor também pode cadastrar-se no sistema e, desta forma, oferecer os seus produtos através do sistema Tanamão. O fornecedor precisa cadastrar o nome, cnpj, senha, endereço, telefone, site, url do Webservice, banco, agência e conta. Após o cadastro, pode oferecer promoções através do serviço de criar promoções para usuários com perfis específicos.

CNPJ	senha	nome	endereço	telefone	site
789654	rSAYkK/YBww0yVb/vlW2jmj7l5k=	Mutli-Coisas	Rua Multivias, 123	7894-7894	www.multicoisas.com.br
123456	AYkK/YsKVyrAWI5KOQBwk	Uau Vênus	Rua Lactea, 4567	1324-7895	www.uauvenus.com.br

WebserviceUrl	banco	agência	conta
http://com.multicoisas.com.br/WebServiceFornecedor/wsFornecedor.aspx	456	654	564654
http://com.uauvenus.com.br/WebServiceFornecedor/wsFornecedor.aspx	987	321	132456

Tabela 10 – Registros de Fornecedores.

5 CONSIDERAÇÕES FINAIS

Apresentamos a seguir as principais conclusões do grupo a respeito do projeto realizado, e indicamos ainda algumas idéias e contribuições que julgamos relevantes para futuros projetistas de sistemas similares ao Sistema Tanamão.

5.1 Conclusões

O foco dos integrantes do grupo, desde o início do projeto, era no desenvolvimento de uma arquitetura inovadora que aplicasse o estado da arte no que tange à Tecnologia da Informação. A utilização de *Web Services* como modelo de comunicação entre sistemas de diferentes plataformas tecnológicas, a aplicação de conceitos de CRM para construir um portal “inteligente” que passasse a conhecer os consumidores através de suas interações, e ainda, a disponibilização de um serviço que oferecesse conforto e praticidade aos usuários de dispositivos móveis, eram os objetivos principais do Sistema Tanamão.

Destes objetivos, o mais difícil de se atingir foi a personalização da IHC baseada em conceitos de CRM. Mesmo a implementação de funcionalidades simplificadas mostrou-se bastante complexa, pois notamos que existe a necessidade de uma quantidade razoável de dados a respeito do consumidor para que a geração de perfis seja feita eficientemente. Talvez se introduzíssemos um *Data Warehouse* e utilizássemos algoritmos de *Data Mining* pudéssemos obter melhores resultados, porém isto poderia tornar inviável a aplicação desta arquitetura para um portal de comércio eletrônico.

Já a meta de integração de *Web Services* de diferentes fornecedores para oferecer serviços a usuários de dispositivos móveis foi alcançada com sucesso. Comprovou-se que através de uma aplicação relativamente simples é possível agregar funcionalidades e disponibilizá-las ao consumidor utilizando uma interface já bastante difundida.

Quanto à metodologia de desenvolvimento, acreditamos que a escolha pelo Processo em Cascata rendeu bons resultados ao nosso projeto. O tamanho reduzido da equipe e a experiência de cada integrante neste tipo de processo de desenvolvimento de *software* foram fatores que contribuíram para que este modelo se ajustasse mais adequadamente às nossas necessidades. No entanto, vale comentar que houve um atraso no cronograma de atividades durante a fase de implementação, mas isto ocorreu devido à diminuição da carga horária disponível para dedicação ao projeto de formatura.

5.2 Trabalhos Futuros

O sistema atualmente implementado foca na arquitetura em quatro camadas baseada em componentes, com acesso a Web Services. Porém, simplifica vários fatores, visto a duração deste projeto, e estes podem ser melhorados em versões futuras, ou adotar-se soluções de tecnologias diferentes.

- Ao invés de utilizar o WAP e JSPs gerando páginas WML, poderia-se utilizar o protocolo de comunicação baseado em mensagens SMS e a camada de apresentação ser realizada em J2ME. Desta forma, minimiza-se os custos de utilização do sistema para o usuário final, mas acarreta em custos adicionais para convênio com alguma operadora de celular que ofereça este serviço de SMS. Contudo, esta solução iria alterar significativamente a arquitetura do sistema e possui uma complexidade elevada;
- Focar mais no sistema de CRM. O sistema atual é simplificado e permite apenas alguns tipos de personalização da interface e há muito campo para ser melhorado;
- Implementar segurança de dados nas transações entre os sistemas de forma mais eficiente e padronizada. Estudar deficiências de segurança de dados do sistema e melhorá-las;

- Detectar o número do celular, evitando-se a necessidade de identificação no sistema através do celular. Isto é um problema complexo, já que necessita de um convênio com as operadoras de celular para passar informações através do gateway que processa as requisições do browser WAP;
- Melhorar a interface com o usuário, permitindo que o usuário digite o menor número de informações possível, eliminando passos repetitivos;
- Permitir que a cobrança do produto fosse incorporada à conta telefônica, seja ela pré ou pós-paga;
- Especificação, definição e padronização dos Web Services dos fornecedores e do Banco.

Utilização de um Web Service que forneça o endereço através do CEP, diminuindo a digitação do consumidor, caso este escolha entregar em um endereço não cadastrado previamente.

ANEXO A – ANÁLISE ORIENTADA A OBJETOS DO SISTEMA

1 Descrição dos casos de uso

1.1 Atores

São três os atores do sistema:

- Consumidor: representa o papel do consumidor no sistema, ativando atividades de pesquisa de produtos, compra, etc.
- Fornecedor: representa o papel do fornecedor de produtos ou serviços no sistema, ativando principalmente casos de uso relacionado à pesquisa de informações sobre os consumidores.
- Administrador: representa o papel do administrador do sistema.

1.2 Subsistema Interface Consumidor

Segue a descrição dos casos de uso relacionados à interface de comunicação entre o sistema e o consumidor.

Caso de Uso 1: Cadastrar Consumidor.

Descrição: Este caso de uso descreve o procedimento de inserção dos dados do consumidor no sistema.

Ator: Consumidor.

Pré-Condição: Consumidor acessa o sistema via Internet.

Seqüência de Eventos:

1. O Consumidor entra no *site* por um *browser* por um micro-computador para facilitar o cadastro (digitar dados).
2. O Consumidor informa o Nome, Senha, CPF, Endereço de entrega, telefone residencial, celular.
3. O sistema verificará se o CPF já está sendo utilizado e então confirma o cadastro.

Pós-Condição: Consumidor está cadastrado no sistema.

Inclusão:

1. Registrar no Histórico (Caso de Uso 14)

Caso de Uso 2: Comparar Preços.

Descrição: Este caso de uso descreve o processo de pesquisa e comparação de preços para o Consumidor.

Ator: Consumidor.

Pré-Condição: Consumidor autenticado no sistema.

Seqüência de Eventos:

1. O Consumidor navega pelo menu e seleciona um tipo de produto ou serviço.
2. Uma lista exibida com o nome dos produtos, preços e fornecedor/empresa. A opção de comprar o produto/serviço também é disponibilizada.

Pós-Condição: Resultados da pesquisa de preços são mostrados ao Consumidor, e opção de compra é disponibilizada.

Inclusões:

1. Consultar Preço no Fornecedor (Caso de Uso 12).
2. Registrar no Histórico (Caso de Uso 14).

Caso de Uso 3: Comprar Produto/Serviço.

Descrição: Este caso de uso descreve o procedimento para compra de um produto ou serviço pelo Consumidor.

Ator: Consumidor.

Pré-Condição: Consumidor está autenticado no sistema e produto escolhido.

Seqüência de Eventos:

1. O Consumidor faz o pedido.
2. O sistema exibe os dados do produto desejado e os dados para entrega.
3. O Consumidor fornece os dados para cobrança e confirma.
4. O sistema efetua a compra, interagindo com os sistemas bancário e do fornecedor.
5. O sistema informa o código do pedido.

Pós-Condição: Compra efetuada.

Inclusões:

1. Registrar no Histórico (Caso de Uso 14).
2. Efetuar Transação Bancária (Caso de Uso 13).
3. Registrar Pedido (Caso de Uso 15).

Caso de Uso 4: Login Consumidor.

Descrição: Este caso de uso descreve a autenticação do Consumidor no sistema.

Ator: Consumidor.

Pré-Condição: Consumidor está cadastrado no sistema.

Seqüência de Eventos:

1. Consumidor entra com CPF e senha.
2. Sistema verifica a existência e validade.
3. Sistema envia mensagem de sucesso.

Extensão:

1. Se CPF e/ou senha estão incorretos, sistema informa que dados estão incorretos.

Pós-Condição: Consumidor está autenticado e pronto para utilizar o sistema.

Inclusão:

1. Registrar no Histórico (Caso de Uso 14).
2. Buscar Perfil (Caso de Uso 10).

Caso de Uso 5: Verificar Status do Pedido.

Descrição: Este caso de uso descreve a verificação do status de um pedido no sistema.

Ator: Consumidor.

Pré-Condição: Pedido efetuado e Consumidor autenticado.

Seqüência de Eventos:

1. O Consumidor solicita a verificação do status de seus pedidos.
2. O Sistema informa o status de cada pedido.

Pós-Condição: Status do pedido informado ao Consumidor.

Inclusão:

1. Verificar Status no Fornecedor (Caso de Uso 16).

1.3 Subsistema Interface Fornecedor

Segue a descrição dos casos de uso relacionados à interface de comunicação entre o sistema e o fornecedor.

Caso de Uso 6: Criar Promoção.

Descrição: Este caso de uso descreve a criação de uma promoção no sistema.

Ator: Fornecedor.

Pré-Condição: Fornecedor autenticado no sistema.

Seqüência de Eventos:

1. Fornecedor seleciona a categoria do produto que deseja promover.
2. Fornecedor seleciona o tipo do produto que deseja promover (opcional).
3. O sistema cria uma promoção, veiculando propagandas para os Consumidores mais adequados.

Pós-Condição: Promoção promovida para os Consumidores que mais se adequarem ao perfil desejado.

Caso de Uso 7: Login Fornecedor.

Descrição: Este caso de uso descreve a autenticação do Fornecedor no sistema.

Ator: Fornecedor.

Pré-Condição: Fornecedor está cadastrado no sistema.

Seqüência de Eventos:

1. Fornecedor entra com CNPJ e senha.
2. Sistema verifica a existência e validade.
3. Sistema envia mensagem de sucesso.

Extensão:

1. Se CNPJ e/ou senha estão incorretos, sistema informa que dados estão incorretos (passo 3).

Pós-Condição: Fornecedor está autenticado e pronto para utilizar o sistema.

1.4 Subsistema Interface Administrador

Segue a descrição dos casos de uso relacionados à interface de comunicação entre o sistema e o administrador.

Caso de Uso 8: Login Administrador

Descrição: Este caso de uso descreve a autenticação do administrador no sistema.

Ator: Administrador.

Pré-Condição: Administrador está cadastrado no sistema.

Seqüência de Eventos:

1. Administrador entra com login e senha.
2. Sistema verifica a existência e validade.
3. Sistema envia mensagem de sucesso.

Extensão:

1. Se login e/ou senha estão incorretos, sistema informa que dados estão incorretos (passo 3).

Pós-Condição: Administrador está autenticado e pronto para utilizar o sistema.

Caso de Uso 9: Cadastrar Fornecedor

Descrição: Este caso de uso descreve o procedimento de inserção dos dados do fornecedor no sistema.

Ator: Administrador.

Pré-Condição: Administrador acessa o sistema via Internet.

Seqüência de Eventos:

1. O Administrador entra no *site* por um *browser* por um micro-computador para facilitar o cadastro (digitar dados).
2. O Administrador informa os dados do fornecedor.
3. O sistema verificará se o fornecedor já está sendo utilizado e então confirma o cadastro.

Pós-Condição: Consumidor está cadastrado no sistema.

1.5 Lógica de Negócio

Segue a descrição dos casos de uso relacionados à implementação da lógica de negócio do sistema, executando funções de consumo de *Web Services*, CRM e persistência de dados.

Caso de Uso 10: Buscar Perfil.

Descrição: Este caso de uso descreve a busca de um perfil os dados armazenados no sistema.

Pré-Condição: Informações do Consumidor registradas no sistema.

Ator: nenhum.

Seqüência de Eventos:

1. O Sistema recebe informações sobre o perfil desejado.
2. O Sistema consulta as informações registradas sobre os usuários e filtra os mesmos.
3. O Sistema retorna uma listagem com os usuários com o perfil desejado.

Pós-Condição: Listagem de consumidores obtidos com o perfil desejado.

Caso de Uso 11: Cadastrar Fornecedor.

Descrição: Este caso de uso descreve o cadastro de um fornecedor no sistema.

Ator: nenhum.

Pré-Condição: Administrador autenticado.

Seqüência de Eventos:

1. Administrador preenche os campos com os dados do fornecedor, tais como Senha, Endereço, CNPJ, etc.
2. O Sistema valida os dados e retorna uma mensagem de sucesso.

Extensão:

1. Se os dados não estiverem corretos, o Sistema solicita o preenchimento do cadastro novamente (Passo 2).

Pós-Condição: Fornecedor cadastrado.

Caso de Uso 12: Consultar Preço no Fornecedor.

Descrição: Este caso de uso descreve a consulta de preço de um produto no fornecedor.

Ator: nenhum.

Pré-Condição: Produto selecionado.

Seqüência de Eventos:

1. O Sistema informa o produto escolhido para um fornecedor específico.
2. O fornecedor retorna o preço unitário do produto.

Pós-Condição: Preço do produto verificado.

Caso de Uso 13: Efetuar Transação Bancária.

Descrição: Este caso de uso descreve o procedimento de transação bancária para a compra de um produto ou serviço.

Ator: nenhum.

Pré-Condição: Consumidor efetuou a compra.

Seqüência de Eventos:

1. Sistema solicita a transação bancária, informando os valores envolvidos.
2. O Consumidor fornece seus dados bancários.
3. O sistema bancário efetua a transação e retorna uma confirmação.

Pós-Condição: Transação financeira confirmada.

Caso de Uso 14: Registrar no Histórico.

Descrição: Este caso de uso descreve o processo de registro de atividades do consumidor no histórico do sistema.

Ator: nenhum.

Pré-Condição: Consumidor interagindo com o Sistema.

Seqüência de Eventos:

1. O Sistema capta informações sobre as ações do consumidor.
2. O Sistema registra essas informações para futura análise.

Pós-Condição: Comportamento do consumidor registrado no sistema.

Caso de Uso 15: Registrar Pedido.

Descrição: Este caso de uso descreve o processo de registro de um pedido no sistema.

Ator: nenhum.

Pré-Condição: Pedido registrado no sistema.

Seqüência de Eventos:

1. O sistema acessa o fornecedor enviando os dados da compra.
2. O sistema recebe uma mensagem de sucesso.

Pós-Condição: Pedido registrado no sistema do fornecedor.

Caso de Uso 16: Verificar Status no Fornecedor.

Descrição: Este caso de uso descreve a verificação do status de um pedido no fornecedor.

Ator: nenhum.

Pré-Condição: Pedido registrado no sistema.

Seqüência de Eventos:

1. O Sistema fornece o código do pedido ao fornecedor.
2. O Fornecedor retorna o status do pedido.

Pós-Condição: Status do pedido verificado.

2 Diagramas de Interação

Apresentamos a seguir os diagramas de interação principais do Sistema Tanamão. Estes modelos facilitam o entendimento do funcionamento das funções mais complexas do sistema.

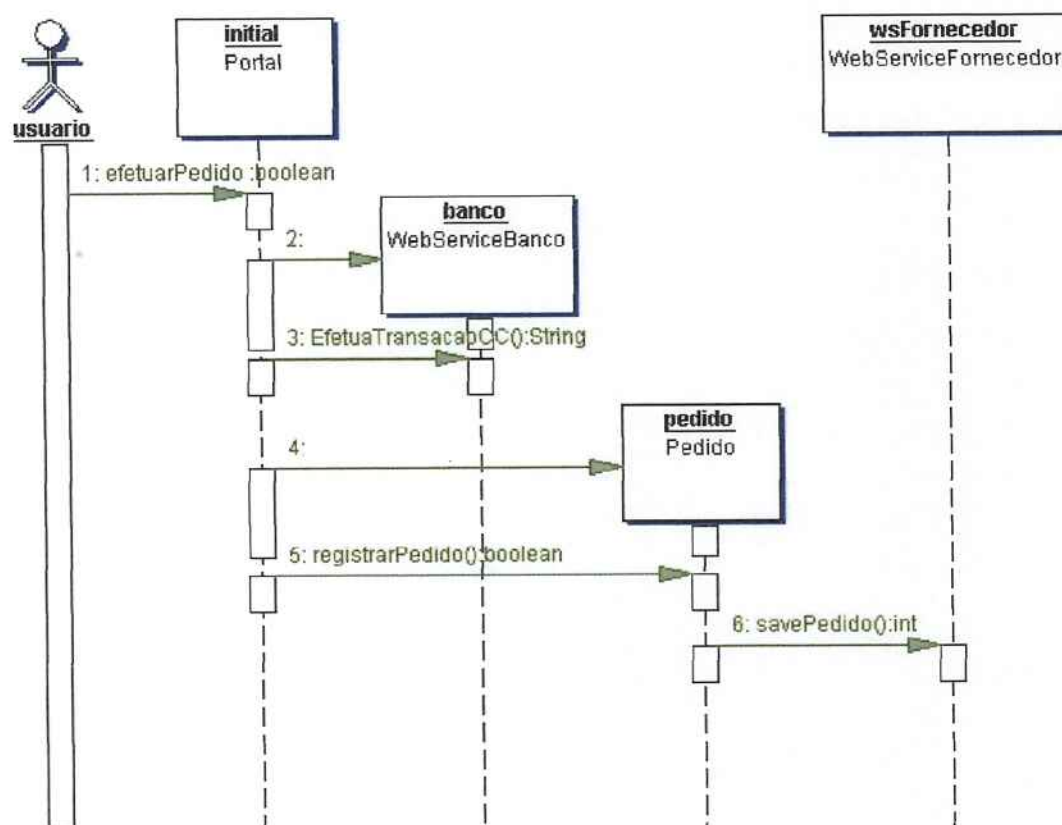


Figura 30 – Diagrama de Interação do método `efetuarPedido` da classe `Portal`.

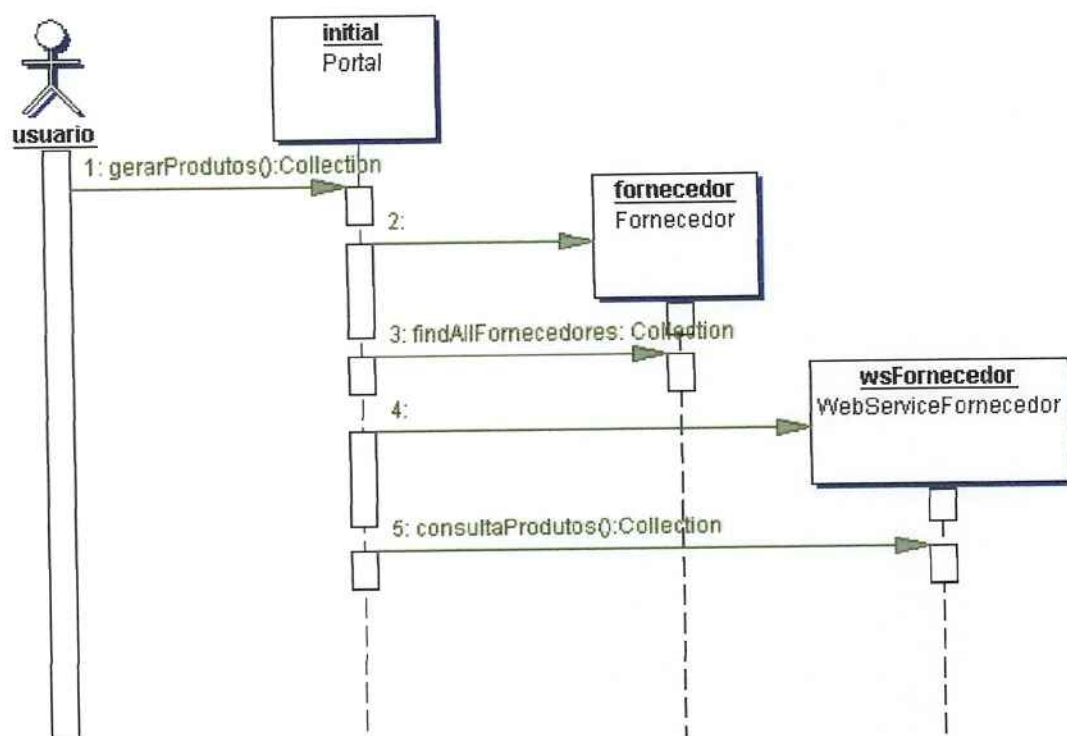


Figura 31 – Diagrama de Interação do método `gerarProdutos` da classe `Portal`.

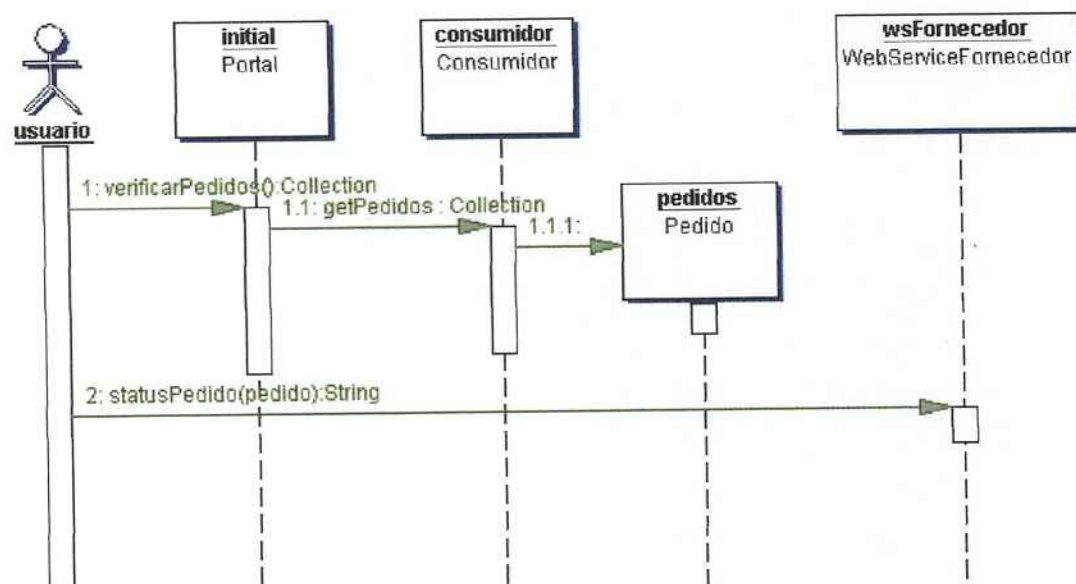


Figura 32 – Diagrama de Interação do método `verificarPedidos` da classe `Portal`.

ANEXO B – ESTUDO DA IHC DO SISTEMA TANAMÃO

Apresentamos a seguir um estudo da interface com o consumidor do Sistema Tanamão, realizado em forma de projeto para a disciplina PCS 2057 – Multimídia e Hipermedia.

1 Estrutura de Hipertexto

A estrutura de navegação para estudo das métricas de hipertexto é apresentada na figura a seguir:

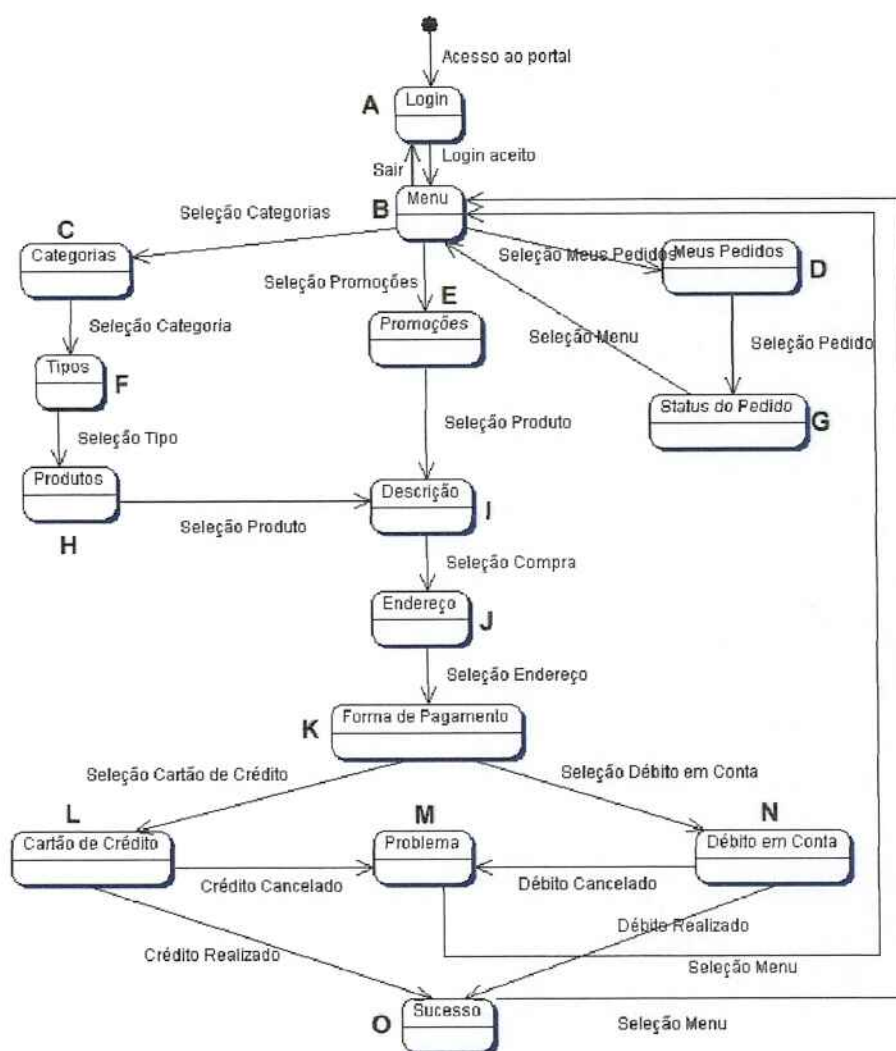


Figura 33 – Estrutura de navegação marcada para análise de métricas de hipertexto.

2 Métricas do Hipertexto

Segue abaixo a Matriz Convertida de Distâncias da estrutura de navegação:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	COD	ROC	
A		0	1	2	2	2	3	2	4	3	4	5	6	7	6	7	54	16
B		1	0	1	1	1	2	2	3	2	3	4	5	6	5	6	42	21
C		9	8	0	2	9	1	2	2	3	4	5	6	7	6	7	71	12
D		3	2	3	0	3	3	1	4	3	4	5	6	7	6	7	57	15
E		7	6	7	7	0	3	2	4	1	2	3	4	5	4	5	60	15
F		8	7	8	8	8	0	3	1	2	3	4	5	6	5	6	74	12
G		2	1	2	2	2	3	0	4	3	4	5	6	7	6	7	54	16
H		7	6	7	7	7	8	8	0	1	2	3	4	5	4	5	74	12
I		6	5	6	6	6	7	7	8	0	1	2	3	4	3	4	68	13
J		5	4	5	5	5	6	6	7	7	0	1	2	3	2	3	61	14
K		4	3	4	4	4	5	5	6	6	7	0	1	2	1	2	54	16
L		3	2	3	3	3	4	4	5	5	6	7	0	1	2	1	49	18
M		2	1	2	2	2	3	3	4	4	5	6	6	0	6	2	48	18
N		3	2	3	3	3	4	4	5	5	6	7	0	1	2	1	49	18
O		2	1	2	2	2	3	3	4	4	5	6	7	8	7	0	56	16
CID	62	49	55	54	57	55	52	61	49	56	63	61	69	65	63	871		
RIC	14	18	16	16	15	16	17	14	18	16	14	14	13	13	14			

Tabela 11 – Tabela das distâncias convertidas da interface WAP/WML.

CD - Converted Distance

CID - Converted In Distance

COD - Converted Out Distance

RIC - Relative In Centrality

ROC - Relative Out Centrality

Nota-se que o nó B possui um alto valor de ROC, portanto é um bom candidato a nó raiz. A compacidade da interface do sistema Tanamão é dada por:

$$C_p = (3150 - 871) / (3150 - 210) = 0,78$$

Isto permite concluir que todos os nós podem ser atingidos com pouca navegação.

3 Usabilidade

A primeira idéia em termos de usabilidade do portal foi a tentativa de detectar automaticamente o número do celular, de forma a identificar o usuário sem a necessidade de nenhum tipo de autenticação. Porém, isto se mostrou inviável, devido à arquitetura do sistema.

Para que o celular tenha acesso à Internet, é necessário que os pacotes de dados passem por um *Gateway* da operadora de telefonia, que encaminha então as requisições. Ao passar por esse *Gateway*, os pacotes perdem a informação de número de celular de origem. Desta forma, é impossível para um sistema qualquer na Internet detectar o número de origem. Isto só seria possível se houvesse um convênio com as operadoras, de forma que as estas repassassem essa informação.

Por isso, esta idéia foi deixada para versões futuras do sistema Tanamão, dado que o foco atual foi o desenvolvimento da arquitetura de *software*, e não a implantação do sistema comercialmente.

Desta forma, decidimos utilizar na primeira versão um método de autenticação explícita (com nome de usuário e senha). Para que a usabilidade não fosse comprometida, decidimos usar somente de números para esses campos, dado à limitação do teclado da maioria dos celulares atuais.

4 Sobrecarga cognitiva

O objetivo do nosso sistema para o usuário final é de poder comparar preços de alguns produtos e efetuar a compra dos mesmos. Para isso, a estrutura de navegação das páginas WML no WAP será intuitiva e simples, evitando que o usuário precise tomar muitas decisões para utilizar o nosso sistema.

A navegação tem a forma de um catálogo de produtos, de forma a ser familiar para a maioria dos usuários, não necessitando de conhecimentos avançados sobre algum tipo de tecnologia.

5 Dinamicidade

As páginas do Sistema Tanamão serão geradas por um *Application Server* (JBoss), através da tecnologia JSP (*Java Server Pages*). Desta forma, a cada acesso ao sistema, o conteúdo será personalizado de acordo com o sistema de CRM (*Customer Relationship Management*) simplificado.

Cada vez que o usuário do sistema realizar uma pesquisa ou comprar um produto, suas ações serão guardadas no banco de dados referente ao usuário e, baseando-se nestas informações, serão apresentadas promoções conforme o seu perfil.

Entretanto, tais alterações afetarão somente o conteúdo específico das páginas. A estrutura, conforme modelada, não sofrerá alterações e, portanto, as métricas derivadas continuam válidas.

ANEXO C – CRONOGRAMA DA FASE DE IMPLEMENTAÇÃO

Início: Segunda 30/8/04

Fim: Quinta 9/12/04

Recursos e Tarefas	Início	Fim
Alberto Abe	Seg 30/8/04	Sex 5/11/04
<i>Caso de Uso: Registrar pedido</i>	Seg 30/8/04	Qua 13/10/04
<i>Caso de Uso: Registrar no histórico</i>	Ter 7/9/04	Sex 29/10/04
<i>Caso de Uso: Consultar preço no fornecedor</i>	Sun 19/9/04	Sex 5/11/04
<i>Relatório Resumido</i>	Seg 6/9/04	Seg 6/9/04
<i>Relação e Programação de Uso de Recursos</i>	Ter 7/9/04	Ter 7/9/04
Fábio Onishi	Seg 30/8/04	Qua 24/11/04
<i>Caso de Uso: Consultar preço no fornecedor</i>	Seg 30/8/04	Sex 17/9/04
<i>Caso de Uso: Verificar status no fornecedor</i>	Sat 18/9/04	Qui 21/10/04
<i>Caso de Uso: Efetuar transação bancária de pagamento</i>	Qui 7/10/04	Sex 12/11/04
<i>Caso de Uso: Verificar status no fornecedor</i>	Sex 22/10/04	Qui 11/11/04
<i>Caso de Uso: Efetuar transação bancária de pagamento</i>	Seg 15/11/04	Ter 23/11/04
<i>Caso de Uso: Buscar perfil</i>	Qui 21/10/04	Ter 9/11/04
<i>Caso de Uso: Login de fornecedor</i>	Ter 12/10/04	Qui 11/11/04
<i>Caso de Uso: Cadastrar consumidor</i>	Qua 20/10/04	Ter 16/11/04
<i>Caso de Uso: Cadastrar fornecedor</i>	Qui 28/10/04	Qui 18/11/04
<i>Caso de Uso: Comprar produto</i>	Qui 4/11/04	Qui 18/11/04
<i>Caso de Uso: Verificar status do pedido</i>	Sex 12/11/04	Sex 19/11/04
<i>Cadastro</i>	Seg 15/11/04	Seg 15/11/04

<i>Login</i>	Ter 16/11/04	Ter 16/11/04
<i>Busca de Produtos</i>	Qua 17/11/04	Qua 17/11/04
<i>Compra de Produtos</i>	Seg 22/11/04	Seg 22/11/04
<i>Personalização da Interface</i>	Ter 23/11/04	Qua 24/11/04
Eduardo Yanai	Seg 30/8/04	Qui 9/12/04
<i>Caso de Uso: Login de consumidor</i>	Ter 28/9/04	Seg 8/11/04
<i>Caso de Uso: Login de administrador</i>	Ter 5/10/04	Qua 10/11/04
<i>Caso de Uso: Login de fornecedor</i>	Ter 12/10/04	Qui 11/11/04
<i>Caso de Uso: Cadastrar consumidor</i>	Qua 20/10/04	Ter 16/11/04
<i>Caso de Uso: Cadastrar fornecedor</i>	Qui 28/10/04	Qui 18/11/04
<i>Caso de Uso: Comprar produto</i>	Qui 4/11/04	Qui 18/11/04
<i>Caso de Uso: Verificar status do pedido</i>	Sex 12/11/04	Sex 19/11/04
<i>Caso de Uso: Comparar preços</i>	Sex 12/11/04	Qui 25/11/04
<i>Caso de Uso: Pesquisar perfis</i>	Seg 15/11/04	Sex 26/11/04
<i>Caso de Uso: Criar campanha</i>	Qui 18/11/04	Qua 1/12/04
<i>Relatório Resumido em Inglês</i>	Sex 17/9/04	Sex 17/9/04
<i>Relatório de Atividades I</i>	Seg 20/9/04	Sex 24/9/04
<i>Pôster</i>	Qua 10/11/04	Qua 10/11/04
<i>Relatório de Atividades II</i>	Sex 12/11/04	Qui 18/11/04
<i>Documentação Final</i>	Seg 30/8/04	Qua 8/12/04
<i>Planejamento da Demonstração</i>	Seg 22/11/04	Qua 24/11/04
<i>Apresentação III</i>	Qui 2/12/04	Sex 3/12/04
<i>Relatório de Atividades III</i>	Seg 6/12/04	Qui 9/12/04

LISTA DE REFERÊNCIAS

- [1] AGÊNCIA NACIONAL DE TELECOMUNICAÇÕES. Endereço da ANATEL na Internet, onde podem ser encontrados dados estatísticos sobre a telefonia no Brasil. Disponível em <<http://www.anatel.gov.br>>. Acesso em: 5 fev 2004.
- [2] WORLD WIDE WEB CONSORTIUM. Endereço do W3C na Internet, onde podem ser encontradas as especificações da arquitetura Web Services e do protocolo WAP. Disponível em: <<http://www.w3c.org>>. Acesso em: 15 mar 2004.
- [3] OPEN MOBILE ALLIANCE / WAP FORUM. Endereço do antigo portal WAP Forum, agora incorporado pela Open Mobile Alliance, onde podem ser encontradas informações a respeito de tecnologias para dispositivos móveis. Disponível em <<http://www.wapforum.org>> ou <<http://www.openmobilealliance.org>>. Acesso em: 3 dez 2004.
- [4] SILVA, P.V.M. GTA / UFRJ – Redes de Computadores I – WAP. Rio de Janeiro. Aula de redes de computadores, onde é abordado o conceito de WAP. Disponível em: <http://www.gta.ufrj.br/grad/00_1/vitor/intro.htm>. Acesso em: 23 mar 2004.
- [5] NETSCAPE DEVEDGE. Web Services, SOAP e Aplicações Web. Página com informações sobre Web Services. Disponível em: <<http://devedge.netscape.com/viewsource/2002/soap-overview>>. Acesso em: 23 mar 2004.
- [6] JANJICEK, R. CRM Architecture for Enterprise Relationship Marketing in the New Millenium. Site da Hewlett-Packard, 2002. Disponível em <http://h71028.www7.hp.com/enterprise/downloads/CRMArchitecture_Whitepaper_HPC.pdf>. Acesso em: 2 nov 2004.

[7] PEPPERS & ROGERS GROUP CONSULTING. Marketing 1 to 1 – Um Guia Executivo para Entender e Implantar Estratégias de Customer Relationship Management. 3^a. edição. Disponível para download em <<http://www.1to1.com.br>>. Acesso em: 3 mar 2004.

[8] IBM GLOBAL SERVICES. Improving Customer Relationships. Disponível em: <<http://www-1.ibm.com/services/us/index.wss/bt/bcs/a1000404>>. Acesso em: 3 mar 2004.

[9] DATAWAREHOUSE!. Home page com artigos interessantes sobre sistemas de Data Mining, Data Warehouse e CRM. Disponível em: <<http://www.datawarehouse.inf.br>>. Acesso em 4 mar 2004.

[10] INTEL NEXT GENERATION CENTER. Curso Online de CRM. Apresenta as estratégias envolvidas na implementação de soluções de CRM. Disponível em <<http://www.nextg.com.br>>. Acesso em: 7 mar 2004.




[11] JAVAFREE.COM.BR. Entendendo Melhor a Arquitetura J2EE. Disponível em <<http://www.javafree.com.br/home/modules.php?name=News&file=article&sid=1944>>. Acesso em 5 nov 2004.

[12] FUNDAÇÃO DA COMPUTAÇÃO. Design Patterns Fundamentais do J2EE. Disponível em <<http://www2.fundao.pro.br/articles.asp?cod=145>>. Acesso em 5 nov 2004.

[13] REVISTA DO LINUX. Alternativa para o .NET. Disponível em <http://labbi.uesc.br/apostilas/revista_do_linux/027/desenvolvimento.html>. Acesso em 5 nov 2004.

- [14] RATIONAL SOFTWARE. Rational Unified Process – Best Practices for Software Development Teams. Apresenta o modelo de processo unificado da Rational Software. Disponível em <<http://www.ibm.com>>. Acesso em: 9 set 2004.
- [15] MELNIKOFF, S.S.S. Modelos de Processo de Software. Escola Politécnica da USP, 2004.
- [16] GREEN, D.; DICATERINO, A. A Survey of System Development Process Models. Fevereiro, 1998. Disponível em <http://www.ctg.albany.edu/publications/reports/survey_of_sysdev>. Acesso em: 15 out 2004.
- [17] NUMERIC ALGORITHM LABORATORIES. M3Gate WAP Browser. Página do emulador de WAP chamado M3Gate. Disponível em: <<http://numeric.ru/m3gate>>. Acesso em: 23 mar 2004.

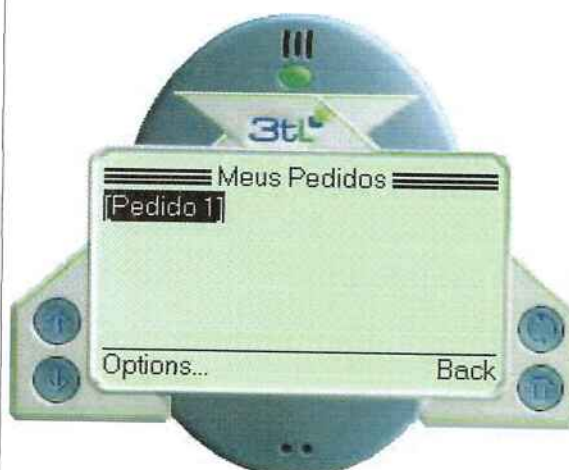
APÊNDICE – MANUAL DO USUÁRIO

	<p>Esta é a tela de início do Tanamão.</p> <p>A partir daqui, podemos selecionar a opção “Login” para entrar no sistema.</p> <p>Se nada for selecionado, em 5 segundos a tela de login aparecerá.</p>
	<p>Na tela de Login, o usuário deve digitar o seu CPF e senha, que devem ter sido cadastrados previamente pela interface web.</p>
	<p>A seguir, são apresentadas 3 opções para o usuário:</p> <ul style="list-style-type: none"> - verificar promoções de interesse - acessar o menu de produtos(categorias) - verificar os pedidos efetuados e seus status <p>Basta selecionar a opção desejada e clicar em “OK” para continuar.</p>



Caso seja escolhida a opção de promoções, o sistema mostra produtos que estejam com descontos no site, de acordo com o seu perfil.

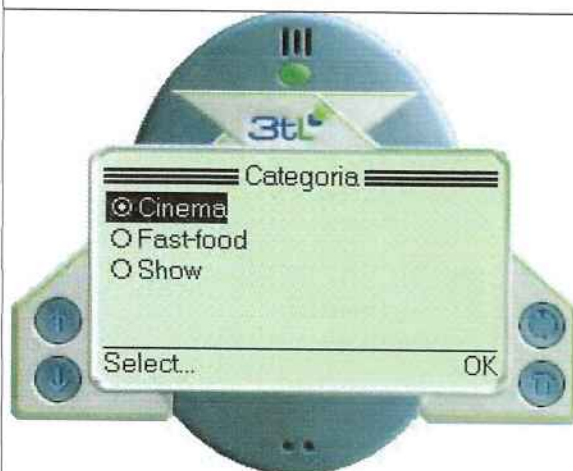
Por exemplo, se o usuário comprar muitos sanduíches pelo sistema, ofertas relacionadas a isto aparecerão. Cada usuário receberá somente promoções de interesse.



Caso a opção de verificar pedidos seja selecionada, é apresentada uma lista dos últimos pedidos registrados no sistema.




Selecionando um deles, é possível ver o status, como neste exemplo.










Selecionando a opção de acessar categorias, é possível navegar pelo menu de produtos, organizados por categoria e tipo de produto.

São mostrados, no último nível, os produtos de diversos fornecedores, com seu respectivo preço. Desta forma, podemos compara-los e efetuar a melhor escolha.

	
	<p>Escolhido o produto, e selecionando-o no menu, é possível compra-lo.</p>
	<p>Para a compra ser efetuada, são confirmados alguns dados. O primeiro deles é o endereço de entrega. São mostrados os endereços já cadastrados no sistema, para escolha, ou então pode-se digitar um novo endereço na caixa de texto.</p>

	<p>A seguir, os dados para o pagamento são solicitados. Duas formas de pagamento são aceitas: cartão de crédito e débito em conta.</p>
	<p>Os dados da forma de pagamento escolhida são solicitados. Para cartões de crédito, é necessário informar ao sistema a bandeira, o número e o código do cartão. E para débito em conta, o banco, a agência, a conta e a senha são necessários.</p>

	
	<p>Se tudo ocorrer como previsto, uma tela final de confirmação é apresentada, com os dados do pedido.</p>
	<p>Após efetuada a compra, o usuário pode decidir sair do sistema ou acessar novamente o menu para efetuar uma nova compra.</p>