



## Trabalho de Graduação

# SERVIÇO DE APRESENTAÇÃO CONSCIENTE DO DISPOSITIVO EM UM AMBIENTE PERVASIVO

---

Rubens Cesar Belusso

Curso de Ciência da Computação

Santa Maria, RS, Brasil

2005

**SERVIÇO DE APRESENTAÇÃO CONSCIENTE DO  
DISPOSITIVO EM UM AMBIENTE PERVASIVO**

---

por

**Rubens Cesar Belusso**

Trabalho de Graduação apresentado ao Curso de Ciência da  
Computação – Bacharelado, da Universidade Federal de  
Santa Maria (UFSM, RS), como requisito parcial para  
obtenção do grau de  
**Bacharel em Ciência da Computação.**

**Curso de Ciência da Computação**

Trabalho de Graduação nº 212

Santa Maria, RS, Brasil

2005

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

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

**SERVIÇO DE APRESENTAÇÃO CONSCIENTE DO  
DISPOSITIVO EM UM AMBIENTE PERVASIVO**

elaborado por  
**Rubens Cesar Belusso**

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

**COMISSÃO EXAMINADORA:**

---

**Prof<sup>a</sup>. Dr<sup>a</sup>. Iara Augustin**  
(Orientador)

---

**Prof. João Carlos Damasceno Lima**  
(Co-orientador)

---

**Prof<sup>a</sup>. Dr<sup>a</sup>. Andréa Schwertner Charão**

---

**Prof<sup>a</sup>. Dr<sup>a</sup>. Márcia Pasin**

Santa Maria, 22 de dezembro de 2005.

# Agradecimentos

Inicialmente agradeço a meus pais, Celino e Margarita, por tudo que me possibilitaram, pelos conselhos, pelo apoio nas horas difíceis, pelos momentos de descontração, por sempre apoiarem as minhas decisões e também por serem essas pessoas maravilhosas das quais tenho muito orgulho de ser filho. Agradeço também a minha irmã Sabrina e ao meu irmão Daniel por tudo que já passamos juntos.

Agradeço a todos os professores, principalmente a professora Iara Augustin, por ter me orientado de forma tranqüila e sempre disposta a ajudar. Agradeço também ao professor Caio, pela co-orientação prestada e pelo alto astral levado as reuniões do GMOB, principalmente aquelas realizadas no posto.

Aos meus grandes e inseparáveis amigos formados dentro da faculdade, Aline, Cristóvão, Juliano e Rafael, agradeço por tudo. Por todo o tempo que passamos juntos e pela grande amizade formada e consolidada nas tardes regadas a café na sala do PET.

Ao colega de GMOB Redin e ao pessoal do LSC Edmar, Elton, Veiga e Tucunduva por todos os auxílios prestados e pelas dezenas de idas ao bar para o lanche da tarde nas quais botávamos os assuntos em dia e planejávamos o que mais iríamos fazer.

Ao Acari, amigão a mais de 16 anos, o qual o considero como um irmão, agradeço por sempre se fazer presente com seu auto astral incomparável.

Outros a quem agradeço são o Mauro, Álvaro, Iuri e o Guilherme pela amizade e por todas as festas, carnavais e viagens que fizemos.

Além destes já citados agradeço ao meu amigo Jacson, por me agüentar e

conseguir dividir um apartamento comigo durante os cinco anos que residi em Santa Maria, mas principalmente pelo companheirismo mostrado em todos os momentos.

Finalizando agradeço também a tantos outros amigos e familiares que se fizeram presentes em minha vida.

# Sumário

<b>Agradecimentos</b>	<b>iii</b>
<b>Lista de Figuras</b>	<b>vii</b>
<b>Resumo</b>	<b>i</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 Revisão da Literatura</b>	<b>3</b>
2.1 Adaptação de Conteúdo na Computação Pervasiva . . . . .	3
2.1.1 Estratégias de Implementação da Adaptação de Conteúdo . . .	4
2.1.2 Estratégia adotada na Implementação do Serviço de Apresentação . . . . .	6
2.2 Identificação do Dispositivo . . . . .	7
2.2.1 User-Agent . . . . .	7
2.2.2 CC/PP . . . . .	7
2.2.3 User Agent Profile . . . . .	8
2.2.4 Modo de Identificação Utilizado na Arquitetura de Serviços pBuy . . . . .	8
2.3 Projetos Relacionados . . . . .	8
2.3.1 WTP . . . . .	8
2.3.2 iMASH . . . . .	9
2.3.3 CAF . . . . .	10
2.3.4 MDAT . . . . .	11

<b>3</b>	<b>Arquitetura de Serviços pBuy</b>	<b>13</b>
3.1	Serviço de Autenticação . . . . .	15
3.2	Servidor do Portal de Compras . . . . .	15
3.3	Serviço de Disseminação de Mensagens . . . . .	16
3.4	Serviço de Comunicação Consciente do Estado da Rede . . . . .	16
3.5	Serviço de Apresentação de Conteúdo . . . . .	17
<b>4</b>	<b>Modelagem do Serviço de Apresentação</b>	<b>18</b>
4.1	Tecnologias Utilizadas pelo Serviço de Apresentação . . . . .	18
4.1.1	XML . . . . .	18
4.1.2	XSL . . . . .	19
4.1.3	RMI . . . . .	22
4.2	Modelagem e Implementação do Serviço de Apresentação . . . . .	22
4.2.1	Decisões sobre o Processo de Adaptação . . . . .	22
4.2.2	Funcionalidade Básica do Serviço de Apresentação . . . . .	23
4.2.3	Etapa de Análise . . . . .	23
4.2.4	Etapa de Aplicação . . . . .	25
4.2.5	Arquivo XML do Sistema pBuy . . . . .	26
4.2.6	Arquivos de Estilo XSL . . . . .	28
<b>5</b>	<b>Resultados</b>	<b>30</b>
5.1	Arquivo XML Utilizado nos Testes . . . . .	30
5.2	Adaptações Realizadas . . . . .	31
<b>6</b>	<b>Conclusão</b>	<b>36</b>
	<b>Referências Bibliográficas</b>	<b>38</b>

# Lista de Figuras

2.1	Propriedades da Computação Pervasiva . . . . .	4
3.1	Relação entre os serviços da arquitetura proposta no projeto pBuy e ciclo normal de atendimento a uma requisição da aplicação . . . . .	14
4.1	Diagrama de Sequência do Serviço de Apresentação . . . . .	24
4.2	Estrutura do arquivo Mensagem.xml . . . . .	26
4.3	Exemplo XML do corpo da mensagem a ser adaptada . . . . .	28
4.4	Exemplo de arquivo de estilo XSL . . . . .	29
5.1	XML usado nos testes do Serviço de Apresentação . . . . .	32
5.2	Resultado da adaptação para HTML . . . . .	33
5.3	Resultado da adaptação para PDF . . . . .	34
5.4	Resultado da adaptação para TXT . . . . .	35

## RESUMO

Trabalho de Graduação  
Ciência da Computação  
Universidade Federal de Santa Maria

### SERVIÇO DE APRESENTAÇÃO CONSCIENTE DO DISPOSITIVO EM UM AMBIENTE PERVASIVO

AUTOR: RUBENS CESAR BELUSSO

ORIENTADOR: PROF<sup>a</sup>. DR<sup>a</sup>. IARA AUGUSTIN

Data e Local da Defesa: Santa Maria, 22 de dezembro de 2005.

A Computação Pervasiva é uma área nova da computação, que fornece uma visão futura onde a computação estará totalmente integrada ao ambiente, com todos os elementos computacionais (código, dados, rede, usuário, etc) podendo ser móveis. Neste cenário, os dados que são enviados aos usuários devem estar disponíveis independentemente do dispositivo em uso no momento. Neste sentido, a contribuição deste trabalho é o desenvolvimento de um serviço responsável pela apresentação dos dados através da adaptação de conteúdo, de acordo com as características do dispositivo usado pelo usuário.

# Capítulo 1

## Introdução

Um novo cenário computacional, conhecido como Computação Pervasiva (*Pervasive Computing*) [SAH 2003], está surgindo a partir das possibilidades introduzidas pelos equipamentos portáteis e pela comunicação sem fio. Esta computação pode ser resumida por 3 A's: *Anywhere, Anytime, Any device*, e visa disponibilizar o ambiente computacional do usuário em qualquer lugar, a qualquer tempo, acessando-o com qualquer dispositivo.

Sistemas pervasivos têm como objetivo prover acesso universal aos seus dados. Para isso devem criar várias formas de apresentação, envolvendo diferentes formatos, dispositivos e redes. Para que seja possível ter acesso às informações de forma correta a partir de vários tipos de dispositivos, é imprescindível a adaptação, via transcodificação de conteúdo, das informações que trafegam no sistema.

O Grupo de Pesquisa em Sistemas de Computação Móvel (GMob) da Universidade Federal de Santa Maria (UFSM) atualmente desenvolve o projeto pBuy. Este projeto tem por finalidade a inserção de tecnologia móvel com acesso pervasivo em uma aplicação real, o Portal de Compras, desenvolvido pela empresa SIG - Soluções em Informática e Gestão, de forma a agregar novas funcionalidades tecnológicas ao sistema, visando atingir o mercado de exportação de software e obter a padronização CMMI (*Capability Maturity Model Integrated*).

No ambiente considerado pelo pBuy, as aplicações são móveis e distribuídas. Isto significa que durante a execução da aplicação esta pode se mover entre dispositivos seguindo a trajetória do usuário (semântica siga-me [AUG 2004]). Para permitir

esta funcionalidade é necessário que a aplicação se adapte ao ambiente corrente, logo, consciência do contexto é um requisito importante na modelagem dessas aplicações. Para isso, verificou-se a necessidade de inserir na arquitetura do Portal de Compras Pervasivo um serviço de apresentação do conteúdo adaptado às características de cada dispositivo.

A contribuição deste trabalho é a modelagem e implementação de um serviço responsável pela adaptação de conteúdo consciente do dispositivo. Esta adaptação se baseia na transformação das mensagens enviadas pelo sistema para o formato que melhor se adequar às características obtidas do dispositivo que o usuário estiver usando no momento em que o sistema lhe enviar alguma informação. Assim, a criação deste serviço oferece como principal vantagem a possibilidade de usuários terem acesso ao sistema independentemente do tipo de dispositivo que estejam utilizando.

Este texto descreve os requisitos, a modelagem e o desenvolvimento deste serviço. O restante deste texto está organizado em 3 capítulos:

- **Capítulo 2 - Revisão da Literatura:** contextualiza a adaptação de conteúdo no ambiente pervasivo, expõe os problemas identificados e apresenta alguns projetos e ferramentas relacionadas.
- **Capítulo 3 - Arquitetura de Serviços pBuy:** contextualiza a Arquitetura de Serviços pBuy
- **Capítulo 4 - Modelagem do Serviço de Apresentação:** aborda as tecnologias usadas na implementação do serviço, além de apresentar a modelagem e detalhes da implementação do Serviço.
- **Capítulo 5 - Resultados:** apresenta os resultados obtidos em alguns testes usando a implementação do serviço.
- **Capítulo 6 - Conclusão:** apresenta as considerações finais deste trabalho e algumas propostas de trabalhos futuros.

# Capítulo 2

## Revisão da Literatura

Este capítulo aborda algumas questões relativas à adaptação de conteúdo, sua importância para o ambiente computacional pervasivo, os problemas para sua implementação e os trabalhos relacionados.

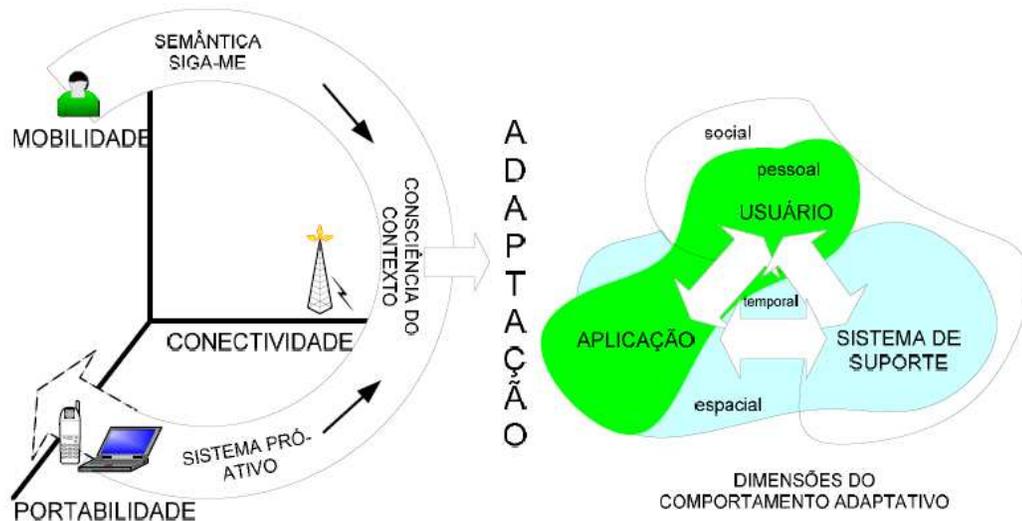
### 2.1 Adaptação de Conteúdo na Computação Pervasiva

A computação pervasiva através da natureza de suas propriedades - portabilidade, mobilidade e conectividade - introduz restrições aos sistemas e aplicações [AUG 2004]. Apesar da evolução natural da tecnologia, a maioria dos pesquisadores considera que essas limitações permanecerão, principalmente se comparadas ao ambiente de rede fixa. Então, as aplicações devem procurar explorar a natureza da mobilidade.

As aplicações para a computação pervasiva fazem uso de um cenário onde todos os elementos computacionais (código, dados, rede, usuário) podem ser móveis, permitindo a mobilidade lógica e física enquanto mantém a conectividade global, seguindo a trajetória do usuário (semântica siga-me)[AUG 2004]. Para permitir esta funcionalidade, é necessário que a aplicação se adapte dinamicamente as características de cada dispositivo. Adaptação é um aspecto fundamental para a Computação Pervasiva e foco recente de muitas pesquisas [SAT 2001]. Esta focaliza, particularmente, o aumento da usabilidade das aplicações em face aos diferentes

dispositivos móveis disponíveis.

A adaptação pode ocorrer em várias dimensões (pessoal, social, temporal e espacial), conforme ilustra a figura 2.1 (Fonte: [AUG 2005]). Além disso, a adaptação pode ser abordada de diversas formas, entre elas: adaptação via código para ajustar-se aos diferentes níveis de recursos disponíveis, adaptação automática da interface das aplicações às capacidades de vídeo dos dispositivos, e adaptação de conteúdo. Neste trabalho, focaliza-se o aspecto de adaptação de conteúdo.



**Figura 2.1:** Propriedades da Computação Pervasiva

Adaptação de conteúdo pode ser entendida como o processo de seleção, geração ou modificação do conteúdo (texto, imagens, áudio e vídeo) satisfazendo as características computacionais do dispositivo em uso, e o contexto de seu uso. O contexto pode ser identificado por qualquer informação que caracteriza o usuário, o lugar ou qualquer outro objeto relevante à interação entre o usuário e a aplicação [DEY 2001].

### 2.1.1 Estratégias de Implementação da Adaptação de Conteúdo

As estratégias mais utilizadas para implementar adaptação de conteúdo são através do uso de múltiplas autorias manuais, de adaptação no servidor, de adapta-

ção no cliente ou adaptação realizada por um proxy.

Múltiplas autorias manuais propõem a codificação de várias versões da mesma aplicação, criadas especificamente para consumir determinado nível de recursos e considerando algumas plataformas [WEB 2004]. Quando uma requisição por informação é recebida, a aplicação mais apropriada é selecionada e enviada ao usuário. Esta proposta é muito usada em aplicações WEB, quando diferentes versões do web site são desenvolvidas para diferentes tipos de navegadores, como por exemplo HTML e WAP.

Múltiplas autorias geralmente apresentam a forma mais otimizada de apresentação do conteúdo, entretanto, têm um custo muito alto de desenvolvimento e manutenção, além da dificuldade de criar diferentes versões para cada combinação possível de dispositivo e contexto [WEB 2004]. Conseqüentemente, esta não é uma solução viável para sistemas que necessitam de formas mais complexas e dinâmicas de adaptação.

A adaptação no servidor pode ser considerada uma estratégia dinâmica de múltiplas autorias [WEB 2004]. Neste caso, é desenvolvida uma única versão com o conteúdo a ser apresentado, enquanto diferentes formas de apresentação são geradas a partir deste conteúdo de acordo com o tipo de dispositivo. Uma maneira muito utilizada de prover esta adaptação é através do armazenamento do conteúdo como *Extensible Markup Language* (XML) [XML 2005] para, então, converter o conteúdo para uma forma apropriada à apresentação usando *Extensible Style Language Transformation* (XSLT) [XSL 2005]. A adaptação no servidor vem a ser uma boa alternativa para a adaptação de conteúdo, com um baixo custo de manutenção.

Adaptação realizada no cliente geralmente envolve a transferência dos dados do servidor para o dispositivo-cliente para que neste seja realizada a adaptação. Adaptações envolvendo as tecnologias XSL [XSL 2005a] e CSS (*Cascading Style Sheets*) [CAS 2005] podem ser aplicadas aqui. Neste caso, o conteúdo e as folhas de estilo (*stylesheets*) são transferidos para o dispositivo do cliente, onde será realizada a transformação. A adaptação no cliente oferece bons resultados. Entretanto, considerando o ambiente pervasivo é necessário avaliar possíveis restrições que en-

volvam a transferência de informações do servidor para o cliente, tais como largura de banda, quantidade de memória e poder computacional e de armazenamento do dispositivo-cliente.

Na adaptação via *proxy*, um *proxy* analisa e transcodifica o conteúdo dinamicamente (*on-the-fly*); após, envia o resultado ao cliente [WEB 2004]. O *proxy* também pode manter uma *cache* de conteúdos adaptados para poder usar depois, assim a transformação é realizada uma única vez. Considerando a capacidade dos dispositivos, o *proxy* pode executar em um dispositivo móvel ou em outro equipamento da rede.

### 2.1.2 Estratégia adotada na Implementação do Serviço de Apresentação

A alternativa que melhor se enquadrou para o desenvolvimento deste serviço foi pela adaptação no servidor, por ser a alternativa que melhor se adequou às características do portal de compras pervasivo e por se acreditar que esta seja a que venha a apresentar os melhores resultados, principalmente por possuir um baixo custo de manutenção quando necessária a alteração do sistema referente a adaptação, além de permitir a adaptação dinâmica de conteúdo necessária em sistemas pervasivos. Além disso, a escolha também se justifica pela análise das demais alternativas:

- A adaptação através de múltiplas autorias manuais não era uma escolha aceitável em se tratando de um sistema com características pervasivas. Se fosse seguida essa estratégia, haveria a necessidade do armazenamento de uma cópia do conteúdo já adaptado para cada dispositivo que viesse a utilizar o sistema, ou seja, o sistema teria um custo muito alto de desenvolvimento e de manutenção;
- A adaptação realizada no cliente nem sempre apresenta bons resultados. Por envolver a transferência de informações adicionais aos clientes, necessita levar em consideração fatores adversos como largura de banda, memória disponível, poder de processamento, além da capacidade do dispositivo de interpretar as

instruções relativas à adaptação. Ou seja, envolve a disponibilidade de recursos no cliente móvel, nem sempre existente;

- A adaptação por meio de um *proxy* não é aplicável no sistema pBuy pelo fato de que o conteúdo adaptado pelo serviço de apresentação deve retornar ao serviço de disseminação e não ser enviado diretamente para o cliente.

## 2.2 Identificação do Dispositivo

Com o aumento do número e da variedade de dispositivos ocorre um aumento correspondente na necessidade de conteúdo adaptado para diversos tipos de dispositivos. A seguir, descrevem-se as abordagens mais utilizadas para identificação de dispositivos.

### 2.2.1 User-Agent

Todo o cliente quando faz alguma requisição a um servidor, deve enviar informações sobre o seu navegador, sistema operacional e sobre seu hardware. Este conjunto de informações é conhecido pelo nome de *User-Agent* [USE 2005]. Entretanto, devido à grande diversidade de navegadores existentes fica difícil a identificação apenas pelo *User-Agent*. Existem dispositivos que enviam informações adicionais nos cabeçalhos http, como informações sobre as capacidades do dispositivo. Por exemplo, o Pocket PC envia informações adicionais que indicam a sua resolução em pixels e a quantidade de cores suportada. Entretanto, essas informações adicionais não são padronizadas e podem variar de acordo com o dispositivo.

### 2.2.2 CC/PP

Os *User-Agents* não são suficientes para identificar a gama de dispositivos disponíveis. Para suprir esta necessidade a W3C desenvolveu o CC/PP (*Composite Capabilities / Preference Profiles*) [CC/P 2005], que é um sistema padrão para expressar capacidades de dispositivos e preferências de usuários. O CC/PP pode ser usado para a comunicação do contexto, através de informações que identificam, por

exemplo, o tamanho da tela, as capacidades de áudio, a largura de banda, dentre outros. O CC/PP é um *framework* extensível que utiliza a RDF (*Resource Description Framework*) para a descrição de informações contextuais. Empresas como IBM, Nokia, Sun, Ericsson mantêm pesquisas sobre a utilização deste padrão.

### 2.2.3 User Agent Profile

*User Agent Profile* (UAProf) [USE 2005a] desenvolvido pela *Open Mobile Alliance* (OMA) é um concreto vocabulário CC/PP dedicado à descrição de celulares. Servidores e *proxies* podem usar essas informações e otimizar o conteúdo para estes dispositivos. As informações são transmitidas em um documento XML e abrangem os seguintes atributos: plataforma de hardware e de software; características da rede, do navegador, dos formatos suportados e do tamanho das mensagens.

### 2.2.4 Modo de Identificação Utilizado na Arquitetura de Serviços pBuy

A identificação das características dos dispositivos que acessam o sistema pBuy está sob a responsabilidade do Serviço de Autenticação (ver seção 3.1). Atualmente a identificação está sendo realizada usando-se os *User-Agent* enviados pelos clientes. Essa identificação limita-se a classificar os dispositivos em celulares, PDAs ou desktops. Para o Serviço de Autenticação, futuramente pretende-se utilizar o CC/PP para a obtenção das características dos dispositivos juntamente com as preferências dos usuários.

## 2.3 Projetos Relacionados

A seguir, apresentam-se alguns projetos que visam prover a o desenvolvimento de aplicações com suporte a adaptação de conteúdo.

### 2.3.1 WTP

O WTP (*WebSphere Transcoding Publisher*) [IBM 2005], desenvolvido pela IBM, possui uma série de funcionalidades voltadas à adaptação de conteúdo, através

de várias metodologias que possibilitam uma transcodificação dinâmica dos dados. O WTP inclui funções para realizar o roteamento de requisições dinamicamente para o adaptador correspondente, baseado na URL de destino, nas características e capacidades de cada dispositivo e nas preferências do usuário. Alguns tipos de adaptação de conteúdo implementados pelo WTP são:

- Simplificação do HTML: modificação do HTML presente nos documentos removendo *tags* não suportadas por alguns dispositivos.
- Conversão HTML-WML: transformação de documentos HTML em WML (*Wireless Markup Language*) para envio a alguns tipos de celulares.
- XSLT (*eXtensible Stylesheet Language Transcoding*): permite a transformação de documentos XML em outros tipos de documentos.
- Fragmentação: consiste na divisão do documento em vários fragmentos para poderem ser enviados a determinados dispositivos.

Atualmente, o WTP utiliza os *User-Agents* transmitidos pelos navegadores para determinação das características dos dispositivos em uso pelos clientes. O WTP pretende vir a utilizar o CC/PP para explorar mais detalhadamente as características dos dispositivos. Embora o WTP possua um série de recursos voltados a adaptação de conteúdo, este não pode ser utilizado pelo pBuy, pelo fato de não possibilitar que o conteúdo adaptado, de forma dinâmica, seja enviado para um outro serviço integrante do projeto pBuy e não diretamente ao usuário como o faz através de um proxy. Outro fator que impede o seu uso está relacionado ao preço cobrado por esta ferramenta que faz parte de um conjunto de vários pacotes de desenvolvimento.

### 2.3.2 iMASH

O projeto iMASH (*Interactive Mobile Application Support for Heterogeneous Clients*) [PHA 2002], desenvolvido na Universidade da Califórnia, tem por objetivo

criar uma infra-estrutura em software e hardware para ser usada por dispositivos móveis. Para isso, fez-se necessário o desenvolvimento de um middleware chamado *Content Adaptation Pipeline* que fornece a adaptação de conteúdo a vários tipos de dados.

Este projeto utiliza XML como padrão para o armazenamento e transmissão de informações, incluindo características do dispositivo e do usuário, dos dados iniciais e dos dados adaptados.

Para um usuário que venha a utilizar o iMASH, a adaptação funcionará da seguinte maneira. Primeiramente, o usuário deverá registrar-se no servidor. Este registro é necessário para a obtenção de informações referentes ao usuário e as características do dispositivo que ficarão armazenadas no servidor. A partir disso, quando o usuário requisitar dados do servidor, este último passará as características dos dados solicitados e as informações referentes ao cliente para um *Gerador de Comandos de Adaptação* (*ACG - Adaptation Command Generator*). O ACG produzirá uma série de comandos a serem aplicados para a adaptação dos dados. Estes comandos serão passados para o *CAE* (*Content Adaptation Executor*) que aplicará os comandos sobre os dados e enviará os dados ao cliente.

Trabalhos futuros propõem a inclusão de novas funcionalidades voltadas à adaptação, além de estudos voltados ao ganho de desempenho e a segurança das informações da aplicação. O fato que desqualifica o iMASH para ser usado pelo pBuy é a necessidade que este tem de possuir registrado no servidor qual o dispositivo que será utilizado pelo usuário, isto vai na contra-mão do que propõe a computação pervasiva onde o usuário pode trocar constantemente de dispositivo.

### 2.3.3 CAF

O *CAF* (*Content Adaptation Framework*) [MOH 99] é um projeto desenvolvido no centro de pesquisas da IBM que apresenta uma ferramenta flexível para adaptação de conteúdo da Internet. Este projeto provê uma adaptação de conteúdo através da conversão dos dados em alternativas que vão ao encontro das características do cliente.

A ferramenta desenvolvida por este projeto é composta por vários componentes que juntos têm por objetivo fornecer o conteúdo adaptado. Estes componentes são responsáveis pelas seguintes funções:

- **Análise:** o dispositivo cliente e o conteúdo requisitado são analisados a fim de se obter informações relevantes aos próximos componentes.
- **Transcodificação:** baseado nas capacidades do dispositivo cliente, diferentes módulos de transcodificação são empregados para gerar versões do conteúdo em diferentes resoluções e modalidades.
- **Descrição:** este componente é responsável pela representação das múltiplas resoluções e modalidades do conteúdo transcodificado.
- **Seleção:** um módulo que melhor se enquadre nas características do dispositivo é selecionado. Cada versão do conteúdo requer diferentes recursos do cliente.
- **Renderização:** o conteúdo selecionado é, então, transformado para um formato adequado, por exemplo HTML. Este conteúdo será enviado a uma *cache* com a finalidade de obter ganho de tempo de resposta.

Este projeto apresenta uma boa estratégia em relação a adaptação de conteúdo, mas esbarra no fato de não suportar adaptação de forma dinâmica. O CAF seria uma boa escolha para o desenvolvimento de várias versões de um mesmo conteúdo que já fosse conhecido pelo sistema.

### 2.3.4 MDAT

MDAT (*Multi-Device Authoring Technology*) [MUL 2005] consiste numa metodologia e num conjunto de ferramentas com a finalidade de reduzir a complexidade da criação de aplicações Web que executem em diversos tipos de dispositivos. O MDAT segue três passos para a geração de suas aplicações:

- Generalização: conversão das páginas HTML existentes num modelo genérico, do qual possa se partir para a adaptação em qualquer outro formato, por exemplo, este modelo genérico pode ser obtido através da conversão para versões anteriores do próprio HTML ou para um XML;
- Especialização: modificação do modelo genérico para o dispositivo alvo específico;
- Refinamento Interativo: nesta fase é realizado o refinamento da aplicação.

O MDAT suporta tanto a adaptação nas fases de desenvolvimento, gerando diversas versões da aplicação para dispositivos já conhecidos, como também suporta adaptação em tempo de execução, gerando a versão a ser enviada ao dispositivo a partir da aplicação genérica no momento em que este requisitar a aplicação ao servidor.

Projetos futuros visam o refinamento do sistema, aumentando a sua usabilidade e a sua utilidade. Outros pontos-alvos de pesquisas referem-se à adaptação de aplicações Web já existentes, e a novos tipos de dados, incluindo som e vídeo. O MDAT é mais um dos projetos que não pode vir a ser utilizado pelo pBuy pelo fato de não suportar a adaptação dinâmica de conteúdo.

## Capítulo 3

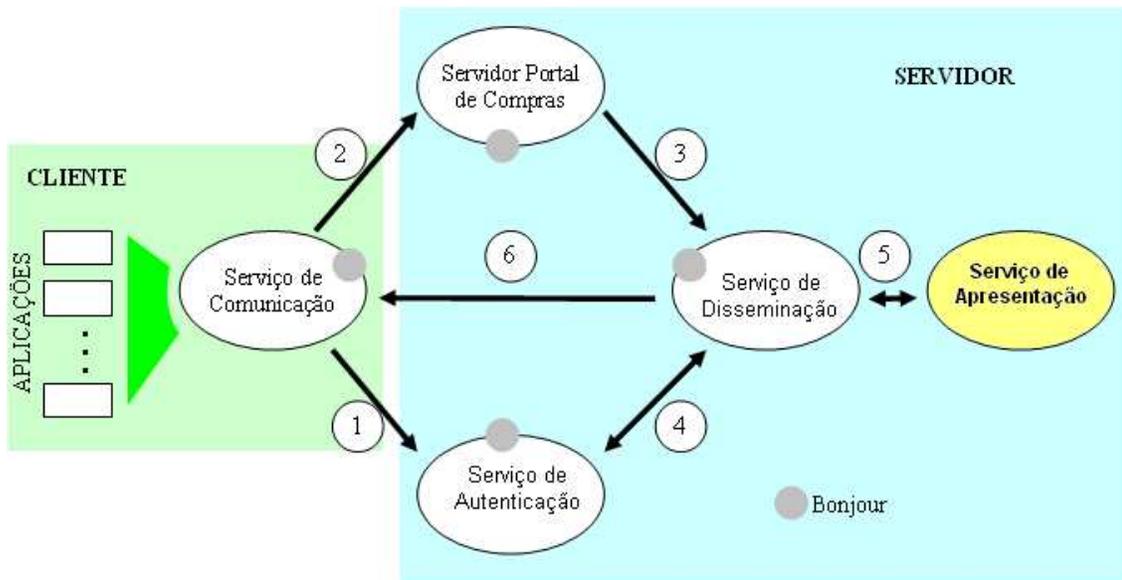
# Arquitetura de Serviços pBuy

O projeto pBuy, financiado pela FINEP, com período de execução de fevereiro/2005 a dezembro/2006, está sendo desenvolvido pelo grupo de pesquisa GMob (Grupo de Pesquisa em Sistemas de Computação Móvel - UFSM). O desafio é inserir tecnologia móvel com acesso pervasivo em um sistema legado, chamado Portal de Compras, desenvolvido pela empresa SIG Soluções em Informática e Gestão ([www.sigbrasil.com.br](http://www.sigbrasil.com.br)). Grande parte das aplicações existentes na área da computação pervasiva é experimental e interna aos grupos de pesquisa. O projeto pBuy é uma oportunidade de avaliar o impacto que essa nova tendência tecnológica traz ao mercado de produção de software.

O software Portal de Compras tem como base um sistema de leilão virtual para a realização de compras. O leilão é dividido em fases, delimitadas por datas ou ações da parte que realiza o leilão. Cada fase gera tanto mensagens do usuário para o sistema quanto mensagens do sistema para o usuário, sendo que estas últimas devem ser adaptadas para o dispositivo que o usuário estiver utilizando no momento.

Essas características geram a necessidade de um serviço de envio de mensagens (Serviço de Disseminação) para o usuário autenticado pelo sistema (Serviço de Autenticação) de forma uniforme e independente do dispositivo que será utilizado. O sistema deve estar ciente do dispositivo do usuário no momento do envio para fazer a adaptação do conteúdo (**Serviço de Apresentação**) de forma que o dispositivo aceite a mensagem e disponibilize-a para o usuário, levando em consideração o estado da rede e o tipo de conexão através do Serviço de Comunicação. A figura

3.1 mostra a integração dos serviços para o sistema proposto, ilustrando também o ciclo normal de funcionamento [BEL 2005].



**Figura 3.1:** Relação entre os serviços da arquitetura proposta no projeto pBuy e ciclo normal de atendimento a uma requisição da aplicação

Inicialmente, quando o serviço de comunicação for ativado no cliente, ele descobrirá, através do protocolo Bonjour [APP 2005], os serviços de autenticação, disseminação e o servidor do portal de compras.

O protocolo Bonjour é uma tecnologia que permite criar uma rede instantânea de computadores e outros dispositivos sem a necessidade de predefinir endereços IP ou fazer configurações em servidores DNS. O Bonjour permite que os serviços e as capacidades de cada dispositivo sejam registrados na rede, e que esses serviços sejam descobertos dinamicamente pelos outros dispositivos da rede [RED 2005]. O Bonjour funciona sob as tecnologias de conexão mais populares do mercado, incluindo Ethernet e IEEE 802.11 e sobre o padronizado e amplamente utilizado protocolo IP.

Após a ativação da comunicação, o usuário deverá então autenticar-se no sistema através do serviço de autenticação (1). Em seguida a aplicação do usuário provavelmente fará uma solicitação a alguma informação do portal (2), que por sua vez enviará o documento solicitado (em formato XML) e informações do usuário destino ao serviço de disseminação (3).

O serviço de disseminação deverá verificar, consultando o serviço de autenticação (4), se o usuário está conectado e autenticado, no caso da informação não ser pública. Essas informações, juntamente com o documento XML, são enviadas ao serviço de apresentação (5) para que sejam transformadas em algo visualizável pelo dispositivo do usuário. Finalmente, as informações solicitadas pelo dispositivo serão entregues pelo serviço de disseminação (6).

Nas próximas seções, os serviços são apresentados em maiores detalhes.

### 3.1 Serviço de Autenticação

Este serviço fará a autenticação dos usuários cadastrados no sistema. Além disso, guardará informações, como o dispositivo utilizado pelo usuário na seção atual e o estado da conexão, a fim de provê-las aos serviços de disseminação de mensagens e apresentação de conteúdo.

Para manter a informação do estado da conexão do dispositivo, o Serviço de Autenticação enviará periodicamente *heartbeats* (batidas cardíacas) [AGU 97]. O não envio, determinado por um (*timeout*), caracteriza o dispositivo como desconectado.

### 3.2 Servidor do Portal de Compras

O servidor do portal de compras é o provedor das informações sobre os leilões virtuais. É ele quem enviará os documentos XML ao serviço de disseminação para que sejam adaptados e enviados ao dispositivo do usuário.

Para que as mensagens sejam entregues aos destinatários corretos, o servidor do portal de compras deverá enviar ao serviço de disseminação, a identificação do usuário (através de um identificador provido pelo próprio usuário) e a mensagem.

### 3.3 Serviço de Disseminação de Mensagens

Os usuários da computação pervasiva são móveis e podem explorar as capacidades de vários tipos de dispositivos. O serviço de disseminação de dados deve ser transparente para quem o utiliza, ou seja, somente o usuário destino e a mensagem propriamente dita devem ser suficientes para a sua entrega ao destino e dispositivo correto. Assim, o serviço de disseminação [RED 2005] deve identificar onde o usuário destino está, qual equipamento está utilizando no momento e enviar a mensagem de forma adaptada ao dispositivo.

No pBuy, este serviço interage com os outros serviços de suporte às aplicações com comportamento pervasivo; em especial, com o serviço de apresentação, que realiza a adaptação do conteúdo, e com o serviço de autenticação, para confirmar que o destinatário esteja *online* e autenticado no sistema.

### 3.4 Serviço de Comunicação Consciente do Estado da Rede

A comunicação no ambiente pervasivo, por causa dos seus requisitos, envolve várias questões e desafios de pesquisa. Neste cenário, onde desconexões são frequentes, o serviço de comunicação [PIR 2005] deve levar em consideração o estado da rede e o tipo de conexão para que se adapte às demandas específicas de cada indivíduo.

As desconexões têm várias causas: voluntária, falta de energia, mudança de área de cobertura (*handoff*). Os terminais móveis são frequentemente desconectados da rede. As unidades móveis podem se desconectar de uma determinada rede caso não concordem com o nível de serviço disponível ou queiram economizar energia. A preocupação com a desconexão faz parte do suporte à computação pervasiva, por isso mecanismos para tratar *handoffs*, recuperar informações e consistência de *caching* são necessários.

O serviço de comunicação está permanentemente disponível no dispositivo móvel (PDA, celular) e é a ponte de comunicação das aplicações do pBuy executando no dispositivo com o mundo externo. Este é responsável por controlar o estado da

rede e identificar o tipo de acesso. Estratégias de *caching* são utilizadas para os momentos de desconexões. Outra questão tratada é como fornecer estas funcionalidade considerando as diferentes capacidades de PDAs e celulares.

### 3.5 Serviço de Apresentação de Conteúdo

A adaptação de conteúdo permite que informações sejam mostradas corretamente em diversos ambientes. No sistema pBuy, o serviço de apresentação está sendo utilizado pelo serviço de disseminação de informações que informa o tipo do dispositivo e os dados que sofrem o processo de adaptação automática. Os dados adaptados ao dispositivo retornam ao serviço de disseminação para o envio ao usuário.

O serviço de apresentação considera a heterogeneidade dos dispositivos portáteis e celulares existentes. Como não foram encontrados soluções (ver 2.3) que se encaixassem nas características do Projeto pBuy por envolver a comunicação entre serviços e a adaptação de conteúdo para mensagens criadas dinamicamente pelo sistema, concluiu-se que seria necessário desenvolver uma solução própria para atender aos requisitos funcionais do serviço. A modelagem e implementação desse serviço são detalhadas nos próximos capítulos.

# Capítulo 4

## Modelagem do Serviço de Apresentação

A seguir, apresenta-se um sumário das tecnologias usadas na implementação do serviço e uma abordagem sobre a modelagem e a implementação do Serviço.

### 4.1 Tecnologias Utilizadas pelo Serviço de Apresentação

A arquitetura do sistema pBuy utiliza a Plataforma Java 2 em suas três versões: J2SE, J2ME e J2SE, sobre a qual os serviços propostos são implementados. Por ser de conhecimento geral, essa tecnologia não será descrita.

#### 4.1.1 XML

*Extensible Markup Language* (XML) [XML 2005] é uma linguagem de marcação que possibilita uma representação estruturada dos dados, sendo amplamente implementável, de fácil desenvolvimento e capaz de descrever diversos tipos de dados.

XML é um subtipo de SGML (Standard Generalized Markup Language - padrão de marcação generalizada), definido pela *World Wide Web Consortium* (W3C), assegurando a uniformidade dos dados independente de aplicações e fornecedores. Uma das mais importantes características do XML se refere ao fato da separação entre o conteúdo e a forma de apresentação. A linguagem HTML especifica como

o documento deve ser apresentado na tela por um navegador. Já o XML define o conteúdo do documento. Por exemplo, em HTML são utilizadas *tags* para definir tamanho e cor de fonte, assim como formatação de parágrafo. No XML, as *tags* são usadas para descrever os dados, como exemplo *tags* de assunto, título, autor, conteúdo, referências, datas, etc...

XML também possibilita a integração de dados de diferentes fontes, interligando bancos de dados distintos. Por exemplo, um banco de dados pode, com o uso de uma aplicação, gerar um arquivo com seus dados, e um outro banco de dados qualquer pode, então, ler estes mesmos dados.

XML nos permite criar um número infinito de *tags*, além de prover um sistema para criar *tags* para dados estruturados. Com isso é possível realizar buscas mais eficientes.

O XML tem por objetivo a flexibilidade e o aumento do poder das aplicações Web. São vários os benefícios para desenvolvedores e usuários. Estes benefícios se referem tanto à realização de buscas mais eficientes quanto ao desenvolvimento de aplicações Web mais flexíveis, incluindo a integração de dados de fontes completamente diferentes; computação e manipulação local dos dados, além de múltiplas formas de visualização e de atualização do conteúdo.

#### 4.1.2 XSL

*Extensible Stylesheet Language* (XSL) [XSL 2005a], um padrão recomendado pelo W3C, é uma família de linguagens para a apresentação de documentos XML. Neste caso, diferentemente do XML, o navegador conhece as instruções do XSL.

O XSL pode ser entendido com uma linguagem que nos permite várias ações sobre documentos XML, tais como, transformar em diversos formatos (inclusive HTML), filtrar e classificar dados XML, considerar apenas algumas partes de um documento XML, formatar os dados baseados em seus valores, dentre outros.

A família XSL é composta de três linguagens: XPath, XSLT, XSL:FO.

##### **XSLT**

*Extensible Stylesheet Language Transformations* (XSLT) [XSL 2005], a parte

mais importante do padrão XSL, é uma poderosa tecnologia para a transformação de documentos XML em representações diferentes. O XSLT recebe como entrada um documento XML. Através da aplicação de *templates* é possível transformá-lo em outro documento XML ou em outros tipos de documentos também baseados em texto.

Além disso, com o uso do XSLT, é possível eliminar, adicionar, reestruturar e classificar elementos. Também é possível especificar quais objetos de formatação XSL serão aplicados a cada classe de elementos. Considerando o XML como um arquivo de dados em formato de árvore, uma transformação expressa em XSLT apresenta regras com a finalidade de transformar uma árvore fonte em uma árvore resultado. A transformação é obtida associando-se padrões encontrados no arquivo XML com padrões pré definidos, os chamados gabaritos. Um padrão é confrontado com elementos na árvore fonte. Um gabarito é instanciado para criar parte da árvore resultado.

Com a finalidade de obter-se ganho de desempenho, na utilização de XSLT no processamento de arquivos XML, é mais vantajoso utilizar o processamento no lado do servidor do que deixar este trabalho a cargo do o cliente, que, na maioria das vezes, é inferior em recursos. É necessária a utilização de um processador XSLT no servidor na execução dessa tarefa.

Muitos desses processadores são escritos em Java, de modo que esta linguagem constitui uma excelente escolha para o desenvolvimento de aplicações com o uso de XML. Um desses processadores é o Xalan [XAL 2005], desenvolvido pelo grupo Apache. Este processador, voltado a transformação de documentos XML, pode ser usado tanto usando linha de comando quanto usando a API. Xalan-Java é responsável pela interpretação do XSLT e do XPath.

### **XPath**

A *XML Path Language* (XPath) [XPA 2005] é parte importante da família de tecnologias XML, e foi projetada para ser usada em conjunto com o XSLT. Sua função principal é definir partes ou padrões de um documento XML através de um conjunto de regras de sintaxe.

A sintaxe adotada pela XPath utiliza expressões de caminhos para a definição de elementos XML. Essas expressões de caminhos são muito parecidas com as utilizadas em um sistema de arquivos de um computador. Documentos XML podem ser representados como árvores de nós (semelhante à visão de árvore das pastas do computador). Um padrão XPath é um caminho separado por barras que descreve as informações no documento XML. O padrão seleciona elementos que combinam com o caminho. Por exemplo, o caminho que começar com uma barra (/) representa um caminho absoluto para um elemento.

O XPath também define uma biblioteca de funções para trabalhar com *strings*, números e expressões *booleanas*, facilitando na seleção e manipulação dos elementos disponíveis no XML.

### **XSL:FO**

O conjunto de especificações *XSL Formatting Objects* XSL:FO [XSL 2005b], também conhecido apenas por XSL, é a tecnologia usada para definir os objetos de apresentação dos dados XML para saídas formatadas.

O processo de formatação ocorre após o XML ter passado pelo processador de XSLT e ter sido gerado a árvore resultante. A formatação é habilitada pela inclusão de semânticas de formatação na árvore resultado. Semânticas de formatação são expressas em função de classes de formatação de objetos. Estas classes podem especificar abstrações como tamanho da página, parágrafos, alinhamento, cor de fundo, espaçamento, bordas, ocupação de áreas, dentre outros. São encontrados 51 tipos de objetos de formatação com um total de 231 propriedades.

O processo de aplicação da formatação é realizado por um processador XSL-FO. Neste trabalho, o processador escolhido foi o FOP. O FOP (*Formatting Objects Processor*) [FOP 2005] é um processador de comandos que lê instruções de formatação e produz em uma página com formato específico. Atualmente, os formatos suportados são: PDF, PCL, PS, SVG, XML, Print, AWT, MIF e TXT.

### 4.1.3 RMI

A invocação Remota de Métodos Java (RMI) [SUN 2005] é uma versão orientada a objetos da *Remote Procedure Call* (RPC). A Utilização do RMI permite que um objeto invoque o método de outro objeto que pode estar localizado em um *host* diferente, de maneira transparente como se fosse feita a chamada a um método local. Para isso, é necessário que os programas Java obtenham a referência dos métodos contidos nos objetos remotos, o que é realizado pelo serviço de nomes do RMI Java.

## 4.2 Modelagem e Implementação do Serviço de Apresentação

Esta seção apresenta as decisões e os passos seguidos para a implementação do serviço de apresentação consciente do dispositivo em um ambiente pervasivo presente no projeto pBuy.

### 4.2.1 Decisões sobre o Processo de Adaptação

O primeiro passo para o desenvolvimento do serviço de apresentação foi escolher a melhor estratégia de implementação a ser seguida, no que se refere ao lugar e a maneira que ocorre a adaptação. Como visto anteriormente (seção 2.1.1), a adaptação irá ocorrer no Servidor.

Decidido pela adaptação no servidor, optou-se pelo desenvolvimento desta solução utilizando a tecnologia XSL (ver seção 4.1.2 por permitir realizar diversas operações sobre arquivos XML (ver seção 4.1.1), desde a adição e remoção de elementos, formatação e conversão para demais formatos de arquivos. Todas estas características são necessárias à adaptação de conteúdo.

Neste trabalho, é armazenada uma única versão com o conteúdo a ser apresentado em um arquivo no padrão XML, enquanto diferentes maneiras de apresentação (estilos XSL) serão criadas e aplicadas de acordo com as requisições e características dos dispositivos usados pelos usuários do sistema.

## 4.2.2 Funcionalidade Básica do Serviço de Apresentação

A partir do momento em que o serviço de disseminação receber uma mensagem, proveniente das aplicações do Portal de Compras, com dados a serem enviados para um determinado usuário, este serviço irá requisitar junto ao serviço de autenticação informações a respeito das características do dispositivo que o usuário estiver usando no momento. Caso o usuário, para o qual os dados devem ser enviados, não estiver *online* no momento do envio da mensagem, esta será armazenada para uma nova tentativa de envio na próxima vez que o usuário ingressar no sistema [RED 2005].

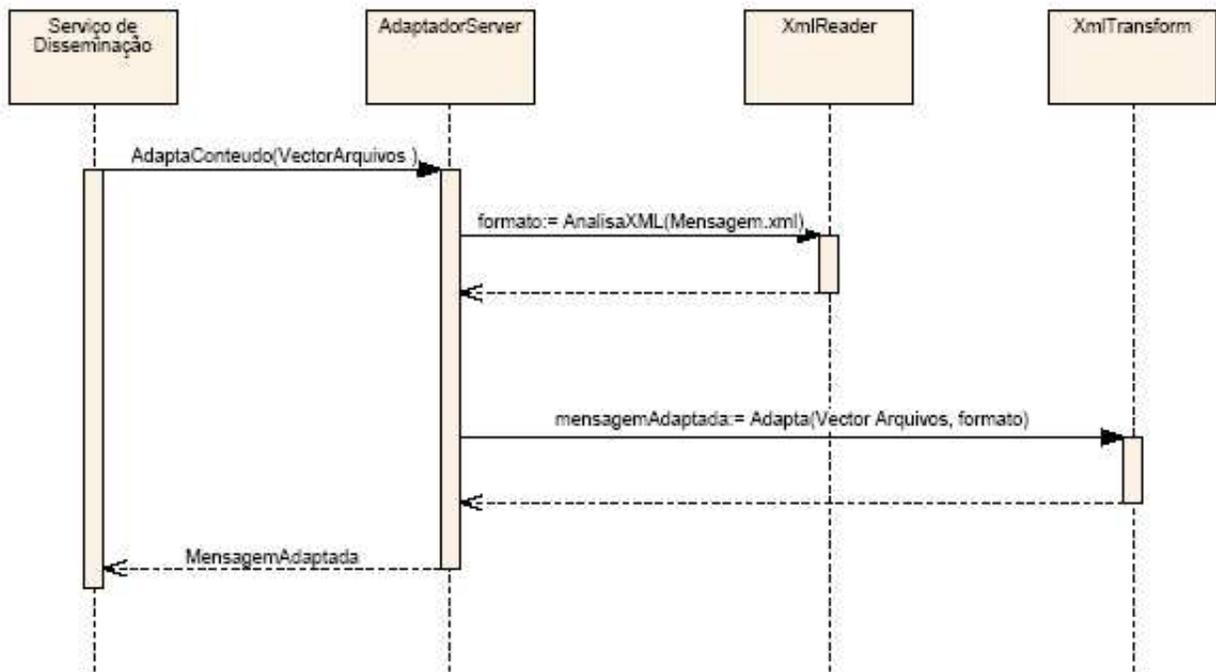
Estando o usuário conectado, o Serviço de Disseminação irá agrupar em um único arquivo XML as características do usuário com o conteúdo a ser adaptado. Além deste XML, pode-se enviar arquivos para que estes sejam anexados à mensagem adaptada. Seguindo as características do Portal de Compras, estes arquivos tipicamente serão imagens, que deverão ser anexadas à mensagem final. Todos estes arquivos serão submetidos ao Serviço de Apresentação através de uma invocação remota de métodos (RMI) 4.1.3.

No Serviço de Apresentação, o processo de adaptação ocorre em duas etapas (Análise, Aplicação). O funcionamento desse sistema é exposto no diagrama de seqüência apresentado na figura 4.1. A modelagem das etapas de adaptação é detalhada a seguir.

### 4.2.3 Etapa de Análise

Nesta primeira fase, todos os arquivos recebidos do Serviço de Disseminação (1) são armazenados em um diretório temporário pelo Servidor RMI do Serviço de Apresentação (2) para que, depois da adaptação, sejam removidos ou encaminhados ao usuário juntamente com o arquivo adaptado.

Após, o conteúdo do documento XML é analisado (3) para a extração das características do dispositivo ao qual o conteúdo irá ser adaptado. Esta análise é realizada pela classe *XmlReader* através do uso da API DOM, disponível a partir do JDK 1.4 da SUN. A escolha desta API está relacionada ao desempenho obtido



**Figura 4.1:** Diagrama de Sequência do Serviço de Apresentação

na leitura do arquivo XML, já que ela mapeia diretamente o XML na memória.

No XML recebido pelo sistema foram definidas duas *tags* auxiliares utilizadas na escolha do formato para o qual os dados serão adaptados. São elas a *tag* retorno e a *tag* dispositivo.

- **Dispositivo:** a *tag* dispositivo é utilizada para informar ao sistema com qual dispositivo o usuário está acessando o Portal de Compras. Esta *tag* pode conter os valores: `null`, `desktop`, `celular`, `pda`. De acordo com o valor encontrado nesta *tag* será realizada a adaptação de conteúdo. Para os valores `desktop` e `pda` o conteúdo está sendo adaptado para HTML, enquanto que para o valor `celular` o conteúdo será adaptado para wml. Se a *tag* chegar com o valor `null` significa que o dispositivo não foi identificado pelo sistema; para este caso, o serviço de apresentação adaptará o conteúdo da mensagem para o formato texto.

- **Retorno:** esta *tag* é a primeira a ser avaliada na leitura do XML, e é utilizada para informar quando o usuário solicitar um formato específico para seus dados, independente do dispositivo que estiver usando. Essa solicitação de formato será usada pelo usuário quando este receber alguma mensagem do sistema e resolver solicitar novamente a mesma mensagem para algum outro formato que julgue ser mais adequando às suas necessidades. Essa *tag* pode conter os valores: `null`, `html`, `wml`, `xml`, `pdf`, `txt`. Quando a *tag* estiver com o valor `null`, significa que o usuário não requisitou nenhum formato específico e o retorno deve proceder com base nas características do dispositivo que está sendo usado pelo usuário.

#### 4.2.4 Etapa de Aplicação

Nesta etapa (4), com base nas características do dispositivo obtidas na etapa anterior, um arquivo de estilo XSL será selecionado dinamicamente. Para esse sistema, foram criados cinco arquivos responsáveis pelos estilos empregados nas mensagens. Cada um desses arquivos é responsável por um formato que pode ser entregue ao usuário. Esses formatos podem ser HTML, WML, PDF, TXT e XML.

Estes arquivos de estilos são aplicados por duas classes do sistema, a `XmlFormat` e a `XmlTransform`.

- **XmlTransform:** esta classe é responsável pela geração dos formatos HTML, WML e XML. Ela aplica o *template* de acordo com o formato solicitado através do processador de comandos Xalan [XAL 2005]. Neste caso, os *templates* intercalam *tags* do HTML, WML ou XML para a formatação, com funções XPath e XSLT para a localização de dados no XML e para a geração do arquivo de saída.
- **XmlFormat:** esta classe é utilizada para a geração dos arquivos em pdf e texto. Estes tipos de arquivos necessitam de instruções adicionais para a definição do layout a ser usado. Para isso ocorre a adição de classes de formatação (XSL:FO) no arquivo de estilos. Esta formatação refere-se ao tamanho da pá-

gina, fonte, alinhamento, parágrafo, cores de fundo, dentre outros. Para esta etapa o processamento dos arquivos de estilo sobre o XML é realizado pelo processador de comandos FOP [FOP 2005].

#### 4.2.5 Arquivo XML do Sistema pBuy

Nesta seção é apresentado o formato de armazenamento, em XML, do arquivo que contém os dados a serem adaptados pelo Serviço de Apresentação. A estrutura principal deste arquivo pode ser visualizada na figura 4.2.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Mensagem>
  <retorno>null</retorno>
  <dispositivo>pda</dispositivo>
  <titulo>Licitação</titulo>
  <corpo>Corpo Da mensagem!</corpo>
</Mensagem>
```

Figura 4.2: Estrutura do arquivo Mensagem.xml

Este arquivo contém as *tags* **dispositivo** e **retorno** utilizadas para informar ao sistema qual o tipo de dispositivo que está sendo utilizado pelo usuário e qual o tipo de retorno solicitado. Estas *tags* foram explicadas na seção anterior.

Além destas, este arquivo também contém a *tag* **titulo** responsável por dar um nome à mensagem a ser adaptada, e à *tag* **corpo** onde é armazenado todo o conteúdo da mensagem.

Para o conteúdo da mensagem, localizado entre as *tags* **corpo**, é permitido o uso de mais uma série de *tags* que foram definidas de acordo com a necessidade das aplicações do Portal de Compras. Estas *tags*, na sua maioria, seguem o padrão HTML em seus atributos. Algumas das principais *tags* que podem ser utilizadas

são:

- **texto:** esta *tag* é utilizada para a adição de algum texto à mensagem que será enviada ao usuário.
- **imagem:** utilizada para a adição de imagens à mensagem final. Nesta *tag* é informado o nome do arquivo de imagem a ser utilizado. Esta imagem é recebida pelo Serviço de Apresentação juntamente com o arquivo XML para que possa ou ser encaminhada para o usuário no caso de uma adaptação para o formato Html, ou para que possa ser integrada ao documento, como no caso da geração de um arquivo em pdf. Para os demais formatos, que não suportam a adição de imagens, apenas será citado na mensagem adaptada a presença de uma imagem, mas o arquivo de imagem não é enviado ao usuário.
- **link:** esta *tag* é composta por outras duas *tags*. São elas as *tags* **arquivo** e **descricao**. A primeira indica o endereço do link e a segunda apresenta o texto que deve aparecer como link.
- **form:** como a maioria das aplicações desenvolvidas pelo Portal de Compras envolve a interação com usuário, é importante a adição da *tag* **form** para a elaboração de formulários. Esta *tag* ainda recebe dois atributos responsáveis por indicar a ação e o método (GET ou POST) a serem utilizados na submissão dos dados.
- **input:** esta *tag* é utilizada para que o usuário possa inserir dados em campos. Esta *tag* possui o atributo *type* que possibilita a escolha de como será feita a entrada de dados. As opções de escolha são, caixas de texto (*text*), campos de seleção do tipo *checkbox* ou *radio*, dentre outros.
- **select:** utilizada para a seleção de valores pré-definidos. Para a utilização desta, se faz uso da *tag* *option*, que define as opções de escolha que o usuário irá ter.

- **textarea**: esta *tag* é utilizada quando se deseja que a mensagem já adaptada possua uma caixa de texto.
- **button**: define a inserção de um botão que, quando pressionado, irá realizar alguma ação pré-definida em um dos seus atributos. Este botão somente é utilizado quando o conteúdo é adaptado para os formatos HTML e WML que suportam a submissão de informações.

Um exemplo do corpo da mensagem, fazendo a utilização destas *tags* pode ser visualizado na figura 4.3.

```
- <corpo>
  <texto>Universidade Federal de Santa Maria</texto>
  - <link>
    <arquivo>http://www.ufsm.br</arquivo>
    <descricao>UFSM</descricao>
  </link>
  <imagem>logo.jpg</imagem>
  - <form action="envia.do" method="GET">
    <texto>Escolha o seu curso:</texto>
    - <select name="curso">
      <option value="CC">Ciência da Computação</option>
      <option value="EE">Engenharia Elétrica</option>
      <option value="EC">Engenharia Civil</option>
    </select>
  </form>
</corpo>
```

Figura 4.3: Exemplo XML do corpo da mensagem a ser adaptada

#### 4.2.6 Arquivos de Estilo XSL

Nesta seção é descrita a estrutura e o funcionamento dos arquivos de estilos XSL. Nestes arquivos são definidos os dados que constarão na versão adaptada, bem como o *layout* que será aplicado. O XSL possui um elemento raiz que declara o documento a ser uma folha de estilos XSL. Este elemento é o `<xsl:stylesheet>`.

Seguido a este, tem-se o elemento `output`, indicando que aquela aplicação de estilo irá gerar uma saída e determinando o formato e a codificação desta.

Após este cabeçalho, uma folha de estilos XSL consiste de um conjunto de regras chamados moldes (*templates*). Cada elemento `<xsl:template>` contém regras para aplicar quando um nó específico do XML é combinado (*matched*). O atributo `match` também é usado para definir um molde para toda uma seção de um documento XML, por exemplo, `match="/"` define o todo o documento XML. A partir disso, são usados laços e instruções condicionais para a seleção dos dados que são obtidos através do elemento `<xsl:value-of>`.

A figura 4.4 apresenta um exemplo de um arquivo de estilo XSL. Neste exemplo, o estilo apenas está sendo utilizado para a obtenção do valor contido em uma *tag* `titulo` presente no xml no qual será aplicado o estilo.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="xml" encoding="ISO-8859-1" indent="yes" />
  - <xsl:template match="Mensagem">
    - <xsl:element name="Mensagem">
      - <xsl:element name="titulo">
        <xsl:value-of select="titulo" />
      </xsl:element>
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

**Figura 4.4:** Exemplo de arquivo de estilo XSL

# Capítulo 5

## Resultados

Com o objetivo de validar o Serviço de Apresentação foram realizados vários testes objetivando a adaptação de conteúdo para diversos formatos. Alguns desses testes são apresentados neste capítulo. Vários dos testes que foram realizados envolvem a participação de todos os serviços integrantes do pBuy. Entretanto, para a apresentação dos resultados apenas serão comentadas as etapas que envolvem desde a requisição da adaptação de conteúdo pelo Serviço de Disseminação, até o envio dos arquivos já adaptados.

### 5.1 Arquivo XML Utilizado nos Testes

Para os testes realizados foi selecionado um arquivo XML com o maior número de tipos de informações para serem adaptadas. Este XML possui textos, links, imagens e um formulário para a inserção de dados. A presença de um formulário se deve ao fato de que a maioria das aplicações do Portal de Compras envolve algum tipo de obtenção de dados do usuário, através do preenchimento de formulários *online*. Além disso, formulários possuem um grande número de elementos (caixas de texto, botões, caixas de seleção, etc) que devem ser adaptados.

O XML selecionado para ser adaptado pelo sistema é apresentado na figura 5.1. Este XML é composto por uma parte inicial, onde são informados o tipo dispositivo usado pelo usuário e o tipo de retorno caso seja realizada uma solicitação pelo usuário para um determinado formato, através das *tags* dispositivo e retorno. Esta

parte inicial está sendo explicada na seção 4.2.5. Após isso, é apresentado o título da mensagem através *tag* titulo e finalizando é apresentado o corpo da mensagem. Para este teste, o corpo da mensagem é composto por um texto intercalado por um link, uma imagem, outro link e um formulário composto por *checkpoints*, campos de texto e caixas de seleção.

O Serviço de Disseminação é o responsável pelo envio deste XML para o Serviço de Apresentação para que possa ser realizada a adaptação. O Serviço de Apresentação é invocado através de uma chamada remota (RMI) para a adaptação de conteúdo. Para que ocorra a adaptação nesta invocação remota de métodos, é passado como parâmetro um *Vector de Arquivos*. Arquivo é uma classe comum a todos os Serviços integrantes do pBuy e é composta por uma *String* que informa o nome do arquivo, e um *Array* de *bytes* com o conteúdo do arquivo.

Este *Vector* enviado para a adaptação contém em sua primeira posição o arquivo XML com a mensagem a ser adaptada, e nas demais posições arquivos (imagens) que podem ser agregados ao conteúdo adaptado e enviados para o usuário. No caso deste teste o Serviço de Disseminação enviou o arquivo XML com a mensagem a ser adaptada, mais um arquivo chamado ufsm.jpg.

Após o recebimento destes arquivos, o Serviço de Apresentação analisa os arquivos e define qual estilo deve ser aplicado ao conteúdo.

## 5.2 Adaptações Realizadas

O Serviço de Apresentação tem a capacidade de gerar cinco tipos diferentes de saídas (html, pdf, txt, wml, xml) de acordo com as características obtidas dos usuários. Nesta seção serão apresentados os resultados obtidos nos formatos html, pdf e txt.

### Adaptação para html

A adaptação para o formato html é realizada utilizando-se o interpretador Xalan para XSLT. O resultado da adaptação do XML (fig 5.1) pode ser visualizado na figura 5.2. Como pode ser observado neste caso, a adaptação foi realizada para o

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
< Mensagem >
  < retorno > null </ retorno >
  < dispositivo > desktop </ dispositivo >
  < titulo > Concorrência </ titulo >
  < corpo >
    < texto > Contratação de pessoa jurídica especializada na prestação de serviços técnicos de informática e telecomunicações, com vistas a desenvolver, otimizar e manter em perfeito funcionamento os sistemas, serviços e a infra-estrutura computacional do Ministério da Educação </ texto >
    < link >
      < arquivo > http://www.mec.gov.br/ </ arquivo >
      < descricao > (MEC) </ descricao >
    </ link >
    < texto > Entrega das propostas: 03/05/2006 às 09h30. </ texto >
    < imagem > ufsm.jpg </ imagem >
    < link >
      < arquivo > http://www.ufsm.br </ arquivo >
      < descricao > Universidade Federal de Santa Maria </ descricao >
    </ link >
    < form action = "cadastraPerfilUsuario.do" method = "GET" >
      < texto > Sexo: </ texto >
      < input type = "checkbox" name = "sexo" value = "M" > Masculino </ input >
      < input type = "checkbox" name = "sexo" value = "F" > Feminino </ input >
      < texto > Endereço: </ texto >
      < input type = "text" name = "endereco" size = "40%" />
      < texto > Cidade: </ texto >
      < input type = "text" name = "cidade" size = "40%" />
      < texto > Estado: </ texto >
      < select name = "estado" >
        < option value = "null" />
        < option value = "PR" > Paraná </ option >
        < option value = "RS" > Rio Grande do Sul </ option >
        < option value = "SC" > Santa Catarina </ option >
        < option value = "00" > Outro </ option >
      </ select >
      < texto > Deixe seu comentário: </ texto >
      < textarea cols = "40" rows = "5" name = "comentario" > ... </ textarea >
      < button name = "botao" > Enviar </ button >
    </ form >
  </ corpo >
</ Mensagem >

```

Figura 5.1: XML usado nos testes do Serviço de Apresentação

formato html ocorrendo a presença de uma imagem. Neste caso, quando o Serviço de Apresentação devolve ao Serviço de Disseminação o arquivo adaptado juntamente com o arquivo da imagem fazendo uso de um *Vector* de Arquivos.

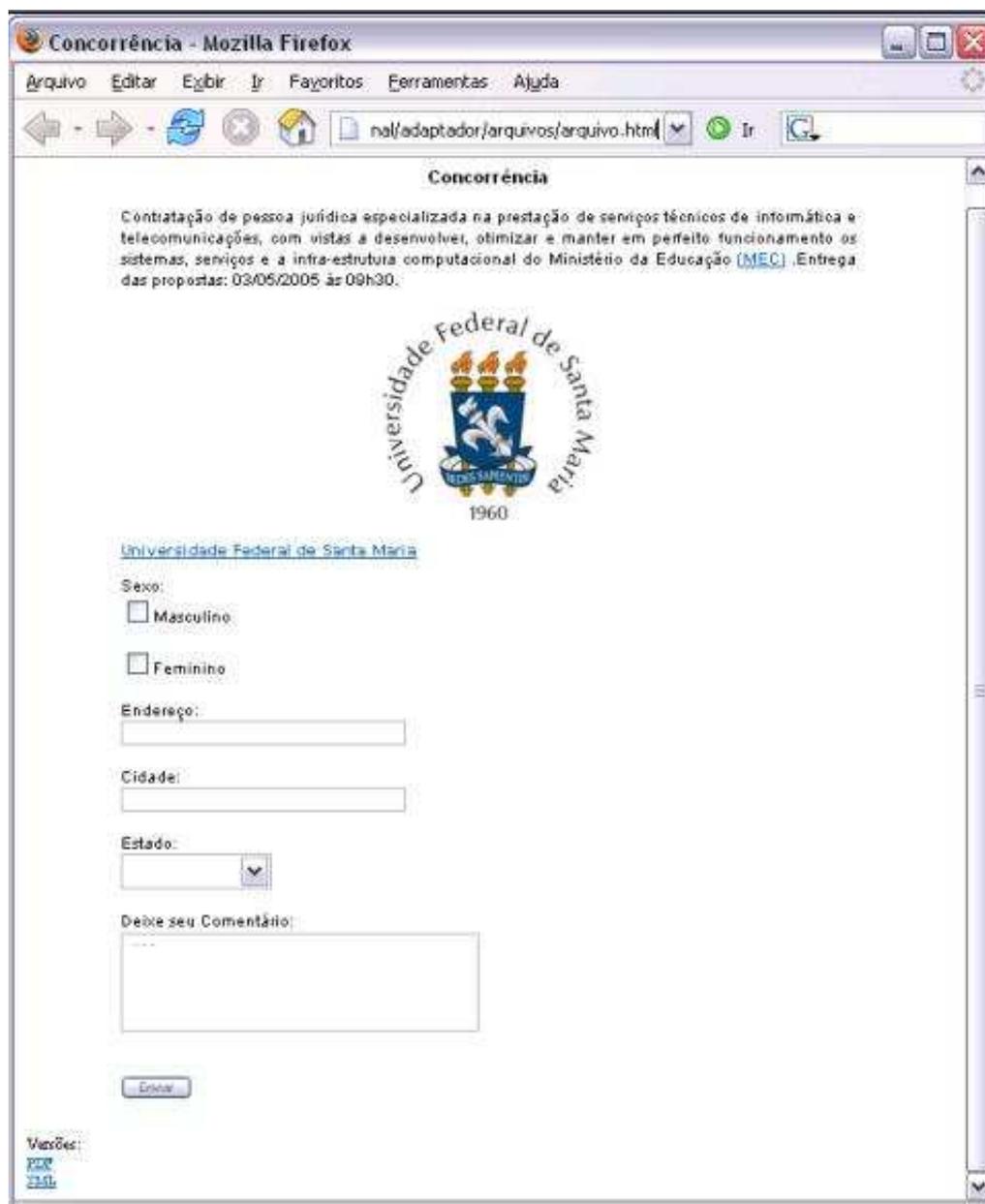


Figura 5.2: Resultado da adaptação para HTML

Além dos dados contidos no arquivo XML, também foram acrescentados *links* para versões em pdf e xml da mesma mensagem para que o usuário possa escolher outro formato que achar mais conveniente.

### Adaptação para PDF

A adaptação para PDF utiliza a tecnologia XSL:FO e com isso a interpretação fica a cargo do interpretador FOP. O resultado obtido pode ser observado na figura

## 5.3.

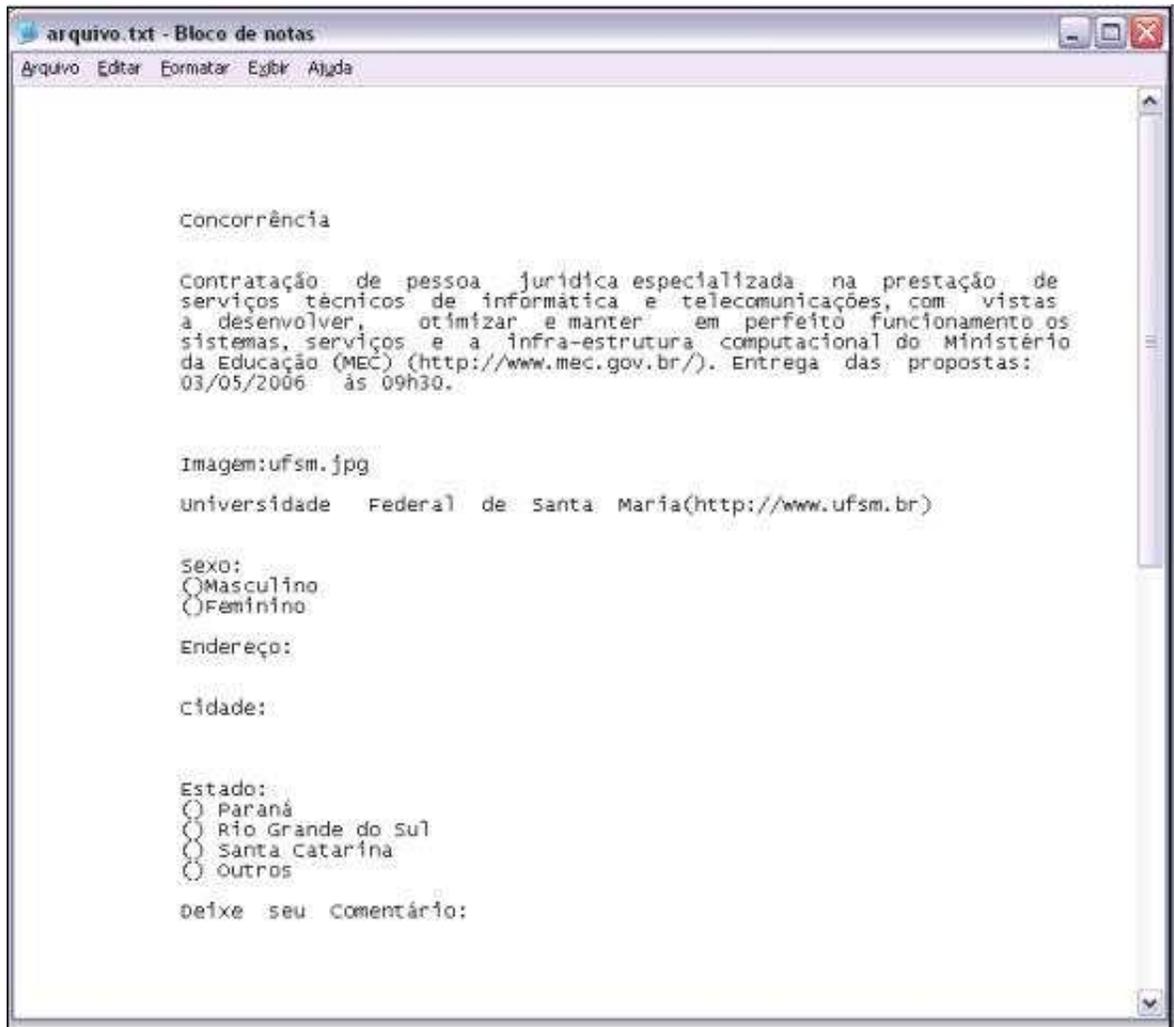


**Figura 5.3:** Resultado da adaptação para PDF

### Adaptação para txt

A adaptação para o formato texto ocorre somente quando o Serviço de Autenticação não conseguir identificar o dispositivo a partir do qual o usuário está acessando o sistema. Na adaptação para o formato texto, quando ocorre a presença de imagens, esta apenas será referenciada com o seu nome no meio do texto, mas o

arquivo da imagem não será enviado para o usuário. A adaptação para o formato texto pode ser observada na figura 5.4.



**Figura 5.4:** Resultado da adaptação para TXT

# Capítulo 6

## Conclusão

A computação pervasiva representa uma visão futura onde todos os elementos computacionais podem ser móveis. Este paradigma tem despertado muita atenção entre pesquisadores e indústria e pode ser facilmente resumido através dos por 3 A's: *Anywhere, Anytime, Any device*, pois visa disponibilizar o ambiente computacional do usuário a qualquer lugar, a qualquer tempo, independentemente do dispositivo utilizado.

Previsto para ser a computação do século 21, somente agora este cenário computacional começa a ser explorado. Resultados iniciais dos projetos de pesquisa identificaram os requisitos e questões a serem resolvidas para a concretização desse ambiente computacional.

No entanto, observa-se que implementações de aplicações voltadas à adaptação de conteúdo, fundamentais na computação pervasiva, vêm sendo exploradas de forma específica a aplicações WEB e não relativas ao ambiente pervasivo, fazendo com que a adaptação voltada ao dispositivo se torne inacessível, impondo restrições no desenvolvimento de aplicações pervasivas.

Este trabalho apresenta a implementação e utilização de um serviço responsável pela apresentação de conteúdo consciente do dispositivo, permitindo que informações sejam apresentadas em diversos ambientes. Este Serviço de Apresentação foi desenvolvido utilizando-se a tecnologia *Java 2 Standard Edition* juntamente com XSL (família de linguagens para a apresentação de documentos XML)

Como os dispositivos móveis são diversos e com variadas capacidades, esta

---

solução pode ser utilizada em outras aplicações e, devido ao tema ser inovador, constitui-se de uma oportunidade de gerar futuros trabalhos.

Algumas sugestões de trabalhos futuros concentram-se em: (i) melhorar a forma como é realizada a identificação dos dispositivos que acessam o sistema, possibilitando assim obter-se um número maior de características dos dispositivos e, conseqüentemente, tornar a adaptação mais rica em detalhes; (ii) aumentar o número de formatos suportados pela adaptação, incluindo a adaptação de imagens e vídeo de acordo com o dispositivo.

# Referências Bibliográficas

- [AGU 97] AGUILERA, M. K.; CHEN, W.; TOUEG, S. Heartbeat: A timeout-free failure detector for quiescent reliable communication. In: LECTURE NOTES IN COMPUTER SCIENCE: DISTRIBUTED ALGORITHMS, PROC. OF 11TH INTERNATIONAL WORLD INTERNATIONAL WORKSHOP, WDAG'97, 1997, Saarbrücken, Germany. **Anais. . .** Springer, 1997. v.1320, p.126–140.
- [APP 2005] APPLE developer connection - Bonjour. <http://developer.apple.com/networking/bonjour/index.html> - acessado em 02/08/2005.
- [AUG 2004] AUGUSTIN, I. **Abstrações para uma linguagem de programação visando aplicações móveis em um ambiente de pervasive computing**. 2004. Tese (Doutorado em Ciência da Computação) — UFRGS - Universidade Federal do Rio Grande do Sul.
- [AUG 2005] AUGUSTIN, I. **Managing the follow-me semantics to build large-scale pervasive applications**. [S.l.: s.n.], 2005. Middleware 2005 - Workshop on Pervasive and Ad-hoc Computing (MPAC'05). Grenoble, France.
- [BEL 2005] BELUSSO, R. et al. Arquitetura de serviços para o portal de compras pervasivo. **Simpósio de Informática da Região Centro do RS**, 2005.

- [CAS 2005] CASCADING style sheets 2.1 specification. w3c candidate recommendation. Disponível em <http://www.w3.org/TR/CSS21/>, acessado em agosto / 2005.
- [CC/P 2005] CC/PP (composite capabilities / preference profiles). Disponível em <http://www.w3.org/Mobile/CCPP/>, acessado em agosto / 2005.
- [DEY 2001] DEY A., A. G. **Towards a better understanding of context and context-awareness**. [S.l.: s.n.], 2001. Technical Report GIT-GVU-99-22, Georgia Institute Technology.
- [FOP 2005] FOP, formatting objects processor. Disponível em <http://xmlgraphics.apache.org/fop/>.
- [IBM 2005] IBM websphere transcoding publisher, ibm corporation. <http://www.ibm.com/software/Webservers/transcoding/>, acessado em julho/2005.
- [MOH 99] MOHAN, R.; SMITH, J. R.; LI, C.-S. Content adaptation framework: bringing the internet to information appliances. **Global Telecommunications Conference**, 1999.
- [MUL 2005] MULTI-DEVICE authoring technology. Disponível em <http://www.ibm.com/software/pervasive/products/>, acessado em julho/2005.
- [PHA 2002] PHAN, T.; ZORPAS, G.; BAGRODIA, R. An extensible and scalable content adaptation pipeline architecture to support heterogeneous clients. **Internacional Conference on Distributed Computing Systems**, 2002.
- [PIR 2005] PIRES, R. **Serviço de comunicação consciente do estado da rede em um ambiente pervasivo**. 2005. Monografia (Graduação em Ciência da Computação) — Universidade Federal de Santa Maria.

- [RED 2005] REDIN, R. **Serviço de suporte a disseminação de informações independente de dispositivo em um ambiente de computação pervasiva**. 2005. Monografia (Graduação em Ciência da Computação) — Universidade Federal de Santa Maria.
- [SAH 2003] SAHA, D.; MUKHERJEE, A. Pervasive computing: a paradigm for the 21<sup>st</sup> century. **IEEE Computer Society**, v.36, n.3, p.25–31, Março 2003.
- [SAT 2001] SATYANARAYANAN, M. Pervasive computing: vision and challenges. In: **IEEE PERSONAL COMMUNICATIONS**, 2001, New York. **Anais...** [S.l.: s.n.], 2001.
- [SUN 2005] SUN microsystems. Java Remote Method Invocation (Java RMI). <http://java.sun.com/products/jdk/rmi>, acessado em agosto/2005.
- [USE 2005] USER agent accessibility guidelines 1.0. Disponível em <http://www.w3.org/TR/UAAG10/>, acessado em agosto / 2005.
- [USE 2005a] USER agent profile specification. Disponível em <http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html>, acessado em julho/2005.
- [WEB 2004] WEB content adaptation. White Paper, MediaLab, TeliaSonera Finland, <http://www.medialab.sonera.fi>.
- [XAL 2005] XALAN, xslt processor. Disponível em <http://xml.apache.org/xalan/>, acessado em agosto / 2005.
- [XML 2005] XML, extensible markup language. Disponível em <http://www.w3.org/XML/>, acessado em agosto / 2005.
- [XPA 2005] XPATH, xml path language. Disponível em <http://www.w3.org/TR/xpath>, acessado em agosto / 2005.

- [XSL 2005a] XSL, extensible stylesheet language. Disponível em <http://www.w3.org/Style/XSL/>, acessado em agosto / 2005.
- [XSL 2005b] XSL:FO, xsl formatting objects. Disponível em <http://www.w3.org/TR/xsl/>, acessado em agosto / 2005.
- [XSL 2005] XSLT, xsl transformations. Disponível em <http://www.w3.org/TR/xslt/>, acessado em agosto / 2005.