

CURSO DE CIÊNCIA DA COMPUTAÇÃO

ARQUITETURA PARA SISTEMA DE ACESSO A DADOS VIA DISPOSITIVOS FIXOS E MOVEIS INDEPENDENTES DE PLATAFORMA.

Tiago Aparecido Teixeira

Monografia desenvolvida durante a disciplina de Projeto Orientado e apresentada ao Curso de Bacharelado em Ciência da Computação da Faculdade Ubaense Ozanam Coelho, como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Luciano Eugênio de Castro Barbosa

Ubá, 03 de dezembro de 2003.

Faculdade Ubaense Ozanam Coelho
Curso de Bacharelado em Ciência da Computação

Diretor:

Prof. Dr. Roberto Santos Barbieri

Coordenador do Curso de Ciência da Computação:

Prof. Marcelo Oliveira Andrade

Coordenado das Disciplinas de Projeto Orientado:

Prof. Marcelo Oliveira Andrade

Data da defesa: 03/12/2003.

Banca Avaliadora composta por:

Prof. Luciano Eugênio de Castro Barbosa (Orientador)

Prof. Marcelo Oliveira Andrade

Prof. Eduardo Vicente do Prado

Revisão Ortográfica:

Profª. Fernanda Cristina Abrão

CIP – Catalogação na Publicação

TEIXEIRA, Tiago Aparecido

Arquitetura para sistema de Acesso a Dados Através de Dispositivos Fixos e Moveis Independentes de Plataforma / Tiago Aparecido Teixeira; orientado por Luciano Eugênio de Castro Barbosa. – Ubá: 2003.

136 p.: il.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação). Faculdade Ubaense Ozanam Coelho, 2003.

1. Convergência 2. Wireless. 3. Multiplataforma. I. Barbosa, Luciano Eugênio de Castro. II. Título Sistema de Acesso a Dados Através de Dispositivos Fixos e Moveis Independentes de Plataforma.

Endereço:

Faculdade Ubaense Ozanam Coelho
R. Adjalme Silva Botelho, 20 – Seminário
CEP 36500-000 – Ubá – MG – Brasil

SUMÁRIO

ÍNDICE DE FIGURAS	8
ÍNDICE DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 Introdução.....	13
2 Tecnologias Analisadas.....	15
2.1. Ferramentas de Desenvolvimento	15
2.1.1. Ferramentas RAD.....	15
2.1.2. Ferramenta CASE	16
2.1.3. Ferramentas RAD versus Case.....	17
2.2. Sistema de Gerenciamento de Banco de Dados ..	19
2.3. Sites Dinâmicos.....	21
2.4. Tecnologia de Comunicação Móvel.....	24
2.5. Processo de Desenvolvimento de Software	26
3. Metodologia	28
3.1. Utilização de Ferramentas RAD e SGBD em Conjunto	28
3.2. Utilização das Tecnologias de Sites dinâmicos e Computação Móvel em Conjunto	30
3.3. Utilização das Tecnologias de Sites dinâmicos e SGBD em Conjunto	31
3.4. Utilização das Tecnologias de Site dinâmicos e Ferramentas RAD em Conjunto.....	32
3.5. Integração das Tecnologias Analisadas.....	32
4. Elaboração da Arquitetura.....	34
4.1. Modo Desktop	36
4.2. Modo Web.....	39
4.2.1. Modo Web Convencional.....	39
4.2.2. Modo Web Móvel	42
5. Estudo de Caso	47
5.1. Tecnologias Utilizadas para o Estudo de Caso ..	49
5.1.1. Ferramenta RAD	49

5.1.2. Sistema de Gerenciamento de Banco de Dados.....	51
5.1.3 Sites Dinâmicos.....	52
5.1.4. Tecnologia de Comunicação Móvel.....	56
6. Conclusão.....	58
7. Bibliografia.....	60
Anexo A – Tutoriais Criados Para o Desenvolvimento da Arquitetura	62
1. Servidor de Banco de dados MySQL:.....	63
1. 1. O que é o MySQL?.....	63
1. 1. 1. Principais características do MySQL:	64
1. 2. Softwares necessários:.....	66
1. 3. Configuração no Microsoft Widows.	67
1. 4. Configuração no Linux.....	67
1. 5. Mini ajuda de comandos MySQL:	69
1. 6. Integração MySQL J2EE	70
1. 7. Integração MySQL Delphi / Kylix.....	71
1. 7. 1. No Windows (Delphi):.....	71
1. 7. 2. No Linux (Kylix):.....	72
1. 8. Referencia bibliográfica:	72
2. Servidor Apache Jakarta-TomCat e J2SDK.....	74
2. 1. Por que JSP?.....	74
2. 2. Softwares nescessarios:	74
2. 3. Configuração do J2SDK.....	75
2. 3. 1. No Windows 9x:.....	75
2. 3. 2. No Windows NT, 2000, XP	76
2. 3. 3. No Linux	76
2. 4. Configuração do servidor Apache Jakarta-TomCat.....	76
2. 4. 1. No Windows 9x:.....	77
2. 4. 2. No Windows NT, 2000, XP	77
2. 4. 3. No Linux	78
2. 5. Integração Apache Jakarta-TomCat / Apache....	80
2. 5. 1. Compatibilidades do TomCat / Apache	80

2. 5. 2. Como funciona o redirecionamento TomCat / Apache?	81
2. 5. 3. Obtendo o mod_jk	82
2. 5. 4. Instalação e configuração do redirecionador TomCat / Apache	83
2. 5. 5. O velho “mod_jserv”	83
2. 5. 6. TomCat usando auto-configuração	84
2. 5. 7. Exemplo de uma configuração simples	84
2. 5. 8. Forwarding	87
2. 5. 9. Configurando o Apache para servir arquivos de aplicação Web estáticos	90
2. 6. Integração Apache Jakarta-TomCat / IIS	91
2. 6. 1. Compatibilidades do TomCat / IIS	91
2. 6. 2. Como funciona o redirecionador do TomCat / IIS ?	91
2. 6. 3. Instalação e configuração do redirecionador TomCat / IIS	92
2. 6. 4. Criação de um arquivo de registro	94
2. 6. 5. Um exemplo do arquivo uriworkermap.properties	95
2. 6. 6. Um exemplo do arquivo worker.properties	95
2. 7. Alguns comandos Java JSP	96
2. 7. 1. Algumas tags úteis:	96
2. 7. 2. Algumas diretivas JSP	96
2. 7. 3. Alguns comandos úteis para programação Java / JSP	97
2. 8. Referência bibliográfica:	98
3. <i>Servidor WAP</i>	99
3. 1. O que é WAP ?	99
3. 2. Softwares necessários:	99
3. 3. Configuração do Nokia Mobile Internet Toolkit	100
3. 4. Conceitos básicos sobre WML	100
3. 5. Desenvolvendo em WML:	101

3. 5. 1. A estrutura de um arquivo WML	101
3. 5. 2. As definições dos elementos da WML.....	102
3. 5. 3. Decks e Cards.....	103
3. 5. 4. Atributos.....	104
3. 5. 5. Eventos	105
3. 5. 6. Tarefas	106
3. 5. 7. Variáveis.....	106
3. 5. 8. Mecanismo de entrada de Usuário	107
3. 5. 9. O elemento <do>	109
3. 5. 10. O elemento <timer>	110
3. 5. 11. O elemento <go>	111
3. 5. 12. O elemento <input>.....	112
3. 5. 13. O elemento <select>.....	114
3. 5. 14. O elemento	115
3. 5. 15. O elemento <table>	117
3. 5. 16. Exemplo de código WML	118
3. 6. Referencia Bibliográfica:	119
Anexo B – Publicação no 3º CONIC / 1º COINT	120
Anexo C – Publicação no 1º CONISLI	123
1. Introdução.....	124
2. Revisão Bibliográfica.....	125
2.1. Ferramentas RAD.....	125
2.2. Sistemas de Gerenciamento de Bancos de Dados	125
2.3. Sites Dinâmicos.....	125
2.4. Sistema de Comunicação Móvel	126
3. Metodologia	126
4. Projeto	127
5. Estudo de Caso	127
5.1. Sistema de Gerenciamento de Banco de Dados	130
5.2. Cliente Desktop	130
5.2.1. Ferramenta RAD	131
5.3. Cliente Web.....	131
5.3.1. Cliente Web Convencional.....	131

5.3.2. Cliente Web Móvel	132
6. Conclusão	132
7. Referências	133
Anexo D – Citação no 4º CIIEE.....	134
Palavras chaves: acesso móvel, tecnologias gratuitas, multiplataforma, voiceml, jsp.....	135

ÍNDICE DE FIGURAS

Figura 2.1 – Arquitetura de sites dinâmicos.....	22
Figura 3.1 – Arquitetura Multiple Tier.....	29
Figura 4.1 – Arquitetura Proposta.....	34
Figura 4.2 – Casos de uso da arquitetura proposta.....	36
Figura 4.3 – Atividades relacionadas ao caso de uso Modo Desktop.	38
Figura 4.4 – Seqüência de estados possíveis para o caso de uso Modo Desktop.....	39
Figura 4.5 – Atividades relacionadas ao caso de uso Modo Web Convencional.....	41
Figura 4.6 – Seqüência de estados possíveis para o caso de uso Modo Web Convencional.	42
Figura 4.7 – Atividades relacionadas ao caso de uso Modo Web Móvel.	45
Figura 4.8 – Seqüência de estados possíveis para o caso de uso Modo Web Móvel.....	46
Figura 5.1 – Componentes utilizados na implementação do estudo de caso.....	48
Figura 5.2 – Construção do Sistema na Plataforma Linux utilizando Borland Kylix.....	50
Figura 5.3 – Recopilação do Sistema na Plataforma Windows utilizando o Borland Delphi.....	50
Figura 5.4 – Utilização do sistema nas Plataformas Windows (a esquerda) e Linux (a direita).....	51
Figura 5.5 – Modelagem Conceitual dos Dados	52
Figura 5.6. – Modelagem Física dos Dados	52
Figura 5.7 – Tela inicial do sistema nas Plataformas Windows(a esquerda) e Linux (a direita).....	54
Figura 5.8 – Tela representando uma iteração com o usuário em Linux.....	55
Figura 5.9 – Tela representando uma iteração com o usuário em Windows	56

Figura 5.10 – Etapas necessárias para a votação via Modo Web Móvel	57
---	----

ÍNDICE DE TABELAS

Tabela 4.1 – Eventos gerados pelo caso de uso Modo Desktop.....	37
Tabela 4.2 – Eventos gerados pelo caso de uso Modo Web Convencional.	40
Tabela 4.3 – Eventos gerados pelo caso de uso Modo Web Móvel.....	43

RESUMO

Há diversas tecnologias de sistemas independentes de plataforma existentes, que geralmente possuem um elevado custo e uma alta dependência de dispositivos específicos e cada vez mais sofisticados. Este trabalho propõe uma arquitetura dentro desta filosofia, a SAD-DMF/IP (Sistema de Acesso a Dados em Dispositivos Fixos e Móveis Independentes de Plataforma), demonstrando que uma elaboração da mesma pode ser implementada a baixo custo, quando se utilizam tecnologias gratuitas disponíveis na internet. A abordagem utilizada busca a convergência de algumas tecnologias existentes para que possibilitem o acesso aos dados de uma organização ou instituição, independente de onde o usuário esteja ou de que dispositivo ou sistema operacional utilize.

Palavras-chaves: Convergência; Wireless; Multi-plataforma.

ABSTRACT

It has diverse technologies of existing independent systems of platform, that generally poses one high cost and high dependence of sophisticated specific devices and each time more. This work inside considers an architecture of this philosophy, the SAD-DFM/IP (System of Access the Data in Independent Fixed and Mobile Devices of Platform), demonstrating that an elaboration of the same one can be implemented the low cost, when uses available gratuitous technologies in the internet. The used boarding search the convergence of some existing technologies so that they make possible the access to the data of an organization or institution, independent of where the user it is or that device or operational system uses.

Key-words: Convergence; Wireless; Multiple-platform

1 Introdução

Uma necessidade observada em muitas organizações é o acesso às informações armazenadas em suas bases de dados, independente de local físico, sistema operacional e dispositivo de acesso. Para exemplificar, podemos citar o caso em que um representante comercial necessita acessar e modificar dados contidos nos sistemas de informação da organização em que trabalha. Tal acesso não pode depender de sistemas de comunicação específicos, pois o representante não sabe que tipo de tecnologia de comunicação poderá encontrar junto ao cliente.

Alguns computadores portáteis possuem navegadores que permitem a visualização de páginas desenvolvidas utilizando como fonte a linguagem HTML, mas para isso a página deverá ter seguido alguns critérios para que a fonte HTML seja a mais simples e compacta possível.

A maioria dos celulares em uso no Brasil atualmente não possui acesso a Web Móvel, mas sim a um Sistema de Mensagens Curtas. Já na Europa e nos Estados Unidos a maioria dos celulares vendidos atualmente possuem um Sistema Global para Comunicações Moveis, o qual suporta vários protocolos de comunicação e começa a ser introduzido no Brasil.

Este trabalho propõe a utilização de tecnologias que interajam entre si a fim de prover aos seus usuários o acesso às informações de sua organização, dentro da filosofia de acesso independente de plataforma e dispositivo (sendo este último fixo ou móvel). Será utilizado o protocolo WAP para o acesso a Web Móvel,

por possuir mais recursos que o Sistema de Mensagens Curtas, e por ser suportado pelo Sistema Global para Comunicações Moveis.

A estrutura desse documento está descrita a seguir:

O segundo capítulo apresenta as tecnologias analisadas para a elaboração da arquitetura. O capítulo três especifica a metodologia utilizada na integração das tecnologias utilizadas no desenvolvimento da arquitetura. Estes dois capítulos apresentam a base tecnológica utilizada na elaboração da arquitetura.

O capítulo quatro apresenta o projeto da arquitetura, tanto no nível lógico quanto físico, e o capítulo cinco a implementação do estudo de caso: Sistema de Votação Eletrônica utilizando um Sistema de Acesso a Dados Através de Dispositivos Fixos e Moveis Independentes de Plataforma. Finalmente, o sexto capítulo apresenta a conclusão do trabalho e o sétimo, a bibliografia.

Após os capítulos, este trabalho conta ainda com um apêndice que contém um Mini-tutorial para a configuração dos Servidores e utilização das linguagens usadas no estudo de caso.

2 Tecnologias Analisadas

2.1. *Ferramentas de Desenvolvimento*

2.1.1. *Ferramentas RAD*

Atualmente, uma nova metodologia – RAD (Rapid Application Development, ou Desenvolvimento Rápido de Aplicações) – com características próprias diferentes das técnicas tradicionais, está sendo cada vez mais usada em produtos comerciais de desenvolvimento de aplicações. O método RAD beneficia-se das vantagens oferecidas por ferramentas poderosas de desenvolvimento para produzir rapidamente sistemas de alta qualidade. Essas novas ferramentas, mais intuitivas e visuais, permitem que o projetista de software construa "visualmente" a interface com o usuário, além de gerar o código numa linguagem orientada a objetos, permitindo o reuso de componentes [1]¹.

2.1.1.1. **Ambientes Integrados**

O conceito de **IDE** (Integrated Development Environment) surgiu com o Turbo Pascal da Borland, e consistia na integração do editor de textos, compilador, linkeditor e depurador em um único aplicativo, formando um ambiente para o desenvolvimento de software. Antes dos IDEs, estes componentes eram programas isolados (como devem estar lembrados os programadores Clipper) e este modelo que ainda é utilizado pela maioria das

¹ Uma Metodologia de Projeto Orientado a Objetos Com Vistas à Reutilização Fabrício André Rubin Prof. Dr. Carlos Alberto Heuser (Orientador)

ferramentas de desenvolvimento tradicionais do Linux como o GCC, Perl, PHP e Python [2]².

Esta idéia evoluiu para o conceito de RAD com o surgimento do Visual Basic, sendo logo adotado por ferramentas como o Watcom C++, SQLWindows e PowerBuilder [2].

Um IDE RAD é focado no desenvolvimento visual de sistemas de informação cliente/servidor, portanto, dois componentes fundamentais de um IDE RAD são o construtor de formulários e os componentes ou classes de acesso a bancos de dados. Além disso, um IDE RAD deve estar baseado em uma linguagem de programação de fácil aprendizado [2].

2.1.2. Ferramenta CASE

Algumas técnicas estruturadas de sistema evoluíram com o avanço da tecnologia dando origem às ferramentas CASE. A sigla CASE (Computer-Aided Software Engineering) significa Engenharia de Software Auxiliada por Computador e é usada para designar os produtos que auxiliam no projeto e desenvolvimento de um software.

Um produto é classificado como uma Ferramenta CASE quando este oferece documentação, automação e racionalização do projeto de software e de sua implementação. Dentre as classes de Ferramentas CASE, podemos destacar os geradores automáticos de aplicações

² Ambientes Integrados por Fernando Lozano – fernando@lozano.eti.br para a revista do Linux

e as ferramentas de modelagem, projeto e implementação de bancos de dados.

Uma ferramenta Case é qualquer Software (SW) que auxilia as pessoas que trabalham em um ADS. Elas existem auxiliando todo ciclo de desenvolvimento (Análise, Projeto, Implementação e Teste) e são também de suma importância para a manutenção do SW. Há também ferramentas Case para apoiar a gerência dos projetos de desenvolvimento.

A tecnologia CASE veio para redefinir o ambiente de desenvolvimento de software com a automação do desenvolvimento e da manutenção, transformando-se na mais avançada e poderosa ferramenta no que diz respeito ao aumento da produtividade. Afinal o objetivo dos sistemas CASE é economizar o trabalho e automatizar todas as fases do ciclo de vida de um software.

A principal característica de uma ferramenta CASE é a construção automática de um Banco de Dados do projeto ou definição de elementos de dados. Este BD é criado dentro da própria ferramenta e é chamado de repositório, onde o mesmo mantém todas as informações sobre os dados a serem implementados e informações relativas aos requisitos do sistema de projeto [3]³[4]⁴.

2.1.3. Ferramentas RAD versus Case

³ Apostila: Ferramentas Case - Trabalho desenvolvido pelos alunos de Tecnologia em Informática da UNIVALE em Outubro/1996.

⁴ Manual Dr. Case Autor: Squadra tecnologia em software LTDA
www.squadra.com

Aqueles que estão prestes a iniciar um projeto de software se defrontam com um dilema. Duas correntes disputam a atenção dos projetistas de software. A primeira advoga o uso de ferramentas RAD, para abreviar o desenvolvimento de software; enquanto a outra, propõe uma abordagem metodológica para a criação de um novo programa de computador, apoiando-se no uso de ferramentas CASE [5]⁵.

Todas as modernas ferramentas de programação visual oferecem um variado conjunto de recursos que permite construir rapidamente um programa aplicativo. Auxiliadas por agentes e bibliotecas com centenas de milhares de linhas de código, tornam o desenvolvimento de software uma tarefa muito mais ágil [5].

Induzidos pela possibilidade de se construir rapidamente um programa, os analistas partem de imediato para a programação do aplicativo, antes mesmo de que o problema esteja razoavelmente entendido. A adoção de um método de engenharia de software para conduzir o desenvolvimento, faz com que o analista seja estimulado a entender o problema e a criar documentos de projeto antes de iniciar a programação. Como os sistemas atuais são bastante complexos, há necessidade de se apoiar o método de desenvolvimento em uma ferramenta CASE. Esta abordagem, aparentemente distancia o programa executável das fases iniciais do projeto, e por isso é combatida pelos adeptos do RAD [5].

⁵ No desenvolvimento de software toda a ajuda é bem-vinda por José Eduardo Zindel Deboni

Muitos consideram que os projetos de software sofrem os males de uma tecnologia imatura. Não se começa a construir uma casa sem uma planta, sem um projeto aprovado pelo cliente, ainda que ela seja pré-fabricada. A engenharia de software não é diferente. Antes de se iniciar a construção de um programa deve-se projetá-lo, analisando os requisitos dos clientes, a complexidade do problema, e avaliando as possíveis soluções, que incluem o uso de ferramentas RAD [5].

No desenvolvimento da arquitetura SAD-DFM/IP, optou-se por usar ferramenta CASE para a modelagem da estrutura de dados utilizada e ferramenta RAD para o desenvolvimento do software aplicativo usado para o acesso aos dados estruturados pela ferramenta CASE.

2.2. Sistema de Gerenciamento de Banco de Dados

Um SGBD – Sistema de Gerenciamento de Banco de Dados é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades. Há, também, bancos de dados baseados em Gerenciadores de Base de Dados e Gerenciadores de Arquivos [6]⁶.

Bancos de dados baseados em gerenciadores de arquivos, como os padrões btrieve e dBase (Fox e

⁶ Apostila de Banco de Dados e SQL Autores: Prof. Jorge Surian
Prof. Luiz Nicochelli SGBD x GA

Clipper), podem no máximo simular as características típicas de um SGBD. Os fabricantes das linguagens Delphi (utiliza opcionalmente o padrão dBase) e o VB (que utiliza o Access), recomendam a utilização de SGBDs em conjunto com suas ferramentas de desenvolvimento, porém permitem acesso àqueles "Bancos de Dados" que possuem algumas características de SGBDs, mas são Gerenciadores de Arquivo.

Algumas regras básicas e claras devem ser definidas para um sistema de manipulação de dados ser considerado um SGBD. Fica implícito que se ao menos uma das características abaixo não estiver presente no nosso "candidato" a SGBD, este poderá ser um GA (Gerenciador de Arquivo) de altíssima qualidade, "quase" um SGBD, mas não um SGBD [4].

Regra 1: Auto-Contenção: Um SGBD não contém apenas os dados em si, mas armazena completamente toda a descrição dos dados, seus relacionamentos e formas de acesso. Normalmente esta regra é chamada de Meta-Base de Dados ou metadados.

Regra 2: Independência dos Dados: Quando as aplicações estiverem realmente imunes a mudanças na estrutura de armazenamento ou na estratégia de acesso aos dados, podemos dizer que esta regra foi atingida.

Regra 3: Abstração dos Dados: Em um SGBD real é fornecida ao usuário somente uma representação conceitual dos dados, o que não inclui maiores detalhes sobre sua forma de armazenamento real. O chamado Modelo de Dados é um tipo de abstração utilizada para fornecer esta representação conceitual.

Regra 4: Visões: Um SGBD deve permitir que cada usuário visualize os dados de forma diferente daquela existente previamente no Banco de Dados.

Regra 5: Transações: Um SGBD deve gerenciar completamente a integridade referencial definida em seu esquema, sem precisar em tempo algum, do auxílio do programa aplicativo. Desta forma, exige-se que o banco de dados tenha ao menos uma instrução que permita a gravação de uma série de modificações simultâneas e uma instrução capaz de cancelar uma série de modificações.

Regra 6: Acesso Automático: Em um GA uma situação típica é o chamado Dead-Lock, o abraço mortal. Se a responsabilidade de evitar esta ocorrência for responsabilidade da aplicação, você não está lidando com um SGBD.

Conclusão: Um SGBD deve obedecer INTEGRALMENTE às seis regras acima. Em caso contrário estaremos diante de um GA ou de um "quase" SGBD [6].

Considerando todas essas características dos SGBDs e de acordo com as características da arquitetura SAD-DFM/IP (que serão definidas no capítulo 4) recomenda-se que ao desenvolver um sistema baseado nesta arquitetura se utilize para a armazenagem e gerenciamento dos dados um SGBD real observando-se todas as regras definidas acima.

2.3. Sites Dinâmicos

São sites cujas páginas internas têm seu conteúdo gerado em tempo real. No servidor, a página apresenta comandos de uma linguagem de programação para sites dinâmicos. No navegador, obtém-se apenas código da linguagem de visualização HTML [7]⁷.

A arquitetura de sites dinâmicos é dividida em 4 camadas, figura 2.1 [7]:

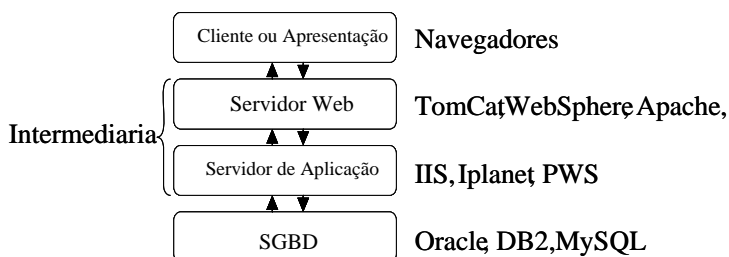


Figura 2.1 – Arquitetura de sites dinâmicos

A primeira, conhecida como Camada de Apresentação, pode ser dividida em cliente leve e cliente pesado:

Clientes Pesados são programas em Java ou Delphi, por exemplo, onde toda a lógica da apresentação e parte da lógica de montagem da informação está codificada no programa. Esses podem se comunicar diretamente com o servidor de aplicação ou SGBD.

Clientes Leves são navegadores (IE, Netscape, Opera) que recebem a codificação da interação (HTML)

⁷ Desenvolvimento de Sites Dinâmicos por Eônio Júnior
eoniojunior@hotmail.com

em tempo de execução. Pode-se usar JavaScript para implementar recursos da lógica de apresentação.

A seguir, tem-se a **Camada Intermediária**, que compreende dois elementos principais: o Servidor Web (**Camada Web**) e o Servidor de Aplicação (**Camada de Aplicação**).

O *Servidor Web* é responsável pelo recebimento das requisições HTTP (ex: POST, GET, etc.). Pode atender a uma requisição isoladamente (no caso de páginas estáticas ".html") ou encaminhar a requisição para um módulo ou servidor de aplicação (no caso de páginas dinâmicas ".asp, .jsp, .php"). Deve ser capaz de ter um bom desempenho (atendimento de várias requisições simultâneas), ser robusto e seguro. Como exemplos, podemos citar: Apache, IIS, IPlanet, IBM HTTP server, PWS.

O *Servidor de Aplicação* (SA) é responsável pelo recebimento das requisições HTTP (ex: POST, GET, etc) e pela execução da aplicação responsável pelo atendimento da requisição. Para cumprir seus objetivos o SA fornece serviços de construção dinâmica de páginas, processamento da lógica de negócios, acesso a dados, integração com aplicações externas, gerência de sessões, balanceamento de carga, recuperação de falhas. As funções do SA e do Servidor HTTP se confundem, sendo que em algumas arquiteturas os dois grupos de funções são realizados pelo mesmo software. Pode-se citar como exemplos: Jakarta Tomcat, WebSphere, Web Logic, Orion, PWS.

A última camada da arquitetura, o SGBD (**Camada de Gerenciamento de Dados**), cuida do armazenamento dos dados, que são manipulados via a interação com o servidor de aplicação.

O uso da arquitetura de sites dinâmicos permite que os sistemas desenvolvidos baseados na arquitetura SAD-DFM/IP possam sofrer manutenções e atualizações em um único servidor e essas alterações visualizadas por todos os clientes ao mesmo tempo sem a necessidade de atualizar cada cópia do sistema em cada cliente onde o sistema esteja presente.

2.4. Tecnologia de Comunicação Móvel

Atualmente, o mercado de Tecnologia de Comunicação Móvel conta com três principais tecnologias de acesso móvel, sendo:

Sistema de Mensagens Curtas SMS (Short Message System): este sistema não possui acesso a Web Móvel, mas é o mais popular usado no Brasil constando do envio e recebimento de mensagens de texto curtas em canais: unidirecional (o usuário é um destinatário passivo) e bidirecional (o usuário pode reenviar a mensagem recebida para o seu remetente ou para outro destinatário).

Dispositivos com acesso via WAP (Wireless Application Protocol), um padrão emergente para marcação e apresentação de conteúdo sem fio dirigido explicitamente a dispositivos de mão pequenos e móveis, como telefones celulares, pagers e terminais de acesso

sem fio [8]⁸. Esse padrão vem se popularizando no Brasil devido a sua tecnologia já estar consolidada em vários países da Europa e Estados Unidos.

Sistema Global para Comunicações Moveis GSM (Global System for Mobile Communications). Esse sistema suporta vários protocolos de comunicação móvel e começa a ser introduzido no Brasil. Um dos sistemas celulares digitais, GSM usa tecnologia de TDMA e permite oito chamadas simultâneas na mesma frequência de rádio[9]⁹.

A especificação do WAP usa os padrões de comunicação sem fios existentes e acrescenta novas extensões, permitindo que os participantes da indústria desenvolvam soluções como interfaces aéreas independentes dispositivos independentes e completamente interoperáveis [8].

A principal característica do padrão WAP é a utilização de vários protocolos já existentes na Internet, como TCP, UDP, XML, HTML, SSL, etc. [10]¹⁰.

O conteúdo e as aplicações WAP foram especificados através de um conjunto de formatos bem conhecido, baseado na Internet tradicional. O browser interpreta e exibe conteúdo desenvolvido para o ambiente

⁸ RISCHPATER, R. (2001) "Desenvolvendo Wireless para WEB: Como Enfrentar os Desafios dos Projetos para a Web Sem Fio", Makron Books, São Paulo.

⁹ Nokia Mobile Internet Toolkit Version 3.1 NMIT User's Guide June 2002.

¹⁰ WAP "Sample Configurations of WAP Technology" in WAP Architecture by <http://www.wapforum.org/what/copyright.htm>.

WAP. Esse conteúdo é criado através da linguagem WML (Wireless Markup Language ou Linguagem de Marcação Sem Fio). WML é semelhante a HTML que é usada na criação de páginas da Internet. Mas, ao contrário da linguagem HTML, a linguagem WML foi criada para atender às necessidades dos dispositivos e redes sem fio com baixo poder computacional. Para acrescentar recursos dinâmicos às aplicações WML, utiliza-se a linguagem WML Script, uma linguagem semelhante a JavaScript. Tanto WML como WML Script são adaptadas e otimizadas para utilização em ambiente sem fio [8].

Por essas características apresentadas, o protocolo WAP será usado através de sua linguagem de marcação WML para implementar a mobilidade dos sistemas desenvolvidos e considerando ainda que a maioria dos clientes Web móvel presentes no Brasil atualmente usam este protocolo para acesso móvel, o que possibilitara uma gama maior de usuários.

2.5. Processo de Desenvolvimento de Software

Dentre os processos de desenvolvimento de softwares existentes optou-se por utilizar o Rational Unified Process (RUP) pelas seguintes características:

A metodologia Rational Unified Process – Processo Unificado Rational é a inclusão das idéias e experiências de líderes da indústria, sócios, e literalmente milhares de projetos de software reais, cuidadosamente sintetizados em um conjunto prático das melhores

práticas, workflows, e artefatos para desenvolvimento interativo de software [11]¹¹.

O RUP está se tornando, rapidamente o padrão da indústria para desenvolvimento interativo; dos "5 grandes" integradores de sistemas globais, para 8 entre os 10 maiores bancos dos E.U.A., milhares de projetos de nível mundial depositam confiança no RUP [11].

Quando usado em combinação, as melhores práticas promovidas pela metodologia RUP – que incluem: Desenvolvimento Interativo, Administração de Exigências, Arquiteturas de Componentes de Uso, Modelagem Visual, Administração de Mudanças, e Continua Verificação de Qualidade – eliminando a causa de problemas no desenvolvimento de software na raiz, enquanto lhe ajudando a evitar armadilhas comuns como as presentes em ferramentas e tecnologias novas. Usando uma metodologia aprovada e compartilhando um único processo inclusivo, seu time poderá comunicar mais efetivamente e trabalhar mais eficazmente [11].

Por essas características recomenda-se que todo o processo de desenvolvimento de sistemas baseado na arquitetura SAD-DFM/IP siga esse modelo, pois por se tratar da integração de varias e diferentes tecnologias um acompanhamento mais próximo do processo de desenvolvimento por parte da equipe de projeto é altamente recomendado para diminuir a possibilidade de falhas de projeto em fases avançadas de desenvolvimento.

¹¹ www.rational.com/products/rup/index.jsp.

3. Metodologia

A metodologia utilizada neste trabalho propõe a utilização das tecnologias analisadas de forma que possam ser utilizadas em conjunto de forma que se complementem cada qual dentro de suas capacidades de utilização individual e em conjunto com outras, quando possível para tentar demonstrar essas capacidades iremos analisá-las de forma mais detalhada nos próximos tópicos.

3.1. Utilização de Ferramentas RAD e SGBD em Conjunto

A utilização de ferramentas RAD para acesso a SGBDs é um modelo de desenvolvimento em camadas conhecido como Multiple Tier: nesse modelo o driver processa as chamadas às funções e passa os comandos SQL para o SGBD ou para um gateway. Estes drivers podem ser "two tiered", "three tiered", etc. O driver é two tiered se ele envia a SQL diretamente para o SGBD. É three tiered se envia a SQL para um gateway que encaminha para o SGBD figura 3.1 [12]¹².

¹² Bate Byte 59 Novembro/99 - Visão Geral sobre Delphi Autor: Vidal Martins – GPT www.pr.gov.br/celepar/celepar/batebyte/

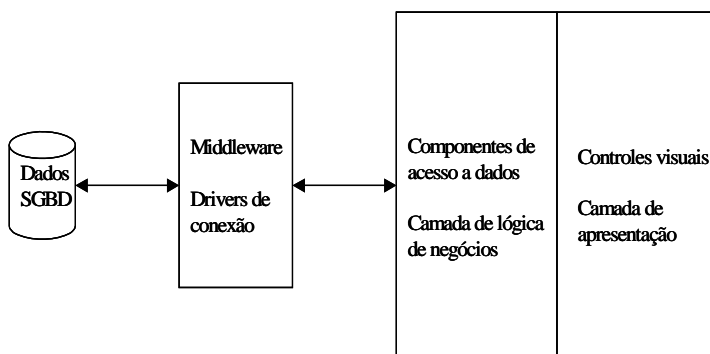


Figura 3.1 – Arquitetura Multiple Tier

Na primeira camada desse modelo encontra-se o IDE e todos os seus componentes. Na segunda camada, a de middleware, encontra-se todos os seus drivers de conexão. E na terceira camada encontram-se as fontes de dados que a ferramenta RAD é capaz de acessar, tanto em ambiente local quanto em ambiente remoto [12].

Em uma aplicação dessa categoria as camadas são: apresentação, regras de negócio, e persistência. Também deve saber que toda a lógica de apresentação fica no cliente, a camada de persistência é implementada através de um SGBD, e que as regras de negócio não têm um lugar bem definido; depende da infra-estrutura disponível [12].

Em aplicações pessoais e de suporte à decisão, recomenda-se que as regras de negócio sejam colocadas no cliente (fat-client). Já para as aplicações missão-crítica sugerem-se regras de negócio no servidor. Se a aplicação não é tão pesada assim (TP-Lite), que pode ser implantada através de uma arquitetura two-tier, a palavra servidor refere-se ao servidor de banco de dados. Porém,

se a aplicação é muito crítica (TP-Heavy), a palavra servidor refere-se a um servidor de aplicações que utiliza um monitor de transações para otimizar o processo [12].

A utilização de ferramentas RAD e SGDB possibilita um desenvolvimento rápido de sistemas de acesso a dados baseados na arquitetura SAD-DFM/IP através do desenvolvimento de aplicações TP-Lite como definido acima sendo recomendado o desenvolvimento de aplicações TP-Heavy para a administração remota do SGDB devido a características intrínsecas da arquitetura, como vários e diferentes clientes on-line simultaneamente.

3.2. Utilização das Tecnologias de Sites dinâmicos e Computação Móvel em Conjunto

Para alcançar esse objetivo usaremos uma linguagem script que possibilita a geração de páginas dinâmicas no lado servidor através do servidor back-end tanto para computadores do tipo desktop quanto páginas com qualidade compatíveis com computadores de mão (pdqSuite, ProxiNet, AvantGO, etc), assim como páginas para celulares que acessam a Web usando o protocolo WAP para ter acesso a páginas WML - que serão geradas pelo mecanismo JSP [8].

A utilização, de mecanismos de sites dinâmicos e computação móvel são encorajadas por algumas características dos dispositivos moveis tais como:

- **Interação Cliente-Servidor:** a forma como um navegador e um servidor Web interagem possibilita que tecnologias como WAP podem ser

usadas com um servidor Web existente para atender a solicitações de clientes distintos.

➤HTML (HyperText Markup Language): a utilização de HTML especialmente adaptado possibilita a visualização de páginas por navegadores de dispositivos sem fio, como computadores de mão (pdqSuite, ProxiNet, AvantGO, etc).

➤CGI (Common Gateway Interface): os sites Web para acesso sem fio têm maiores chances de ser dinâmicos do que sites convencionais

As integrações dessas tecnologias suprem as expectativas da arquitetura SAD-DFM/T P no requisito de suprir um meio de integrar dispositivos fixos e moveis utilizando um ponto de convergência comum o servidor back-end.

3.3. Utilização das Tecnologias de Sites dinâmicos e SGBD em Conjunto

A utilização de sites dinâmicos para acesso a SGBDs é um modelo de desenvolvimento em camadas Multiple Tier.

Na primeira camada desse modelo encontra-se o Servidor Web e todos os seus componentes. Na segunda camada, a de middleware, encontra-se o Servidor de Aplicação e todos os seus drivers de conexão. E na terceira camada encontram-se as fontes de dados que o Servidor de Aplicação é capaz de acessar, tanto em ambiente local quanto em ambiente remoto.

3.4. Utilização das Tecnologias de Site dinâmicos e Ferramentas RAD em Conjunto

A utilização do desenvolvimento em camadas presentes nas duas tecnologias possibilita uma base de desenvolvimento comum, convergindo sobre tudo na camada de acesso a dados comum entre elas.

Desconsiderando as diferenças estruturais básicas entre essas duas tecnologias e considerando que a forma de acesso a dados nas duas se assemelha. É possível o desenvolvimento de sistemas independentes que se comuniquem através dessa camada, como por exemplo, o desenvolvimento de uma versão aplicativo através de uma ferramenta RAD para uso administrativo/gerencial de nível servidor. E uma versão no formato de páginas Web dinâmicas para uso de inclusão e análise estatística de nível cliente.

3.5. Integração das Tecnologias Analisadas

A integração de Sites Dinâmicos e SGBDs possibilita a manipulação dos dados contidos no SGBD pelas páginas dinâmicas geradas pelos servidores de aplicação inclusive às páginas geradas em WML para dispositivos Wireless possibilitando, assim que aplicativos gerados por ferramentas RAD, páginas geradas por mecanismos de Sites Dinâmicos e Dispositivos de Computação Móvel utilizem um canal comum de acesso ao SGBD.

Com o uso de SGBD será possível que um dado gerado por software desenvolvido através de uma ferramenta RAD seja acessado por um Dispositivo de Computação Móvel através de um Servidor de Aplicação

que tenha gerado um Site Dinâmico que possua acesso ao SGBD.

4. Elaboração da Arquitetura

Este trabalho propõe uma Arquitetura para Sistemas de Acesso a Dados Através de Dispositivos Fixos e Moveis Independentes de Plataforma, conforme mostrado na figura 4.1.

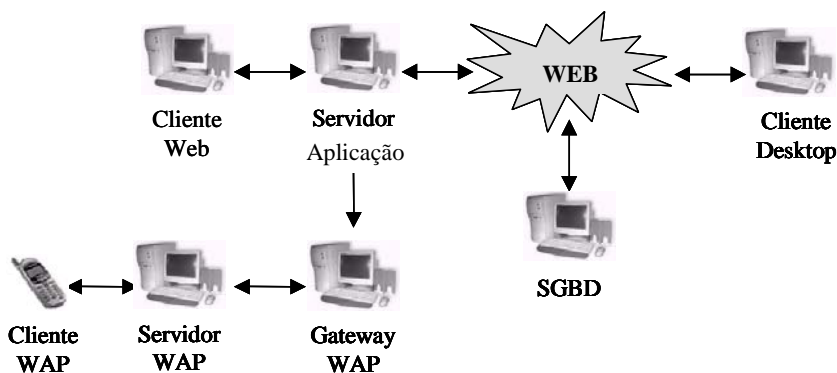


Figura 4.1 – Arquitetura Proposta

Ao analisarmos as tecnologias utilizadas individualmente e em conjunto nos foi revelado que, a Camada de Gerenciamento de Dados pode ser utilizada para unir tanto os softwares gerados por ferramentas RAD, na forma de Clientes Pesados, quanto paginas Web geradas de forma dinâmica por Clientes Leves. Sendo que estes possibilitam a geração de paginas tanto para usuários de computadores ligados a Web quanto usuários de outros dispositivos ligados a Web sem fio, ou seja, grande parte da dependência de plataforma está na interface utilizada pelo usuário final do sistema e na forma como são utilizadas algumas plataformas de desenvolvimento por alguns desenvolvedores.

Para que o usuário possa utilizar qualquer forma de interface de software ou hardware que lhe seja conveniente em determinado momento, é necessário que essa interface consiga ter acesso a Camada de Gerenciamento de Dados.

Para uma melhor utilização dessa Arquitetura é necessário subdividir o Sistema em duas categorias de acesso: acesso em Modo Desktop (ou Workstation, ou Localhost) ou acesso em Modo Web, esse último pode ser subdividido em Modo Web Convencional e Modo Web Móvel.

Os modos de acesso propostos pela arquitetura podem ser observados no diagrama de casos de uso da figura 4.2, e a especificação de cada caso de uso será feita nos referidos tópicos em sequência.

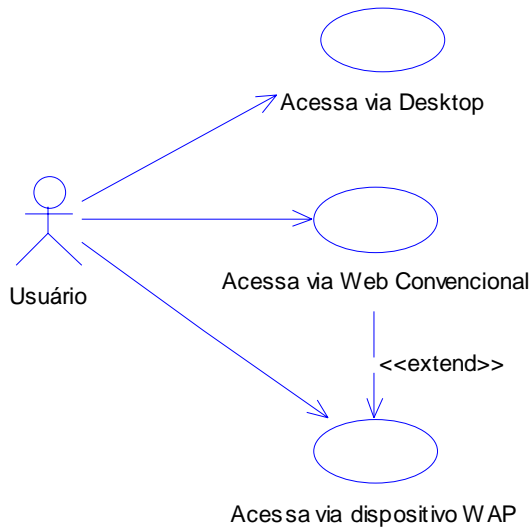


Figura 4.2 – Casos de uso da arquitetura proposta

4.1. Modo Desktop

O usuário em Modo Desktop poderá utilizar o Sistema através de um Cliente Pesado (software aplicativo) modelado a partir de uma ferramenta CASE e implementado através de uma ferramenta RAD. Esse Cliente Pesado terá acesso a Camada de Gerenciamento de Dados de forma direta, ou seja, utilizando um SGBD instalado na própria estação de trabalho – ou em um servidor de banco de dados instalado na intranet local ou disponível na Web – utilizando para isto uma conexão direta ao SGBD.

A utilização desse modo de acesso poderá incorrer em falha de segurança caso seja utilizado por terceiros, pois possibilita o acesso total ao SGBD.

No entanto, é recomendado em situações de controle por parte da gerencia do sistema, em caso de sistemas isolados e onde o acesso seja restrito a um numero mínimo de usuários.

As pré-condições o fluxo de eventos e as pós-condições para o acesso em Modo Desktop podem ser aferidos na Tabela 4.1.

Tabela 4.1 – Eventos gerados pelo caso de uso Modo Desktop.

<i>Acessa via Desktop</i>
Atores: Usuário
Pré-Condições: O usuário acessa ao sistema de forma direta através de software projetado por ferramenta Case e implementado utilizando ferramenta Rad
Fluxo de eventos: <i>Principal</i> 1 - O usuário possui acesso privilegiado ao sistema (como administrador, por exemplo). 2 - Ao acessar o sistema possui privilégios especiais de alteração de dados e consultas privilegiadas 3 - O usuário usa esses privilégios para administrar o sistema <i>Alternativo</i> 1 - O usuário não possui acesso ao sistema 2 - O usuário loga como usuário remoto fornecendo identificação e senha para acesso ao sistema 3 - Logado ao sistema o usuário passa a ter privilégios administrativos

Pós-Condições: o usuário possui todas as informações contidas no sistema e as administra

Um exemplo de algumas das atividades exercidas pelo caso de uso Modo Desktop pode ser observado na figura 4.2.

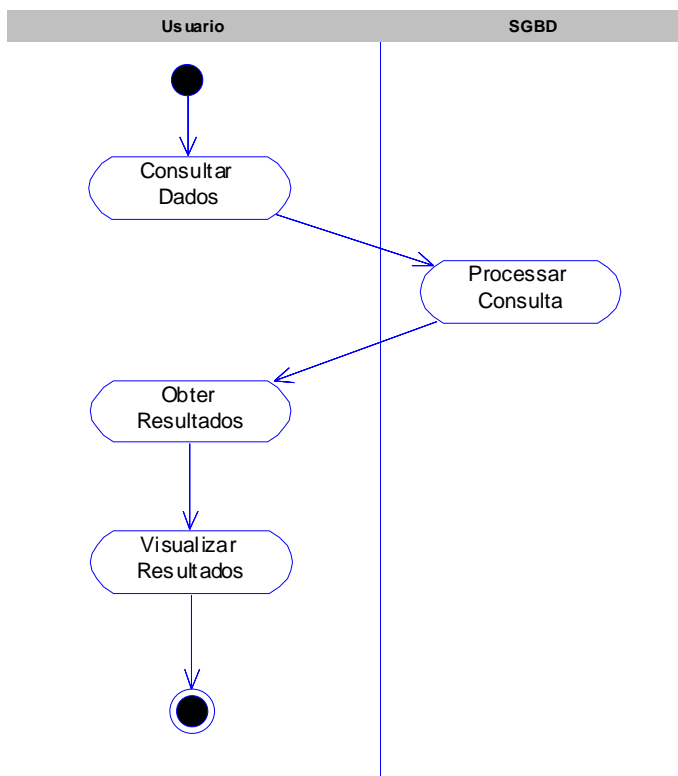


Figura 4.3 – Atividades relacionadas ao caso de uso Modo Desktop.

Durante a utilização do caso de uso analisado o mesmo poderá apresentar os estados indicados na figura 4.3.

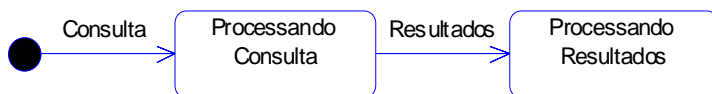


Figura 4.4 – Seqüência de estados possíveis para o caso de uso Modo Desktop.

4.2. Modo Web

O usuário em Modo Web poderá utilizar o Sistema através de dois outros modos: Modo Web Convencional ou Modo Web Móvel. A escolha de um desses modos dependerá, basicamente, do tipo de dispositivo que o usuário estiver usando no momento do acesso ao sistema.

Independente do Modo Web (Convencional ou Móvel) usado pelo usuário do sistema a comunicação com a Camada de Gerenciamento de Dados será feita com o uso de uma Camada de Aplicação (Servidor de Aplicação) que irá gerar conteúdo dinâmico a partir de interações com a Camada de Gerenciamento de Dados.

4.2.1. Modo Web Convencional

O usuário em Modo Web Convencional poderá utilizar o sistema através de um Cliente Leve que acessara a Camada Web (Servidor HTTP).

O usuário utilizando um Cliente Leve fará requisições a Camada de Gerenciamento de Dados, sendo que essas requisições serão tratadas pela Camada de

Aplicação e o resultado obtido por essas requisições será exibido ao usuário através da Camada Web acessada pelo Cliente Leve utilizado pelo usuário.

As pré-condições o fluxo de eventos e as pós-condições para o acesso em Modo Web Convencional podem ser aferidos na Tabela 4.2.

Tabela 4.2 – Eventos gerados pelo caso de uso Modo Web Convencional.

<i>Acesso via Web Convencional</i>
Atores: Usuário
Pré-Condições: O usuário acessa ao sistema de forma indireta através de um navegador Web convencional (como os existentes no mercado, Internet Explorer, Netscape, etc) para ter acesso a um servidor de aplicação onde o sistema está instalado.
Fluxo de eventos: <i>Principal</i> 1 - O usuário acessa ao sistema via navegador 2 - O sistema identifica a origem do usuário e o tipo de navegador utilizado e gera uma pagina Web de acordo com essas características 3 - O usuario interage com o sistema de acordo com as paginas geradas para o seu perfil <i>Alternativo</i> 1 - O usuário utiliza o sistema para obter informações sobre os dados armazenados, tais como estatísticas e gráficos dos dados. 2 - Em implementações específicas do sistema o usuário poderá administrar o sistema via Web, sendo não recomendado por questões de segurança.
Pós-Condições: o usuário adiciona dados ao sistema e analisa estatísticas desses dados

Algumas das seqüências de atividades desenvolvidas pelos processos envolvidos no caso de uso Modo Web Convencional podem ser observadas na figura 4.4.

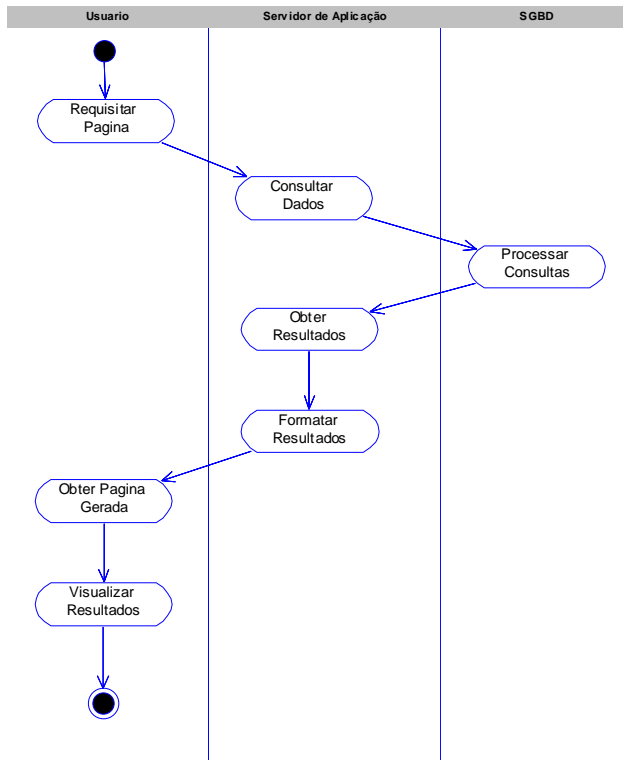


Figura 4.5 – Atividades relacionadas ao caso de uso Modo Web Convencional.

Durante a utilização do caso de uso analisado o mesmo poderá apresentar alguns estados como os estados indicados na figura 4.5.

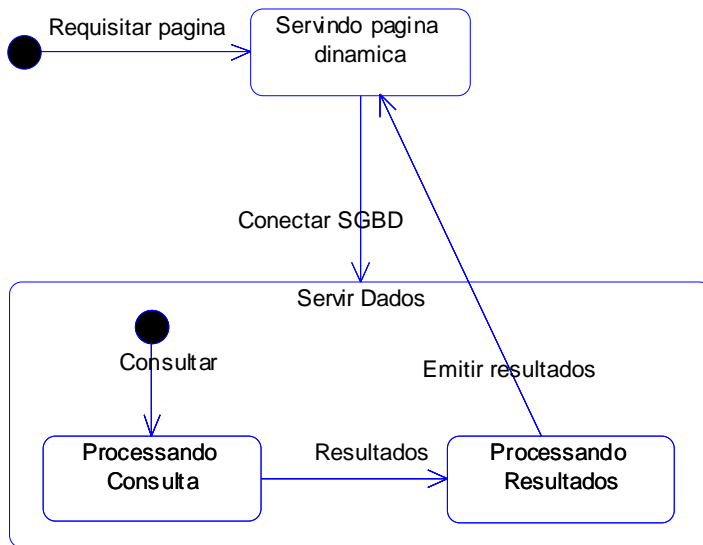


Figura 4.6 – Sequência de estados possíveis para o caso de uso Modo Web Convencional.

4.2.2. Modo Web Móvel

O usuário para ter acesso ao Modo Web Móvel irá necessitar de um dispositivo que permita o acesso a Web Móvel, normalmente um celular, hand held, ou outro tipo de computador portátil.

Para o dispositivo móvel usado pelo usuário acessar o Modo Web Móvel será necessário o uso de um Cliente Leve especialmente projetado para Web Móvel.

O usuário utilizando esse Cliente Leve fará requisições a Camada de Gerenciamento de Dados, sendo que essas requisições serão recebidas por uma Camada

de Acesso Móvel (Servidor WAP) e transmitidas a uma Camada de Modulação Móvel Fixo e Demodulação Fixo Móvel (Gateway WAP) para que essas requisições sejam transmitidas via Web Convencional para serem tratadas pela Camada de Aplicação e o resultado obtido por essas requisições será exibido através da Camada Web acessada pela Camada de Modulação Móvel Fixo e Demodulação Fixo Móvel e transmitida ao Cliente Leve utilizado pelo usuário pela Camada de Acesso Móvel.

As pré-condições o fluxo de eventos e as pós-condições para o acesso em Modo Web Móvel podem ser aferidos na Tabela 4.3.

Tabela 4.3 – Eventos gerados pelo caso de uso Modo Web Móvel.

<i>Acesso via Web Móvel</i>
Atores: Usuário
Pré-Condições: O usuário acessa ao sistema de forma indireta através de um dispositivo móvel que implementa alguma forma de acesso a Web
Fluxo de eventos: <i>Principal</i> 1 - Ao solicitar o acesso a Web o provedor de acesso a Web móvel o direciona a um servidor de aplicação através de gateway wap 2 - Durante a requisição os dados do usuário modulados do protocolo wap para o http e durante as respostas demulados de http para wap 3 - Estabelecida à conexão o sistema identifica a origem do usuário e o tipo de navegador utilizado e gera uma pagina Web de acordo com essas características 4 - O usuário interage com o sistema de acordo com essas características

Alternativo

1 - O usuário utiliza o sistema em situações onde, por alguma razão, não possua um meio fixo para o acesso ao sistema.

2 - O usuário pode utilizar ainda o sistema para obter algumas estatísticas simples e emergenciais do sistema

Pós-Condições: o usuário adiciona dados ao sistema e analisa estatística simples desses dados

Algumas das seqüências de atividades envolvidas nos processos internos ao caso de uso Modo Web Móvel podem ser visualizadas na figura 4.6

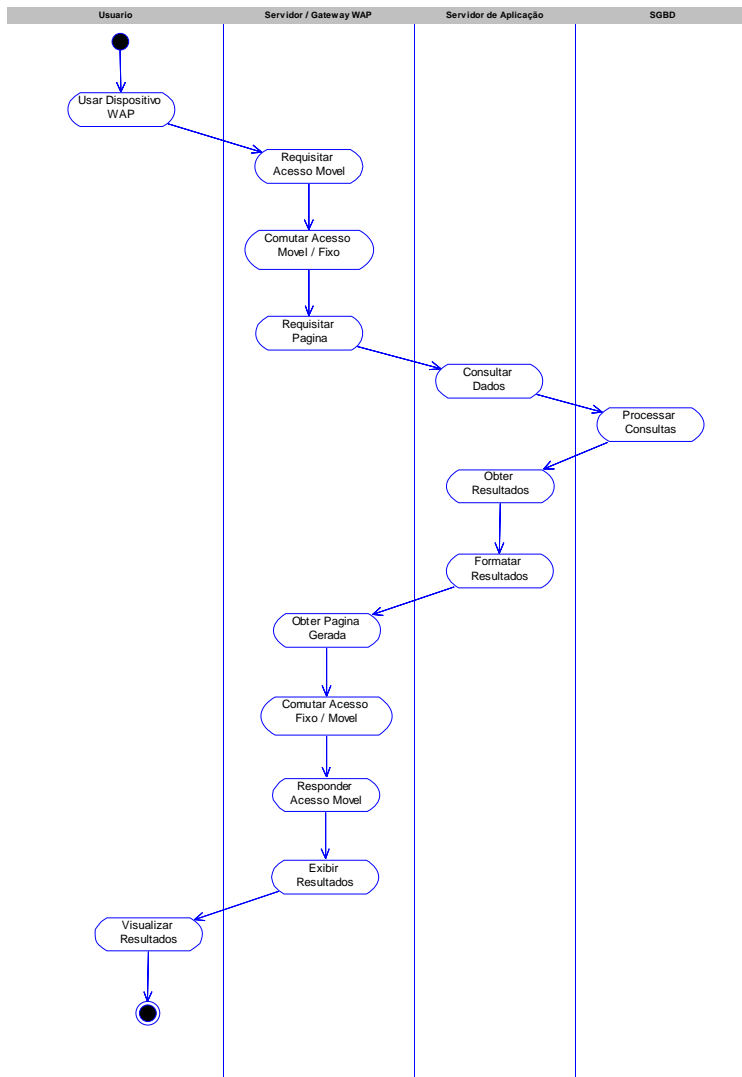


Figura 4.7 – Atividades relacionadas ao caso de uso Modo Web Móvel.

Durante a utilização do caso de uso analisado o mesmo poderá apresentar os estados indicados na figura 4.7.

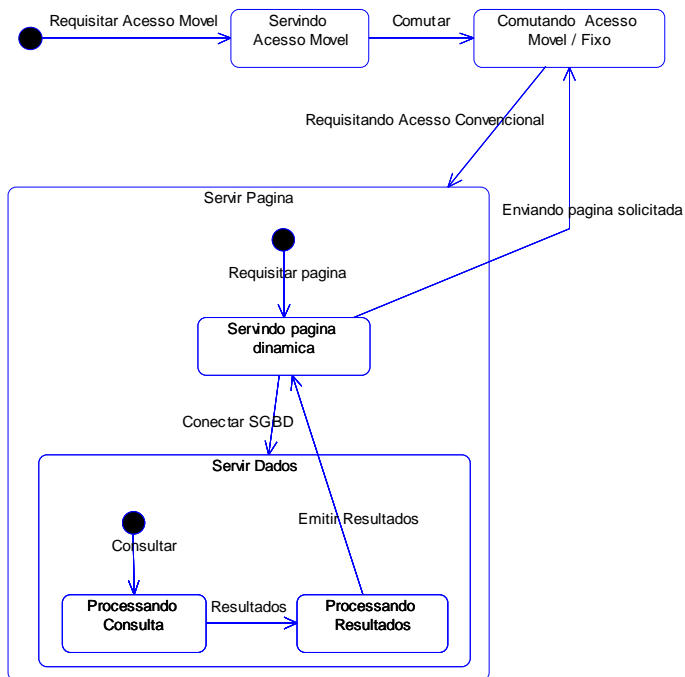


Figura 4.8 – Sequência de estados possíveis para o caso de uso Modo Web Móvel.

5. Estudo de Caso

A implementação de um estudo de caso visa testar a aplicabilidade e viabilidade da Arquitetura que está sendo proposta neste trabalho. Para isto o sistema produzido usou como base de desenvolvimento a Arquitetura proposta, como modelagem de dados uma ferramenta Case, e como processo de desenvolvimento de software o Processo Unificado Rational RUP (Rational Unified Process).

A elaboração, desse estudo de caso específico, procurou utilizar tecnologias independentes de plataforma e – sempre que possível – tecnologias gratuitas para diminuir o custo de desenvolvimento do sistema.

Os componentes utilizados neste estudo de caso podem ser observados no diagrama da figura 5.1.

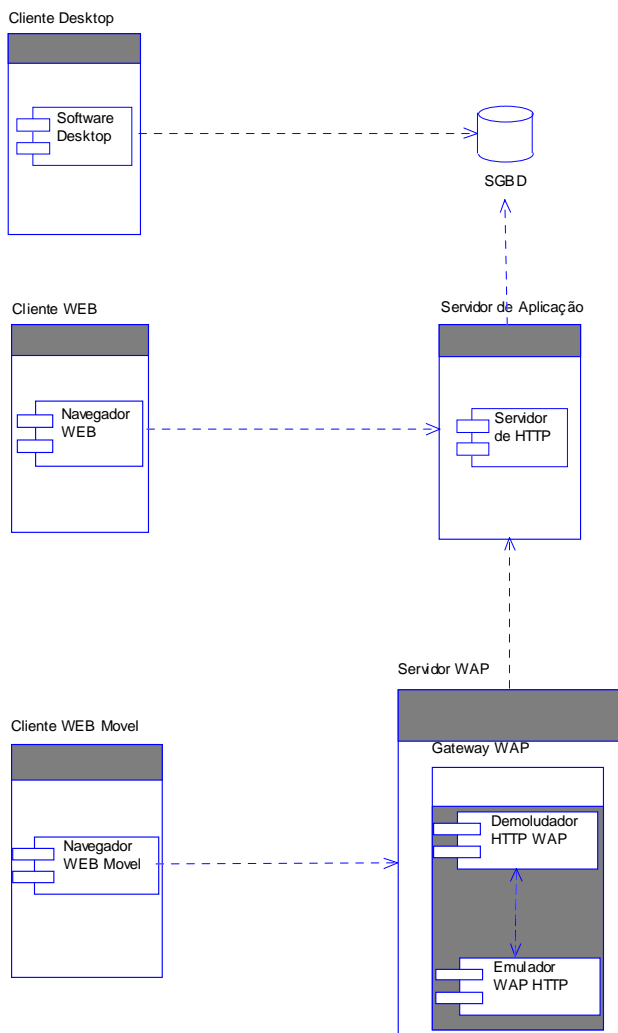


Figura 5.1 – Componentes utilizados na implementação do estudo de caso.

5.1. Tecnologias Utilizadas para o Estudo de Caso

5.1.1. Ferramenta RAD

Para possibilitar uma maior usabilidade do sistema desenvolvido, optou-se por usar a ferramenta RAD Borland Kylix 3 Open Edition. Esta ferramenta possibilita o desenvolvimento de sistemas compatíveis com pelo menos duas plataformas operacionais (Linux e Windows) sendo necessário apenas que se compile o sistema na plataforma onde ele for utilizado.

Após a construção do sistema citado ele foi recompilado na plataforma Windows utilizando-se uma versão trial do Borland Delphi.

Para possibilitar o acesso a dados pelo sistema foi utilizado componentes de acesso a dados compatíveis como SGBD MySQL disponíveis na Web na forma de componentes freeware.

Algumas telas representado as fases de: desenvolvimento, recompilação e execução podem ser visualizadas nas figuras a seguir.

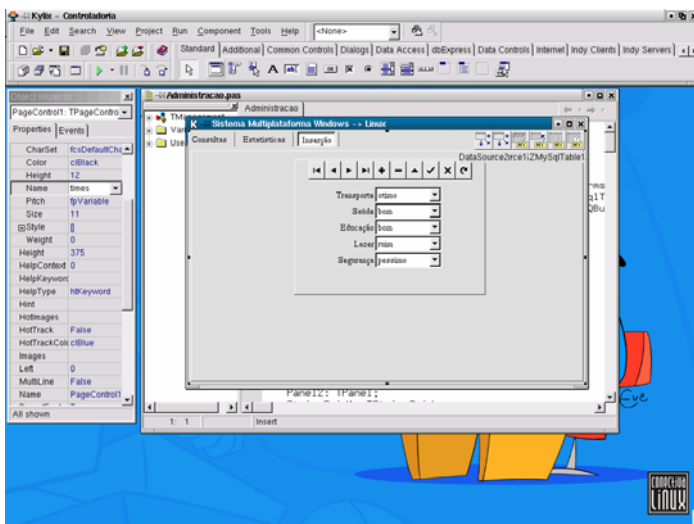


Figura 5.2 – Construção do Sistema na Plataforma Linux utilizando Borland Kylix

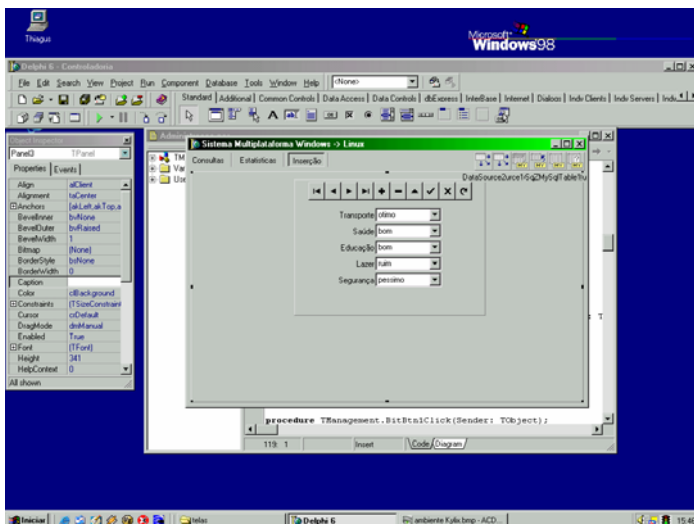


Figura 5.3 – Recopilação do Sistema na Plataforma Windows utilizando o Borland Delphi

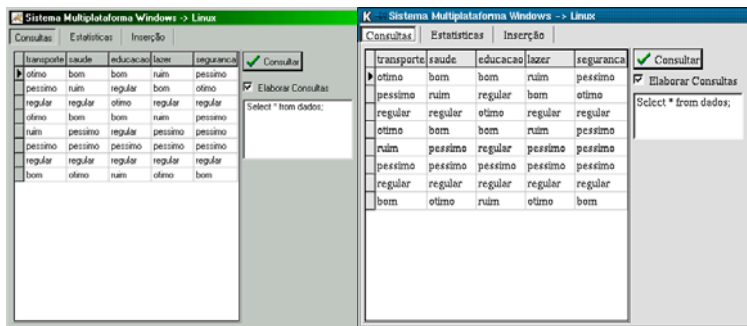


Figura 5.4 – Utilização do sistema nas Plataformas Windows (a esquerda) e Linux (a direita)

5.1.2. Sistema de Gerenciamento de Banco de Dados

Dentro da proposta de desenvolvimento de se utilizar tecnologias gratuitas disponíveis na Web e de uma maior independência de plataforma operacional optou-se por utilizar como SGBD o MySQL por este possuir as seguintes características:

O MySQL é um SGBD (Sistema de Gerenciamento de Banco de Dados) relacional com código fonte aberto. Foi originalmente desenvolvido para manipular bases de dados muito grandes, com maior rapidez que as soluções existentes, e tem sido utilizado com sucesso em ambientes de produção com alta demanda. Apesar de estar em constante desenvolvimento, o MySQL oferece um rico e muito útil conjunto de funções. Conectividade, velocidade e segurança tornam o MySQL altamente apropriado para acesso a bancos de dados na Web [13]¹³.

¹³ www.mysql.com/documentation em maio de 2001

O MySQL é um sistema cliente/servidor que consiste de um servidor SQL multi-threaded que suporta diferentes backends, vários programas clientes e bibliotecas, ferramentas administrativas e uma interface de programação [13].

As próximas figuras representam a modelagem conceitual e física de dados usada na elaboração do caso de uso

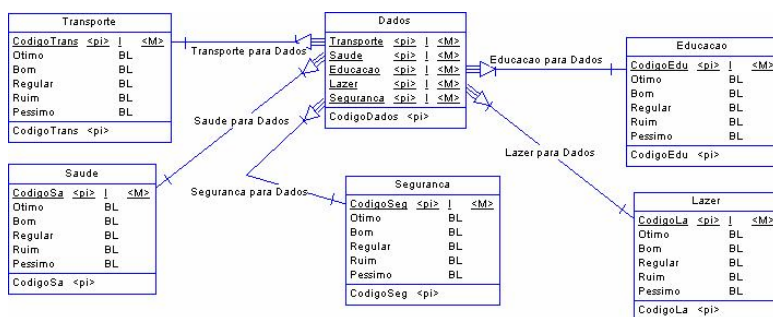


Figura 5.5 – Modelagem Conceitual dos Dados

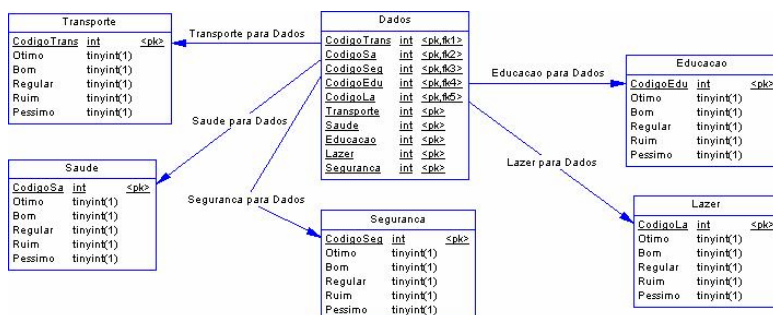


Figura 5.6. – Modelagem Física dos Dados

5.1.3 Sites Dinâmicos

Para a geração de páginas Web de forma dinâmica foi utilizada a plataforma de desenvolvimento J2EE (Java Enterprise Edition) através de sua extensão para a Web JSP (Java Server Pages) os motivos que levaram para a utilização dessa tecnologia em específico estão descritos abaixo.

JSP (Java Server Pages) é uma tecnologia para desenvolvimento de aplicações WEB semelhante ao Microsoft Active Server Pages (ASP), porém tem a vantagem da portabilidade de plataforma podendo ser executado em outros Sistemas Operacionais além dos da Microsoft. Ela permite ao desenvolvedor de sites produzir aplicações que permitam o acesso à banco de dados, o acesso a arquivos-texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e loops entre outras coisas [14]¹⁴.

Por definição, JSP usa Java como sua linguagem de scripts. Por esse motivo, O JSP se apresenta mais flexível e mais robusto do que outras plataformas baseadas simplesmente em JavaScripts e VBScripts [14].

As Java Server Pages combinam HTML ou XML com partes de código Java para produzirem páginas Web dinâmicas. Cada página é compilada automaticamente em uma Servlet pelo mecanismo JSP na primeira vez que ela é solicitada e, a seguir, é executada. As JSP fornecem

¹⁴ JSPBrasil - Tutorial JSP O que é JSP por: Ulisses Telemaco Neto
ulisses@jspbrasil.com.br

diversas maneiras de se dialogar com classes, servlets e applets Java, bem como com o servidor Web [15]¹⁵.

As próximas figuras apresentam a utilização do sistema desenvolvido em duas plataformas distintas

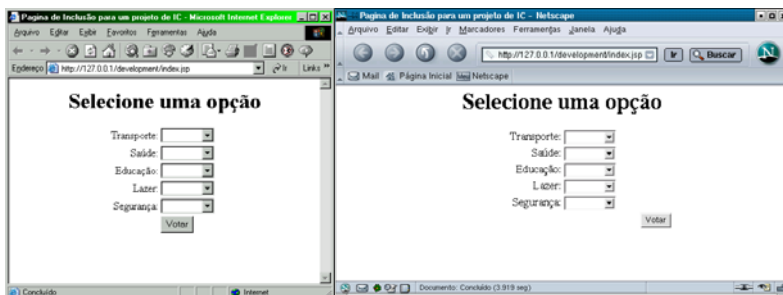


Figura 5.7 – Tela inicial do sistema nas Plataformas Windows(a esquerda) e Linux (a direita)

¹⁵ MUKHI, V. and MUKHI, S. and KOTOCHA, N. (2002) “Java Servlets JSP”, Makron Books, São Paulo.

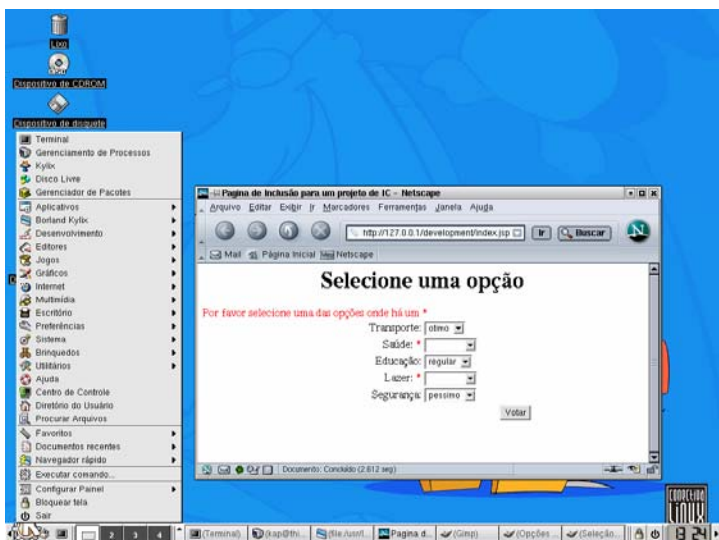


Figura 5.8 – Tela representando uma interação com o usuário em Linux

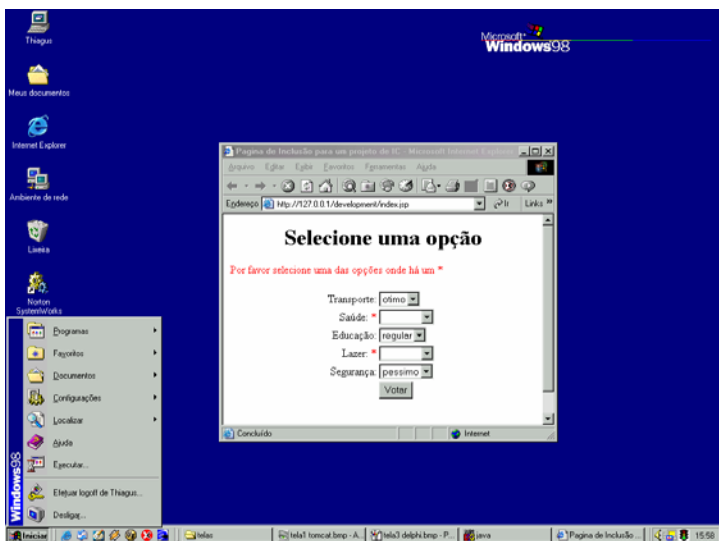


Figura 5.9 – Tela representando uma interação com o usuário em Windows

5.1.4. Tecnologia de Comunicação Móvel

Para uma maior mobilidade e para que uma gama maior de usuários possam ter acesso móvel ao sistema o mecanismo de sites dinâmicos possibilita a geração de paginas dinâmicas especialmente projetadas para o acesso móvel e no caso de celulares as paginas geradas são na linguagem WML.

As próximas figuras demonstram o mesmo sistema gerado de forma dinâmica no formato WML para o acesso via celular.

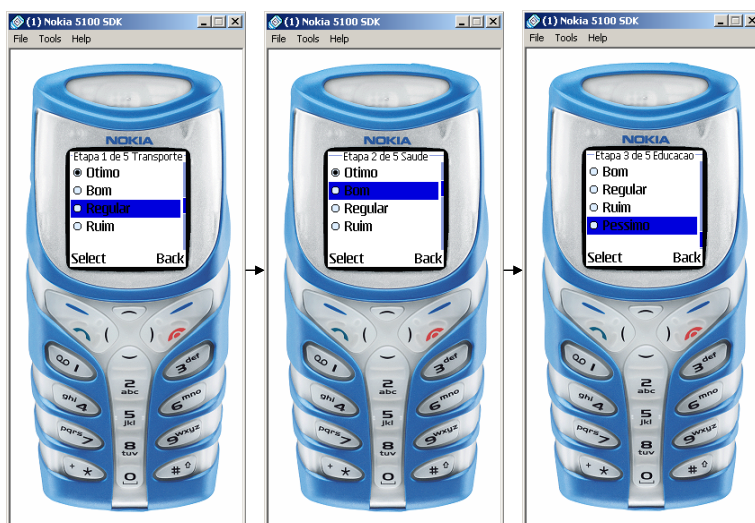




Figura 5.10 – Etapas necessárias para a votação via Modo Web Móvel

6. Conclusão

Este trabalho abre uma nova perspectiva para o desenvolvimento de software tornando possível que diferentes plataformas operacionais e diferentes dispositivos possam compartilhar informações antes presas a sistemas fechados.

Usando-se dessa arquitetura definida, neste trabalho, possibilita-se o desenvolvimento de sistemas que sejam uma espécie de Esperanto utilizando-se várias “línguas” é possível construir um sistema que seja compreendido por todos, plataformas operacionais e de hardware.

Atingido o objetivo proposto por esse trabalho criam-se novos desafios desenvolver uma arquitetura que além de possuir todas as características definidas nesta acrescentar a ela meios para que todas as pessoas tenham acesso aos sistemas desenvolvidos a partir dela.

O primeiro passo já foi dado o estudo de caso desenvolvido a partir da arquitetura utilizou-se exclusivamente de tecnologias e ferramentas de software disponíveis livremente na Web com isso combate a pirataria de software e torna o software acessível a populações de baixa renda já que o único custo do software é o do tempo gasto para desenvolvê-lo.

Os próximos passos incluem desenvolver uma arquitetura que possibilite o desenvolvimento de softwares cujo uso seja independente de sentido, ou seja, um software que seja facilmente usável por pessoas que

possuam algum tipo de privação de algum sentido como cegueira, mudes, surdes, tetraplégia, paraplegia, etc.

Conseguimos um meio de desenvolvimento onde todas as máquinas conseguem ter acesso agora devemos conseguir desenvolver um meio onde todas as pessoas consigam ter acesso ao sistema gerado por esse meio.

7. Bibliografia

- [1] Uma Metodologia de Projeto Orientado a Objetos Com Vistas à Reutilização Fabrício André Rubin Prof. Dr. Carlos Alberto Heuser (orientador)
- [2] Ambientes Integrados por Fernando Lozano - fernando@lozano.eti.br para a revista do linux
- [3] Apostila: Ferramentas Case - Trabalho desenvolvido pelos alunos de Tecnologia em Informática da UNIVALE em Outubro/1996.
- [4] Manual Dr. Case Autor: Squadra tecnologia em software LTDA www.squadra.com
- [5] No desenvolvimento de software toda a ajuda é bem-vinda por José Eduardo Zindel Deboni
- [6] Apostila de Banco de Dados e SQL Autores: Prof. Jorge Surian Prof. Luiz Nicochelli SGBD x GA
- [7] Desenvolvimento de Sites Dinâmicos por Eônio Júnior eoniojunior@hotmail.com
- [8] RISCHPATER, R. (2001) "Desenvolvendo Wireless para WEB: Como Enfrentar os Desafios dos Projetos para a Web Sem Fio", Makron Books, São Paulo.
- [9] Nokia Mobile Internet Toolkit Version 3.1 NMIT User's Guide June 2002
- [10] WAP "Sample Configurations of WAP Technology" in WAP Architecture by <http://www.wapforum.org/what/copyright.htm>
- [11] www.rational.com/products/rup/index.jsp
- [12] Bate Byte 59 Novembro/99 - Visão Geral sobre Delphi Autor: Vidal Martins – GPT www.pr.gov.br/celepar/celepar/batebyte/
- [13] www.mysql.com/documentation em maio de 2001

- [14] JSPBrasil - Tutorial JSP O que é JSP por: Ulisses Telemaco Neto ulisses@jspbrasil.com.br
- [15] MUKHI, V. and MUKHI, S. and KOTOCHA, N. (2002) “Java Servlets JSP”, Makron Books, São Paulo.

Anexo A – Tutoriais Criados Para o Desenvolvimento da Arquitetura

1. Servidor de Banco de dados MySQL:

1. 1. O que é o MySQL?

Um dos mais rápidos programas para servidores de SQL (do inglês, "Linguagem de pesquisa simples"), hoje no mercado, é o MySQL, desenvolvido pela T.c.X. DataKonsultAB. Este programa está disponível para download em sua versão original em www.mysql.com, em inglês, e também em www.mysql.com.br para sua versão brasileira, onde encontram-se projetos de tradução e documentação do MySQL em português.

Além de oferecer vários recursos não existentes em outros servidores, o MySQL tem a vantagem de ser totalmente gratuito para uso tanto comercial, quanto privado, em conformidade com a licença pública GPL.

As principais metas da equipe de desenvolvimento do MySQL é construir um servidor rápido e robusto.

Os recursos acima mencionados incluem:

- Capacidade de lidar com um número ilimitado de usuários;
- Capacidade de manipular mais de cinquenta milhões (50.000.000) de registros;
- Execução muito rápida de comandos, provavelmente o mais rápido do mercado;
- Sistema de segurança simples e funcional.

Quem usa o MySQL:

- Silicon Graphics (www.sgi.com)
- Siemens (www.siemens.com)

- Yahoo (www.yahoo.com)
- IFX Networks (www.ifx.com.br)
- Dezenas de Web hosting e Provedores devido ao enorme sucesso que o MySQL vem fazendo

Existem sistemas rodando servidor MySQL com bases de dados de 200 Gigabytes! Uma lista completa de usuários do MySQL no site oficial - www.mysql.com

1. 1. 1. Principais características do MySQL:

- Totalmente multi-threaded usando threads do kernel. O que significa que pode facilmente fazer uso de múltiplas CPUs se disponíveis;
- Suporta APIs de C, C++, Eiffel, Java, Perl, PHP, Python e Tcl;
- Roda em praticamente todos os sistemas operacionais atuais;
- Suporte a vários tipos de dados: inteiros com ou sem sinal de 1, 2, 3, 4, e 8 bytes e os tipos FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET e ENUM;
- Junções rápidas utilizando one-sweep multi-join;
- Suporte completo a operadores e funções nas cláusulas SELECT e WHERE das consultas. Ex.:
SELECT CONCAT(first_name, " ", last_name)
FROM tbl_name WHERE income/dependents > 10000 AND age > 30;
- As funções SQL são implementadas através de uma biblioteca de classes altamente otimizadas e são tão rápidas quanto podem ser;
- Suporte completo para as cláusulas GROUP BY e ORDER BY. Suporta para funções de grupo

(COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() e MIN());

- Suporta LEFT OUTER JOIN e RIGHT OUTER JOIN com sintaxe SQL ANSI e ODBC;
- Suporte a consultas com tabelas de diferentes bases de dados;
- Um sistema de privilégios e senhas flexível e seguro onde todo tráfego de senhas é encriptado quanto da conexão com o servidor;
- Suporte para ODBC (Open Data Base Connectivity) para Win32. Todas as funções do ODBC 2.5 e muitas outras;
- Permite até 32 índices por tabela. Cada índice pode ser formado por 1 a 16 colunas ou partes de colunas. O tamanho máximo dos índices é de 500 bytes (isso pode ser alterado no momento da compilação do MySQL);
- Registros de tamanho fixo ou variável;
- Tabelas hash em memória que são usadas como tabelas temporárias;
- Manipula grandes bases de dados. Existem bases de usuários em MySQL com 60.000 tabelas e cerca de 5.000.000.000 de registros;
- Escrito em C e C++. Testado com uma ampla gama de diferentes compiladores;
- Um sistema rápido de alocação de memória baseado em threads;
- Suporte completo a vários character sets, incluindo ISO-8859-1 (Latin1), big5, ujis, e mais;
- Todos os dados são salvos no character set escolhido. Todas as comparações de colunas string normais são case insensitive;

- A ordenação é feita de acordo com o character set escolhido. É possível alterá-lo quando o servidor MySQL é iniciado;
- São permitidos aliases em tabelas e colunas conforme o padrão SQL92;
- DELETE, INSERT, REPLACE e UPDATE retornam o número de linhas afetadas;
- Nomes de funções não colidem com nomes de tabelas ou colunas. Por exemplo, ABS é um nome de coluna válido. A única restrição é que para uma chamada de função, não são permitidos espaços entre o nome da função e o '(' que o segue;
- Clientes podem se conectar ao MySQL usando sockets TCP/IP, sockets Unix (Unixes) ou Names Pipes (NT);
- O comando SHOW, específico do MySQL, pode ser usado para resgatar informações sobre bases de dados, tabelas e índices. O comando EXPLAIN pode ser usado para determinar como o otimizador resolve uma consulta;

1. 2. Softwares necessários:

mysql-3.23.56 (versão para Windows e Linux).

mysqlgui (versão para Windows e Linux).

mysql-connector-java (para integração MySQL / J2EE).

MyODBC (versão para Windows e Linux) -> Opcional.

Ambos em www.mysql.com

zeosdbo (para integração MySQL / Delphi -> Kylix).

Em: <http://zeoslib.sourceforge.org>

1. 3. Configuração no Microsoft Windows.

Definir o usuário e a senha com o comando:

```
mysql -u <usuário (root)> -p <senha>
```

Através do prompt do MySQL no DOS ou através do WinMySQLadmin redefinindo as opções do arquivo my.ini, aproveitar para selecionar a língua informando o caminho: language=C:/mysql/share/portuguese

Para criar a base de dados use o comando CREATE DATABASE <nome da base de dados>

Acesse a base de dados usando a interface gráfica configurando Option na aba Server com o nome da máquina e a aba Client com o nome do usuário, ou use o comando:

```
use (ou \u) <base de dados> no prompt do MySQL
```

Crie as tabelas com o comando:

```
CREATE TABLE <nome da tabela>(<nome do campo> <tipo do campo>);
```

Após fazer isso use o comando:

```
GRANT ALL PRIVILEGES ON <base de dados>.<tabela (ou *)> to '<usuário>'@'<máquina (localhost)>' identified by '<senha>';
```

E o comando: FLUSH PRIVILEGES;

Para que todos os processos “saibam” das mudanças.

1. 4. Configuração no Linux.

Logo que terminar a instalação do MySQL utilizar os comandos:

```
./bin/mysqladmin -u root --password 'new-password'
```

```
./bin/mysqladmin -u root -h localhost.localdomain  
--password 'new-password' ou
```

```
./bin/mysqladmin -u root -h localhost --password  
'new-password'
```

Para redefinir a senha do usuário root.

E para iniciar o servidor MySQL usar o comando:

```
mysqld -u root &
```

Para acessar ao cliente MySQL usar o comando:

```
mysql -u root -p <senha>
```

Quando logar criar uma base de dados com o comando:

```
create database <base de dados>
```

Acessar a base de dados com o comando: \u <base de dados>

Criar as tabelas com o comando:

```
create table <nome da tabela>(<nome do campo>  
<tipo do campo>);
```

E para que determinado usuário tenha todos os privilégios no banco de dados usar o comando:

```
grant all privileges on <base de dados>.<tabela  
(ou *)> to '<usuário>'@'<maquina (localhost)>'  
identified by '<senha>';
```

E o comando: flush privileges;
Para que todos os processos “saibam” das mudanças.

1. 5. Mini ajuda de comandos MySQL:

Insira valores nas tabelas com o comando:

```
INSERT INTO <nome da tabela> VALUES  
(<valor1>,...,<valorN>);
```

Pesquise resultados com o comando:

```
SELECT * FROM <tabela> WHERE (<campo>  
= “<valor>”);
```

Altere valores com o comando:

```
UPDATE <tabela> SET <campo> = '<valor  
novo>'  
WHERE <campo> = '<valor antigo>';
```

Apague valores com o comando:

```
DELETE FROM <tabela> WHERE (<campo> =  
'<valor>');
```

Apague tabelas com o comando:

```
DROP TABLE <tabela>;
```

Tipos de campos:

CHAR(M) tipo caractere de tamanho M (fixo)

VARCHAR(M) tipo caractere de tamanho M
(variável, mas mais lento)

INT [Unsigned] tipo numérico inteiro sendo
Unsigned usado para definir o intervalo dos números
positivos.

FLOAT[(M,D)] tipo flutuante onde M tamanho e D precisão decimal (opcionais)

DATE tipo data no formato AAAA(ano) MM(mês) DD(dia).

1. 6. Integração MySQL J2EE

Para utilização do SGBD MySQL com a plataforma J2EE será necessário o driver mysql-connector-java-X.X.XX.zip.

Para utilização desse driver é necessário instala-lo (ou simplesmente copia-lo) no diretório:

\$JAVA_HOME/jre/lib/ext (Linux) ou
%JAVA_HOME%\jre\lib\ext (DOS)

Para uso com JDK e J2SDK.

Caso esteja desenvolvendo para J2EE instale o driver no diretório:

Webapp/WEB-INF/lib (Linux) ou
Webapp\WEB-INF\lib (DOS)

E para utilização com JSP ou Servlets utilize a sequência de comandos:

```
<% @ page import="java.sql.*" %>
try {
    Class          mysql          =
Class.forName("com.mysql.jdbc.Driver");
    Connection con =
DriverManager.getConnection("jdbc:mysql://localhost/pe
quisa?user=root&password=teste");
    Statement stmt = con.createStatement();
    < . . . programação da servlet . . . >
```

```

        stmt.executeUpdate(<comando sql>); // Para
inclusão de dados no banco de dados
        ResultSet          <variavel>          =
stmt.executeQuery(<consulta sql>);
        // Para executar uma consulta no banco de dados
        con.close();
    } catch (Exception e) {out.println("Ocorreu o
seguinte erro "+e);}

```

1. 7. Integração MySQL Delphi / Kylix

Para integração do SGBD MySQL com os ambientes de desenvolvimento Delphi e Kylix foi usado o componente ZeosDBO sendo que a sua utilização segue a seguinte sequência de instalação:

1. 7. 1. No Windows (Delphi):

1. Após descompactar o arquivo zipado copiar o arquivo libmysql.dll para a pasta c:\windows\system (no windows 9X) ou c:\winnt\system (no windows NT / 2000 / XP).
2. Editar o arquivo zeos.inc para definir as opções de linguagem e outras características técnicas.
3. Adicionar no Delphi Library Path os caminhos para XXX, XXX\common, XXX\dbase, onde XXX é o diretório ZeosDBO.
4. Compilar e instalar os arquivos dpk nesta sequência:
ZCommonXXX, ZDbwareXXX, ZMySqlXXX {
ZPgSqlXXX | ZIbSqlXXX | ZMsSqlXXX |
ZOraSqlXXX }
5. Sendo que os arquivos entre parênteses não são necessários para a integração com o SGBD

MySQL , mas servem para o uso com outros SGBDs e caso se deseje usar o PostgreSQL deve-se copiar o arquivo libpq.dll para o mesmo diretório que o arquivo libmysql.dll

1. 7. 2. No Linux (Kylix):

1. Descompactar XXX/kylix/desprop50562.tar.gz no diretório kylix/lib (Os arquivos dbdesign.dcp e std.dcp devem ficar lá). Além de os servidores MySQL ou PostgreSQL devem estar instalados e os clientes com suas bibliotecas compartilhadas.
2. Editar o arquivo zeos.inc para definir as opções de linguagem e outras características técnicas.
3. Adicionar no Kylix Library Path os caminhos para XXX, XXX/common, XXX/dbase, onde XXX é o diretório ZeosDBO.
4. Compilar e instalar os arquivos dpk nesta seqüência:
ZCommonXXX, ZDbwareXXX, ZMySQLXXX {
ZPgSqlXXX | ZIbSqlXXX }
5. Sendo que os arquivos entre parênteses não são necessários para a integração com o SGBD MySQL , mas servem para o uso com outros SGBDs.

1. 8. Referencia bibliográfica:

Este material sobre MySQL foi compilado através de diversos textos combinados e adaptados de diversas fontes sendo as principais e de caráter original apresentadas a seguir:

Tutorial MySQL básico <http://www.linuxqos.cjb.net>

Documentação oficial do MySQL
<http://www.mysql.com/documentation/>

MySQL Connector/J 2.0.14 (Formerly MM.MySQL -
Mark Matthews JDBC Driver for MySQL) Copyright (c)
2002 MySQL-AB

Zeos Database Objects is distributed as freeware product
with LGPL licence. <http://zeoslib.sourceforge.org> or
<https://sourceforge.net/projects/zeoslib>

2. Servidor Apache Jakarta-TomCat e J2SDK

2. 1. Por que JSP?

Sem dúvida, o J2EE (Java 2 Enterprise Edition) é o nicho de tecnologia que vêm despertando maior furor. Mas o que é o J2EE? Inicialmente, o Java no servidor era limitado aos servlet engines, que permitiam que um código Java executasse como um programa CGI. Com o sucesso de tecnologias como o ASP e o PHP, o Java incorporou a tecnologia JSP, que foi agregada com um diferencial: ela serve como desacoplamento entre código (servlet) e a apresentação (JSP). Esta idéia é chamada Model X View X Controller, e já foi explorada em algumas edições da DevMag. Esta foi apenas a ponta do iceberg. Nenhuma outra tecnologia existente atualmente, pelo menos as que já saíram do estado de Beta, implementa de forma tão eficiente o desacoplamento entre apresentação e código.

Para aplicações de cliente magras, J2EE têm Java ServerPages (JSP) para etiqueta orientando o HTML dinâmico página e servlets para HTML da programação de páginas.

Classes de fundação do Java (JFC) são utilizadas para clientes complexos, ricos.

Serviços de Web são utilizados para acesso da programação.

2. 2. Softwares necessários:

j2sdk1.4.1_01 (versão para Windows e Linux).

Em <http://java.sun.com/j2se/1.4.1/download.html>

j2sdk1.4.1_01-doc (documentação do j2sdk1.4.1_01).

Em <http://java.sun.com/j2se/1.4.1/download.html#docs>

tomcat-4.1.24 (versão para Windows e Linux).

Em <http://jakarta.apache.org/tomcat>

isapi_redirect.dll (integração TomCat / IIS)

mod_jk.so ou mod_jk.dll (integração TomCat / Apache)

Em <http://jakarta.apache.org/builds/jakarta-tomcat-connectors/jk/release/v1.2.4/bin/win32/>

(windows)

Em <http://jakarta.apache.org/builds/jakarta-tomcat-connectors/jk/release/v1.2.4/bin/linux/>

(linux)

2. 3. Configuração do J2SDK

Depois de devidamente instalado o J2SDK no Windows 9x, NT, 2000 XP e Linux seguindo o guia de instalação disponível para cada sistema operacional no qual a instalação for efetuada há início a configuração do mesmo.

2. 3. 1. No Windows 9x:

Setar a variável de ambiente JAVA_HOME editando o arquivo autoexec.bat de acordo com o caminho onde o j2sdk foi instalado, por exemplo: set JAVA_HOME=C:\j2sdk1.4.1_01

Depois inserir na PATH a referencia a variável de ambiente JAVA_HOME como por exemplo: PATH=<. . .>;%JAVA_HOME%\bin

2. 3. 2. No Windows NT, 2000, XP

Setar a variável de ambiente JAVA_HOME indo até o “painel de controle” no ícone sistema na aba avançado selecionando a opção variáveis de sistema incluindo a variável JAVA_HOME (com o valor C:\j2sdk1.4.1_01, por exemplo) e editando a variável PATH adicionando o valor %JAVA_HOME%\bin

2. 3. 3. No Linux

Usar o comando para definir a variável de ambiente JAVA_HOME:

```
export JAVA_HOME=/<diretorio onde foi instalado o j2sdk>
```

```
export JAVA_HOME=/usr/java/j2sdk1.4.1_01,
```

por exemplo

E o comando para adicionar a variável de ambiente JAVA_HOME na path do sistema

```
PATH=$PATH:$JAVA_HOME/bin
export PATH
```

Para conferir se a variável foi corretamente alterada use o comando:

```
echo $PATH
```

2. 4. Configuração do servidor Apache Jakarta-TomCat

Depois de devidamente instalado o TomCat no Windows 9x, NT, 2000 XP e Linux seguindo o guia de instalação disponível para cada sistema operacional no qual a instalação for efetuada há início a configuração do mesmo.

2. 4. 1. No Windows 9x:

Setar a variável de ambiente CATALINA_HOME editando o arquivo autoexec.bat de acordo com o caminho onde o TomCat foi instalado, por exemplo: set CATALINA_HOME=C:\Arquivos de programas\Apache Group\Tomcat 4.1

Depois inserir na PATH a referência a variável de ambiente CATALINA_HOME como por exemplo: PATH=<. . .>;% CATALINA_HOME%\bin

Para iniciar o servidor TomCat no Windows 9x:

Acessar o menu iniciar / programas / Apache Tomcat 4.1 / Start TomCat

ou

Digitar o comando: C:\j2sdk1.4.1_01\bin\java.exe -jar -Duser.dir="C:\Arquivos de programas\Apache Group\Tomcat 4.1" "C:\Arquivos de programas\Apache Group\Tomcat 4.1\bin\bootstrap.jar" start

Para parar o servidor TomCat no Windows 9x basta seguir os mesmos passos que para iniciar, mudando, é claro, o comando ou opção Start por Stop

2. 4. 2. No Windows NT, 2000, XP

Setar a variável de ambiente CATALINA_HOME indo até o “painel de controle” no ícone sistema na aba

avariado selecionando a opção variáveis de sistema incluindo a variável CATALINA_HOME (com o valor C:\"Arquivos de programas"\Apache Group"\Tomcat 4.1, por exemplo) e editando a variável PATH adicionando o valor % CATALINA_HOME%\bin

Ao instalar e configurar o servidor TomCat no Windows com tecnologia NT é instalado automaticamente um serviço de sistema que garante o início do servidor assim que o Windows é carregado, mas se for necessário pode-se iniciar o serviço manualmente como no Windows 9x

2. 4. 3. No Linux

Usar o comando para definir a variável de ambiente JAVA_HOME:

```
export CATALINA_HOME=/<diretorio onde foi
instalado o TomCat>
```

```
export CATALINA_HOME=/usr/jakarta-tomcat-
4.1.24, por exemplo
```

E o comando para adicionar a variável de ambiente CATALINA_HOME na path do sistema

```
PATH=$PATH:$ CATALINA_HOME/bin
export PATH
```

Para conferir se a variável foi corretamente alterada use o comando:

```
echo $PATH
```

Para iniciar o servidor TomCat no Linux pode-se usar a seguinte Shell:

```
start_tomcat
#!/bin/bash
```

```
#####
#####
# define as variáveis de ambiente necessárias para a
# execução do servidor
#
export JAVA_HOME=/usr/java/j2sdk1.4.1_01
export CATALINA_HOME=/usr/jakarta-tomcat-4.1.24
#
# adiciona as variáveis de sistema definidas na path do
# sistema
#
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HOME/bin
#
export PATH
#
cd $CATALINA_HOME/bin
#
# inicia o servidor TomCat
#
./startup.sh
#
#####
#####
```

Para parar o servidor TomCat pode-se usar a seguinte Shell:

```
stop_tomcat
#!/bin/bash
#####
#####
```

```

# define as variáveis de ambiente necessárias para a
execução do servidor
#
export JAVA_HOME=/usr/java/j2sdk1.4.1_01
export CATALINA_HOME=/usr/jakarta-tomcat-4.1.24
#
# adiciona as variáveis de sistema definidas na path do
sistema
#
PATH=$PATH:$JAVA_HOME/bin:$CATALINA_HO
ME/bin
#
export PATH
#
cd $CATALINA_HOME/bin
#
# inicia o servidor TomCat
#
./shutdown.sh
#
#####
#####

```

2. 5. Integração Apache Jakarta-TomCat / Apache

2. 5. 1. Compatibilidades do TomCat / Apache

O módulo mod_jk foi desenvolvido e testado em:

- Linux, FreeBSD, AIX, HP-UX, MacOS X, e deve trabalhos em plataformas de Unixes principais que apóiam o Apache 1.3 e/ou 2.0

- WinNT4.0-i386 SP4/SP5/SP6a (deveria poder trabalhar com outros pacotes de serviço), Win2K, WinXP e Win98
- Cygwin (até que você possua um servidor Apache com ferramentas que apoiem autoconf/automake)
- NETWARE
- ISeries V5R1 e V5R2 com Apache 2.0.39. Esteja seguro em ter o mais recente Apache PTF instalado.
- TomCat 3.2.x, TomCat 3.3.x, TomCat 4.0.x, TomCat 4.1.x e TomCat 5

2. 5. 2. Como funciona o redirecionamento TomCat / Apache?

Em um acesso a Web um servidor Web está esperando por pedidos HTTP do cliente. Quando estes pedidos chegam o servidor faz tudo que é preciso para servir aos pedidos provendo o conteúdo necessário.

Adicionando um servlet container algumas mudanças podem ocorrer no comportamento do servidor. Agora o servidor Web também precisa executar o seguinte:

- Carregar e inicializar a biblioteca do adaptador da servlet container (antes de servir os pedidos).
- Quando um pedido chega, é preciso conferir se o pedido pertence a um servlet, nesse caso é preciso deixar o adaptador controla-lo.

O adaptador precisa saber por outro lado que pedidos vai servir, normalmente baseado em algum padrão no URL de pedido, e para onde dirigir estes pedidos.

Configurações mais complexas ocorrem quando o usuário quer utilizar uma configuração que usa vários hosts virtuais, ou quando ele quer que múltiplos desenvolvedores trabalhem no mesmo servidor Web mas em recipiente de servlet diferente utilizando varias JVMs.

2. 5. 3. *Obtendo o mod_jk*

O mod_jk pode ser obtido em dois formatos - binário e fonte. Dependendo da plataforma que você estiver rodando seu servidor Web.

É recomendado usar a versão binária se esta estiver disponível. Se o binário não está disponível, siga as instruções para construir mod_jk a partir do código fonte.

Os binários para o mod_jk agora estão disponíveis, para várias plataformas, em uma área separada do Tomcat Binary Release. Os binários ficam situados em subdiretórios de acordo com a plataforma.

Para algumas plataformas, como Windows, este é o modo típico de obter mod_jk pois a maioria dos sistemas Windows não possuem compiladores de C em sua instalação básica.

Por exemplo JK 1.2.4 pode ser achado **aqui** e pode ser contido o seguinte:

Localização	Conteúdo
aix	SAVF inclusive mod_jk para Apache 2.0 para iSeries V5R1/V5R2
iseries	SAVF inclusive mod_jk para Apache 2.0 para iSeries V5R1/V5R2
linux	mod_jk.so(Apache 1.3 standard API e EAPI e Apache 2.0) para algumas arquiteturas Linux

macosx	Contém o mod_jk.so para MacOS X
netware	mod_jk.nlm e nsapi.nlm para Netware
rpms	Contém o rpms (inclusive fontes e códigos para arquitetura i386/ppc)
solaris6	Contém o mod_jk.so para Solaris 6
solaris8	Contém o mod_jk.so para Solaris 8
win32	Contém o mod_jk.dll para Windows como outros binários úteis.

2. 5. 4. *Instalação e configuração do redirecionador TomCat / Apache*

mod_jk requer duas entidades:

- mod_jk.xxx - O módulo Apache, dependendo de seu sistema operacional, será mod_jk.so, mod_jk.dll, mod_jk.nlm ou QZTCJK.SRVPGM (veja a seção de construção).
- workers.properties - Um arquivo que descreve o host(s) e porta(s) usados pelos trabalhadores (processos do TomCat). Uma amostra de workers.properties pode ser achado no diretório conf.

Também como com outros módulos Apaches, deveria ser instalado mod_jk primeiro no diretório de módulos de seu Webserver Apache, ie,: /usr/lib/Apache e você deveria atualizar seu arquivo httpd.conf.

2. 5. 5. *O velho “mod_jserv”*

Se você configurou o Apache previamente para usar mod_jserv, remova qualquer diretiva de ApJServMount de seu httpd.conf.

Se você está incluindo TomCat*-Apache.conf ou tomcat.conf, você irá querer os remover bem como - eles são específicos a mod_jserv.

As diretivas de configuração de mod_jserv não são compatíveis com mod_jk!

2. 5. 6. TomCat usando auto-configuração

O modo mais simples para configurar o Apache para usar mod_jk é usar o Apache no modo auto-configuração setado em TomCat e inclua a seguinte diretiva ao término de seu arquivo httpd.conf Apache (tenha certeza de substitui TOMCAT_HOME com o caminho correto para sua instalação do TomCat):

Ser adicionado ao término de seu arquivo http.conf a linha

Include TOMCAT_HOME/conf/jk/mod_jk.conf-auto

Isto dirá para o Apache que use diretivas do arquivo mod_jk.conf-auto na configuração Apache. Este arquivo é criado habilitando a auto-configuração Apache como descrito na documentação do TomCat.

2. 5. 7. Exemplo de uma configuração simples

Eis uma configuração simples:

Carregar o modulo mod_jk

LoadModule jk_module libexec/mod_jk.so

Declarar o modulo para <se diretiva do modulo>

```

        AddModule mod_jk.c
# Onde procurar por workers.properties
        JkWorkersFile /etc/httpd/conf/workers.properties
# Onde colocar os logs do jk
        JkLogFile /var/log/httpd/mod_jk.log
# Definir o nível do log do jk [debug/error/info]
        JkLogLevel info
# Selecionar o formato do log
        JkLogStampFormat "[%a %b %d %H:%M:%S
%Y] "
# JkOptions indicações de envio SSL KEY SIZE,
        JkOptions                                +ForwardKeySize
+ForwardURICompat -ForwardDirectories
# JkRequestLogFormat definindo o formato da
requisição
        JkRequestLogFormat "%w %V %T"
# Enviando servlet para contexto /exemplos para worker
nomeado worker1
        JkMount /examples/servlet/* worker1
# Enviando JSPs para contexto /exemplos para worker
nomeado worker1
        JkMount /examples/*.jsp worker1

```

Definindo os trabalhadores:

JkWorkersFile especifica o local onde mod_jk achará as definições de trabalhadores.

```

        JkWorkersFile /etc/httpd/conf/workers.properties

```

Logando:

JkLogFile especifica o local onde mod_jk vai colocar seus arquivo de log.

```

        JkLogFile /var/log/httpd/mod_jk.log

```

JkLogLevel fixou o nível de log em:

- info log vai conter atividade de mod_jk standard (falta).
- error log vai também conter relatórios de erro.
- debug log vai conter todas as informações das atividades de mod_jk

JkLogLevel info

info deveria ser sua seleção para operações normais.

JkLogStampFormat configurará o formato de data/hora no arquivo de log de mod_jk. Usando o strftime () sendo a string construída no formato" [%a %b %d %H:%M:%S %Y]"

JKLOGSTAMPFORMAT " [%A %B %D %H:%M:%S %Y]"

JkRequestLogFormat configurará o formato de mod_jk anotando cada pedido individual. Pedido anotando é configurado e habilitado em um host virtual base. Habilitar um pedido para um host virtual adicionando a um JkRequestLogFormat config. A sintaxe do formato da string é similar ao comando Apache LogFormat, abaixo há uma lista das opções de formato de log disponíveis:

Descrição de opções

Opções	Descrição
%b	Bytes enviados, enquanto excluindo cabeçalhos de HTTP (formato CLF)
%B	Bytes enviados, enquanto excluindo cabeçalhos de http
%H	O protocolo do pedido
%m	O método do pedido

%p	O Porto canônico do servidor que serve o pedido
%q	A query string (precedida por um? se há uma string, caso contrário uma string vazia)
%r	Primeira linha de pedido
%s	Requisição do código de estado do HTTP
%T	Requisição da duração, do tempo decorrido para controlar o pedido em segundos '.' micro-segundos
%U	O caminho da URL pedida, não incluindo qualquer query string
%v	O nome canônico do servidor que serve o pedido
%V	O nome do servidor de acordo com a colocação de UseCanonicalName
%w	O nome do trabalhador TomCat

JKREQUESTLOGFORMAT " %W %V %T "

2. 5. 8. *Forwarding*

As diretivas do JkOptions lhe permitem fixar opções de remetentes que serão habilitadas (+) ou desabilitadas (-) nas seguintes opções.

JkOptions **ForwardKeySize**, questiona mod_jk, quanto ao uso do ajp13, para depois questionar quanto à SSL Key Size como é requerido por Servlet API 2.3. Este flag não deveria ser fixado quando servlet engine for TomCat 3.2.x (como por default).

JkOptions +ForwardKeySize

JkOptions **ForwardURISCompat**, você solicitou que mod_jk enviasse o URI normalmente a TomCat é menos compatível com mod_rewrite, para aumentar a compatibilidade com TomCat 3.2.x engines (como por default).

JkOptions +ForwardURISCompat

JkOptions **ForwardURISCompatUnparsed**, o URI remetido não é analisado gramaticalmente, é compilado evocando mod_rewrite

JkOptions +ForwardURISCompatUnparsed

JkOptions **ForwardURIEscaped**, o URI remetido é escapado e TomCat (como 3.3 rc2) fará a parte de decodificação.

JkOptions +ForwardURIEscaped

JkOptions **ForwardDirectories** é usado junto com a diretiva do servidor Web Apache **DirectoryIndex**. Como tal mod_dir deveria estar disponível ao Apache sendo, statically ou dynamically (DSO)

Quando DirectoryIndex é configurado, o Apache criará sub-pedidos para cada url local especificada pela diretiva, se houver um arquivo local que se assemelha (isto é determinado através da declaração do arquivo).

Se ForwardDirectories é fixado para falso (default) e o Apache não acha qualquer arquivo que se assemelha, o Apache servirá o conteúdo do diretório (se as diretivas de Opções especificam Índices para aquele diretório) ou uma resposta Proibida 403 (se as diretivas de Opções não especificam Índices para aquele diretório).

Se ForwardDirectories é fixado para retificar e o Apache não acha qualquer arquivo que se assemelha, o pedido

será remetido a TomCat para resolução. Isto é usado em casos quando o Apache não pode ver que o índice arquiva no sistema de arquivo por várias razões: TomCat está rodando em uma máquina diferente, o arquivo JSP foi precompilado etc.

Note que arquivos localmente visíveis só levarão precedência em cima dos primeiros visíveis para TomCat (i.e. se o Apache pode ver o arquivo que é o que vai ser servido)

`JkOptions +ForwardDirectories`

O `JkEnvVar` é uma diretiva que permite remeter uma variável de ambiente de servidor Apache para máquinas TomCat.

`JKENVVAR SSL_CLIENT_V_START`

Assinalando URLs para TomCat

Se você criou `mod_jk.conf-local` em um local padrão, você pode mudar colocações como os trabalhadores ou prefixo de URL.

A diretiva `JkMount` assinala URLs específicos ao TomCat. Em geral a estrutura de uma diretiva `JkMount` é:
#envie todos os pedidos que terminam em `.jsp` a `worker1`

`JkMount / * worker1 de .jsp`

#envie todos os pedidos que terminam `/servlet` a `worker1`

`JkMount / */servlet / worker1`

#envie todo o `.jsp` de pedidos para arquivos localizados em `/otherworker` irá para `worker2`

`JkMount /otherworker / *.jsp worker2`

Você pode usar a diretiva `JkMount` em nível superior ou dentro de seções `<VirtualHost>` de seu arquivo `httpd.conf`.

2. 5. 9. Configurando o Apache para servir arquivos de aplicação Web estáticos

Se o TomCat host appBase diretório (Webapps) é acessível pelo servidor Web Apache, o Apache pode ser configurado para servir para Web aplicações, contexto de diretórios, arquivos estáticos ao em vez de passar o pedido ao TomCat.

Precaução: Se o Apache é configurado para servir páginas estáticas para uma aplicação Web que evita qualquer contrato de segurança que você pode ter configurado em sua aplicação Web Web.xml.

Use uma diretiva de Alias do Apache para traçar um único contexto de diretório Web para aplicação no espaço de documento do Apache para um VirtualHost:

#Arquivos estáticos no Webapp de exemplos são servidos através de Apache

Alias /examples /vat/tomcat3/Webapps/examples

#A linha seguinte proíbe aos usuários de acesso direto a WEB-INF

<Location " /examples/WEB-INF /">

AllowOverride None

deny from all

</Location>

#Todo o JSP vai para worker1

JkMount / * worker1 de .jsp

#Todo o servlets vai para worker1

JkMount / * /servlet / worker1

Use a diretiva JkAutoAlias de mod_jk para traçar todos os contextos de diretórios de aplicações Web no espaço de documento do Apache.

Tentativas para ter acesso ao WEB-INF ou diretórios de META-INF dentro de um contexto de aplicação Web ou um Arquivo Web * .war dentro do TomCat Host appBase diretório (Webapps) falhará com um HTTP 403, Acesso Proibido

#Arquivos estáticos em todos os diretórios de contexto TomCat Webapp serão servidos através do Apache

JkAutoAlias /var/tomcat3/Webapps

JkMount / * .ajp13 de .jsp

JkMount / * /servlet / .ajp13

2. 6. Integração Apache Jakarta-TomCat / IIS

2. 6. 1. Compatibilidades do TomCat / IIS

O redirecionador de TomCat / IIS foi desenvolvido e testado para:

- WinNT4.0-i386 SP4/SP5/SP6a (deveria poder trabalhar com outros pacotes de serviço), Win2K , WinXP e Win98
- IIS4.0 e PWS4.0
- TomCat 3.2.x, TomCat 3.3.x, TomCat 4.0.x, TomCat 4.1.x e TomCat 5

O redirecionador usa ajp12 e ajp13 para enviar pedidos aos recipientes do TomCat. Também há uma opção para usar TomCat em curso.

2. 6. 2. Como funciona o redirecionador do TomCat / IIS ?

1. O redirecionador do TomCat / IIS é um plugin do IIS (filtro + extensão), o IIS carrega o plugin de

- redirecionamento e chama a sua função de filtro para cada pedido requisitado.
2. O filtro testa o URL do pedido então encontra uma lista de caminhos de URI segura dentro do arquivo `uriworkermapping.properties`. Se o pedido atual coincide com uma das entradas na lista de caminhos de URI válidos, o filtro transfere o pedido para a extensão adequada.
 3. A extensão armazena os parâmetros do pedido e os remete ao trabalhador apropriado que usa o protocolo definido como `ajp13`.
 4. A extensão armazena a resposta do trabalhador e envia o resultado para o browser.

2. 6. 3. Instalação e configuração do redirecionador TomCat / IIS

Considerando `CATALINA_HOME` sendo o endereço: `C:\jakarta-tomcat\` o arquivo `isapi_redirect.dll` estará instalado em `c:\jakarta-tomcat\bin\win32\i386\isapi_redirect.dll` e foram criados os arquivos de propriedades em `c:\jakarta-tomcat\conf`.

1. No registro, crie uma nova chave nomeada `[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Foundation\Jakarta Isapi Redirector\1.0]`
2. Adicione um novo valor de sequência com o nome `extension_uri` com o valor de `/jakarta/isapi_redirect.dll`
3. Adicione um novo valor de sequência com o nome `log_file` com um valor que aponta para onde você quer seu arquivo de log (por exemplo, `c:\jakarta-tomcat\logs\isapi.log`).

4. Adicione um novo valor de seqüência com o nome `log_level` com um valor para seu nível de log (pode ser `depure`, `info`, `erro` ou `emerg`).
5. Adicione um novo valor de seqüência com o nome `worker_file` com um valor que seja o caminho completo do seu arquivo `workers.properties` (por exemplo `c:\jakarta-tomcat\conf\workers.properties`)
6. Adicione um novo valor de seqüência com o nome `worker_mount_file` com um valor que seja o caminho completo do seu arquivo `uriworkermap.properties` (por exemplo `c:\jakarta-tomcat\conf\uriworkermap.properties`)
7. Usando o console de administração do IIS acrescente um novo diretório virtual ao seu IIS/PWS Web site. O nome do diretório virtual deve ser `jakarta`. Seu caminho físico deveria ser o diretório onde você colocou o `isapi_redirect.dll` (em nosso exemplo é `c:\jakarta-tomcat\bin\win32\i386`). Após criando este novo diretório virtual assinale este diretório com acesso de execução.
8. Usando o console de administração do IIS adicione `isapi_redirect.dll` como um filtro em seu IIS/PWS Web site. O nome do filtro deveria refletir sua tarefa (eu uso o `jakarta` de nome), seu executável deve ser o endereço `c:\jakarta-tomcat\bin\win32\i386\isapi_redirect.dll`. Para o PWS, você precisará usar `regedit` e editar a lista de DLLs da chave `[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W3SVC\Parameters]`. Esta

chave contém uma lista separada de dlls separadas por (,) (caminhos completos) - você precisa inserir o caminho completo a isapi_redirect.dll.

9. Reinicie IIS (parar e recomeçar o IIS funciona), para ter certeza que o filtro jakarta foi carregado ele deve estar marcado com uma seta verde apontando para cima. No Win98 você pode precisar acessar ao endereço cd WINDOWS\SYSTEM\inetsrv e digitar PWS /stop (assim são fechados os DLLs e os arquivos de log - até mesmo se você clicar o botão de parada, PWS ainda manterá os DLLs em memória.). Digite pws para começar recomeçar novamente.

Caso tenha experiência com o registro do Windows crie o arquivo tomcat.reg com as seguintes chaves:

2. 6. 4. Criação de um arquivo de registro

REGEDIT4

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache  
Software Foundation\Jakarta Isapi Redirector\1.0]  
"extension_uri"="/jakarta/isapi_redirect.dll"  
"log_file"="C:\jakarta-tomcat\logs\isapi.log"  
"log_level"="info"  
"worker_file"="C:\jakarta-  
tomcat\conf\workers.properties"  
"worker_mount_file"="C:\jakarta-  
tomcat\conf\uriworkermap.properties"
```

2. 6. 5. Um exemplo do arquivo uriworkermap.properties

Um exemplo do arquivo uriworkermap.properties pode ser obtido ao se tentar incluir um novo contexto como:

 /context/*=worker_name é definido o nome
do local de trabalho

 /loja/*=defworker é definido um novo
contexto

ou

 /examples/*=defworker este contexto é
substituído pelas duas próximas linhas

 que tornam explícito o uso de arquivos JSP e
servlets

 /examples/*.jsp=defworker

 /examples/servlet/*=defworker

2. 6. 6. Um exemplo do arquivo worker.properties

Um exemplo do arquivo worker.properties pode ser obtido ao se tentar incluir um novo espaço de trabalho sobre um novo contexto como:

Uma lista de espaços de trabalho como:

 worker.list=worker1, worker2

Entradas que definem o host a porta associada a este host e o valor de ajp por exemplo

 worker.worker1.host=localhost

 worker.worker1.port=8009

 worker.worker1.type=ajp13

 worker.worker2.host=otherhost

 worker.worker2.port=8009

worker.worker2.type=ajp13

Um exemplo usando um fragmento do arquivo uriworkermap.properties onde os espaço de trabalho são definidos:

/examples/*=worker1

/Webpages/*=worker2

E como mostrado acima usados no arquivo work.properties

2. 7. Alguns comandos Java JSP

2. 7. 1. Algumas tags úteis:

TAG	Significado
<% . . . %>	Qualquer coisa entre essas tags é interpretada como um comando JSP
<%= . . . %>	Informa que tudo que está entre essas tags é texto e se estiver entre aspas é considerado um parâmetro
<%! . . . %>	Tudo que é declarado entre essas tags é considerado de escopo global como uma variável ou constante
<%- . . . - %>	Essas são tags de comentário caso queira comentários HTML use <!-- . . . --> ou comentários java <% // %> <% /* . . . */ %> <% /** . . . */ %>

2. 7. 2. Algumas diretivas JSP

Diretivas	Significado
<%@ page info="IC CC" %>	Retorna o valor definido a info pela função

	getServletInfo()
<% @page import="java.applet.*" %>	Será importado todo o conteúdo de java.applet
<% @page language="java" %>	Define a linguagem da pagina como Java
<% @page session="true" %>	Cria uma nova sessão a partir da pagina corrente
<% @page errorPage="f1.jsp" %>	Indica qual deverá ser a pagina de erro que deve ser usada
<% @page isErrorPage="true" %>	Indica que a pagina é uma pagina de erro
<% @include file="aa.html" %>	Inclui o conteúdo do arquivo especificado na posição especificada

2. 7. 3. Alguns comandos úteis para programação Java / JSP

Comando	Resultado
request.getMethod()	GET
request.getRequestURI()	/examples/servlet/zzz
request.getProtocol()	HTTP/1.1
request.getServletPath()	/servlet/zzz
request.getQueryString()	Null
request.getContentLength()	-1
request.getServerName()	127.0.0.1
request.getServerPort()	8080
request.getRemoteUser()	Null
request.getRemoteAddr()	127.0.0.1
request.getRemoteHost()	127.0.0.1
request.getScheme()	http

2. 8. Referência bibliográfica:

Este material sobre Apache Jakarta-TomCat e J2SDK foi compilado através de diversos textos combinados e adaptados de diversas fontes sendo as principais e de caráter original apresentadas a seguir:

j2sdk1.4.1_01-doc (documentação do j2sdk1.4.1_01).
Em <http://java.sun.com/j2se/1.4.1/download.html#docs>

tomcat-4.1.24 (versão para Windows e Linux). Em
<http://jakarta.apache.org/tomcat>

isapi_redirect.dll (integração TomCat / IIS). Em
<http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/jk/iishowto.html>

mod_jk.so ou mod_jk.dll (integração TomCat / Apache).
Em <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/jk2/jk/aphowto.html>

3. Servidor WAP

3. 1. O que é WAP ?

Tendo visto que a maioria dos dispositivos utiliza arquiteturas que permitem a interação do usuário com páginas geradas em Hipertext Markup Language (HTML), a arquitetura WAP (Wireless Application Protocol), através de sua linguagem de programação específica denominada WML (Wireless Markup Language) permite que uma gama maior de dispositivos possa ter acesso aos dados através de consultas ao SGBD e alteração de suas informações conforme necessário.

A especificação do WAP usa os padrões de comunicação sem fio, existentes e acrescenta novas extensões, permitindo que os participantes da indústria desenvolvam soluções como interfaces aéreas independentes, dispositivos independentes e completamente interoperáveis [RISCHPATER 2001].

A principal característica do padrão WAP é a utilização de vários protocolos já existentes na Internet, como TCP, UDP, XML, HTML, SSL, etc. [WAP 2001].

A configuração do servidor WAP é de responsabilidade da empresa de telefonia celular que provê o serviço de acesso a paginas Web via telefonia móvel, mas esse detalhe não pode ser empecilho para que nenhum ser vivo pensante possa desenvolver suas próprias paginas para Web móvel por isso levante você mesmo seu próprio Servidor WAP.

3. 2. Softwares necessários:

É recomendado que se faça o download do seguinte kit:
Nokia Mobile Internet Toolkit Version 3.1 em
<http://www.forum.nokia.com>.

Este kit da Nokia possui um Servidor WAP e um Gateway WAP / HTTP / WAP integrados que possibilitam o desenvolvimento de paginas wap sem a necessidade de se conectar a um servidor convencional economizando tempo e recursos do projeto

3. 3. Configuração do Nokia Mobile Internet Toolkit

Este kit da Nokia possui versão desenvolvida em Java executável sobre a plataforma Windows, portanto sua configuração se dá de forma quase que automatizada e sua integração com outros servidores Web se dá de forma transparente através do Gateway WAP incluso a partir desses princípios o restante desta sessão se dará sobre aspectos da linguagem WML que faz parte do conteúdo servido por este servidor.

3. 4. Conceitos básicos sobre WML

A linguagem WML (Wireless Markup Language) é da mesma família da linguagem XML (eXtensible Markup Language) que por sua vez evoluiu da SGML (Structured General Markup Language) linguagem que também deu origem a HTML (HiperText Markup Language). WML possui características únicas como a nomenclatura sendo que cada pagina Web desenvolvida em WML possui decks (na maioria das vezes um deck por pagina) e cards,

sendo o deck caracterizado pelas tags <wml> </wml> e os cards pelas tags <card> </card>, e como em todo “baralho” cada deck pode possuir diversos cards que no mundo físico costumam representar uma “tela”, ou seja, para cada tela que se deseja exibir em seu dispositivo Wireless cria-se um card para aquela tela.

Observando-se que o tamanho total do arquivo WML compilado para ser interpretado pelo dispositivo de destino não deve passar de 1 Kbyte é recomendado que cada deck possua o numero suficiente de cards para que use esse espaço de forma consistente e sempre que possível que a correção de erros dos dados fornecidos pelos usuários seja feita no dispositivo do usuário para evitar ao máximo a interação do usuário com o provedor tornando a navegação na pagina a mais produtiva e econômica possível.

3. 5. Desenvolvendo em WML:

3. 5. 1. A estrutura de um arquivo WML

1. A primeira linha especifica o número da versão da XML.
2. A segunda linha especifica o identificador de documento publico da SGML.
3. A terceira linha especifica o local da Definição de Tipo de Documento (DTD) da WML. O DTD pode ser localizado na rede, ou você pode armazenar-lo localmente isto fará o acesso se tornar mais rápido.

4. A quarta linha define o cabeçalho do deck WML. Todas os decks WML têm que começar com uma tag <wml> e terminar com uma tag </wml>.
5. As linhas seguintes definem um card que contém uma tag de início e uma tag de fim e o texto ser exibido para o usuário.
6. A última linha é o rodapé do deck. O agente de usuário trata tudo entre o cabeçalho do deck e o rodapé do deck como um único deck WML.

3. 5. 2. As definições dos elementos da WML

A próxima tabela classifica os tipos de tags encontrados em WML

Categoria	Elementos WML
Decks e cards wml	card template head access meta
Eventos	do ontimer onenterforward onenterbackward onpick onevent postfield
Tarefas	go prev refresh noop

Variaveis	setvar
Entrada de usuario	input select option optgroup fieldset
Ancoras, imagens, e cronômetros	a anchor img timer
Formatação de texto	br p table tr td pre (WAP 1.2 and 1.3)

3. 5. 3. *Decks e Cards*

A próxima tabela dá uma breve explicação dos elementos usados para definir os cards dentro de um deck.

Elemento	Explicação
wml	Define um deck e inclui todas as informações e cards no deck.
card	Indica o esquema geral e o planejamento do usuário no agente de usuário.
template	Declara um modelo para card no deck.
head	Contém informação como um todo relativo ao deck, inclusive meta-dados e elementos de controle de acesso.
access	Especifica informação de controle de acesso

	para o deck inteiro.
meta	Contém informação do tipo meta genérica que relacionado ao deck.

3.5.4. Atributos

A próxima tabela descreve os atributos do elemento card.

Atributo	Significado
id = " card_name "	Especifica um nome que lhe permite a navegação entre cards no mesmo deck.
título = " título "	Especifica um titulo para um card.
newcontext = " false "	Se você colocar este atributo como true o contexto do browner será reinicializado atualizando a entrada do card. O contexto do browner é usado para gerenciar todos os parâmetros e estados do agente de usuários, incluindo variáveis, histórico de navegação e outras implementações - dependendo da informação relatada do estado corrente do agente de usuário.
ordered=boolean	Dá uma indicação ao agente de usuário como o conteúdo de cartão é organizado.
onenterforward = " href "	Navega ao local especificado por href. Acontece quando o usuário navegar em um cartão usando uma tarefa GO.
onenterbackward	Navega ao local especificado por

= “ href “	href. Acontece quando o usuário navegar em um cartão usando uma tarefa PREV.
ontimer = “ href “	Navega ao local especificado por href depois de uma quantia especificada de tempo.
xml:lang	Especifica o idioma natural ou formal dentro o qual o documento é escrito.
class	Afilia um elemento com uma ou mais classes.

3.5. 5. *Eventos*

A tabela a seguir descreve os eventos dos elementos brevemente.

Evento	Explicação
go	Provê um mecanismo geral para executar ações no cartão atual.
ontimer	Especifica um evento intrínseco que acontece quando um cronômetro expira.
onenterforward	Especifica um evento intrínseco que acontece quando o usuário entra em um cartão debaixo de circunstâncias específicas.
onenterbackward	Especifica um evento intrínseco que acontece quando o usuário navega em um cartão usando um URI retirado da pilha do histórico.
onpick	Especifica um evento intrínseco que acontece quando o usuário seleciona

	ou deseleciona um artigo em qual o evento é especificado.
onevent	Liga uma tarefa a um evento intrínseco particular.
postfield	Especifica um nome de campo e avalia para transmissão para um servidor de origem durante um pedido de URI.

3. 5. 6. *Tarefas*

A próxima tabela se refere às tarefas WML

Tarefa	Explicação
go	Indica navegação a um local especificado por href. Se o href nomeia um card WML ou deck, será exibido no display do dispositivo.
prev	Indica navegação ao URI anterior presente na pilha do histórico.
refresh	Indica uma atualização do contexto de agente de usuário como especificado pela tag setvar.
noop	Indica que nada deveria ser feito.

3. 5. 7. *Variáveis*

Referencias a variáveis

Variável	Explicação
\$var ou \$(var)	Substitutos o valor de var. O agente de usuário escapa a variável usando no URL escapado a convenção da variável no contexto apropriado.

<code>\$(var:e)</code> ou <code>\$(var:escape)</code>	Substitui o valor de var, escapando valores não alfanuméricos , de acordo com convenções de URL.
<code>\$(var:unesq)</code>	Substitui o valor de var, sendo valores não alfanuméricos escapados, de acordo com convenções de URL.
<code>\$(var:N)</code> ou <code>\$(var:noesc)</code>	Substitui o valor de var, sem escapar, caráter não alfanuméricos.

3. 5. 8. Mecanismo de entrada de Usuário

Os elementos de entrada de usuário provêm um mecanismo para captura de entrada de usuário. A tabela a seguir lista estes elementos.

Elemento	Descrição
<code>input</code>	Especifica um objeto de entrada de texto.
<code>select</code>	Permite ao usuário escolher de uma lista de opções. WML suporta escolha única e múltiplas escolha em listas.
<code>option</code>	Especifica uma única opção escolhida em um elemento <code>select</code> .
<code>optgroup</code>	Deixa se agrupar elementos de opção relacionados em uma hierarquia para facilitar o layout e apresentação pelo agente de usuário.
<code>fieldset</code>	Deixa se agrupar campos relacionados e texto.

3. 5. 8. 1. Formatação de texto

O atributo format lhe permite especificar o tipo e a caixa do carácter que o usuário pode entrar. Você pode usar uma combinação dos caracteres especiais a seguir:

Carácter	Explicação
A	Permite qualquer maiúscula alfabética ou carácter de pontuação que é, carácter não-numérico maiúsculo.
a	Permite qualquer minúscula alfabética ou carácter de pontuação que é, carácter não-numérico minúsculo.
N	Permite qualquer carácter numérico.
X	Permite qualquer carácter de maiúsculo.
x	Permite qualquer carácter minúsculo.
M	Permite qualquer carácter. O agente de usuário pode escolher assumir que o carácter é maiúsculo com a finalidade de entrada de dados simples, mas tem que permitir entrada de qualquer carácter. Este é o formato de default.
m	Permite qualquer carácter. O agente de usuário pode escolher assumir que o carácter é minúsculo com a finalidade de entrada de dados simples, mas tem que permitir entrada de qualquer carácter.
*f	Permite qualquer número de caracteres; f é de um dos códigos de formato anterior e especifica em que tipo de carácter pode ser entrado. Nota este formato só pode ser especificado uma vez e tem que aparecer ao fim da string de formato.
nf	Permite carácter de n onde n é um número de 1 a 9; f é de um dos códigos de formato anterior e

	especifica que tipo de caráter pode seja entrado. Note que este formato só pode ser especificado uma vez e tem que aparecer ao fim da string de formato.
\c	Exibe o próximo caráter, c, no campo de entrada. Permite citar os códigos de formato assim eles podem ser exibidos na área de entrada. Por exemplo, nos deixamos supor que você use o formato especificado. \ (3N \). Isto mostra que o agente de usuário insira automaticamente um parêntese esquerdo antes do usuário entrar com qualquer coisa, e um parêntese direito depois que o usuário entrasse com três dígitos.

As tags de ênfase são definidas a seguir:

Tag	Explicação
em	Faça com ênfase.
strong	Faça com ênfase forte.
i	Faça com uma fonte itálica.
b	Faça com uma fonte negrita.
u	Faça com uma fonte sublinhada.
big	Faça com uma fonte grande.
small	Faça com uma fonte pequena.

3. 5. 9. O elemento <do>

A próxima tabela descreve o elemento <do>

Componente	Explicação
------------	------------

<do>	A tag de início do elemento <do>. Elementos <do> podem aparecer dentro de elementos cards e templates. Nota você só pode especificar uma única tarefa para um elemento <do>.
type = tasktype “	Especifica o tipo de tarefa para executar. Por exemplo, accept para navegação ao próximo card, e prev para o card prévio. O atributo de tipo é requerido.
label = " label "	Especifica o rótulo a ser exibido.
name = " name "	Especifica o nome da tarefa de ligação.
optional = “ false “	Se você alterar este atributo para true, o agente de usuário, poderia ignorar este elemento.

3. 5. 10. O elemento <timer>

A próxima tabela descreve o elemento <timer>

Componente	Explicação
<timer>	A tag de início do elemento <timer>.
name = " name "	Especifica o nome da variável a ser alterada com o valor do <timer>. A variável nomeada pelo atributo name será fixada com o valor atual do cronômetro quando o card sair ou quando o cronômetro expirar.
valor = “ 50 “	Especifica o valor de intervalo em unidades de um-décimo de um segundo. Este atributo indica o valor de default da

	variável nomeada no atributo name. Este atributo é requerido.
--	---

3. 5. 11. O elemento <go>

A próxima tabela se refere aos atributos do elemento <go>

Atributo	Explicação
sendreferer=boolean	Se configurado como true, o agente de usuário tem que especificar, para o benefício do servidor, o URI do deck, contendo esta tarefa.
method = “ post get “	Especifica o método de submissão de HTTP. Atualmente, os valores de submissão são post e get aceitos pelo agente de usuário para executar um HTTP get ou post respectivamente.
enctype = “ ContentType “	Quando o atributo de método é post, enctype especifica um dos dois tipos de conteúdo codificando: application/x-wwwform-urlencoded (por default) ou multipart/form-data (somente em WAP 1.2). Quando o método é get, só o tipo de conteúdo, application/x-www-form-urlencoded é permitido.
accept-charset = “	Especifica a lista de codificações

chest “	de caráter para dados que o servidor Web tem que aceitar quando processa a entrada. O valor de default é unknown. O agente de usuário usa o caráter codificando isso foi usado para transmitir o deck WML que contém este atributo.
id	Proporciona a um elemento um nome sem igual dentro de um único deck.
class	Afilia um elemento com uma ou mais classes.

3.5.12. O elemento <input>

A próxima tabela se refere aos atributos do elemento <input>

Atributo	Explicação
name = “ varname “	O nome da variável definido como o resultado da entrada de texto do usuário. Este atributo é requerido.
default = " default "	Especifica uma string de texto de default no qual se aparenta como campo de entrada.
format = “ fmt “	Especifica uma máscara de entrada para o usuário. O formato de default permite para o usuário entrar com qualquer caráter.
emptyok = “ boolean “	Se emptyok é definido como true, nenhuma entrada de usuário é

	requerida; se false (o default), o usuário tem que entrar algum texto.
size = "input_size"	Especifica a largura, em caráter, do texto, área entrada.
maxlength = "maxchar"	Especifica o número máximo de caracteres que o usuário pode entrar na área de entrada de texto. O default é ilimitado.
tabindex = "number"	Especifica o posicionamento do campo de entrada na tabela. A posição na tabela indica a ordem relativa na qual são atravessados elementos quando há tabelas dentro de um único card WML.
value = "value"	Especifica o valor da variável nomeada dentro do atributo name. O default e o valor dos atributos são idênticos no comportamento e sintaxe.
type = "input_type"	Especifica o tipo da área de entrada de texto. Os valores permitidos são text e password. No tipo text, input é exibido ao usuário de uma forma legível. No tipo password, input em cada caráter ecoa uma forma obscurecida. Nota o tipo password é inseguro e crítico a aplicação não deveria ser dependente disto. O tipo de default é texto.
title = "title"	Especifica um título para o elemento de entrada. O título pode ser usado na apresentação deste objeto.
accesskey = "	Nomeia uma chave de acesso

key "	(telefone tipicamente usa um carácter " singular 0 " - 9 " ou " * ou " #") para introduzir o elemento, enquanto permitindo assim para o usuário ativar um elemento particular usando uma única chave.
xml:lang	Especifica o idioma natural ou formal em qual o documento é escrito.
id	Proporciona para um elemento um nome sem igual dentro de um único deck.
classe	Afilia um elemento com uma ou mais classes.

3. 5. 13. O elemento <select>

A próxima tabela se refere aos atributos do elemento <select>

Atributo	Explicação
mutiple= boolean "	Quando fixado true, múltiplas seleções de usuário são permitidas; quando false, só uma única seleção de usuário é permitida. O valor de default é false.
name = varname "	O nome da variável recebe o valor do item escolhido.
default = default "	O valor de default da variável especificada pelo atributo name. Se a variável name já contém um valor, o atributo name é ignorado.
title = " title "	Um título para o elemento select que pode ser usado na apresentação deste

	objeto.
iname = “ varname “	O nome da variável com a que é fixado o resultado do índice da seleção de usuário. Quando o usuário seleciona um artigo, o agente de usuário fixa a variável especificada por iname ao número do artigo escolhido. O número 1 especifica o primeiro artigo, o numero 2 o segundo artigo, e assim por diante. O número 0 indica que nenhuma opção foi selecionada.
ivalue = " default "	O índice do elemento da opção selecionada por default. Se a variável especificada pelo atributo iname não é fixada quando o cartão é exibido, isto é nomeada a entrada selecionada por default. Se a variável já contém um valor, o atributo ivalue, é ignorado.
tabindex = " número "	Especifica o posicionamento na tabela do campo de entrada. A posição na tabela indica a ordem relativa na qual são atravessados elementos quando a tabela está dentro de um único card WML.
xml:lang	Especifica o idioma natural ou formal em qual o documento é escrito.
id	Proporciona para um elemento um nome sem igual dentro de um único deck.
class	Afilia um elemento com uma ou mais classes.

3. 5. 14. O elemento

A próxima tabela se refere aos atributos do elemento ``

Atributo	Explicação
alt	Especifica um texto alternado para exibir no evento quando o agente de usuário não pode exibir imagens.
src	Especifica o URI da imagem para ser exibida.
localsrc=filename	Especifica uma representação interna alternada da imagem. Se esta imagem existir, é usado em vez da imagem especificada pelo elemento de src.
hspace=length vspace=length	Especifica a quantia de espaço em branco a ser inserido a esquerda e a direita (hspace) e sobre e debaixo (vspace) da imagem. O default é 0. O valor do comprimento pode ser expresso como uma porcentagem do espaço disponível.
Align=(top middle bottom)	Especifica alinhamento da imagem dentro do fluxo do texto. Topo da imagem alinhado com o topo do texto atual; mediano o meio da imagem alinha com o meio do texto; fundos o fundo da imagem alinha com o fundo do texto.
height=length width=length	Estes atributos provêem uma sugestão ao agente de usuário (que pode os ignorar) sobre o tamanho da

	imagem entrante de forma que isto poderia pôr de lado o espaço para isto enquanto continuando processando texto entrante.
--	---

3. 5. 15. O elemento <table>

Atributos e sub-elementos do elemento <table>

Atributos	Explicação
<tr>	Especifica um container para uma única linha da tabela. Linhas de tabela podem estar vazias (por exemplo, todas as células na linha estão vazias). Linhas vazias são significantes e não devem ser ignoradas.
<td>	Especifica um container para uma única célula de dados da tabela dentro de uma linha da tabela. Células de dados da tabela podem estar vazias. Células vazias são significantes, e não devem ser ignoradas. O agente de usuário deveria fazer um melhor esforço para lidar com múltiplas linhas de dados células que possam ser o resultado de usar imagens ou quebras de linha.
columns = numero	Especificam o número de colunas para o conjunto de linhas.
align = L	Especifica o plano do texto e imagens dentro da tabela. "C", alinhamento esquerdo o valor na célula "L", e alinhamento certo com o valor na célula

	"R" alinhamento direito com o valor na célula.
--	--

3. 5. 16. Exemplo de código WML

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD
WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <template>
    <do type="prev" name="Previous"
label="Previous">
      <prev/>
    </do>
  </template>
  <card id="card1" title="First Card"
newcontext="true">
    <p>
      Card 1 ... <br/>
      <do type="accept" label="Next Card">
        <go href="#card2"/>
      </do>
      <!-- Must override the DO/PREV in the
template to prevent the PREV element from going back
to the previous deck -->
      <do type="prev" name="Previous">
        <noop/>
      </do>
    </p>
  </card>
  <card id="card2" title="Second Card">

```

```

<p>
  Card 2 ... <br/>
  <do type="accept" label="Next Card">
    <go href="#card3"/>
  </do>
</p>
</card>
<card id="card3" title="Third Card">
  <p>
    Card 3 ... <br/>
    <!-- NOTE: the following DO element will go to
a new deck. The name of this deck is embedded in the
URL.-->
    <do type="accept" label="Next Deck">
      <go href="deck2b.wml"/>
    </do>
  </p>
</card>
</wml>

```

3. 6. Referencia Bibliográfica:

A maioria dos materiais, de qualidade sobre WAP, WML e WMLScript (não abordado neste material) são traduções e interpretações de matérias publicadas pelo WAP Fórum (www.wapforum.com), por esta razão citar autores, livros entre outros se torna desnecessário uma vez que estes obtiveram o seu material junto ao WAP Fórum.

Anexo B – Publicação no 3º CONIC / 1º COINT

Resumo Aprovado em setembro de 2003

Sistema de Acesso a Dados via Dispositivos Fixos e Móveis Independentes de Plataforma

Autor: Tiago Aparecido Teixeira

Orientador: Luciano Eugênio de Castro Barbosa

Co-Orientador: Delcio Nonato Araújo da Silva

Faculdade Ubaense Ozanam Coelho – FAGOC

Introdução:

Há diversas tecnologias de sistemas independentes de plataforma existentes, que possuem um elevado custo e uma alta dependência de dispositivos específicos cada vez mais sofisticados.

Objetivos:

Este trabalho propõe-se a desenvolver um “Sistema de Acesso a Dados via Dispositivos Fixos e Móveis Independentes de Plataforma”, possibilitando o acesso aos dados de uma organização ou instituição, independente do lugar em que o usuário esteja ou do dispositivo (fixo / móvel) ou sistema operacional que utilize. O usuário de tal sistema terá como meios de acesso: via software tipo Desktop (com acesso a Internet), via navegador web e via dispositivo móvel (como celular, por exemplo). Objetiva demonstrar que é possível aproveitar-se de tais tecnologias a baixo custo, quando são utilizados seus recursos gratuitos disponíveis na Internet.

Metodologia:

A abordagem a ser utilizada é a convergência de tecnologias existentes, como a arquitetura Java Enterprise Edition (J2EE), através de sua extensão para a Internet Java Server Pages

(JSP) levando à implementação do dinamismo no ambiente da Internet; a utilização do protocolo Wireless Application Protocol (WAP), através de sua linguagem Wireless Markup Language (WML) para o acesso móvel; a utilização da ferramenta RAD Borland Kylix 3 para o desenvolvimento do cliente Desktop portátil para, pelo menos, duas plataformas distintas e o “Sistema de Gerenciamento de Banco de Dados” (SGBD) MySQL para a armazenagem dos dados que serão acessados por todas as tecnologias analisadas. Será utilizada, ainda, a metodologia Rational Unified Process (RUP) para o desenvolvimento do projeto de software.

Conclusão:

A implementação desse sistema alcançou os resultados esperados quanto à viabilidade técnica da execução de um projeto dessa natureza, possibilitando a integração das tecnologias analisadas. O resultado foi alcançado utilizando-se de ferramentas gratuitas, o que comprovou a viabilidade econômica. A partir dessa implementação, conclui-se é possível elaborar uma arquitetura que generaliza o desenvolvimento de sistemas dessa natureza.

Anexo C – Publicação no 1º CONISLI

Trabalho Aprovado em outubro de 2003

Sistema de Acesso a Dados via Dispositivos Fixos e Móveis Independentes de Plataforma

Autor: Tiago Aparecido Teixeira

Orientador: Luciano Eugênio de Castro Barbosa

1. Introdução

No cenário atual em que vivemos a utilização de softwares gratuitos para o desenvolvimento de pesquisas é algo tão natural quanto o ato de pesquisar. Este artigo faz parte de um dos possíveis estudos de caso de uma arquitetura para o desenvolvimento de Sistemas de Acesso a Dados via Dispositivos Fixos e Móveis Independentes de Plataforma, para atingir esse objetivo a arquitetura proposta visa a elaboração de sistemas que possam ser construídos apenas uma única vez e portados para uma gama infinita de plataformas, para atingir esse objetivo esse estudo de caso utilizou-se inteiramente de softwares gratuitos dentro de cada tecnologia analisada. Esse uso se deu sobre dois aspectos de igual importância. São eles:

- 1) Os softwares usados são portáteis entre diversas plataformas operacionais.
- 2) Os custos de um projeto de pesquisa desse nível seriam muito elevados caso se usasse softwares proprietários semelhantes dentro das tecnologias analisadas.

2. Revisão Bibliográfica

2.1. Ferramentas RAD

O método RAD beneficia-se das vantagens oferecidas por ferramentas poderosas de desenvolvimento para produzir rapidamente sistemas de alta qualidade. Essas novas ferramentas, mais intuitivas e visuais, permitem que o projetista de software construa "visualmente" a interface com o usuário, além de gerar o código numa linguagem orientada a objetos, permitindo o reuso de componentes [1].

2.2. Sistemas de Gerenciamento de Bancos de Dados

Um SGBD – Sistema de Gerenciamento de Banco de Dados é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades. Há, também, bancos de dados baseados em Gerenciadores de Base de Dados e Gerenciadores de Arquivos [2].

2.3. Sites Dinâmicos

JSP (Java Server Pages) é uma tecnologia para desenvolvimento de aplicações WEB semelhante ao Microsoft Active Server Pages (ASP), porém tem a vantagem da portabilidade de plataforma podendo ser executado em outros Sistemas Operacionais além dos da Microsoft. Ela permite ao desenvolvedor de sites produzir aplicações que permitam o acesso à banco de dados, o acesso a arquivos-texto, a captação de informações a partir de formulários, a captação de informações sobre o visitante e sobre o servidor, o uso de variáveis e loops entre outras coisas [3].

Por definição, JSP usa Java como sua linguagem de scripts. Por esse motivo, O JSP se apresenta mais flexível e mais robusto do que outras plataformas baseadas simplesmente em JavaScripts e VBScripts [3].

As Java Server Pages combinam HTML ou XML com partes de código Java para produzirem páginas Web dinâmicas. Cada página é compilada automaticamente em uma Servlet pelo mecanismo JSP na primeira vez que ela é solicitada e, a seguir, é executada. As JSP fornecem diversas maneiras de se dialogar com classes, servlets e applets Java, bem como com o servidor Web [4].

2.4. Sistema de Comunicação Móvel

O conteúdo e as aplicações WAP foram especificadas através de um conjunto de formatos bem conhecido, baseado na Internet tradicional. O browser interpreta e exibe conteúdo desenvolvido para o ambiente WAP. Esse conteúdo é criado através da linguagem WML (Wireless Markup Language ou Linguagem de Marcação Sem Fio). WML é semelhante a HTML que é usada na criação de páginas da Internet. Mas, ao contrário da linguagem HTML, a linguagem WML foi criada para atender às necessidades dos dispositivos e redes sem fio com baixo poder computacional. Para acrescentar recursos dinâmicos às aplicações WML, utiliza-se a linguagem WML Script, uma linguagem semelhante a JavaScript. Tanto WML como WML Script são adaptadas e otimizadas para utilização em ambiente sem fio [5].

3. Metodologia

A abordagem a ser utilizada é a convergência de tecnologias existentes, como a arquitetura Java Enterprise Edition (J2EE), através de sua extensão para a Internet Java Server Pages (JSP) levando à implementação do

dinamismo no ambiente da Internet; a utilização do protocolo Wireless Application Protocol (WAP), através de sua linguagem Wireless Markup Language (WML) para o acesso móvel; a utilização da ferramenta RAD Borland Kylix 3 para o desenvolvimento do cliente Desktop portátil para, pelo menos, duas plataformas distintas e o “Sistema de Gerenciamento de Banco de Dados” (SGBD) MySQL para a armazenagem dos dados que serão acessados por todas as tecnologias analisadas. Será utilizada, ainda, a metodologia Rational Unified Process (RUP) para o desenvolvimento do projeto de software.

4. Projeto

A integração de Sites Dinâmicos e SGBDs possibilita a manipulação dos dados contidos no SGBD pelas páginas dinâmicas geradas pelos servidores de aplicação inclusive às páginas geradas em WML para dispositivos Wireless possibilitando, assim que aplicativos gerados por ferramentas RAD, páginas geradas por mecanismos de Sites Dinâmicos e Dispositivos de Computação Móvel utilizem um canal comum de acesso ao SGBD.

Com o uso de SGBD será possível que um dado qualquer por software desenvolvido por uma ferramenta RAD seja acessado por Dispositivo de Computação Móvel através de um Servidor de Aplicação que tenha gerado um Site Dinâmico que possua acesso ao SGBD.

5. Estudo de Caso

A implementação de um estudo de caso visa testar a aplicabilidade e viabilidade da Arquitetura que está sendo proposta neste trabalho. Para isto o sistema produzido usou como base de desenvolvimento a Arquitetura proposta, como modelagem de dados uma ferramenta

Case, e como processo de desenvolvimento de software o Processo Unificado Rational RUP (Rational Unified Process).

A elaboração, desse estudo de caso específico, procurou utilizar tecnologias independentes de plataforma e – sempre que possível – tecnologias gratuitas para diminuir o custo de desenvolvimento do sistema.

Os componentes utilizados neste estudo de caso podem ser observados no diagrama da figura 1.

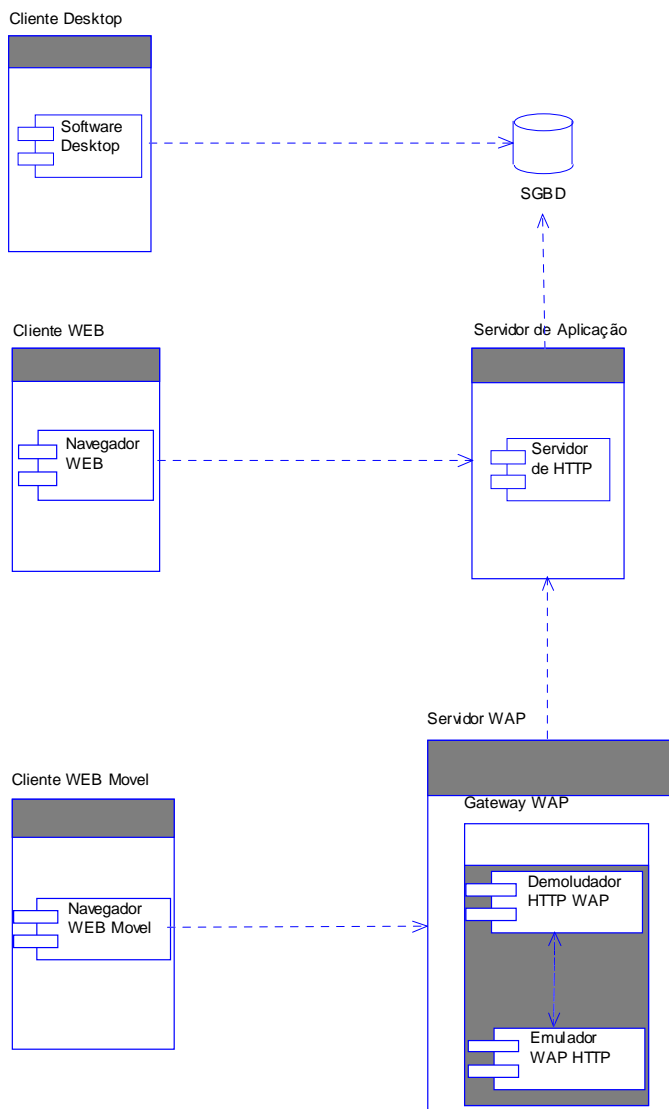


Figura 1 – Componentes utilizados na implementação do estudo de caso.

5.1. Sistema de Gerenciamento de Banco de Dados

Dentro da proposta de desenvolvimento de se utilizar tecnologias gratuitas disponíveis na Web e de uma maior independência de plataforma operacional optou-se por utilizar o SGBD o MySQL (disponível em www.mysql.com) por este possuir as seguintes características:

O MySQL é um SGBD (Sistema de Gerenciamento de Banco de Dados) relacional com código fonte aberto. Foi originalmente desenvolvido para manipular bases de dados muito grandes, com maior rapidez que as soluções existentes, e tem sido utilizado com sucesso em ambientes de produção com alta demanda. Apesar de estar em constante desenvolvimento, o MySQL oferece um rico e muito útil conjunto de funções. Conectividade, velocidade e segurança tornam o MySQL altamente apropriado para acesso a bancos de dados na Web [6].

O MySQL é um sistema cliente/servidor que consiste de um servidor SQL multi-threaded que suporta diferentes backends, vários programas clientes e bibliotecas, ferramentas administrativas e uma interface de programação [6].

5.2. Cliente Desktop

O Cliente Desktop possibilita a administração remota do sistema gerado de forma segura, pois algumas áreas do banco de dados usado, assim como varias informações

administrativas, só podem ser acessadas desse modo por um Cliente Desktop implementado utilizando-se ferramenta RAD

5.2.1. Ferramenta RAD

Para possibilitar uma maior usabilidade do sistema desenvolvido optou-se pela ferramenta RAD Borland Kylix 3 Open Edition. Para possibilitar o acesso a dados pelo sistema foi utilizado componentes de acesso a dados compatíveis com o SGBD MySQL disponíveis na Web na forma de componentes freeware (ZeosDBO <http://zeoslib.sourceforge.org>).

5.3. Cliente Web

O usuário em Modo Web poderá utilizar o Sistema através de dois outros modos: Modo Web Convencional ou Modo Web Móvel. A escolha de um desses modos dependerá, basicamente, do tipo de dispositivo que o usuário tiver disponível no momento do acesso ao sistema.

5.3.1. Cliente Web Convencional

O usuário em Modo Web Convencional poderá utilizar o sistema através de um Cliente Leve que acessara a Camada Web (Servidor HTTP). Utilizando para isso sites gerados de forma dinâmica.

5.3.1.1. Sites Dinâmicos

Para a geração de páginas Web de forma dinâmica foi utilizada a plataforma de desenvolvimento J2EE (Java Enterprise Edition) através de sua extensão para a Web

JSP (Java Server Pages, disponível em <http://jakarta.apache.org/tomcat/>).

5.3.2. Cliente Web Móvel

O usuário para ter acesso ao Modo Web Móvel irá necessitar de um dispositivo que permita o acesso a Web Móvel, normalmente um celular, hand held, ou outro tipo de computador portátil.

Para o dispositivo móvel usado pelo usuário acessar o Modo Web Móvel será necessário o uso de um Cliente Leve especialmente projetado para Web Móvel.

5.3.2.1. Tecnologia de Comunicação Móvel

Para que uma maior mobilidade e para que uma gama maior de usuários possa ter acesso móvel ao sistema o mecanismo de sites dinâmicos possibilita a geração de páginas dinâmicas especialmente projetadas para o acesso móvel e no caso de celulares as páginas geradas são na linguagem WML.

As páginas geradas de forma dinâmica foram testadas utilizando-se o Nokia Mobile Internet Toolkit 4.0 esse kit de desenvolvimento é disponibilizado gratuitamente pela Nokia em <http://www.forum.nokia.com>

6. Conclusão

A implementação desse sistema alcançou os resultados esperados quanto à viabilidade técnica da execução de um projeto dessa natureza, possibilitando a integração das tecnologias analisadas. O resultado foi alcançado utilizando-se de ferramentas gratuitas, o que comprovou a viabilidade econômica.

7. Referências

- [1] Uma Metodologia de Projeto Orientado a Objetos Com Vistas à Reutilização Fabrício André Rubin Prof. Dr. Carlos Alberto Heuser (orientador)
- [2] Apostila de Banco de Dados e SQL Autores: Prof. Jorge Surian Prof. Luiz Nicochelli SGBD x GA
- [3] JSPBrasil - Tutorial JSP O que é JSP por: Ulisses Telemaco Neto ulisses@jspbrasil.com.br
- [4] MUKHI, V. and MUKHI, S. and KOTOCHA, N. (2002) “Java Servlets JSP”, Makron Books, São Paulo.
- [5] RISCHPATER, R. (2001) "Desenvolvendo Wireless para WEB: Como Enfrentar os Desafios dos Projetos para a Web Sem Fio", Makron Books, São Paulo.
- [6] www.mysql.com/documentation em maio de 2001

Anexo D – Citação no 4º CIEE

Resumo Aprovado em novembro de 2003

Sistema Multisensorial De Acesso A Dados Independente De Plataforma

Autor: Tiago Aparecido Teixeira, aluno formando ciência
da computação, FAGOC, Ubá – MG – Brasil
thiagus@fagoc.br, Divinésia – MG – Brasil

**Orientador: Prof. M.Sc. Luciano Eugênio de Castro
Barbosa**, FAGOC, Ubá – MG – Brasil
lecb0@fagoc.br, Viçosa – MG – Brasil

**Palavras chaves: acesso móvel, tecnologias gratuitas,
multiplataforma, voiceml, jsp**

Introdução:

Há diversas tecnologias de sistemas independentes de plataforma existentes, que possuem um elevado custo e uma dependência cada vez maior de dispositivos específicos sofisticados.

Objetivos:

Este trabalho propõe-se a desenvolver um “Sistema Multisensorial de Acesso a Dados Independente de Plataforma”, possibilitando o acesso aos dados de uma organização ou instituição por usuários com necessidades especiais, independente do lugar em que o usuário esteja ou do dispositivo (fixo / móvel) ou sistema operacional que utilize. O usuário de tal sistema terá como meios de acesso um navegador web e um dispositivo móvel (como celular, por exemplo). O trabalho objetiva demonstrar que é possível aproveitar-se de tais tecnologias a baixo custo, quando são utilizados recursos gratuitos disponíveis na Internet.

Metodologia:

A abordagem a ser utilizada é a convergência de três tecnologias existentes. O framework do Java 2 Enterprise Edition (J2EE), através de sua extensão Java Server Pages (JSP), será usado para implementar o dinamismo no ambiente da Internet. O protocolo Wireless Application Protocol (WAP), através de sua linguagem Wireless Markup Language (WML) será usado para implementar o acesso móvel. O “Sistema de Gerenciamento de Banco de Dados” (SGBD) MySQL será utilizado para a armazenagem dos dados que serão acessados por todas as tecnologias de acesso a dados mencionadas. O acesso para usuários com necessidades especiais será obtido com a inclusão de um modulo desenvolvido de forma dinâmica em VoiceML através do mecanismo JSP. Para o desenvolvimento do projeto do software, será utilizado o processo Rational Unified Process (RUP).

Conclusão:

A implementação desse sistema deverá alcançar os resultados esperados quanto à viabilidade técnica da execução de um projeto dessa natureza, possibilitando a integração das tecnologias analisadas. Usuários com deficiência visual serão beneficiados com uma aplicação prática do VoiceML, que servirá de modelo para outros desenvolvedores de software o utilizarem em qualquer projeto. O processo de desenvolvimento fará uso de ferramentas gratuitas, o que comprovará a viabilidade econômica.